IBM

IBM WebSphere QualityStage Methodologies, Standardization, and Matching

IBM WebSphere QualityStage architecture

Merger and acquisition business scenario

IBM Information Server overview

Nagraj Alur
Alok Kumar Jha
Barry Rosen
Torben Skov

Redbooks

ibm.com/redbooks

International Technical Support Organization

IBM WebSphere QualityStage Methodologies, Standardization, and Matching

June 2008

SG24-7546-00

Note: Before using this information and the product it supports, read the information in "Notices" on page xxxv.

First Edition (June 2008)

This edition applies to Version 1, Release 8, Modification 0 of IBM Information Server (5724-Q36).

Contents

Figures

Tables

Examples

Notices

Trademarks

Preface

This IBM® Redbooks® publication documents the procedures for implementing IBM WebSphere® QualityStage and related technologies using a typical merger and acquisition financial services business scenario.

It is aimed at IT architects, Information Management specialists, and Information Integration specialists who are responsible for developing IBM WebSphere QualityStage on a Red Hat Enterprise Linux® 4.0 platform.

The book offers a step-by-step approach to implementing IBM WebSphere QualityStage on Red Hat Enterprise Linux 4.0 platform accessing information that is stored on IBM z/OS® and IBM AIX® platforms.

This book is organized as follows:

- Chapter 1, “IBM WebSphere QualityStage overview” on page 1 provides a detailed description of IBM WebSphere QualityStage, its architecture, configuration flow, and runtime flow.
- Chapter 2, “Financial services business scenario” on page 479 describes a step-by-step approach to implementing IBM WebSphere QualityStage on a Red Hat Enterprise Linux 4.0 platform using a typical merger and acquisition financial services business scenario that involves migration and data integration.
- Appendix A, “IBM Information Server overview” on page 857 provides a detailed description of IBM Information Server, its architecture, configuration flow, and runtime flow.
- Appendix B, “Code and scripts used in the financial services business scenario” on page 887 documents some of the code and scripts that are used in the migration and data integration business scenarios.
- Appendix C, “Additional material” on page 909 explains how to locate and download the additional materials that accompany this book.

The team that wrote this book

This book was produced by a team of specialists from around the world working at the International Technical Support Organization (ITSO), San Jose Center.

Nagraj Alur is a a Project Leader with the IBM ITSO, San Jose Center. He holds a master’s degree in computer science from the Indian Institute of Technology (IIT), Mumbai, India. He has more than 33 years of experience in database management systems (DBMSs) and has been a programmer, systems analyst, project leader, independent consultant, and researcher. His areas of expertise include DBMSs, data warehousing, distributed systems management, database performance, information integration, and client/server and Internet computing. He has written extensively on these subjects and has taught classes and presented at conferences all around the world. Before joining the ITSO in November 2001, he was on a two-year assignment from the Software Group to the IBM Almaden Research Center, where he worked on Data Links solutions and an eSourcing prototype.

Alok Kumar Jha is currently working as a BI Solution Architect in Industry Solutions at the IBM India Software Lab (ISL) in Bangalore. His expertise includes designing and developing BI Solutions across the verticals. He earned his MBA degree from FMS-IIRM, Jaipur, and masters in Statistics from Hindu College, Delhi University. He earned his bachelors in Statistics from Kirori Mal College, Delhi University. He has eight years of experience in Data Warehousing/BI. He has worked with IBM for the past four years. Prior to joining the ISL at IBM he worked at Cognizant Technology Solutions, iGate Global Solutions, and SPSS in various roles. Alok’s core area of expertise includes end-to-end Data Warehousing/BI. His skills include Dimensional Modeling, Data Integration (ETL, EII), Relational Database Reporting, Multidimensional Modeling and Analysis (OLAP), Data Mining, and Statistical Analysis. Alok has designed ETL processes that involved application data such as SAP® R/3 for designing and developing Profitability Analysis (CO-PA) for multidimensional analysis for different KPIs. He has worked extensively on different BI platforms, including DB2® Framework for Business Intelligence, Cognos Framework for Business Intelligence, Microsoft® Framework for Business Intelligence, SPSS, and SAS technologies.

Barry Rosen is currently serving as Director of Best Practices and Data Management in the Center of Excellence for Data Integration and a “champion” of Data Profiling, Data Quality, and Metadata at IBM. His past roles include Director of Enterprise Architecture/Data Warehousing, Technical Architect, and Principal Consultant. He holds a master’s degree in Engineering Management and Computer Information Systems from Northeastern University. Barry has designed, managed, and implemented multiple highly available, large scale transaction, data warehousing/mining and business intelligence solutions in various vertical market segments worldwide including financial, telecom, insurance, pharmaceutical and retail sectors. He has over 25 years of technical architecture and customer focused information management expertise. He has provided technical leadership for business system architecture and applications including ERP, CRM, and data warehousing. Barry has consulted on various complex, high-risk, time sensitive systems for companies such as Epsilon Data Management, Fidelity Investments, Wellington Management Company, Investors Bank and Trust, and Harte-Hanks Data Technologies.

Torben Skov is working as an IT Specialist within the IOD department in IBM Denmark. His current area of interest is the IBM WebSphere Information Server product suite focusing on QualityStage and DataStage®. Torben holds a Master of Science in Economics and Business Administration from the University of Southern Denmark, specializing in Organizational Theory and Operations Research. Torben has 10 years of experience in ETL and application development using the SAS System. Torben has for the past two years been working for IBM within the BI/IOD area. While working at IBM on previous assignments, Torben was engaged in projects at some of the largest companies within the telecommunication and agricultural sector of Denmark. The tasks include designing and developing a system for financial modelling, Project Manger for building a Web portal with secured content for the Agricultural Industry and Project Manager re-factoring a company-wide budgeting system using Web Services.

Thanks to the following people for their contributions to this project:

Elizabeth Dial
Stewart Henna
Harald Smith
IBM Westbboro

Atul Chadha
Asim Singh
IBM Silicon Valley Laboratory, San Jose

Denis Vasconcelos
IBM Brazil

Become a published author

Join us for a two- to six-week residency program! Help write a book dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an e-mail to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

1

IBM WebSphere QualityStage overview

In this chapter, we provide an overview of IBM WebSphere QualityStage, its architecture, and the configuration and execution of its main functions.

The topics that we cover in this chapter are:

- IBM WebSphere QualityStage overview
- IBM WebSphere QualityStage architecture
- IBM WebSphere QualityStage main functions
- IBM WebSphere QualityStage in a project context
- Investigate stage
- Standardize stage
- Match stage
- Survive stage
- Mailing list scenario

1.1 Introduction

The data that drives today's business systems often comes from a variety of sources and disparate data structures. As organizations grow, they retain old data systems and augment them with new and improved systems. Thus, data becomes difficult to manage and use, and a clear picture of a customer, product, or buying trend can be practically impossible to ascertain.

Data quality can be broadly defined from a qualitative viewpoint as information that you can trust or ensuring that data at a particular point in time is suitable for its purpose. A more specific quantitative definition can include the level of compliance that is attained by an enterprise's data environment to independently define rules that describe that data environment. The emphasis is on the business user's perception of data quality—what is delivered to the user and the semantics of the data that is presented. In this case, data quality might well depend upon the data itself (such as correct data types, consistent formatting, retrievability, and usability) and the processes and applications that deliver this data to the business user.

The source of data quality issues is a lack of common standards on how to store data and an inconsistency in how the data is input. Different business operations are often very creative with the data values that they introduce into your application environments. Inconsistency across sources makes understanding relationships between critical business entities such as customers and products very difficult. In many cases, there is no reliable and persistent key that you can use across the enterprise to get all the information that is associated with a single customer or product.

Without high-quality data, strategic systems cannot match and integrate all related data to provide a complete view of the organization and the interrelationships within it. CIOs can no longer count on a return on the investments made in critical business applications. The solution calls for a product that can automatically re-engineer and match all types of customer, product, and enterprise data, in batch or at the transaction level in real time.

Important: To assess and improve data quality, you need a concerted cooperative effort from a number of individuals such as subject matter experts (SME) and IT Data Analysts (DA) using state of the art tools and technologies. These tools and technologies need to assist an individual detect poor quality, fix poor quality data, identify the causes and entry points of poor quality data, develop processes to trap poor quality data at the point of origin, and then monitor the success of the efforts to improve data quality.

IBM Information Server addresses these requirements with an integrated software platform that provides the full spectrum of tools and technologies that are required to address data quality issues. These technologies include data profiling[1] (IBM WebSphere Information Analyzer and IBM WebSphere AuditStage), data cleansing (IBM WebSphere QualityStage), and data movement and transformation (IBM WebSphere DataStage) as follows:

- IBM WebSphere Information Analyzer (the focus of *IBM WebSphere Information Analyzer and Data Quality Assessment*, SG24-7508) is a new module of IBM Information Server that represents the next generation data profiling and analysis tool. It is designed to help business and data analysts understand the content, quality, and structure of their data sources by automating the data discovery process. The IBM WebSphere Information Analyzer product module helps increase the productivity of data personnel and improve return on investment (ROI) and time to benefit of data-intensive projects.
- IBM WebSphere DataStage (which will be covered in an upcoming IBM Redbooks publication) integrates data across multiple and high volumes of data sources and target applications. It integrates data on demand with a high performance parallel framework, extended metadata management, and enterprise connectivity.
- IBM WebSphere QualityStage (the focus of this book) enables customers to use the following processes to make a significant impact in the data that drives an organization's success.
  - Investigation of source data to understand the nature, scope, and detail of data quality challenges.
  - Standardization to ensure that data is formatted and conforms to organization-wide specifications, including name and firm standards as well as address cleansing and verification.
  - Matching of data to identify duplicate records within and across data sets.
  - Survivorship to eliminate duplicate records and create the "best record view" of data.

  1.2, "IBM WebSphere QualityStage overview" on page 4 provides further details about these processes.

Note: For an overview of IBM Information Server, refer to:

http://www.ibm.com/software/data/integration/info_server/

[1] *Data profiling* is a data management practice that reveals the content, quality, and structure of the data. It is vital to the success of any business integration effort.

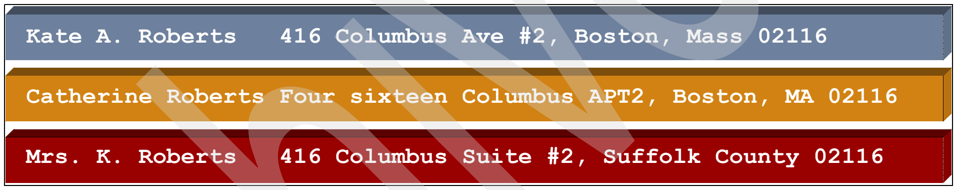
In the following sections, we provide an overview of IBM WebSphere QualityStage, its architecture, main functions, project context, main components, and a mailing list scenario to showcase its main as well as commonly used features.

1.2 IBM WebSphere QualityStage overview

As briefly introduced in 1.1, "Introduction" on page 2, the five types of issues that generally occur within enterprise data stores are as follows:

- Lack of information standards

  Names, addresses, part numbers, and other data are entered in inconsistent ways, particularly across different systems. These differences make records look different even when they are actually the same as shown in Figure 1-1, where *Kate Roberts* is represented in three different ways, with different address standards.

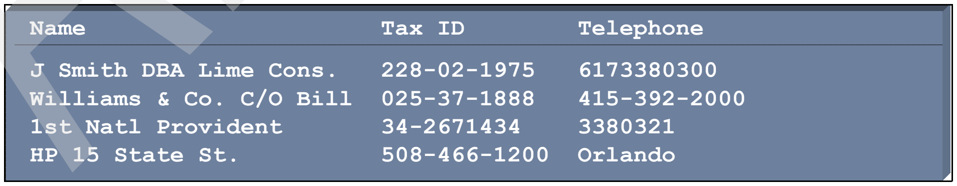


```
Kate A. Roberts    416 Columbus Ave #2, Boston, Mass 02116
Catherine Roberts Four sixteen Columbus APT2, Boston, MA 02116
Mrs. K. Roberts    416 Columbus Suite #2, Suffolk County 02116
```

Figure 1-1 Lack of information standards example

- Data surprises in individual fields

  Data in a database is often misplaced, while in other cases some fields are used for multiple purposes. Figure 1-2 shows an example where the Name field contains company and address information, the Tax ID field contains telephone numbers, and the Telephone field has a variety of mistakes. This often leads to program and application errors and results in the mis-identification of key products and customers.

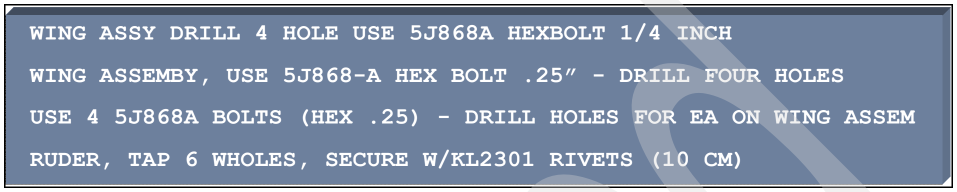


Name	Tax ID	Telephone
J Smith DBA Lime Cons.	228-02-1975	6173380300
Williams & Co. C/O Bill	025-37-1888	415-392-2000
1st Natl Provident	34-2671434	3380321
HP 15 State St.	508-466-1200	Orlando

Figure 1-2 Data surprises in individual fields example

- Information is buried in free-form fields

  In this case, valuable information is hidden away in text fields as shown in Figure 1-3. Because these fields are difficult to query using SQL, this information is often not taken advantage of, although it most likely has value to the business. This type of problem is common in product information and help desk case records.



WING ASSY DRILL 4 HOLE USE 5J868A HEXBOLT 1/4 INCH

WING ASSEMBY, USE 5J868-A HEX BOLT .25" - DRILL FOUR HOLES

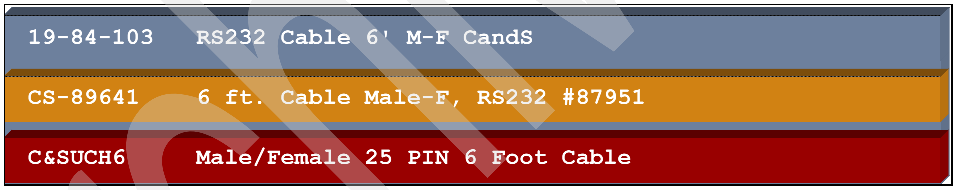
USE 4 5J868A BOLTS (HEX .25) - DRILL HOLES FOR EA ON WING ASSEM

RUDER, TAP 6 WHOLES, SECURE W/KL2301 RIVETS (10 CM)

Figure 1-3 Information buried in free-form fields example

- Data *myopia* (our term for the lack of consistent identifiers across different systems)

  Without adequate foreign-key relationships, it is impossible to get a complete view of information across systems. Figure 1-4 shows three products that look very different but that are actually the same.

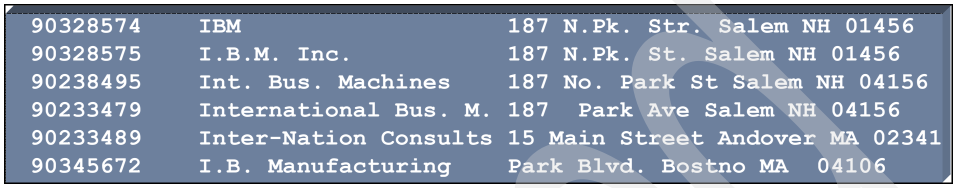


19-84-103	RS232 Cable 6' M-F CandS
CS-89641	6 ft. Cable Male-F, RS232 #87951
C&SUCH6	Male/Female 25 PIN 6 Foot Cable

Figure 1-4 Data myopia example

- Redundancy within individual tables

  This issue is extremely common, where data is re-entered into systems because the data entry mechanism is not aware that the original record is already there. This issue is a common side effect of the lack of standards, but it is one of the worst data quality problems, because it links directly to costs and customer dissatisfaction. Figure 1-5 shows an example of redundant IBM information that is represented differently in a data store.



```
90328574   IBM                      187 N.Pk. Str. Salem NH 01456
90328575   I.B.M. Inc.              187 N.Pk. St. Salem NH 01456
90238495   Int. Bus. Machines       187 No. Park St Salem NH 04156
90233479   International Bus. M.    187  Park Ave Salem NH 04156
90233489   Inter-Nation Consults    15 Main Street Andover MA 02341
90345672   I.B. Manufacturing       Park Blvd. Bostno MA  04106
```

Figure 1-5 Redundancy within individual tables example

IBM WebSphere QualityStage helps to identify and resolve all these issues for any type of data. It helps to ensure that systems deliver accurate and complete information to business users across the enterprise. IBM WebSphere QualityStage is a data re-engineering environment that is designed to help programmers, programmer analysts, business analysts, and others cleanse and enrich data to meet business objectives and data quality management standards.

A process for re-engineering data can help accomplish the following goals:

- Resolve conflicting and ambiguous meanings for data values
- Identify new or hidden attributes from free-form and loosely controlled source fields
- Standardize data to make it easier to find
- Identify duplication and relationships among such business entities as customers, prospects, vendors, suppliers, parts, locations, and events
- Create one unique view of the business entity
- Facilitate enrichment of re-engineered data, such as adding information from vendor sources or applying standard postal certification routines

> **Note:** You can use a data re-engineering process in batch or real time for continuous data quality improvement.

IBM WebSphere QualityStage provides data quality functions on an easy-to-use, design-as-you-think flow diagram, which allows data quality to be embedded in any information integration process.

IBM WebSphere QualityStage data quality functions include:

- Free-form text investigation that allows you to recognize and parse out individual fields of data from free-form text

  Investigation provides the ability to determine the number and frequency of the unique values found in either single-domain or free-form columns:

  – For *single-domain* columns, the unique values can represent complete data values, partial data values (that is prefixes, suffixes, or substrings), simple data formats (alpha, numeric, blanks, or special characters), or combinations of values and simple data formats.

  – *Free-form* columns often contain multiple (usually related) data elements such as person names, business names, or postal addresses. For free-form columns, the unique values can represent individual tokens (words) that are determined by parsing the free-form data or lexical patterns providing context to help understand the free-form data.

- Standardization that allows individual fields to be parsed and made uniform according to your own standards

  Standardization provides the ability to normalize your information to defined standards. This incorporates the ability to parse free-form columns into single-domain data elements to create a consistent representation of the input data and to ensure data values conform to standard representation.

- Address verification and correction that uses postal information to standardize, validate, and enrich address data

  For addresses, standardization incorporates postal standards with delivered rule sets. It also prepares data elements for more effective matching.

- Record linkage and matching that allows duplicates to be identified from individual sources, and common records across sources to be identified and linked

  The central strength of IBM WebSphere QualityStage is its ability to match data from different records, even when it appears very different. IBM WebSphere QualityStage utilizes a statistical matching technique called *Probabilistic Record Linkage*[2] that provides the highest optimization of each contributing data element to a match and an unlimited number of elements for the highest confidence in your information. This matching allows each individual field score to be summed to produce a final score that precisely measures the information content of the matching fields. That final score, or *match weight*, is an accurate gauge of the probability of a match.

[2] Probabilistic Record Linkage is familiar to computer science professionals who must perform highly precise matching when there is great liability and consequence from errors. This method evaluates each field, taking into account frequency, discriminating value, and data reliability, and produces a score that is a numerical representation of the amount of information that is produced by the single pair of values.

Note: The design of these matching rules is very important, because it determines which records are brought together. These match rules are designed using a visual, business-centric interface that provides instant feedback on match rule changes to allow the rules to be fine tuned quickly and easily.

Because of this ability to match records, IBM WebSphere QualityStage is a key enabler of creating a single view of customers or products.

- Survivorship that allows the best data from across different systems to be merged into a consolidated record

  Survivorship is the process of aggregating or consolidating a group of records into a single unique representation of data, a single consolidated record for retaining the best of breed information from the given individual records.

  Survivorship incorporates a flexible process of assessing data at the record or individual field level based on specific survivorship rules. These rules can include completeness of data, frequency of data, or logical conditions.

These capabilities of IBM WebSphere QualityStage support the re-engineering of data to meet complex business goals and challenges.

Note: A key opportunity exists in these investigation and standardization phases to extend the reach of IBM WebSphere QualityStage with the addition of IBM Global Name Recognition (GNR).[a] The challenge is that representations of names vary dramatically according to the cultures from which they come. A cultural understanding of a name can go a great length towards the effective, accurate processing of client information. We do not discuss IBM Global Name Recognition in this book. For more information, see:

http://www.ibm.com/software/data/ips/products/masterdata/globalname/

a. A unique competitive advantage is achieved with the addition of IBM Global Name Recognition, the industry's leading solution for comprehensive multi-cultural name genderization, cultural classification, parsing, recognition, and analysis. This unique solution is built upon years of extensive linguistic research and a knowledge base of approximately one billion names, representing over 200 countries from around the world. The result is a deeper understanding of multi-cultural names and the information that these names contain to increase business insight and further enhance the quality of data.

Attention: In this book, we do not cover all the functions and features of IBM WebSphere QualityStage. Refer to the resources that are described in "Related publications" on page 911 for more details about IBM WebSphere QualityStage.

1.3 IBM WebSphere QualityStage architecture

IBM WebSphere QualityStage is built around a services-oriented vision for structuring data quality tasks that are used by many new enterprise system architectures. As part of the integrated IBM Information Server platform, it is supported by a broad range of shared services and benefits from the reuse of several suite components.

IBM WebSphere QualityStage (and IBM WebSphere DataStage) share the same infrastructure for importing and exporting data, for designing, deploying, and running jobs, and for reporting. The developer uses the same design canvas to specify the flow of data from preparation to transformation and delivery.

Multiple discrete services give IBM WebSphere QualityStage the flexibility to match increasingly varied customer environments and tiered architectures. Figure 1-6 on page 10 shows how IBM WebSphere QualityStage Designer (labeled *Development interface*) interacts with other elements of the IBM Information Server platform to deliver enterprise data analysis services.

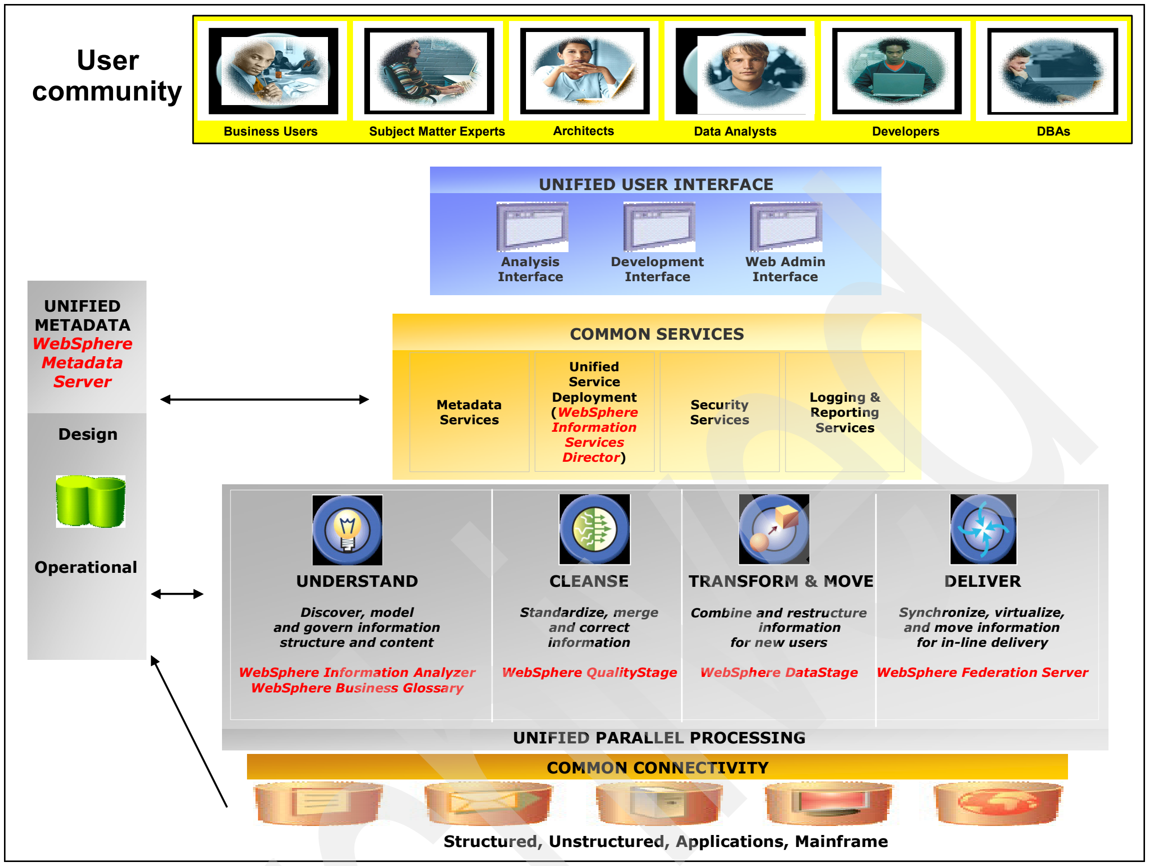

User community
Business Users
Subject Matter Experts
Architects
Data Analysts
Developers
DBAs
UNIFIED USER INTERFACE
Analysis Interface
Development Interface
Web Admin Interface
UNIFIED METADATA
WebSphere Metadata Server
Design
Operational
COMMON SERVICES
Metadata Services
Unified Service Deployment (WebSphere Information Services Director)
Security Services
Logging & Reporting Services
UNDERSTAND
Discover, model and govern information structure and content
WebSphere Information Analyzer
WebSphere Business Glossary
CLEANSE
Standardize, merge and correct information
WebSphere QualityStage
TRANSFORM & MOVE
Combine and restructure information for new users
WebSphere DataStage
DELIVER
Synchronize, virtualize, and move information for in-line delivery
WebSphere Federation Server
UNIFIED PARALLEL PROCESSING
COMMON CONNECTIVITY
Structured, Unstructured, Applications, Mainframe

Figure 1-6 IBM Information Server architecture

With reference to Figure 1-6, the following suite components are shared between IBM WebSphere QualityStage and IBM Information Server:

- **Unified user interface**

  The IBM WebSphere DataStage and QualityStage Designer provides a development environment. The IBM WebSphere DataStage and QualityStage Administrator provides access to deployment and administrative functions. IBM WebSphere QualityStage is integrated tightly with IBM WebSphere DataStage and shares the same design canvas, which enables users to design jobs with data transformation stages and data quality stages in the same session.

- **Common services**

  IBM WebSphere QualityStage uses the common services in IBM Information Server for logging and security. Because metadata is shared “live” across tools, you can access services such as impact analysis without leaving the design environment. You can also access domain-specific services for enterprise data cleansing such as investigate, standardize, match, and survive from this layer.

- **Common repository**

  The repository holds data to be shared by multiple projects. Clients can access metadata and results of data analysis from the respective service layers.

- **Common parallel processing engine**

  The parallel processing engine addresses high throughput requirements for analyzing large quantities of source data and handling increasing volumes of work in decreasing time frames.

- **Common connectors**

  Any data source that is supported by IBM Information Server can be used as input to a IBM WebSphere QualityStage job by using connectors. The connectors also enable access to the common repository from the processing engine.

IBM WebSphere QualityStage uses one or more of the stages shown in Figure 1-7 to improve an organization's data quality.

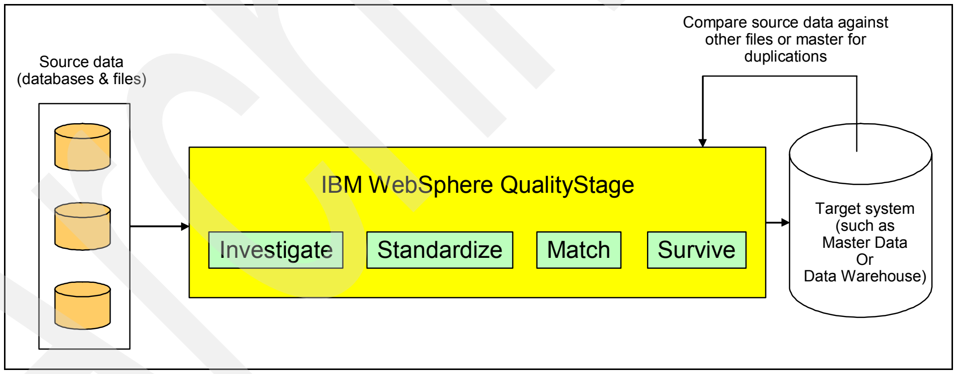

Source data
(databases & files)
Compare source data against other files or master for duplications
IBM WebSphere QualityStage
Investigate
Standardize
Match
Survive
Target system (such as Master Data Or Data Warehouse)

Figure 1-7 IBM WebSphere QualityStage process overview

The main functions are investigate, standardize, match, and survive as follows:

- *Investigate* source data to understand the nature, scope, and detail of data quality challenges.
- *Standardize* data to ensure that it is formatted and conforms to organization-wide specifications including name and firm standards as well as address cleansing and verification. Investigate can be used to assess the

effectiveness of Standardize. The standard rules can then be augmented to improve the Standardize data.

- *Match* data to identify duplicate records within and across data sets.
- *Survive* appropriate data by eliminating duplicate records and creating the best record view of data.

IBM WebSphere QualityStage comprises a set of stages, a Match Designer, and related capabilities that provide a development environment for building data-cleansing tasks called *jobs*. The IBM WebSphere QualityStage components include the Match Designer for designing and testing match passes and the IBM WebSphere QualityStage stage types, including Investigate, Standardize, Match Frequency, Reference Match, Unduplicate Match, and Survive. (This is not a comprehensive list.)

When a IBM WebSphere QualityStage job is compiled, the Designer (client) transfers the logic (developed on the client) to the Server. On execution, the IBM WebSphere QualityStage load modules perform the actual data engineering tasks such as investigate, standardize, matching, and survive. These load modules operate through control parameter statements that are passed to the modules during processing. Each IBM WebSphere QualityStage operation is referred to as a *stage*. Complex data engineering tasks can be performed by linking individual stages together in a job comprising multiple job steps or stages.

Note: Using IBM WebSphere QualityStage in batch mode allows for the bulk processing of data. IBM WebSphere QualityStage jobs can also be published as Web services through the WebSphere Information Services Director component of IBM Information Server. For more information, see *SOA Solutions Using IBM Information Server*, SG24-7402.

1.4 IBM WebSphere QualityStage main functions

IBM WebSphere QualityStage comes with customizable rules to prepare complex information about your business entities for a variety of transactional, operational, and analytical purposes.

IBM WebSphere QualityStage automates the conversion of data into verified standard formats including the use of probabilistic matching, in which variables that are common to records (such as the given name, date of birth, or gender) are matched when unique identifiers are not available.

IBM WebSphere QualityStage components include the Match Designer, for designing and testing match passes, and a set of data-cleansing operations called *stages*. Information is extracted from the source system and then measured, cleansed, enriched, consolidated, and loaded into the target system. At run time, data cleansing jobs typically consist of the following sequence of stages:

- *Standardize* stage parses free-form or fixed-format columns into single-domain data elements to create a consistent representation of the input data. This stage ensures that each data element has the same content and format and also standardizes spelling formats and abbreviations.

  Note: The *Investigate* stage is generally used as a development tool to obtain complete visibility into the actual condition of data prior to developing the data cleansing process. However, this stage could potentially be part of the data cleansing job after standardization to determine the effectiveness of standardization. It could also be used for checks and audits.

- *Match* stages ensure data integrity by linking records from one or more data sources that correspond to the same customer, supplier, or other entity. Matching can be used to identify duplicate entities that are caused by data entry variations or account-oriented business practices. For example:
  - Unduplicate Match jobs group records into sets (within a single data set) that have similar attributes.
  - Reference Match stage matches reference data to source data between two data sets.

  The probabilistic matching capability and dynamic weighting strategies of IBM WebSphere QualityStage help you to create high-quality, accurate data and identify core business information consistently, such as customer, location, and product throughout the enterprise. IBM WebSphere QualityStage standardizes and matches any type of information.
- *Survive* stage ensures that the best available data survives and is consequently prepared for the target correctly.

Business intelligence packages that are available with IBM WebSphere QualityStage provide data enrichment that is based on business rules. These rules can resolve issues with common data quality problems such as invalid address fields throughout multiple geographies. The following packages are available:

- Worldwide Address Verification and Enhancement System (WAVES) matches address data against standard postal reference data that helps you verify address information for 233 countries and regions.
- Multinational geocoding is used for spatial information management and location-based services by adding longitude, latitude, and census information to location data.
- Postal certification rules provide certified address verification and enhancement to address fields to enable mailers to meet the local requirements to qualify for postal discounts.

By ensuring data quality, IBM WebSphere QualityStage reduces the time and cost to implement CRM, business intelligence, ERP, and other strategic customer-related IT initiatives.

Some of the scenarios for data cleansing include:

- Obtaining one view of households

  Knowing that your customers (stored in different source systems) belong to the same household (share the same mailing address) can facilitate marketing and mail campaigns.
- Obtaining a consolidated view of an entity

  Knowing the total quarterly sales from the prescriptions of one doctor can help pharmaceutical companies effectively market to them. This information needs to be extracted and consolidated from existing systems with different standards and formats, including information buried in free-form fields, incorrect data values, discrepancies between field metadata and actual data in the field, and duplicates.
- Obtaining a single real-time view of a customer

  Customer Service Representatives (CSR) and customers using self-service portals require a single view of all their health, dental, and benefit plans with the insurance company. Typically, these plans are implemented in different systems, and the data resides in different sources in different formats.

IBM WebSphere QualityStage performs the preparation stage of enterprise data integration, often referred to as *data cleansing*. IBM WebSphere QualityStage takes advantage of the source systems analysis that is performed by IBM WebSphere Information Analyzer and supports the transformation functions of

IBM WebSphere DataStage. Working together, these products automate what was previously a manual or neglected activity within a data integration effort—data quality assurance.

Attention: Before you can perform the various functions that are provided by IBM WebSphere QualityStage, you must set up your system. This setup involves the following tasks:

- Creating and opening an IBM WebSphere QualityStage project as described in 1.10.1, “Create a project” on page 121.
- Importing metadata as described in 1.10.3, “Import table definitions” on page 126.
- Creating a parameter set object as described in 1.10.4, “Create a parameter set object” on page 130.
- Configuring the project, establishing connectivity to a data source, configuring system resources, and setting up security. Most of these tasks are described in 1.10, “Mailing list scenario” on page 117.

1.5 IBM WebSphere QualityStage in a project context

The design and development of an IBM WebSphere QualityStage process should always fit within the context of a broader project.

A data re-engineering development effort should be started only after a definitive project is identified and detailed project requirements are developed through a detailed Data Quality Assessment (DQA) review as described in *IBM WebSphere Information Analyzer and Data Quality Assessment*, SG24-7508. The deliverables created at the end of a DQA are used to define the scope of the data re-engineering engagement and to guide the design and development phases. Any necessary changes or additions to the DQA requirements and high-level design need to be reviewed with the business users and project management to determine the impact on project scope.

A data re-engineering effort is structured around a standard methodology for conducting data conditioning, standardization, matching, enrichment, survivorship, and output formatting for a given project. The project business requirements and target platform need to be defined and documented in the preceding DQA. The Investigate Stage of IBM WebSphere QualityStage is typically used in this context to fill out specific gaps in the DQA.

A typical project addresses a specific business initiative such as the integration of multiple (operations-oriented) line-of-business systems into a consolidated master data system. The new data structures can allow for a cross-reference or single representation of the targeted data. Using a project structure, the design, development, testing, and implementation of a data re-engineering effort using IBM WebSphere QualityStage follows well-defined project methodology to ensure accuracy and completeness. For example, existing systems that use account or policy numbers as primary keys typically do not allow for an easy and complete identification of all accounts and policies for a single unique customer. To further complicate record linkage, these data structures usually have different field formats or data quality levels. To link related records effectively, the systems must be re-engineered to restructure the data for matching and loading to the new target database.

The design of the data re-engineering process needs to account for the data sources and their data quality considerations as well as the target data structure. The development effort follows the design by only applying those techniques that are required to ensure that the project goals are met. Testing of the developed process needs to ensure that the results meet defined project expectations according to an established test plan. Production implementation needs to address standard considerations for process deployment and execution, as well as post-implementation assessment of results and success.

The following methods can be used in the data re-engineering process to restructure the data. The methods are usually performed in the order listed:

1. Source access or extraction*[3]
2. Conditioning
3. Standardization
4. Address verification*
5. Matching
6. Group association*
7. Survivorship*
8. Data enrichment*
9. Output formatting
10. Auditing the load process

We describe each of these methods briefly in the following sections.

[3] An asterisk (*) indicates optional methods. Data enrichment can take place prior to matching to add additional match fields (for example, a Dun and Bradstreet number for organization matching).

1.5.1 Source access or extraction (optional)

An extract process might be needed from specific systems, depending on the type of data to integrate and re-engineer. Existing mainframe systems or production systems that require high-availability are examples where extraction is recommended.

The extract process needs to pull all fields (as defined by the logical data model for the existing systems) to create an extract or *staged* file. This process should, when possible, ensure the removal of all binary or non-printable characters. Data extracts can be tagged with individual line of business identifiers. The extract procedure can be created and executed by the resources that are responsible for the maintenance of the source applications to ensure accurate data mapping. Whenever possible, a common input data format needs to be created to facilitate the input of additional data sources in the future.

As an ongoing process, records can be extracted either by:

- Comparing complete extracts to determine Add, Change, or Delete record types.
- Capture Add, Change, or Delete record types for each of the existing systems through existing fields and processes (that is, change date fields) or acquired third-party software.
- Some combination of these two methods.

Extraction is handled in IBM WebSphere QualityStage through standard connectivity stages. Such a discussion is beyond the scope of this book.

When possible, a common layout needs to be used for all input sources. This layout facilitates the introduction of new sources and simplifies future executions when some input sources might not produce an input file. These extract files can then be joined together easily for processing.

Each extract file must contain a Source ID, usually one or two bytes, that uniquely identifies the source from which that input originated. If available, each input needs to also contain a unique source record key that can be used to trace a single record from the source system to the target data source. This source input key is important for quality assurance and audit tracking, cross-reference file creation, and for identifying Add, Change, or Delete records (if necessary).

The extract process must also filter out rows and columns that are invalid or unnecessary for the matching and load requirements. When in doubt, however, extract a given row or column, and filter it out later. Changes to the extract files later in the development phase results in rework of the layout definitions, coding, and testing. This rework can result in project timeline delays when the IBM WebSphere QualityStage development has begun.

Typically, only the most recent information for a given source key needs to be extracted. Transaction and history processing must be captured in the source systems and data warehouse and not introduced into the data re-engineering application. In general, you need to perform entity matching only on the most recent data available for all records, with prior record versions used only to identify add, change, or deleted records.

1.5.2 Conditioning

The conditioning process decomposes the input data to its lowest common denominators, based on specific data value occurrences. It then identifies and types the component data properly in terms of its business meaning and value.

Data conditioning is performed on name, address, and other data elements that were determined through the DQA phase to contribute value to the entity integrity. Additional fields can also be created to condition some data for loading the target data structures only. These fields add value to the target data structures but are of no value to the matching process. The conditioning process can consist of multiple modular steps to identify and separate the input data or record into the following categories:

- Country of origin (U.S., Canada, U.K., and so forth.)
- Domain (Name, Address, Area)
- Separate domain-specific parsers (Name only, Address only, Area only, and so forth.)

The conditioning process is performed on each individual record and can be set up to run in multiple streams.

Conditioning utilizes a specific Standardize stage with the goal of decomposing complex data. For example, the pre-built Country rule set conditions name and address data based on country of origin.

1.5.3 Standardization

Standardization of the data removes anomalies and standardizes spellings, abbreviations, punctuation, and logical structures (domains). This process is performed after the conditioning process has determined the proper domain and parsed the data (if necessary) into its proper lowest common denominator. This timing is critical. For example, a value such as *North* can have multiple meanings and should be standardized only to *N* when it is identified as a directional value and not when it is determined to be a street name or the middle initial of a personal name.

As a result, Standardization usually occurs in the domain-specific parsing procedures which also use the Standardize stage.

1.5.4 Address verification (optional)

An address verification routine is usually an address look-up that uses postal files as reference files to check and standardize mailing and street address information. It is recommended that any data re-engineering application integrate country-specific, postal look-up processing into the data re-engineering job stream.

This optional process improves and enhances the quality of the address information for downstream entity matching and load. If this verification effort is targeted around discounted mailing, then it must use a software component that is certified for the targeted country, for example, CASS in the U.S., SERP for Canada, and DPID for Australia.

1.5.5 Matching

After standardization and conditioning are accomplished, development of the entity level processing (such as the creation of a unique client ID) also called the *match processing*, begins. The objective of match processing is the establishment of entity-level relationships (client, household, vendor, product, or parts) across all input records, generating each record's appropriate logical relationships. Unique entity keys are created to allow the organization to create entity-oriented views in addition to their existing operational views. The matching process can be set up to run independently for each defined entity relationship and can also be run in multiple parallel streams, depending on business rules.

This processing utilizes the IBM WebSphere QualityStage Match stage to perform the match or record linkage.

1.5.6 Group association (optional)

When two separate match processes are created using different match criteria, there might be a need to consolidate the Group (entity) IDs that are assigned by these distinct processes. The group association process performs this function. It reads in a unique record key and two Group IDs (one from each entity match) and outputs the unique record key and a consolidated Group ID. This consolidated Group ID also includes any additional associative matches between the two match sets.

There are differing techniques that can be applied for such group association, which include the Match stage but can use other Aggregation or Join stages.

1.5.7 Survivorship (optional)

The survivorship process creates a single representation of an entity across business lines with the best of breed data.

This process can be performed at:

- The record level
- The logical domain level (that is, Name, Address, Product, and so forth)
- The field level
- Any combination of these levels

By storing survivorship records in the target data store for each match process, a common institutional representation of data can be viewed across the organization. In storing the data in this fashion, each of the lines of business can access data or other lines of business data efficiently and easily.

This process utilizes the Survive stage.

1.5.8 Data enrichment (optional)

Occasionally, an organization needs to add additional information to some input records to improve the amount and quality of data that is available in the target data store. This can include:

- Geocoding information, such as latitude and longitude coordinates
- Data propagation of an input field/fields from one entity record to another matched entity record whose fields do not contain that same information
- Third-party entity identifier references

This data enrichment might require the use of the Match stage or a Lookup stage to add the additional data.

1.5.9 Output formatting

Finally, the resultant records need to be formatted for load to the target data store. This might require the creation of multiple load files, reformatting of parsed fields, creation of unique table keys, and other load related issues. As with the initial access or extraction, it utilizes standard connectivity or load stages. A discussion of this topic is beyond the scope of this book.

1.5.10 Auditing the load process

Use an investigation tool such as IBM WebSphere Information Analyzer to profile and verify that the values that are entered in each field match the data mapping models, metadata descriptions, and field sizes for the target data structures.

Also trace a sampling of the input records from the extraction file to the load files to ensure that all business requirements have been followed.

Finally, check the referential integrity within and between load files, based on Entity IDs and primary key fields, to identify duplicate primary keys that are flagged by the database load process as table key violations. This duplication usually occurs on records that have joint owners that are broken out into multiple records. The input information about the records might have conflicts that can cause them to be identified as the same individual (such as having the same Tax ID). Data entry errors such as these need to be identified and corrected in the source systems, because the data meets the match criteria or definition of a unique entity. If the data entered shows this type of situation, the source systems must be corrected, or primary key violations are flagged when loading the database, causing unpredictable results and a rapid degradation of the quality of the database over time.

1.6 Investigate stage

Understanding your data is a necessary precursor to cleansing and consolidation. You can use a data profiling product such as IBM WebSphere Information Analyzer to perform a comprehensive analysis across many columns and tables to create a direct input into the cleansing and matching process by using shared metadata. The Investigate stage supports the initial level of column or domain analysis and extends it through the analysis of free-form text fields to create this input.

The Investigate stage shows the actual condition of data in existing sources and identifies and corrects data problems before they corrupt new systems. Investigation parses and analyzes free-form fields creating patterns to reveal field formats, counts unique values, and classifies, or assigns a business meaning to each occurrence of a value within a field. The Investigate stage also allows users to validate field contents in a specific column or domain. The actual full-volume data files must be investigated to assure that as many data anomalies as possible are identified. The Investigate stage involves performing one or more of the following functions:

- Parsing
- Classifying
- Creation of Word or Pattern Distributions

Specific objectives of the investigation process include, but are not limited to the following:

- Investigate data values and anomalies to be considered for special handling during design or development. Any recommended special handling is documented in a Business Requirements Document. Examples include:
  - Out of range, default (such as all nines in the telephone number) or unique values.
  - Free-form text that requires parsing to improve “addressability” to key components for optimal matching.
  - Values that do not match metadata labels and require new field definitions for matching and load format (such as DBA, C/O, Attention text, drivers license numbers, or other comment information in a name or address field).
  - Values that overlap adjacent fields and thus require a re-alignment of field content. For example, name information or city, state, and postal code that extend into street and mailing address fields.
  - Invalid formats that can cause conversion problems such as alphanumeric in character or numeric only fields.
  - Blank or missing data.
  - Special character and punctuation handling.
- Discover additional tokens (key words) to add to the classification table, such as name prefixes, name suffixes, street types, unit types, and business words.
- Verify the usefulness of data fields for the purpose of entity matching (record linkage). For example, does a Tax ID field contain primarily blank or invalid values, making it an unreliable field to use in the identification of a unique customer?
- Validate business requirements or assumptions about the data. Results can lead to changes in the business requirements and can assist in driving the technical solution.

The Investigate stage takes a single input, which can be a link from any database connector that is supported by IBM WebSphere DataStage, from a flat file or data set or from any processing stage. Inputs to the Investigate stage can be fixed length or variable.

Use of investigation (or data profiling) is critical to assess how data can be used for further steps of standardization, matching, and survivorship or consolidation. Data that is used in these additional steps is stored in two types of fields—*single-domain* and *multiple-domain*.

During the investigation process, it is important to document fields that require parsing and standardization, standardization only, or no standardization at all. You must also consider whether data content will be used for matching, survivorship, pass-through (static data) or whether it will not be carried forward into the new system or target database.

- Static fields
  - Can be used in the matching process.
  - Might require some standardization of format and default or out of range values.
  - Typically re-joined with re-engineered data at prior to matching or load formatting.

  Character Investigate is usually sufficient to evaluate.
- Single-domain fields
  - Categorized as either *Entity Identifiers* or *Entity Clarifiers*:
    - Entity Identifiers

      Examples include ZIP Code (U.S. postal code), Social Security Number (SSN), TIN, and telephone numbers. These can serve as critical match fields if their quality is high. These fields typically have a specific format (such as 9-digit numeric for SSN, or Alpha-Numeric-Alpha-Numeric-Alpha-Numeric for Canadian postal code).

      Character Discrete Investigate is commonly used, with the mask set to all *C*s.
    - Entity Clarifiers

      Examples include name prefix, gender, and marital status. This type of field usually has a limited number of known values.

      Character Discrete Investigate is commonly used, with the mask set to all *T*s.

– Typically used for a single purpose.
– Can exist in several common formats.
– No parsing is usually necessary.
– Can serve as strong blocking and match fields if quality is high.
– Standardize Stage usually only requires removal of special characters, creation of a common default value and format and concatenating data.

11. Multiple-domain fields
   – Typically these are large free-form fields such as multiple Address fields.
   – Comprised of multiple single-domain values, such as house number, street name, unit, and first name.
   – Intelligent parsing is required on these fields for optimal matching.
   – Can serve as good match fields and multiple-component blocking candidates.
   – Standardization requires removal of most special characters and strong data typing.
   – Occasionally contain business word interference, such as reference numbers, comments, and so forth.
   – Can contain additional relationships such as Attn, C/O, DBA, and Trustee.

   Word Investigate is commonly used for multi-domain fields.

We describe the Character Investigate and Word Investigate options in the following sections.

1.6.1 Character Investigate option

The Character Investigate option parses a single-domain field (one that contains one data element or token, such as SSN, telephone number, date, or ZIP code) to analyze and classify data. Character investigation provides you with the option of investigating multiple columns individually (Character Discrete) or integrated as one unit of data (Character Concatenate).

The investigation process generates a column frequency report that presents information about frequency values for a specific column.

- Character Discrete Investigate

  The Character Discrete Investigate analyzes multiple single-domain columns. This option allows you to investigate a large number of columns with little effort. A Character Discrete Investigate produces a column frequency report that treats each column as a separate token for frequency count and analysis.

- Character Concatenate Investigate

  The Character Concatenate Investigate option performs cross-column correlations between multiple columns to determine relationships. With this option, you select two or more columns from anywhere in the record (the columns do not have to be contiguous) to be investigated as a single data column. To create the pattern analysis, the tokens are concatenated with no spaces between the tokens.

The Character Investigate process generates a column frequency report that presents information about frequency values for a specific column. A pattern report is prepared for all types of investigations and displays the count, percentage of data that matches this pattern, the generated pattern, and sample data. This output can be presented in a wide range of formats to conform to standard reporting tools.

For character investigations, you use column masks to select the characters that are included in the frequency count or pattern analysis and the characters that are displayed as part of samples in the pattern report. You apply a mask symbol to *each character* in the selected columns. You can use the following mask characters:

- Mask *C*

  Displays the character and includes it in the frequency count and pattern analysis. Use the *C* column mask to inspect the values in your columns and to certify that false data does not appear in a column such as 99999 for a postal code or 111111111 for a national ID number.

- Mask *T*

  Displays the type of character[4] and includes the character in the frequency count and pattern analysis. Use the *T* column mask when you want to inspect the type of data in a character position, for example 9-digit telephone numbers such as *nnn-nnn-nnnn* or (*nnn*)-*nnn-nnnn*.

- Mask *X*

  Excludes the character in the frequency count or the pattern analysis; however, it includes it in the sample data. Use the *X* column mask to include data from the column in the sample but not as a token or part of the token for investigation.

  For example, you set up an Investigate job to analyze the first two characters of a ZIP code[5] to determine the frequency distribution based on a state (each state is defined by the first two characters of the ZIP code). You set the column mask for the ZIP code to *CCXXX*. The qsInvPattern column of the

[4] The character “n” for numeric, “a” for alpha, “(“ for parenthesis, “-” for hyphen, and “b” for blank.
[5] For details on the structure of a ZIP code, refer to:
http://en.wikipedia.org/wiki/ZIP_Code

pattern report (such as that shown in Figure 1-9 on page 27) displays only the first two characters. The frequency count is based on the number of records in the file that start with the first two characters of the ZIP code. In the qsInvSample column of the pattern report (such as that shown in Figure 1-9 on page 27), you see all five characters of the ZIP code.

You can also use the *X* column mask with the Character Concatenate option to specify one or more columns to appear as part of the sample only. From the previous example, you can also select the state columns setting the column mask to *X* for all characters. The pattern report displays the frequency counts for the first two characters of the ZIP code and the full five characters of the ZIP code along with the state in the sample column.

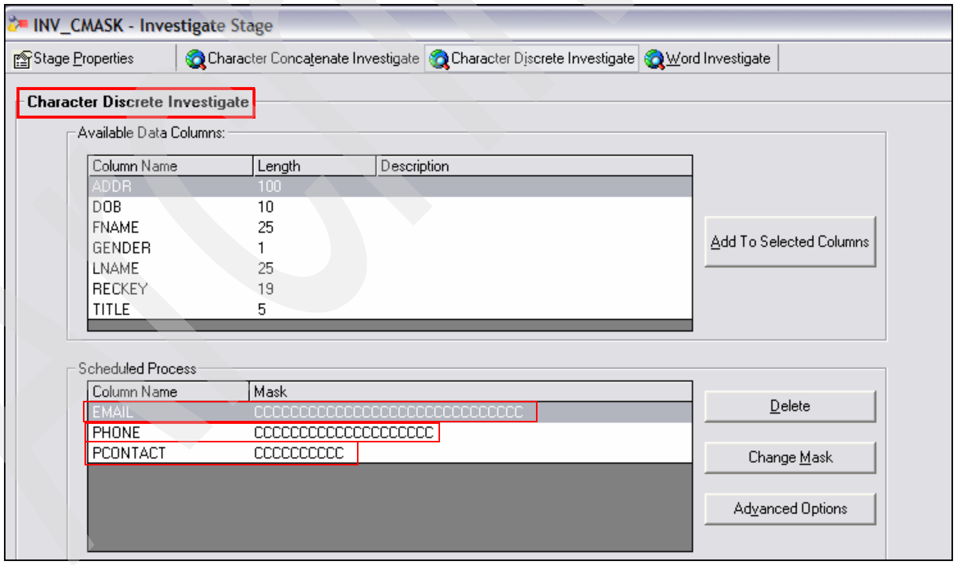
Figure 1-8 through Figure 1-13 on page 30 show examples of Character Discrete and Character Concatenate options:

- Character Discrete option with *C* mask

  Figure 1-8 shows the specification of the *C* mask for three columns:

  – EMAIL
  – PHONE
  – PCONTACT (preferred method of contact)

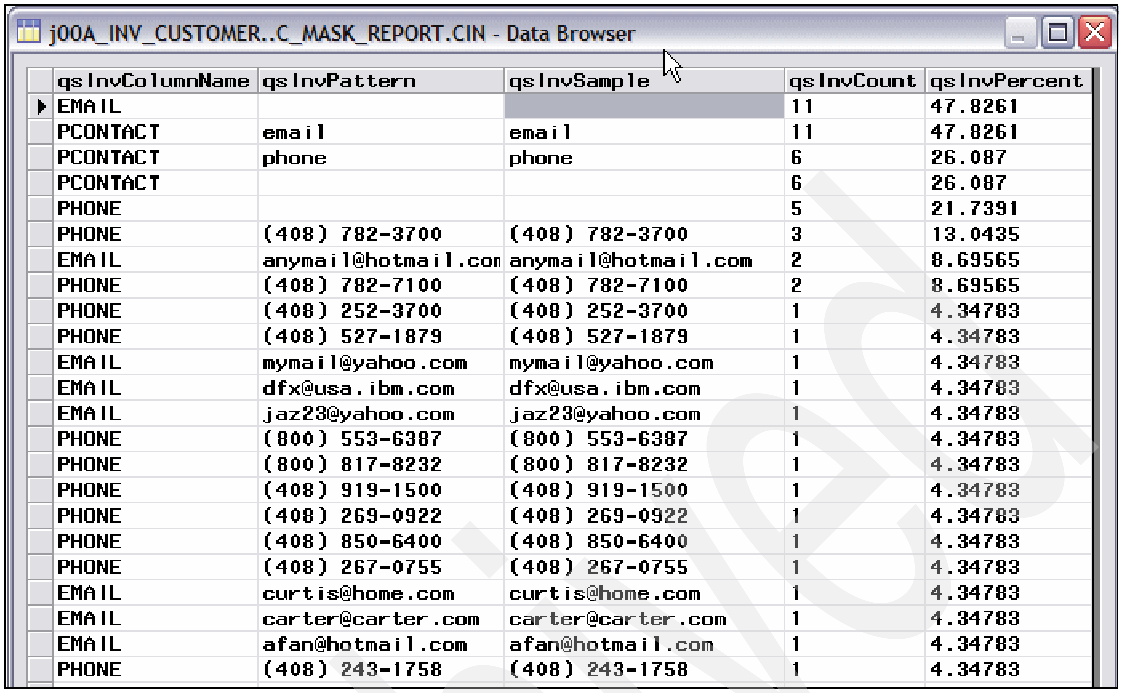
  Figure 1-9 on page 27 shows the corresponding report for this specification.



Column Name	Length	Description
ADDR	100	
DOB	10	
FNAME	25	
GENDER	1	
LNAME	25	
RECKEY	19	
TITLE	5	

Column Name	Mask
EMAIL	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
PHONE	CCCCCCCCCCCCCCCCCCCC
PCONTACT	CCCCCCCCCC

Figure 1-8 Specification of the C mask for discrete columns



j00A_INV_CUSTOMER..C_MASK_REPORT.CIN - Data Browser

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
EMAIL			11	47.8261
PCONTACT	email	email	11	47.8261
PCONTACT	phone	phone	6	26.087
PCONTACT			6	26.087
PHONE			5	21.7391
PHONE	(408) 782-3700	(408) 782-3700	3	13.0435
EMAIL	anymail@hotmail.com	anymail@hotmail.com	2	8.69565
PHONE	(408) 782-7100	(408) 782-7100	2	8.69565
PHONE	(408) 252-3700	(408) 252-3700	1	4.34783
PHONE	(408) 527-1879	(408) 527-1879	1	4.34783
EMAIL	mymail@yahoo.com	mymail@yahoo.com	1	4.34783
EMAIL	dfx@usa.ibm.com	dfx@usa.ibm.com	1	4.34783
EMAIL	jaz23@yahoo.com	jaz23@yahoo.com	1	4.34783
PHONE	(800) 553-6387	(800) 553-6387	1	4.34783
PHONE	(800) 817-8232	(800) 817-8232	1	4.34783
PHONE	(408) 919-1500	(408) 919-1500	1	4.34783
PHONE	(408) 269-0922	(408) 269-0922	1	4.34783
PHONE	(408) 850-6400	(408) 850-6400	1	4.34783
PHONE	(408) 267-0755	(408) 267-0755	1	4.34783
EMAIL	curtis@home.com	curtis@home.com	1	4.34783
EMAIL	carter@carter.com	carter@carter.com	1	4.34783
EMAIL	afan@hotmail.com	afan@hotmail.com	1	4.34783
PHONE	(408) 243-1758	(408) 243-1758	1	4.34783

Figure 1-9 Frequency distribution report for discrete columns and the C mask

A brief description of the report shown in Figure 1-9 follows:

- *qsInvColumnName* identifies the names of the column that is investigated.
- *qsInvPattern* displays the character and includes the character in the frequency count and pattern analysis.
- *qsInvSample* shows one or more samples of the content of this column. The number to be displayed is configurable. In Figure 1-9, the maximum number of samples requested was one.

> **Note:** You can configure Investigate to specify the number of samples of source data to include in the report for each pattern. You can also skip patterns that appear infrequently by specifying a frequency cutoff level. Patterns with frequencies under the specified frequency cutoff level do not appear in the report. This configuration is not shown here.

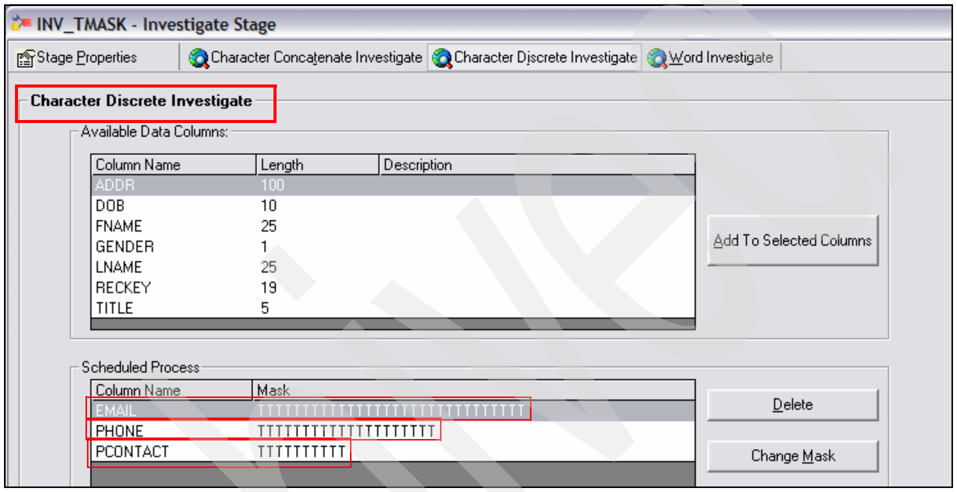
- *qsInvCount* shows the actual number of occurrences of the value in the qsInvPattern column.
- *qsInvPercent* shows the percentage occurrences of the value in the qsInvPattern column to the total number of records on this file.

- Character Discrete option with *T* mask

  Figure 1-10 shows the specification of the *T* mask for the same three columns:

  – EMAIL
  – PHONE
  – PCONTACT

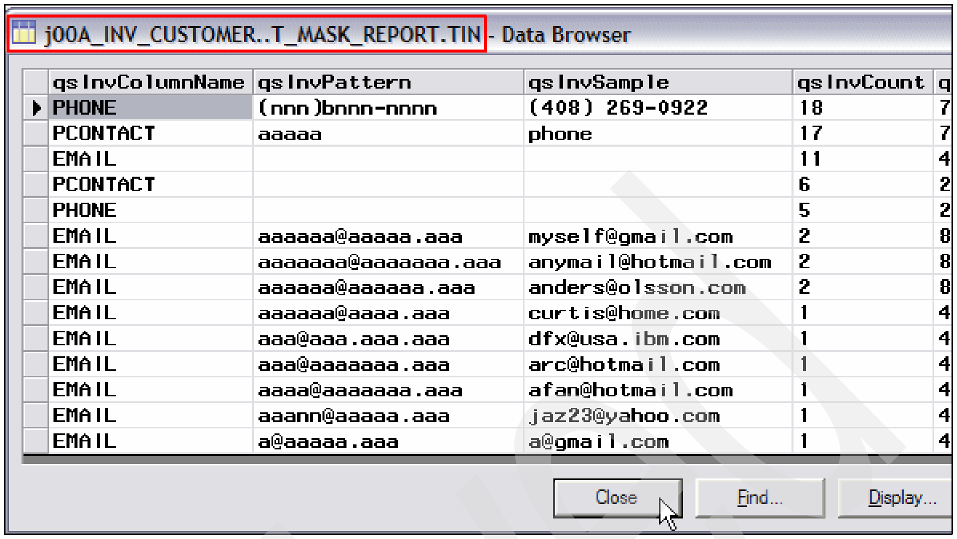
  Figure 1-11 on page 29 shows the corresponding report for this specification.


INV_TMASK - Investigate Stage
Stage Properties
Character Concatenate Investigate
Character Discrete Investigate
Word Investigate
Character Discrete Investigate
Available Data Columns:
Add To Selected Columns
Scheduled Process
Delete
Change Mask

Column Name	Length	Description
ADDR	100	
DOB	10	
FNAME	25	
GENDER	1	
LNAME	25	
RECKEY	19	
TITLE	5	

Column Name	Mask
EMAIL	TTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
PHONE	TTTTTTTTTTTTTTTTTTTT
PCONTACT	TTTTTTTTTT

Figure 1-10 Specification of the T mask for discrete columns



qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	q
PHONE	(nnn)bnnn-nnnn	(408) 269-0922	18	7
PCONTACT	aaaaa	phone	17	7
EMAIL			11	4
PCONTACT			6	2
PHONE			5	2
EMAIL	aaaaaa@aaaaa.aaa	myself@gmail.com	2	8
EMAIL	aaaaaaa@aaaaaaa.aaa	anymail@hotmail.com	2	8
EMAIL	aaaaaa@aaaaaa.aaa	anders@olsson.com	2	8
EMAIL	aaaaaa@aaaa.aaa	curtis@home.com	1	4
EMAIL	aaa@aaa.aaa.aaa	dfx@usa.ibm.com	1	4
EMAIL	aaa@aaaaaaa.aaa	arc@hotmail.com	1	4
EMAIL	aaaa@aaaaaaa.aaa	afan@hotmail.com	1	4
EMAIL	aaann@aaaaa.aaa	jaz23@yahoo.com	1	4
EMAIL	a@aaaaa.aaa	a@gmail.com	1	4

Figure 1-11 Frequency distribution with discrete columns and the T mask

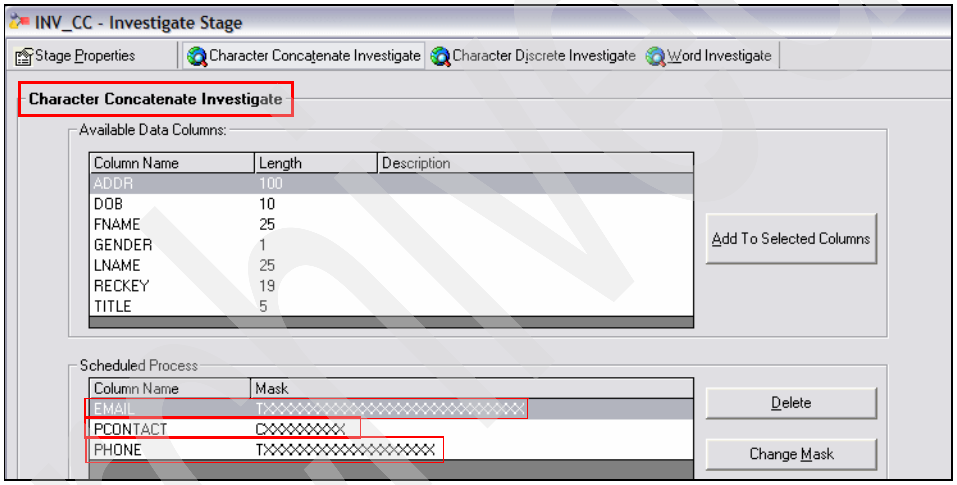
The only difference in Figure 1-11 from the explanation of the report shown in Figure 1-9 on page 27 is that column qsInvPattern displays the *type of character*. As before, this character is included in the frequency count and pattern analysis.

- Character Concatenate option with mix of *C*, *T*, and *X* masks

  Figure 1-12 shows the specification of a mix of the *C*, *T*, and *X* masks for the same three columns:

  – EMAIL
  – PHONE
  – PCONTACT

  The objective of this investigation is to determine whether the preferred method of contact (PCONTACT column) when specified (not null or blank) has the corresponding contact information available. For example, if the preferred contact is using the telephone, then the PHONE column cannot be blank or null. Figure 1-13 shows the corresponding report for this specification.


INV_CC - Investigate Stage

Stage Properties | Character Concatenate Investigate | Character Discrete Investigate | Word Investigate

Character Concatenate Investigate

Available Data Columns:

Column Name	Length	Description
ADDR	100	
DOB	10	
FNAME	25	
GENDER	1	
LNAME	25	
RECKEY	19	
TITLE	5	

Add To Selected Columns

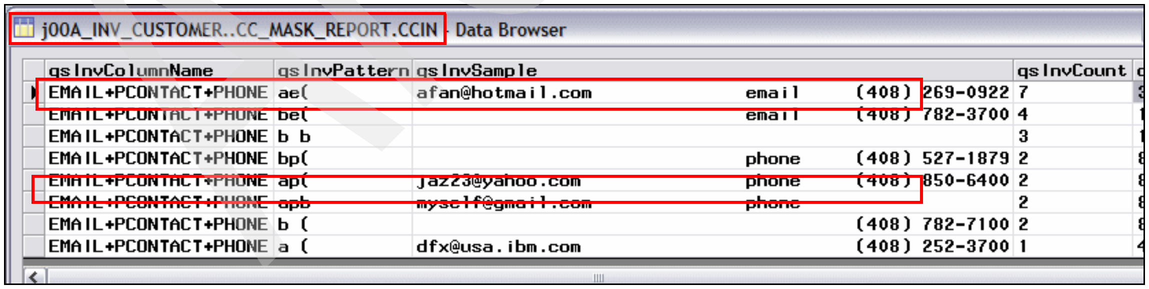
Scheduled Process

Column Name	Mask
EMAIL	TXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PCONTACT	CXXXXXXXXX
PHONE	TXXXXXXXXXXXXXXXXXXX

Delete

Change Mask

Figure 1-12 Specification of the T, C, and X masks for concatenated columns


j00A_INV_CUSTOMER..CC_MASK_REPORT.CCIN - Data Browser

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount
EMAIL+PCONTACT+PHONE	ae(	afan@hotmail.com email (408) 269-0922	7
EMAIL+PCONTACT+PHONE	be(	email (408) 782-3700	4
EMAIL+PCONTACT+PHONE	b b		3
EMAIL+PCONTACT+PHONE	bp(	phone (408) 527-1879	2
EMAIL+PCONTACT+PHONE	ap(	jaz23@yahoo.com phone (408) 850-6400	2
EMAIL+PCONTACT+PHONE	apb	myself@gmail.com phone	2
EMAIL+PCONTACT+PHONE	b (	(408) 782-7100	2
EMAIL+PCONTACT+PHONE	a (	dfx@usa.ibm.com (408) 252-3700	1

Figure 1-13 Frequency distribution report with the T, C, and X masks for concatenated columns

You can interpret the qsInvPattern column in report in Figure 1-13 on page 30 as follows. If the middle pattern is not blank, it indicates the following preferred contact information:

- If the value is *e*, it indicates that the preferred contact method is using e-mail. Therefore, you need to verify that the first character in the qsInvPattern column is an “a,” which indicates that the first character in the EMAIL column is an alphabetic character. We assume that this alphabetic character means that an e-mail address is present but that it does not indicate that the e-mail address is valid.
- If the value is *p*, it indicates that the preferred contact method is using a telephone. Therefore, you need to verify that the first character in the qsInvPattern column is a parenthesis “(”, which indicates that the first character in the PHONE column is a left parenthesis. We assume that this parenthesis means that a 9-digit telephone number corresponding to the general format “(*nnn*) *nnn nnnn*” is present but that it does not indicate that the telephone number is valid.

Note: Character Concatenate Investigate is particularly useful to investigate the output of Standardize to assess how effectively the rule sets used in Standardize worked on the data.

1.6.2 Word Investigate option

The Word Investigation option parses free-form data fields into individual tokens and analyzes them to create patterns. For example, to create the patterns in address data, the Word Investigation option uses a set of rules for classifying personal names, business names, and addresses. The Word Investigation option also provides frequency counts on the tokens.

The Investigate stage provides pre-built rule sets for investigating patterns on names and postal addresses for a number of different countries.[6] For example, for the U.S., the Investigate stage parses the following components:

- USPREP

  Name, address, and area data.
- USNAME

  Individual and organization names.
- USADDR

  Street and mailing addresses.

[6] For information about managing rule sets refer to *IBM WebSphere QualityStage Version 8 User Guide*, SC18-9922.

- USAREA

  City, state, ZIP code, and so on.

The test field “123 St. Virginia St.” is analyzed in the following way:

1. Field parsing would break the address into the individual tokens of 123, St., Virginia, and St.
2. Lexical analysis determines the business significance of each piece:
   a. 123 = number
   b. St. = street type
   c. Virginia = alpha
   d. St. = Street type
3. Context analysis identifies the various data structures and content as 123 St. Virginia, St.
   a. 123 = House number
   b. St. Virginia = Street address
   c. St. = Street type

When you specify Word Investigate, you select the rule by which you want the columns investigated and then select one or more columns to examine.

Word Investigate parses the free-form data column into individual elements or tokens, which is a word, a number, or a mixture separated by one or more spaces or special characters. The process compares each token with classified tokens in the Classification table for that rule set. (Example 1-1 on page 33 shows partial contents of the Classification table of the USPREP rule set.) If the token matches the word in the Classification table, Investigate assigns the class for that token to represent it in the pattern. For tokens that do not match any classified token, Investigate examines the pattern and assigns classes as shown in the Table 1-1 on page 37.

Note: You can gain valuable insight by browsing the Classification tables to determine the classification codes, as well as the different literals supported such as ZQMIXAZQ and ZQNAMEZQ.

Example 1-1 Classification table for the USPREP rule set

```
..........
;------------------------------------------------------------------------------
; USPREP Classification Table
;------------------------------------------------------------------------------
; Classification Legend
;------------------------------------------------------------------------------
; B - Box Types
; C - Common Words (AND, OF, THE, etc.)
; D - Directionals
; E - Name Indicators (MR, MRS, JR, SR, CORP, INC, etc.)
; F - First Names
; I - Initials (E, N, S, W are not included)
; M - Tokens with multiple semantics (CO, FL, DR, etc.)
; S - States (includes military and territories)
; T - Street Types (includes rural routes and Puerto Rican urbanizations)
; U - Unit Types (includes floors and buildings)
;------------------------------------------------------------------------------
; Domain Masks
;------------------------------------------------------------------------------
; A - ADDRESS (Reserved Class)
; N - NAME    (Reserved Class)
; R - AREA    (Reserved Class)
;------------------------------------------------------------------------------
; Meta Data Delimiters
;------------------------------------------------------------------------------
; For all delimiters, input overrides are applied first.
; ZQPUTNZQ: automatically defaults the entire field to the Name Domain.
; ZQMIXNZQ: field overrides and field modifications are applied, then it checks
;           the field for name, address, and area data (in that order).
;           Any information that is not assigned a domain is defaulted to Name.
; ZQNAMEZQ: field overrides and field modifications are applied, then it checks
;           for common Name patterns.  If not found, it checks for Address and
;           Area patterns.  If not found, the field is defaulted to Name.
; ZQPUTAZQ: automatically defaults the entire field to the Address Domain.
; ZQMIXAZQ: field overrides and field modifications are applied, then it checks
;           the field for name, address, and area data (in that order).
;           Any information that is not assigned a domain is defaulted to Addres
; ZQADDRZQ: field overrides and field modifications are applied, then it checks
;           for common Address patterns.  If not found, it checks for Name and
;           Area patterns.  If not found, the field is defaulted to Address.
; ZQPUTRZQ: automatically defaults the entire field to the Area Domain.
; ZQMIXRZQ: field overrides and field modifications are applied, then it checks
;           the field for name, address, and area data (in that order).
;           Any information that is not assigned a domain is defaulted to Area.
; ZQAREAZQ: field overrides and field modifications are applied, then it checks
;           for common Area patterns.  If not found, it checks for Name and
;           Address patterns.  If not found, the field is defaulted to Area.
;-------------------------------------------------------------------------------
ZQADDRZQ                 DELIMITER                A
ZQMIXAZQ                 DELIMITER                A
ZQPUTAZQ                 DELIMITER                A
ZQNAMEZQ                 DELIMITER                N
ZQMIXNZQ                 DELIMITER                N
ZQPUTNZQ                 DELIMITER                N
ZQAREAZQ                 DELIMITER                R
ZQMIXRZQ                 DELIMITER                R
ZQPUTRZQ                 DELIMITER                R
;-------------------------------------------------------------------------------
CALLER                   "PO BOX"                 B
DRAW                     "PO BOX"                 B
.......
PODRAWER                 "PO BOX"                 B
POST                     "PO BOX"                 B
APARTADO                 APARTADO                 B
BO                       BOX                      B
.......
BOX                      B
BXO                      BOX                      B
BUZON                    BUZON                    B
MSC                      MSC                      B
PMB                      PMB                      B
AND                      AND                      C
OF                       OF                       C
THE                      THE                      C
E                        E                        D
EAST                     E                        D
.......
NO                       N                        D
NORTE                    N                        D
.......
NOROESTE                 NW                       D
NORTHWEST                NW                       D 850
NW                       NW                       D
S                        S                        D
........
SUR                      S                        D
SE                       SE                       D
SOUTHEAST                SE                       D 850
```

```
SURESTE                  SE                       D
SOUTHWEST                SW                       D 850
SUROESTE                 SW                       D
SW                       SW                       D
OESTE                    W                        D
W                        W                        D
WEST                     W                        D
II                       GENERATION               E
.......
SNR                      GENERATION               E
SR                       GENERATION               E
ASOC                     ORGTYPE                  E
ASOCS                    ORGTYPE                  E
........
CLINIC                   ORGTYPE                  E
........
INCORPORATD              ORGTYPE                  E
........
UNIVERSITY               ORGTYPE                  E
........
ADMIRAL                  PREFIX                   E
ATTORNEY                 PREFIX                   E
.......
MISS                     PREFIX                   E
MR                       PREFIX                   E
MRS                      PREFIX                   E
.......
SHERIFF                  PREFIX                   E
.......
ATTENTION                QUALIFIER                E
.......
FOR                      QUALIFIER                E
AMUL                     SUFFIX                   E
.......
CUSTODIAN                SUFFIX                   E
.......
TTEES                    SUFFIX                   E
AARON                    AARON                    F
.......
CARLOS                   CARLOS                   F
CARMELA                  CARMELA                  F
.......
STEPHEN                  STEPHEN                  F
STEVE                    STEVE                    F
.......
```

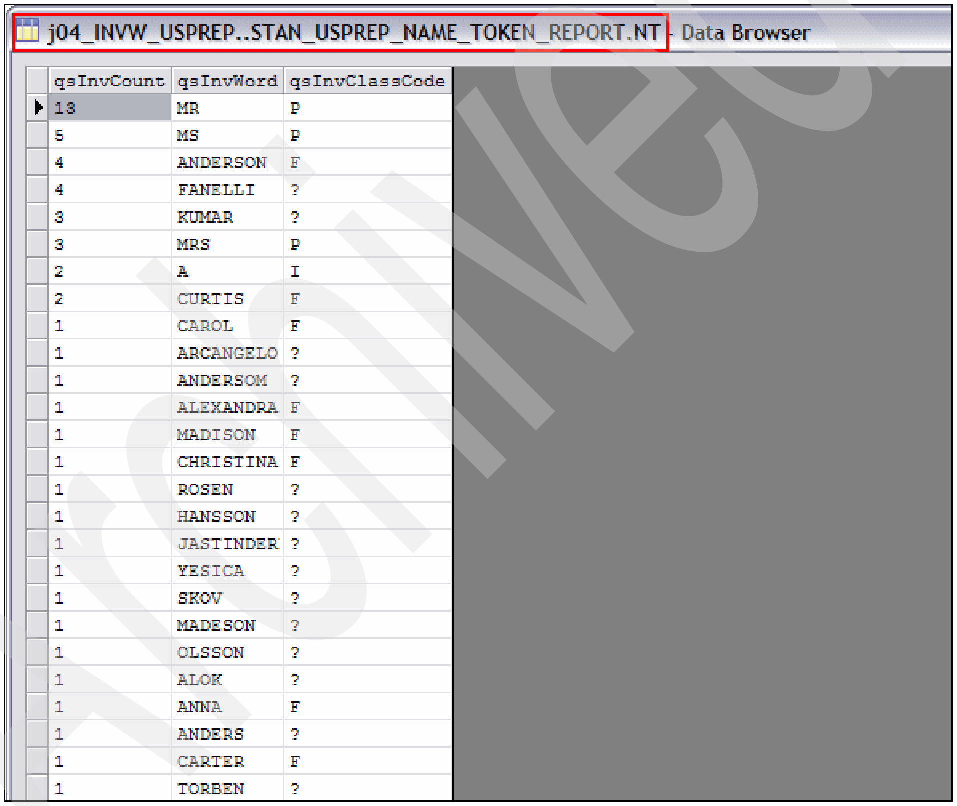
```
YVONNE                  YVONNE                  F
ZACHARY                 ZACHARY                 F
.......
J                       J                       I
K                       K                       I
.......
Z                       Z                       I
.......
CENTRE                  CTR                     M
CNTER                   CTR                     M
CNTR                    CTR                     M
CTER                    CTR                     M
.......
AZ                      AZ                      S
CA                      CA                      S
.......
WV                      WV                      S
CR                      "COUNTY ROAD"           T
TSR                     "TOWNSHIP ROAD"         T
.......
BOULV                   BLVD                    T
BEND                    BND                     T
.......
WLS                     WLS                     T
CROSSING                XING                    T 850
CRSSING                 XING                    T 850
.......
BLD                     BLDG                    U
BLDG                    BLDG                    U
.......
WAREHOUSE               WHS                     U 850
WHS                     WHS                     U
ONE                     1                       ^
TEN                     10                      ^
.......
FIVE                    5                       ^
NINE                    9                       ^
```

Table 1-1 Class and description associated with a token

Class	Description
^	Numeric containing all digits, such as 1234
?	Unknown token containing one or more words, such as CHERRY HILL
>	Leading numeric containing numbers followed by one or more letters, such as 123A
<	Leading alpha containing letters followed by one or more numbers, such as A3
@	Complex mix containing alpha and numeric characters that do not fit into either of the above classes, such as 123A45 and ABC345TR
0	Null
-	Hyphen
\	Slash
&	Ampersand
#	Number sign
(	Left parenthesis
)	Right parenthesis
~	Special containing special characters that are not generally found in addresses, such as !, \, @, ~, %, and so forth

Word Investigate generates two reports as follows:

- Frequency word as shown in Figure 1-14 presents the most commonly occurring word values from columns that are analyzed for patterns.
  - *qsInvCount* indicates the number of times this token was encountered across the entire input data.
  - *qsInvWord* identifies the individual token or word value that is found inside the selected input columns.
  - *qsInvClassCode* identifies the classification of the token that is based on the selected rule set classification table. Unclassified tokens, if selected, get a question mark "?" for alpha or a carat "^" for numeric.

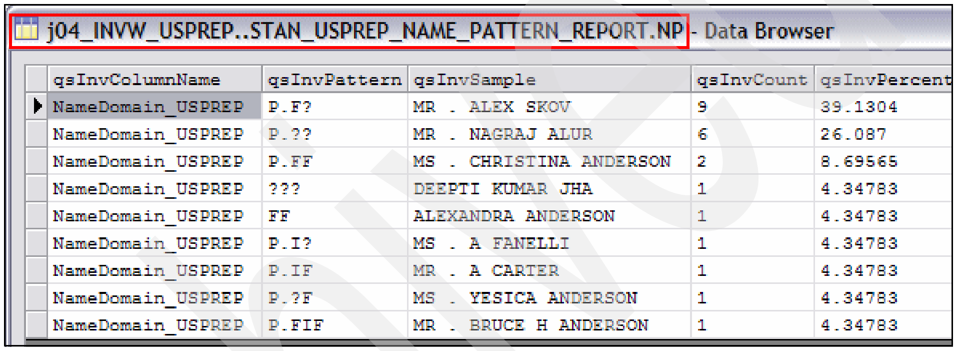

j04_INVW_USPREP..STAN_USPREP_NAME_TOKEN_REPORT.NT - Data Browser

qsInvCount	qsInvWord	qsInvClassCode
13	MR	P
5	MS	P
4	ANDERSON	F
4	FANELLI	?
3	KUMAR	?
3	MRS	P
2	A	I
2	CURTIS	F
1	CAROL	F
1	ARCANGELO	?
1	ANDERSOM	?
1	ALEXANDRA	F
1	MADISON	F
1	CHRISTINA	F
1	ROSEN	?
1	HANSSON	?
1	JASTINDER	?
1	YESICA	?
1	SKOV	?
1	MADESON	?
1	OLSSON	?
1	ALOK	?
1	ANNA	F
1	ANDERS	?
1	CARTER	F
1	TORBEN	?

Figure 1-14 Token (frequency word) report

- Frequency pattern as shown in Figure 1-15 is similar to the description corresponding to Figure 1-11 on page 29.

> **Note:** As mentioned earlier with Character Investigate, you can also configure Word Investigate to specify the number of samples of source data to include in the report for each pattern. You can also skip patterns that appear infrequently by specifying a frequency cutoff level. Patterns with frequencies under the specified frequency cutoff level do not appear in the report. You can also choose how tokens appear in the report—for example, strip out spaces between unclassified words (concatenating them into one word) or include unclassified numeric tokens.


j04_INVW_USPREP..STAN_USPREP_NAME_PATTERN_REPORT.NP - Data Browser

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
NameDomain_USPREP	P.F?	MR . ALEX SKOV	9	39.1304
NameDomain_USPREP	P.??	MR . NAGRAJ ALUR	6	26.087
NameDomain_USPREP	P.FF	MS . CHRISTINA ANDERSON	2	8.69565
NameDomain_USPREP	???	DEEPTI KUMAR JHA	1	4.34783
NameDomain_USPREP	FF	ALEXANDRA ANDERSON	1	4.34783
NameDomain_USPREP	P.I?	MS . A FANELLI	1	4.34783
NameDomain_USPREP	P.IF	MR . A CARTER	1	4.34783
NameDomain_USPREP	P.?F	MS . YESICA ANDERSON	1	4.34783
NameDomain_USPREP	P.FIF	MR . BRUCE H ANDERSON	1	4.34783

Figure 1-15 Pattern (frequency pattern) report

> **Attention:** 1.10, “Mailing list scenario” on page 117 provides usage examples of both word and character investigations.

1.7 Standardize stage

Standardizing data involves moving free-form data (columns that contain more than one data entry) into fixed columns and manipulating data to conform to standard conventions. The process identifies and corrects invalid values, standardizes spelling formats and abbreviations, and validates the format and content of the data.

The Standardize stage builds on the interpretation of the data during the Investigate stage. The Standardize stage uses the same prebuilt tables and rule sets that the Investigate stage used to investigate the data to standardize the data. Standardize reformats data from multiple systems and creates a consistent data presentation with fixed and discrete columns, according to your company requirements.

The Standardize stage processes the data with the following outcome:

- Creates fixed-column, addressable data
- Facilitates effective matching
- Enables output formatting

The Standardize stage uses the data content and placement within the record context to determine the meaning of each data element. To correctly parse and identify each element or token and place them in the appropriate column in the output file, the Standardize stage uses rule sets to support specific conditioning and standardization requirements. These can be standard rule sets that are designed to comply with the name (individual and business) and address conventions of a specific country. Alternatively, they can be custom rule sets to focus on challenges such as product or part descriptions, consistent metadata definitions, insurance claim data, or other industry specific challenges. The Standardize rule sets can assimilate the data and append additional information from the input data, such as gender. 1.7.1, “Standardize rule sets” on page 42 provides further details on the Standardize stage rule sets.

These rule sets are the same as those used in the Investigate stage. You can run the rules as they are shipped, or you can customize them to process obscure data not covered by the standard rule sets. You can also create your own custom rule sets from scratch.

Standardize ensures that each data type has the same content and format. Standardized data is important for the following reasons:

- Effectively matches data
- Facilitates a consistent format for the output data

The Standardize stage parses free-form and fixed-format columns into single-domain columns to create a consistent representation of the input data.

- Free-form columns contain alphanumeric information of any length as long as it is less than or equal to the maximum column length defined for that column.
- Fixed-format columns contain only one specific type of information, such as only numeric, character, or alphanumeric, and have a specific format.

The following rules are examples of the types of rules that the Standardize stage might support depending on the rule set in use:

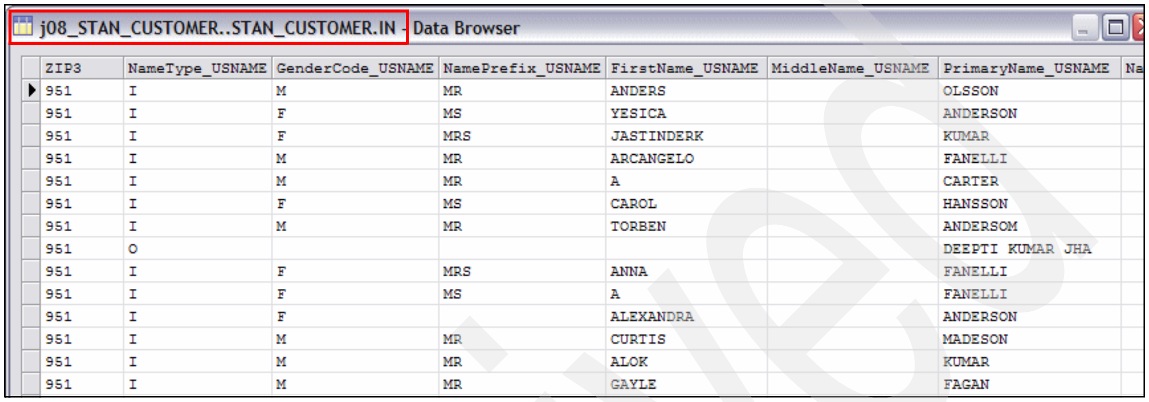
- Assign data to its appropriate metadata fields. Standardize ensures that the data within a specific field is being used for the business purpose defined in the metadata. For example, credit records might have a driver’s license number in the address line 1 field and a customer's address in the address line 2 field. To synchronize data with its appropriate metadata field, the driver's license number can be moved to a separate field for the driver's licenses.

- Decompose free-form fields into single component fields. For example, the customer's address can be decomposed into House number, Street name, PO Box, Rural Route, and other smaller component fields.
- Identify new data fields based on the underlying data. New fields, that do not exist on input, such as Gender Flag, Individual/Business Record Indicator, or Nickname, can be populated by the application, based on table or file look-ups.
- Break up records storing multiple entities. It might be necessary to create a separate record for each person or entity that is represented on a single input record (such as joint accounts). A separate record allows for a more complete linkage of all entities in the input files.
- Exclude records that do not meet minimum criteria. Based on defined business rules, the application can be required to exclude or reject records that do not meet basic requirements (for example, records that do not contain a name or address).

The Standardize stage takes a single input, which can be a link from any database connector supported by IBM WebSphere DataStage, a flat file or data set, or any processing stage. It is not necessary to restrict the data to fixed-length columns. Standardize will accept all basic data types (non-vector or non-aggregate) other than binary.

The Standardize stage has only one output link. This link can send the raw input and the standardized output to any other stage. The Standardize stage creates multiple columns that you can send along with the input columns to the output link. Any columns from the original input can be written to the output along with additional data created by the Standardize stage based on the input data (such as a SOUNDEX phonetic or NYSIIS codes). The Match stage and other stages can use the output from the Standardize stage—you can use any of the additional data for blocking and matching columns in Match stages (more on this in 1.8, "Match stage" on page 82).

Figure 1-16 shows the output of the Standardize stage using the domain-specific USNAME rule set. Some of the columns added by the Standardize stage shown in Figure 1-16 include NameType_USNAME, GenderCode_USNAME, NamePrefix_USNAME, FirstName_USNAME, MiddleName_USNAME, and PrimaryName_USNAME. The relevant tokens from the input file are moved to the appropriate domain column.


j08_STAN_CUSTOMER..STAN_CUSTOMER.IN - Data Browser

ZIP3	NameType_USNAME	GenderCode_USNAME	NamePrefix_USNAME	FirstName_USNAME	MiddleName_USNAME	PrimaryName_USNAME	Na
951	I	M	MR	ANDERS		OLSSON	
951	I	F	MS	YESICA		ANDERSON	
951	I	F	MRS	JASTINDERK		KUMAR	
951	I	M	MR	ARCANGELO		FANELLI	
951	I	M	MR	A		CARTER	
951	I	F	MS	CAROL		HANSSON	
951	I	M	MR	TORBEN		ANDERSOM	
951	O					DEEPTI KUMAR JHA	
951	I	F	MRS	ANNA		FANELLI	
951	I	F	MS	A		FANELLI	
951	I	F		ALEXANDRA		ANDERSON	
951	I	M	MR	CURTIS		MADESON	
951	I	M	MR	ALOK		KUMAR	
951	I	M	MR	GAYLE		FAGAN	

Figure 1-16 Partial list of columns added by the Standardize stage using the domain-specific USNAME rule set

1.7.1 Standardize rule sets

The Standardize stage has three categories of rule sets:

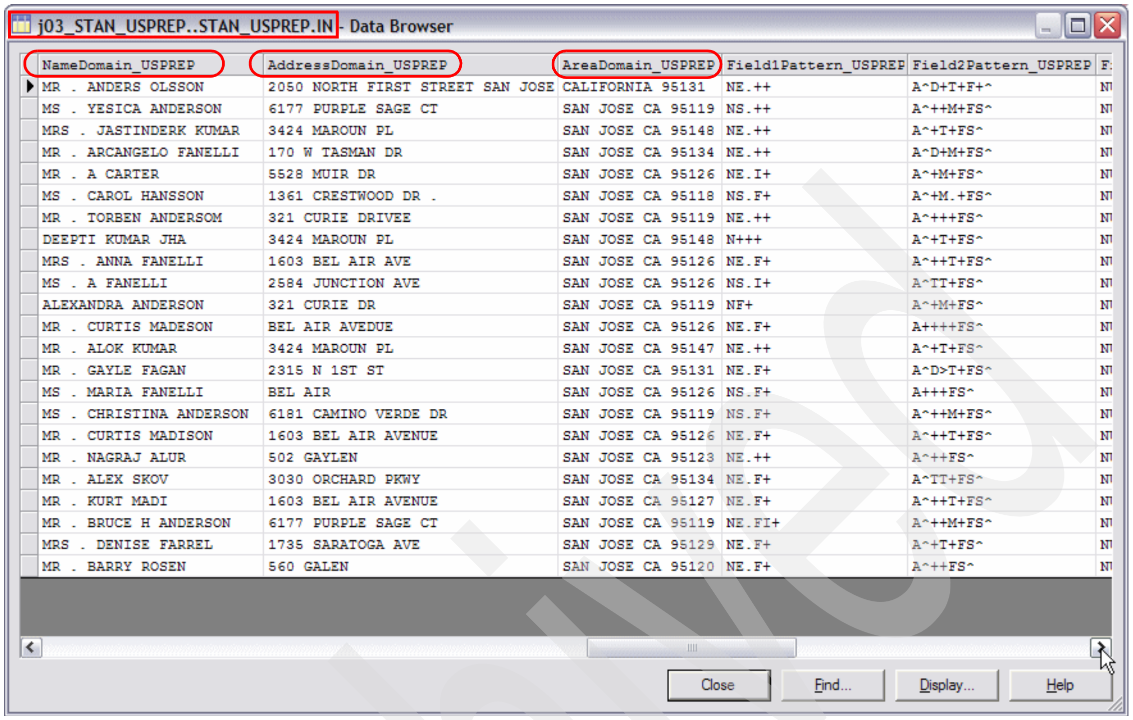
- ► Domain Pre-processor

  For a specific country, the Standardize stage identifies and assigns a data domain to the name, address, and area columns in each record. You can use the output from this file as the input to the country-appropriate Domain-Specific rule sets.

  There is one rule set for each country.

  These rule sets evaluate the mixed-domain input from a file for a specific country. Domain-preprocessor rule sets follow a naming convention that starts with a country abbreviation and ends with prep (an abbreviation for preprocessor) such as USPREP for the U.S. and GBPREP for Great Britain.

  Attention: These rule sets do *not* perform standardization but parse the columns in each record and each token into one of the appropriate domain-specific column sets, which are Name, Area, or Address as shown in Figure 1-17 on page 43.



NameDomain_USPREP	AddressDomain_USPREP	AreaDomain_USPREP	Field1Pattern_USPREP	Field2Pattern_USPREP	F
MR . ANDERS OLSSON	2050 NORTH FIRST STREET SAN JOSE	CALIFORNIA 95131	NE.++	A^D+T+F+^	N
MS . YESICA ANDERSON	6177 PURPLE SAGE CT	SAN JOSE CA 95119	NS.++	A^++M+FS^	N
MRS . JASTINDERK KUMAR	3424 MAROUN PL	SAN JOSE CA 95148	NE.++	A^+T+FS^	N
MR . ARCANGELO FANELLI	170 W TASMAN DR	SAN JOSE CA 95134	NE.++	A^D+M+FS^	N
MR . A CARTER	5528 MUIR DR	SAN JOSE CA 95126	NE.I+	A^+M+FS^	N
MS . CAROL HANSSON	1361 CRESTWOOD DR .	SAN JOSE CA 95118	NS.F+	A^+M.+FS^	N
MR . TORBEN ANDERSOM	321 CURIE DRIVEE	SAN JOSE CA 95119	NE.++	A^+++FS^	N
DEEPTI KUMAR JHA	3424 MAROUN PL	SAN JOSE CA 95148	N+++	A^+T+FS^	N
MRS . ANNA FANELLI	1603 BEL AIR AVE	SAN JOSE CA 95126	NE.F+	A^++T+FS^	N
MS . A FANELLI	2584 JUNCTION AVE	SAN JOSE CA 95126	NS.I+	A^TT+FS^	N
ALEXANDRA ANDERSON	321 CURIE DR	SAN JOSE CA 95119	NF+	A^+M+FS^	N
MR . CURTIS MADESON	BEL AIR AVEDUE	SAN JOSE CA 95126	NE.F+	A++++FS^	N
MR . ALOK KUMAR	3424 MAROUN PL	SAN JOSE CA 95147	NE.++	A^+T+FS^	N
MR . GAYLE FAGAN	2315 N 1ST ST	SAN JOSE CA 95131	NE.F+	A^D>T+FS^	N
MS . MARIA FANELLI	BEL AIR	SAN JOSE CA 95126	NS.F+	A+++FS^	N
MS . CHRISTINA ANDERSON	6181 CAMINO VERDE DR	SAN JOSE CA 95119	NS.F+	A^++M+FS^	N
MR . CURTIS MADISON	1603 BEL AIR AVENUE	SAN JOSE CA 95126	NE.F+	A^++T+FS^	N
MR . NAGRAJ ALUR	502 GAYLEN	SAN JOSE CA 95123	NE.++	A^++FS^	N
MR . ALEX SKOV	3030 ORCHARD PKWY	SAN JOSE CA 95134	NE.F+	A^TT+FS^	N
MR . KURT MADI	1603 BEL AIR AVENUE	SAN JOSE CA 95127	NE.F+	A^++T+FS^	N
MR . BRUCE H ANDERSON	6177 PURPLE SAGE CT	SAN JOSE CA 95119	NE.FI+	A^++M+FS^	N
MRS . DENISE FARREL	1735 SARATOGA AVE	SAN JOSE CA 95129	NE.F+	A^+T+FS^	N
MR . BARRY ROSEN	560 GALEN	SAN JOSE CA 95120	NE.F+	A^++FS^	N

Figure 1-17 Domain pre-processor rule set USPREP output

Because input files are rarely domain-specific, these rule sets are critical when preparing a file for standardization. Columns can contain data that does not match their metadata description as shown in Table 1-2.

Table 1-2 Metadata description versus data content

Metadata label	Data content
NAME1	John Doe
NAME2	123 Main Street Apt. 456
ADDR1	C/O Mary Doe
ADDR2	Boston, MA 02111

Column sets and metadata labels do not necessarily provide hard information about data content as shown in Table 1-2, where the name (C/O Mary Doe) is included in the ADDR1 column. Preprocessing categorizes the input data into domain-specific column sets of Name, Address, and Area as shown in

Table 1-3. It shows the tokens from the various columns (NAME1, NAME2, ADDR1, and ADDR2) parsed and moved into the domain-specific columns of NAME (John Doe C/O Mary Doe), ADDRESS (123 Main Street Apt. 456), and AREA (Boston, MA 02111).

Table 1-3 Domain specific columns and their data content after preprocessing

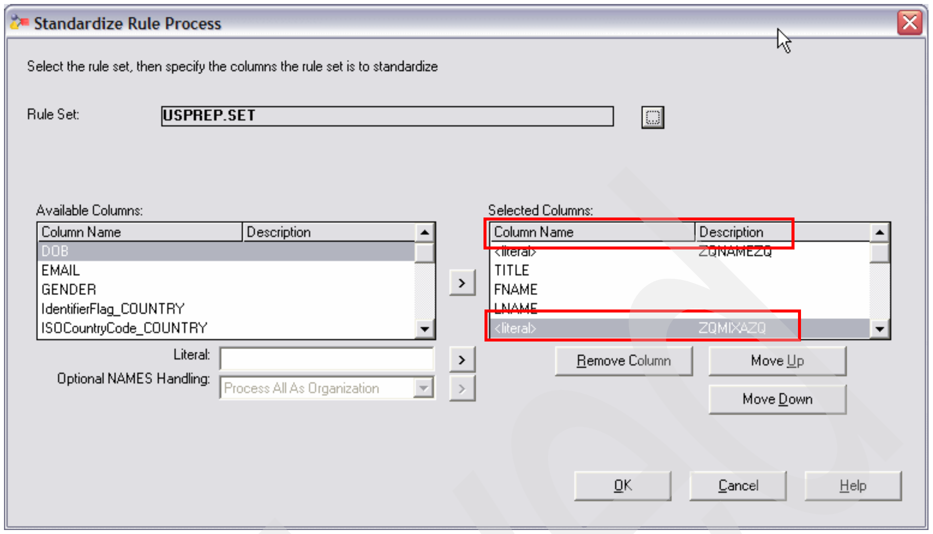
Metadata label	Data content
NAME	John Doe C/O Mary Doe
ADDRESS	123 Main Street Apt. 456
AREA	Boston, MA 02111

Note: Conventions used in name and address vary from one country to the next so the domain-preprocessor is configured for a single country.

The domain-preprocessor rule sets do not assume a data domain with a column position. You need to delimit every column or group of columns with literals as shown in the Standardize Rule Process windows in Figure 1-18 on page 45 and Figure 1-19 on page 45. These figures show the use of delimiters ZQNAMEZQ, ZQADDRZQ, and ZQMIXAZQ.

Important: The delimiter indicates the type of data that you are expecting to find in the column, based on metadata descriptions, investigation results (from the Investigate stage), or an informed guess.

Example 1-1 on page 33 shows the list of possible delimiters available with the USPREP rule set. They begin and end with the characters *ZQ*, such as ZQNAMEZQ, ZQADDRZQ, ZQAREAZQ, and ZQMIXAZQ.


Standardize Rule Process

Select the rule set, then specify the columns the rule set is to standardize

Rule Set: USPREP.SET

Available Columns:

Column Name	Description
DOB	
EMAIL	
GENDER	
IdentifierFlag_COUNTRY	
ISOCountryCode_COUNTRY	

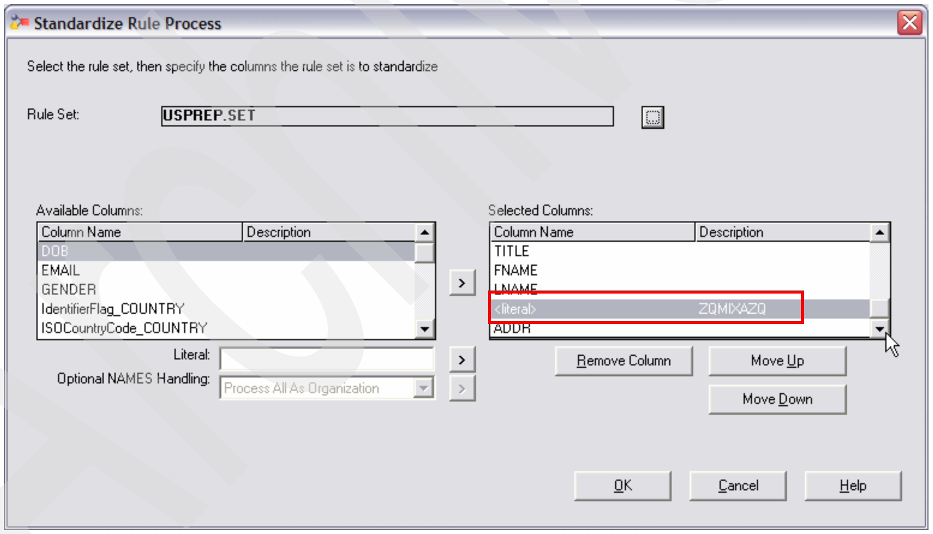
Selected Columns:

Column Name	Description
<literal>	ZQNAMEZQ
TITLE	
FNAME	
LNAME	
<literal>	ZQMIXAZQ

Literal:

Optional NAMES Handling: Process All As Organization

Figure 1-18 Specifying delimiters for columns 1/2


Standardize Rule Process

Select the rule set, then specify the columns the rule set is to standardize

Rule Set: USPREP.SET

Available Columns:

Column Name	Description
DOB	
EMAIL	
GENDER	
IdentifierFlag_COUNTRY	
ISOCountryCode_COUNTRY	

Selected Columns:

Column Name	Description
TITLE	
FNAME	
LNAME	
<literal>	ZQMIXAZQ
ADDR	

Literal:

Optional NAMES Handling: Process All As Organization

Figure 1-19 Specifying delimiters for columns 2/2

- Domain Specific

  These rule sets evaluate the domain-specific input from a file for a specific country. There are three domain-specific rules sets for each country as follows:

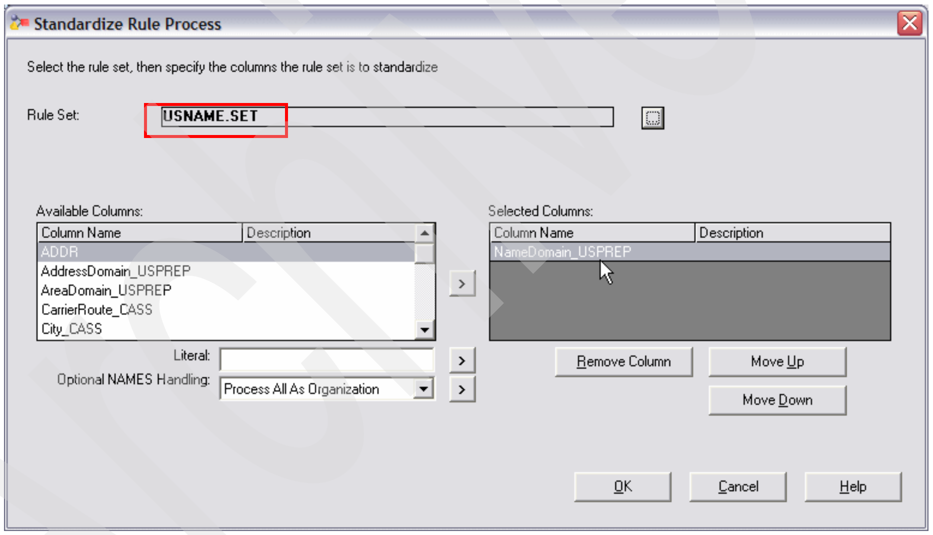
  – Name including individual names, organization names, attention instructions, and secondary names.

  – Address including unit number, street name, type, and directionals.

  – Area including cities, states, region, and other locale information.

  This category creates consistent and industry-standard data storage structures, and matching structures such as blocking keys and primary match keys.

  Figure 1-20 shows the specification of the USNAME rule set for the NameDomain_USPREP column (generated by the domain-preprocessor rule set) in the Standardize Rule Process window.



Standardize Rule Process

Select the rule set, then specify the columns the rule set is to standardize

Rule Set: USNAME.SET

Available Columns:

Column Name	Description
ADDR	
AddressDomain_USPREP	
AreaDomain_USPREP	
CarrierRoute_CASS	
City_CASS	

Literal:

Optional NAMES Handling: Process All As Organization

Selected Columns:

Column Name	Description
NameDomain_USPREP	

Figure 1-20 Domain-specific USNAME rule set

- Validation

  The Validation rule sets are used to standardize common business data including Date, Email Address, Phone Number, and Taxpayer ID/Social Security Number. These rules are configured for U.S. formats.

  The rule sets output two types of columns as follows:

  – Business Intelligence columns which help focus on critical information contained within output data. For example, valid data (VD) and valid flag (VF) columns.

  – Reporting/Error columns which provide details on the data value that fails validation and the reason code detailing the validation error.

  There are four rule sets as shown in Table 1-4 on page 48.

Table 1-4 Validation rule sets for the U.S.

Rule Set Name	Comments
VDATE	Dates that include day, month, and year The following information pertains to the VDATE validation rule set: ► Punctuation, such as hyphens or slashes, are removed during the parsing step. ► The rule set outputs two types of columns: Business Intelligence columns and Reporting/Error columns. ► There are no significant default Classification table entries used with this rule set. ► The standard output format is CCYYMMDD. There are formats for dates that are required for the input data. Expected input date formats include any of the following formats: ► mmddccyy (09211991) ► mmmddccyy (OCT021983) ► mmmdccyy (OCT21983) ► mmddccyy (04101986) ► mm/dd/ccyy (10/23/1960) ► m/d/ccyy (1/3/1960) ► mm/d/ccyy (10/3/1960) ► m/dd/ccyy (1/13/1960) ► mm-dd-ccyy (04-01-1960) ► m-d-ccyy (1-3-1960) ► mm-d-ccyy (10-3-1960) ► m-dd-ccyy (1-13-1960) ► ccyy-mm-dd (1990-10-22O) Examples of the output string corresponding to a particular input format are as follows: ► Input format 1990-10-22; output result 19901022 ► Input formation 1/13/1960; output result 19600113 ► Input format OCT021983; output result 19831002 If a data value passes validation, this rule set populates the following two Business Intelligence column values: ► The valid date {VD} data column which is populated with the eight numeric bytes ► The valid flag {VF} field which is populated with the value *T* If a data value fails validation, this rule set populates the Reporting Error fields "Invalid Data" (which has the invalid data value) and "Invalid Reason" which has a code such as IF (invalid input format), IM (invalid month), IT (invalid table such as a date of 11111111), MM (invalid numeric month), FB (invalid day of February—leap year), M0 (invalid day for months with 30 days), and M1 (invalid day for months with 31 days).

Rule Set Name	Comments
VEMAIL	The VEMAIL rule set identifies the format, components and completeness of e-mail addresses as follows: ► All e-mail addresses should have a user, domain, and top-level qualifier. ► Punctuation such as hyphens (-), at signs (@), and periods (.) are used as key delimiters during the parsing step. ► The default classification table for this rule set contains common domain (for instance, ORG, COM, EDU, GOV, and so forth.) and sub-domain qualifiers (for example, country and state codes). The parsing parameters parse the address into multiple tokens (for example, the e-mail address John_Smith@abccorp.comas has three tokens: John_Smith, abccorp, and comas).[a] If a data value is validated, this rule set populates the following Business Intelligence fields: ► User {US} ► Domain {DM} ► Top-level Qualifier {TL} ► URL {RL} If a data value fails validation, this rule set outputs the Reporting Error fields “Unhandled Data” (contains the unhandled data value) and “Unhandled Patterns” (contains the unhandled pattern).
VPHONE	The VPHONE rule set validates the value and standardizes the format of a U.S. telephone number. Punctuation such as hyphens (-) and parentheses ( ), are removed during the parsing step. Input formats support include (617) 338-0300, (617) 338-0300 X316, and (617) 338-0300 EXT 316. The hyphen, space, and parentheses are used to separate the data. After the data is parsed the hyphen, spaces, and parentheses are dropped. If the data value passes validation, this rule set outputs the following Business Intelligence field values: ► Valid Phone Number {VD} field which is populated with the numeric telephone number ► Phone Number Extension {VX} field which is populated with the extension number ► Valid flag {VF} field which is populated with *T* If the data value fails any one of the validation requirements the “Invalid Data” and the “Invalid Reason” fields are populated. Invalid Reason has a code such as IL (invalid length—main telephone without extension must be 7 or 10 bytes.), IT (invalid value), and IP (invalid pattern or format).

Rule Set Name	Comments
VTAXID	The VTAXID rule set validates the value and standardizes the format of a tax ID or national ID number as follows: Punctuation such as hyphens (-) are removed during the parsing step. There are no significant default Classification table entries used with this rule set. If a data value passes the listed criteria then it is considered a valid value and outputs two Business Intelligence field values: ► TAX_ID/SSN valid data {VD} field which is populated with the nine numeric bytes ► Valid flag {VF} field which is populated with the value *T*. If the data value fails any one of the validation requirements the “Invalid Data” and the “Invalid Reason” fields are populated. Invalid Reason codes include IP (data value did not contain nine, and only nine, numeric characters), IT (data value was found on the Invalid Data table), and Z3 (first three numeric characters are all zeros). *This rule set does not check whether the SSN is actually valid. It only checks the validity of the format and structure of the number and some basic validation rules, such as no zeros in the first three digits.*

a. The ampersand (@) and period (.) are used to separate the data. These separators are removed during the parsing process.

Figure 1-21 on page 51 through Figure 1-31 on page 59 show the use of the VPHONE and VTAXID rule sets against different data sources as follows:

- Figure 1-21 on page 51 shows the VPHONE_VTAXID_JOB that has the two Standardize stages (VPHONE and VTAXID) accessing the North American Bank’s (customer) contact information and driver information.
- Figure 1-22 on page 51 shows the Standardize Rule Process window with the VPHONE rule set for the selected column HOME_PHONE.
- Figure 1-23 on page 52 shows the Standardize Rule Process window with the VTAXID rule set for the selected column SSN.
- Figure 1-24 on page 52 shows the results of the execution of the VPHONE_VTAXID_JOB.
- Figure 1-25 on page 53 through Figure 1-27 on page 55 show the results of the execution of the VPHONE rule set indicating the last five rows (all having all nines) as having invalid data with a reason code of IT (invalid value).
- Figure 1-28 on page 56 through Figure 1-31 on page 59 show the results of the execution of the VTAXID rule set indicating that all the rows have valid social security numbers.

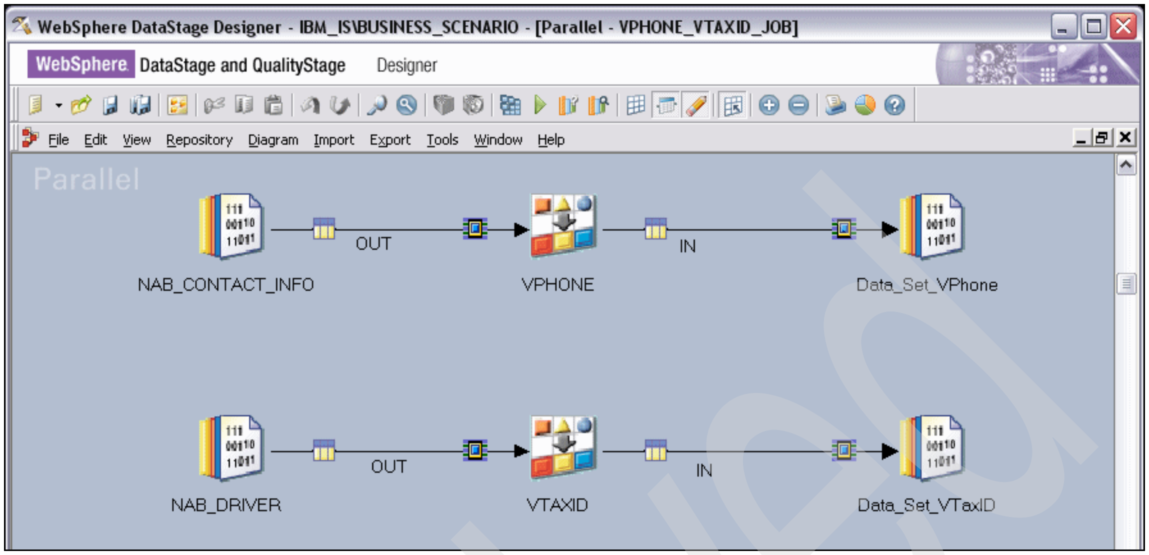
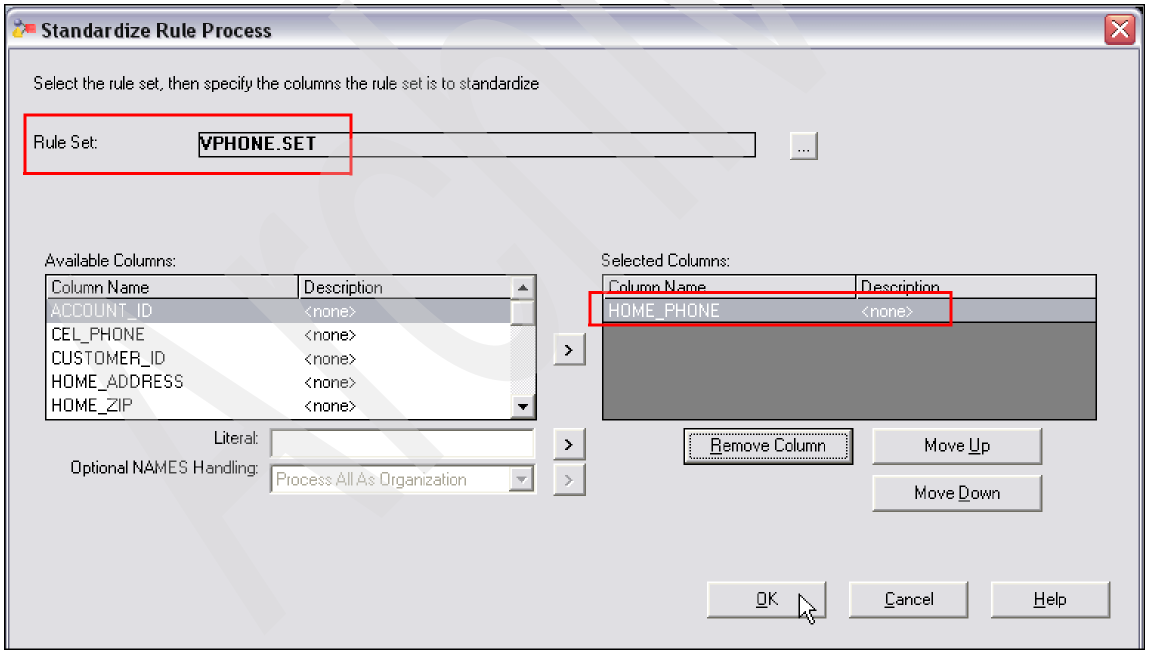

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - VPHONE_VTAXID_JOB]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NAB_CONTACT_INFO
OUT
VPHONE
IN
Data_Set_VPhone
NAB_DRIVER
OUT
VTAXID
IN
Data_Set_VTaxID

Figure 1-21 Validation rule set 1/11



Select the rule set, then specify the columns the rule set is to standardize

Rule Set: VPHONE.SET

Available Columns:

Column Name	Description
ACCOUNT_ID	<none>
CEL_PHONE	<none>
CUSTOMER_ID	<none>
HOME_ADDRESS	<none>
HOME_ZIP	<none>

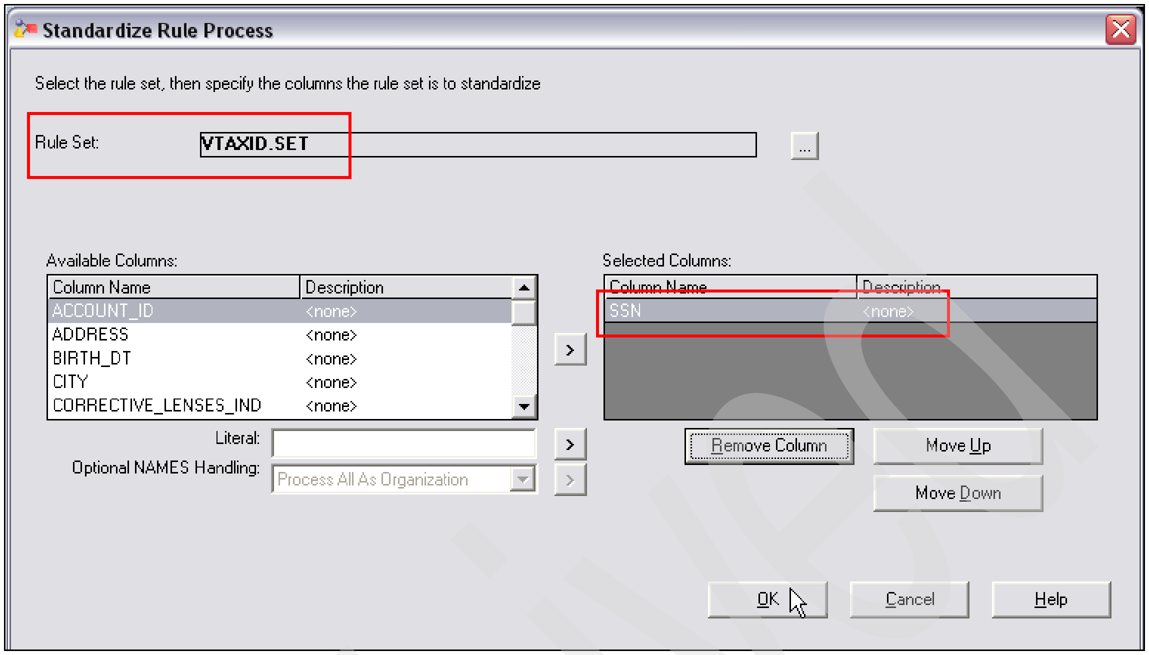
Selected Columns:

Column Name	Description
HOME_PHONE	<none>

Literal:

Optional NAMES Handling: Process All As Organization

Figure 1-22 Validation rule set 2/11



Standardize Rule Process

Select the rule set, then specify the columns the rule set is to standardize

Rule Set: VTAXID.SET

Available Columns:

Column Name	Description
ACCOUNT_ID	<none>
ADDRESS	<none>
BIRTH_DT	<none>
CITY	<none>
CORRECTIVE_LENSES_IND	<none>

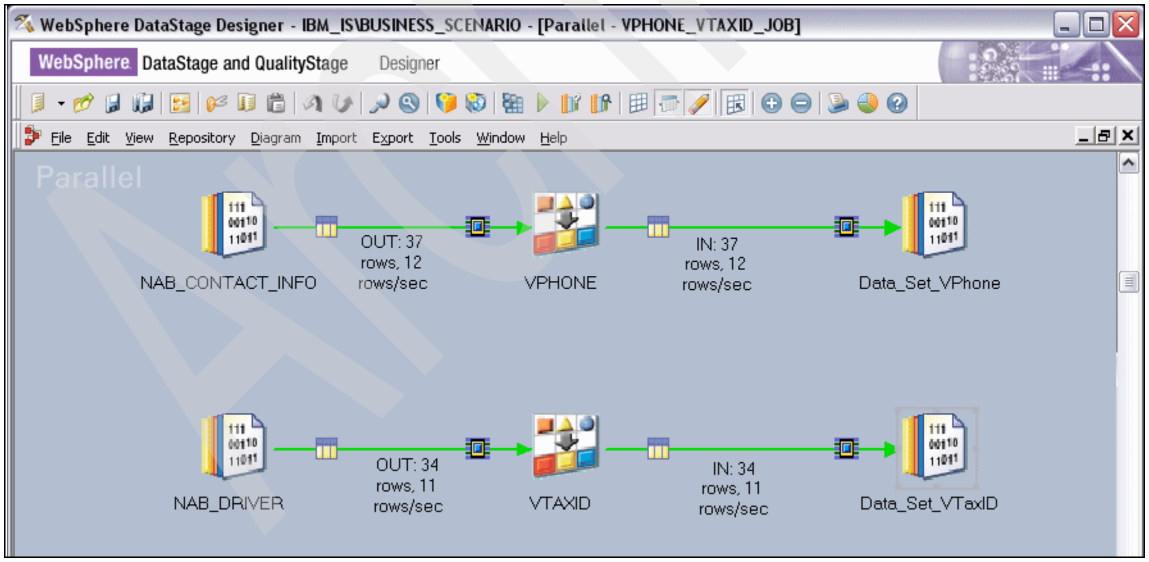
Selected Columns:

Column Name	Description
SSN	<none>

Literal:

Optional NAMES Handling: Process All As Organization

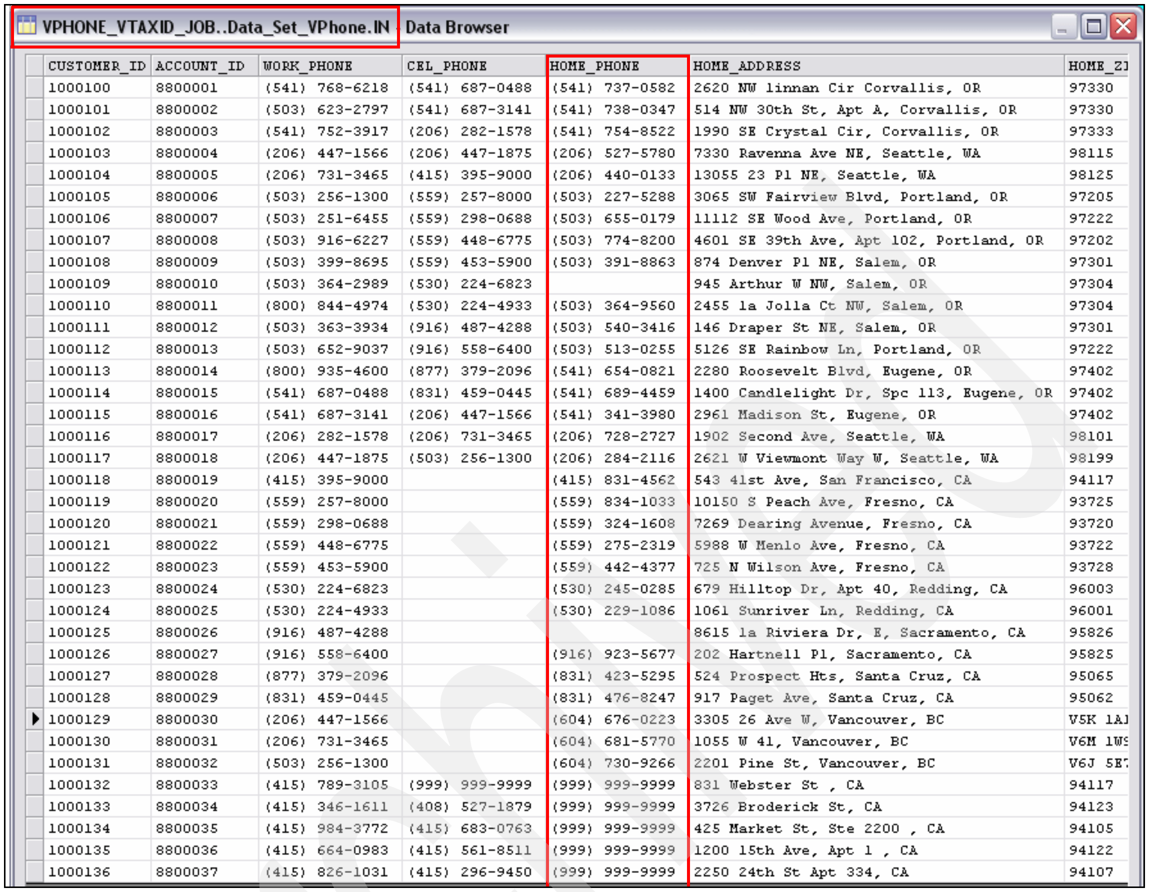
Figure 1-23 Validation rule set 3/11



WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - VPHONE_VTAXID_JOB]

WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NAB_CONTACT_INFO
OUT: 37 rows, 12 rows/sec
VPHONE
IN: 37 rows, 12 rows/sec
Data_Set_VPhone
NAB_DRIVER
OUT: 34 rows, 11 rows/sec
VTAXID
IN: 34 rows, 11 rows/sec
Data_Set_VTaxID

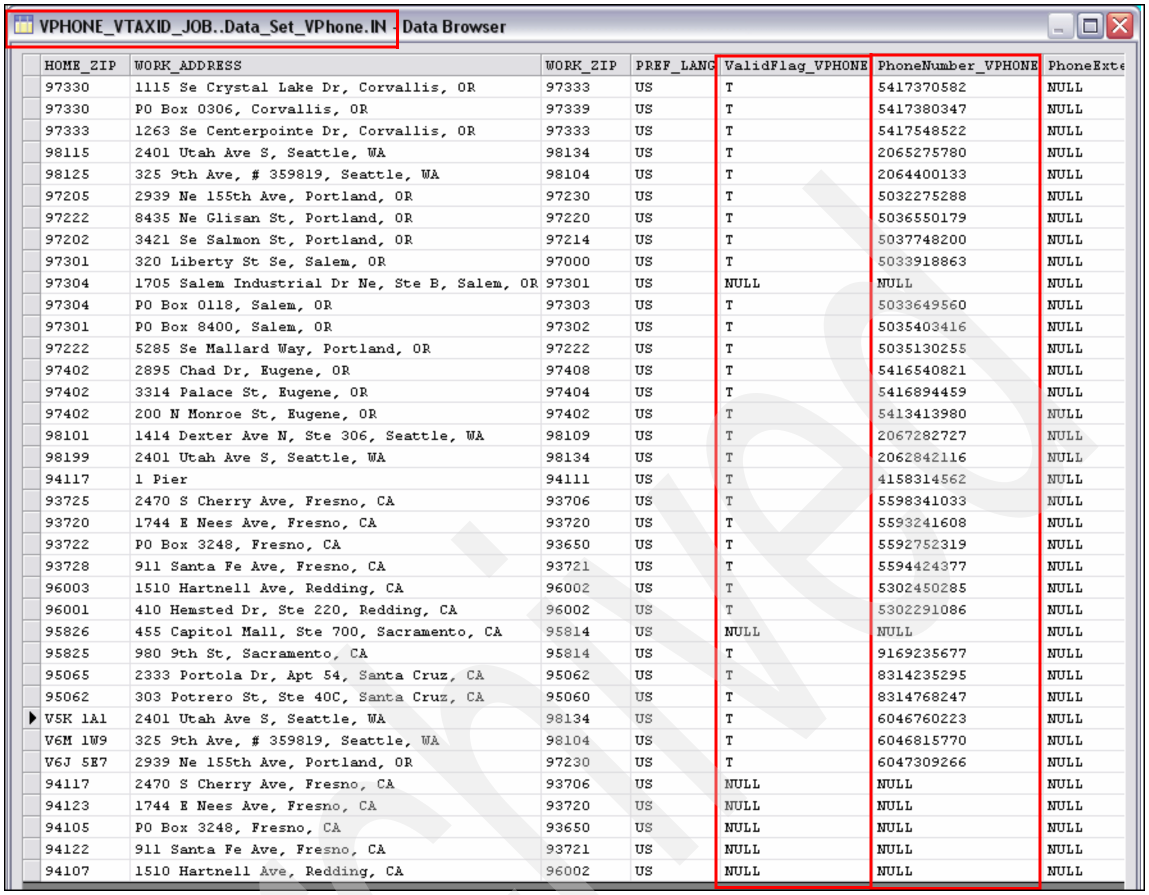
Figure 1-24 Validation rule set 4/11



VPHONE_VTAXID_JOB..Data_Set_VPhone.IN Data Browser

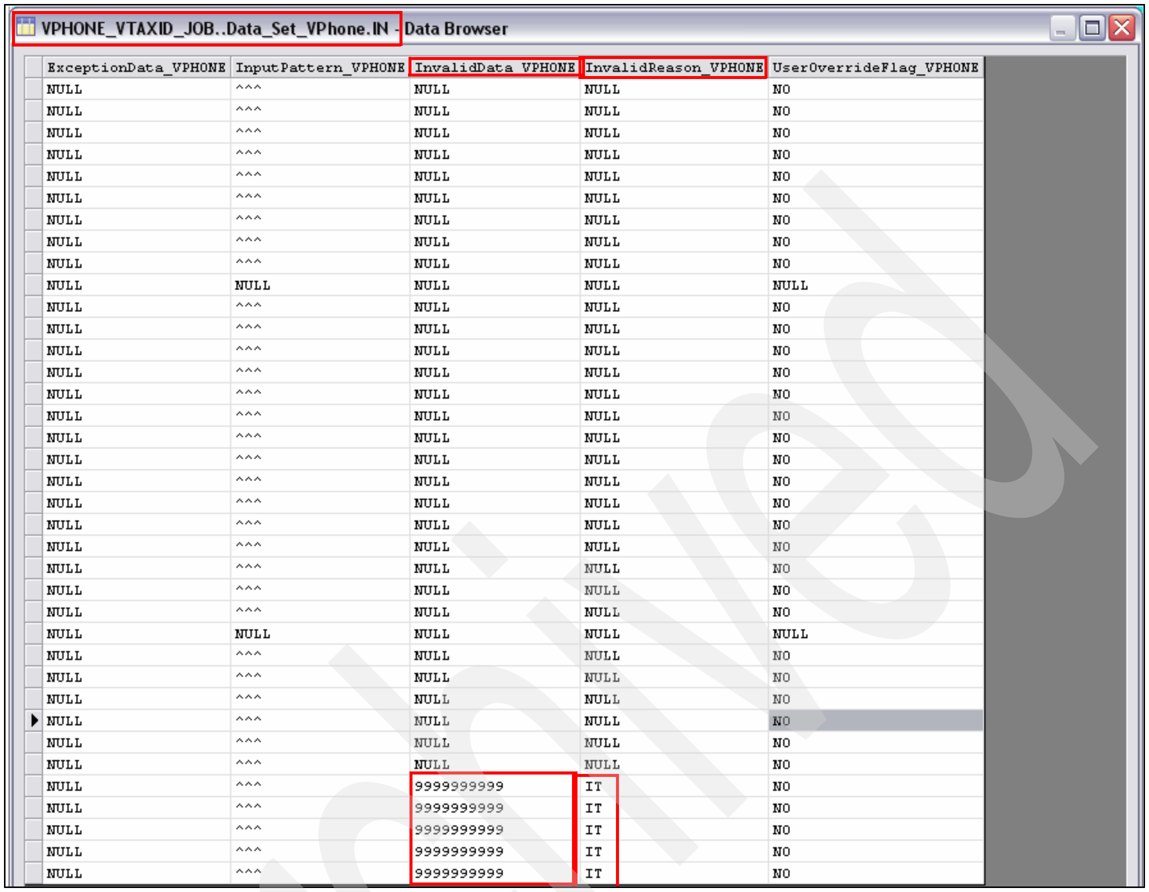
CUSTOMER_ID	ACCOUNT_ID	WORK_PHONE	CEL_PHONE	HOME_PHONE	HOME_ADDRESS	HOME_ZI
1000100	8800001	(541) 768-6218	(541) 687-0488	(541) 737-0582	2620 NW linnan Cir Corvallis, OR	97330
1000101	8800002	(503) 623-2797	(541) 687-3141	(541) 738-0347	514 NW 30th St, Apt A, Corvallis, OR	97330
1000102	8800003	(541) 752-3917	(206) 282-1578	(541) 754-8522	1990 SE Crystal Cir, Corvallis, OR	97333
1000103	8800004	(206) 447-1566	(206) 447-1875	(206) 527-5780	7330 Ravenna Ave NE, Seattle, WA	98115
1000104	8800005	(206) 731-3465	(415) 395-9000	(206) 440-0133	13055 23 Pl NE, Seattle, WA	98125
1000105	8800006	(503) 256-1300	(559) 257-8000	(503) 227-5288	3065 SW Fairview Blvd, Portland, OR	97205
1000106	8800007	(503) 251-6455	(559) 298-0688	(503) 655-0179	11112 SE Wood Ave, Portland, OR	97222
1000107	8800008	(503) 916-6227	(559) 448-6775	(503) 774-8200	4601 SE 39th Ave, Apt 102, Portland, OR	97202
1000108	8800009	(503) 399-8695	(559) 453-5900	(503) 391-8863	874 Denver Pl NE, Salem, OR	97301
1000109	8800010	(503) 364-2989	(530) 224-6823		945 Arthur W NW, Salem, OR	97304
1000110	8800011	(800) 844-4974	(530) 224-4933	(503) 364-9560	2455 la Jolla Ct NW, Salem, OR	97304
1000111	8800012	(503) 363-3934	(916) 487-4288	(503) 540-3416	146 Draper St NE, Salem, OR	97301
1000112	8800013	(503) 652-9037	(916) 558-6400	(503) 513-0255	5126 SE Rainbow Ln, Portland, OR	97222
1000113	8800014	(800) 935-4600	(877) 379-2096	(541) 654-0821	2280 Roosevelt Blvd, Eugene, OR	97402
1000114	8800015	(541) 687-0488	(831) 459-0445	(541) 689-4459	1400 Candlelight Dr, Spc 113, Eugene, OR	97402
1000115	8800016	(541) 687-3141	(206) 447-1566	(541) 341-3980	2961 Madison St, Eugene, OR	97402
1000116	8800017	(206) 282-1578	(206) 731-3465	(206) 728-2727	1902 Second Ave, Seattle, WA	98101
1000117	8800018	(206) 447-1875	(503) 256-1300	(206) 284-2116	2621 W Viewmont Way W, Seattle, WA	98199
1000118	8800019	(415) 395-9000		(415) 831-4562	543 41st Ave, San Francisco, CA	94117
1000119	8800020	(559) 257-8000		(559) 834-1033	10150 S Peach Ave, Fresno, CA	93725
1000120	8800021	(559) 298-0688		(559) 324-1608	7269 Dearing Avenue, Fresno, CA	93720
1000121	8800022	(559) 448-6775		(559) 275-2319	5988 W Menlo Ave, Fresno, CA	93722
1000122	8800023	(559) 453-5900		(559) 442-4377	725 N Wilson Ave, Fresno, CA	93728
1000123	8800024	(530) 224-6823		(530) 245-0285	679 Hilltop Dr, Apt 40, Redding, CA	96003
1000124	8800025	(530) 224-4933		(530) 229-1086	1061 Sunriver Ln, Redding, CA	96001
1000125	8800026	(916) 487-4288			8615 la Riviera Dr, E, Sacramento, CA	95826
1000126	8800027	(916) 558-6400		(916) 923-5677	202 Hartnell Pl, Sacramento, CA	95825
1000127	8800028	(877) 379-2096		(831) 423-5295	524 Prospect Hts, Santa Cruz, CA	95065
1000128	8800029	(831) 459-0445		(831) 476-8247	917 Paget Ave, Santa Cruz, CA	95062
1000129	8800030	(206) 447-1566		(604) 676-0223	3305 26 Ave W, Vancouver, BC	V5K 1A
1000130	8800031	(206) 731-3465		(604) 681-5770	1055 W 41, Vancouver, BC	V6M 1W
1000131	8800032	(503) 256-1300		(604) 730-9266	2201 Pine St, Vancouver, BC	V6J 5E
1000132	8800033	(415) 789-3105	(999) 999-9999	(999) 999-9999	831 Webster St , CA	94117
1000133	8800034	(415) 346-1611	(408) 527-1879	(999) 999-9999	3726 Broderick St, CA	94123
1000134	8800035	(415) 984-3772	(415) 683-0763	(999) 999-9999	425 Market St, Ste 2200 , CA	94105
1000135	8800036	(415) 664-0983	(415) 561-8511	(999) 999-9999	1200 15th Ave, Apt 1 , CA	94122
1000136	8800037	(415) 826-1031	(415) 296-9450	(999) 999-9999	2250 24th St Apt 334, CA	94107

Figure 1-25 Validation rule set 5/11


VPHONE_VTAXID_JOB..Data_Set_VPhone.IN - Data Browser

HOME_ZIP	WORK_ADDRESS	WORK_ZIP	PREF_LANG	ValidFlag_VPHONE	PhoneNumber_VPHONE	PhoneExte
97330	1115 Se Crystal Lake Dr, Corvallis, OR	97333	US	T	5417370582	NULL
97330	PO Box 0306, Corvallis, OR	97339	US	T	5417380347	NULL
97333	1263 Se Centerpointe Dr, Corvallis, OR	97333	US	T	5417548522	NULL
98115	2401 Utah Ave S, Seattle, WA	98134	US	T	2065275780	NULL
98125	325 9th Ave, # 359819, Seattle, WA	98104	US	T	2064400133	NULL
97205	2939 Ne 155th Ave, Portland, OR	97230	US	T	5032275288	NULL
97222	8435 Ne Glisan St, Portland, OR	97220	US	T	5036550179	NULL
97202	3421 Se Salmon St, Portland, OR	97214	US	T	5037748200	NULL
97301	320 Liberty St Se, Salem, OR	97000	US	T	5033918863	NULL
97304	1705 Salem Industrial Dr Ne, Ste B, Salem, OR	97301	US	NULL	NULL	NULL
97304	PO Box 0118, Salem, OR	97303	US	T	5033649560	NULL
97301	PO Box 8400, Salem, OR	97302	US	T	5035403416	NULL
97222	5285 Se Mallard Way, Portland, OR	97222	US	T	5035130255	NULL
97402	2895 Chad Dr, Eugene, OR	97408	US	T	5416540821	NULL
97402	3314 Palace St, Eugene, OR	97404	US	T	5416894459	NULL
97402	200 N Monroe St, Eugene, OR	97402	US	T	5413413980	NULL
98101	1414 Dexter Ave N, Ste 306, Seattle, WA	98109	US	T	2067282727	NULL
98199	2401 Utah Ave S, Seattle, WA	98134	US	T	2062842116	NULL
94117	1 Pier	94111	US	T	4158314562	NULL
93725	2470 S Cherry Ave, Fresno, CA	93706	US	T	5598341033	NULL
93720	1744 E Nees Ave, Fresno, CA	93720	US	T	5593241608	NULL
93722	PO Box 3248, Fresno, CA	93650	US	T	5592752319	NULL
93728	911 Santa Fe Ave, Fresno, CA	93721	US	T	5594424377	NULL
96003	1510 Hartnell Ave, Redding, CA	96002	US	T	5302450285	NULL
96001	410 Hemsted Dr, Ste 220, Redding, CA	96002	US	T	5302291086	NULL
95826	455 Capitol Mall, Ste 700, Sacramento, CA	95814	US	NULL	NULL	NULL
95825	980 9th St, Sacramento, CA	95814	US	T	9169235677	NULL
95065	2333 Portola Dr, Apt 54, Santa Cruz, CA	95062	US	T	8314235295	NULL
95062	303 Potrero St, Ste 40C, Santa Cruz, CA	95060	US	T	8314768247	NULL
V5K 1A1	2401 Utah Ave S, Seattle, WA	98134	US	T	6046760223	NULL
V6M 1W9	325 9th Ave, # 359819, Seattle, WA	98104	US	T	6046815770	NULL
V6J 5E7	2939 Ne 155th Ave, Portland, OR	97230	US	T	6047309266	NULL
94117	2470 S Cherry Ave, Fresno, CA	93706	US	NULL	NULL	NULL
94123	1744 E Nees Ave, Fresno, CA	93720	US	NULL	NULL	NULL
94105	PO Box 3248, Fresno, CA	93650	US	NULL	NULL	NULL
94122	911 Santa Fe Ave, Fresno, CA	93721	US	NULL	NULL	NULL
94107	1510 Hartnell Ave, Redding, CA	96002	US	NULL	NULL	NULL

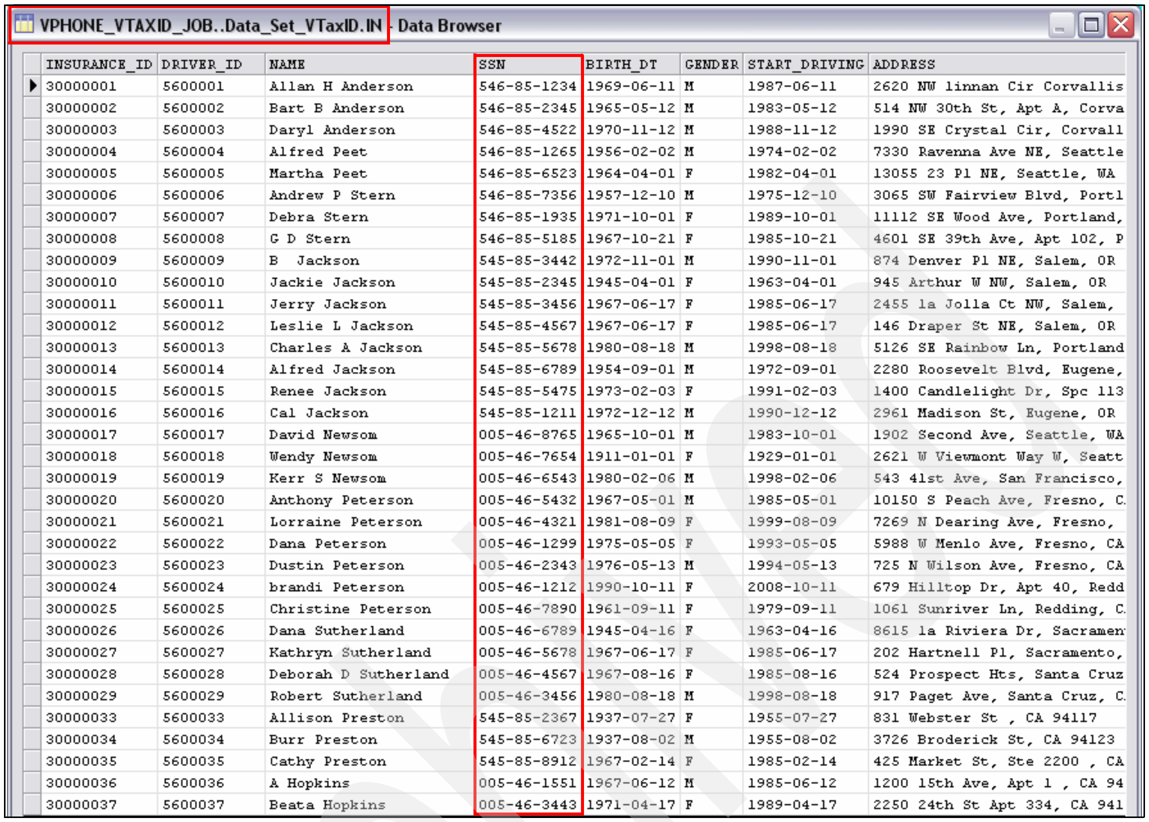
Figure 1-26 Validation rule set 6/11



VPHONE_VTAXID_JOB..Data_Set_VPhone.IN - Data Browser

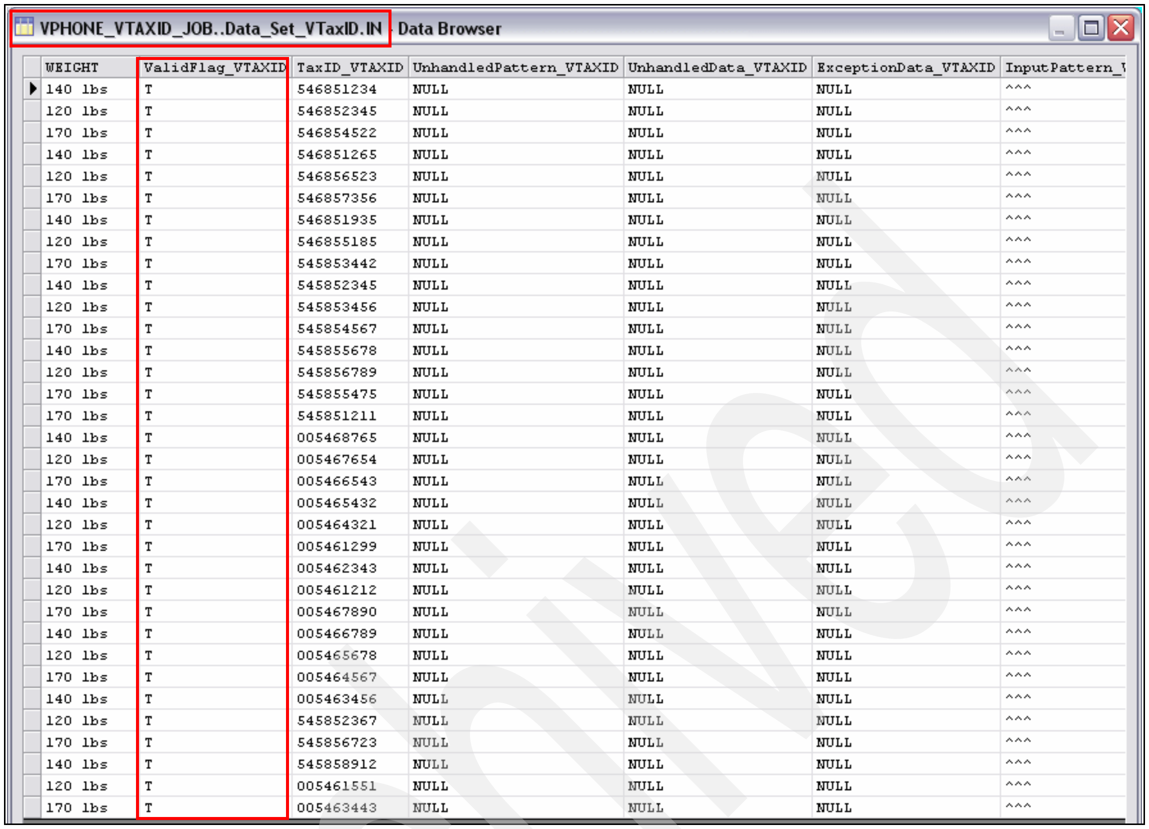
ExceptionData_VPHONE	InputPattern_VPHONE	InvalidData_VPHONE	InvalidReason_VPHONE	UserOverrideFlag_VPHONE
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	NULL	NULL	NULL	NULL
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	NULL	NULL	NULL	NULL
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	9999999999	IT	NO
NULL	^^^	9999999999	IT	NO
NULL	^^^	9999999999	IT	NO
NULL	^^^	9999999999	IT	NO
NULL	^^^	9999999999	IT	NO

Figure 1-27 Validation rule set 7/11


VPHONE_VTAXID_JOB..Data_Set_VTaxID.IN - Data Browser

INSURANCE_ID	DRIVER_ID	NAME	SSN	BIRTH_DT	GENDER	START_DRIVING	ADDRESS
30000001	5600001	Allan H Anderson	546-85-1234	1969-06-11	M	1987-06-11	2620 NW linnan Cir Corvallis
30000002	5600002	Bart B Anderson	546-85-2345	1965-05-12	M	1983-05-12	514 NW 30th St, Apt A, Corva
30000003	5600003	Daryl Anderson	546-85-4522	1970-11-12	M	1988-11-12	1990 SE Crystal Cir, Corvall
30000004	5600004	Alfred Peet	546-85-1265	1956-02-02	M	1974-02-02	7330 Ravenna Ave NE, Seattle
30000005	5600005	Martha Peet	546-85-6523	1964-04-01	F	1982-04-01	13055 23 Pl NE, Seattle, WA
30000006	5600006	Andrew P Stern	546-85-7356	1957-12-10	M	1975-12-10	3065 SW Fairview Blvd, Portl
30000007	5600007	Debra Stern	546-85-1935	1971-10-01	F	1989-10-01	11112 SE Wood Ave, Portland,
30000008	5600008	G D Stern	546-85-5185	1967-10-21	F	1985-10-21	4601 SE 39th Ave, Apt 102, P
30000009	5600009	B Jackson	545-85-3442	1972-11-01	M	1990-11-01	874 Denver Pl NE, Salem, OR
30000010	5600010	Jackie Jackson	545-85-2345	1945-04-01	F	1963-04-01	945 Arthur W NW, Salem, OR
30000011	5600011	Jerry Jackson	545-85-3456	1967-06-17	F	1985-06-17	2455 la Jolla Ct NW, Salem,
30000012	5600012	Leslie L Jackson	545-85-4567	1967-06-17	F	1985-06-17	146 Draper St NE, Salem, OR
30000013	5600013	Charles A Jackson	545-85-5678	1980-08-18	M	1998-08-18	5126 SE Rainbow Ln, Portland
30000014	5600014	Alfred Jackson	545-85-6789	1954-09-01	M	1972-09-01	2280 Roosevelt Blvd, Eugene,
30000015	5600015	Renee Jackson	545-85-5475	1973-02-03	F	1991-02-03	1400 Candlelight Dr, Spc 113
30000016	5600016	Cal Jackson	545-85-1211	1972-12-12	M	1990-12-12	2961 Madison St, Eugene, OR
30000017	5600017	David Newsom	005-46-8765	1965-10-01	M	1983-10-01	1902 Second Ave, Seattle, WA
30000018	5600018	Wendy Newsom	005-46-7654	1911-01-01	F	1929-01-01	2621 W Viewmont Way W, Seatt
30000019	5600019	Kerr S Newsom	005-46-6543	1980-02-06	M	1998-02-06	543 41st Ave, San Francisco,
30000020	5600020	Anthony Peterson	005-46-5432	1967-05-01	M	1985-05-01	10150 S Peach Ave, Fresno, C.
30000021	5600021	Lorraine Peterson	005-46-4321	1981-08-09	F	1999-08-09	7269 N Dearing Ave, Fresno,
30000022	5600022	Dana Peterson	005-46-1299	1975-05-05	F	1993-05-05	5988 W Menlo Ave, Fresno, CA
30000023	5600023	Dustin Peterson	005-46-2343	1976-05-13	M	1994-05-13	725 N Wilson Ave, Fresno, CA
30000024	5600024	brandi Peterson	005-46-1212	1990-10-11	F	2008-10-11	679 Hilltop Dr, Apt 40, Redd
30000025	5600025	Christine Peterson	005-46-7890	1961-09-11	F	1979-09-11	1061 Sunriver Ln, Redding, C.
30000026	5600026	Dana Sutherland	005-46-6789	1945-04-16	F	1963-04-16	8615 la Riviera Dr, Sacramen
30000027	5600027	Kathryn Sutherland	005-46-5678	1967-06-17	F	1985-06-17	202 Hartnell Pl, Sacramento,
30000028	5600028	Deborah D Sutherland	005-46-4567	1967-08-16	F	1985-08-16	524 Prospect Hts, Santa Cruz
30000029	5600029	Robert Sutherland	005-46-3456	1980-08-18	M	1998-08-18	917 Paget Ave, Santa Cruz, C.
30000033	5600033	Allison Preston	545-85-2367	1937-07-27	F	1955-07-27	831 Webster St , CA 94117
30000034	5600034	Burr Preston	545-85-6723	1937-08-02	M	1955-08-02	3726 Broderick St, CA 94123
30000035	5600035	Cathy Preston	545-85-8912	1967-02-14	F	1985-02-14	425 Market St, Ste 2200 , CA
30000036	5600036	A Hopkins	005-46-1551	1967-06-12	M	1985-06-12	1200 15th Ave, Apt 1 , CA 94
30000037	5600037	Beata Hopkins	005-46-3443	1971-04-17	F	1989-04-17	2250 24th St Apt 334, CA 941

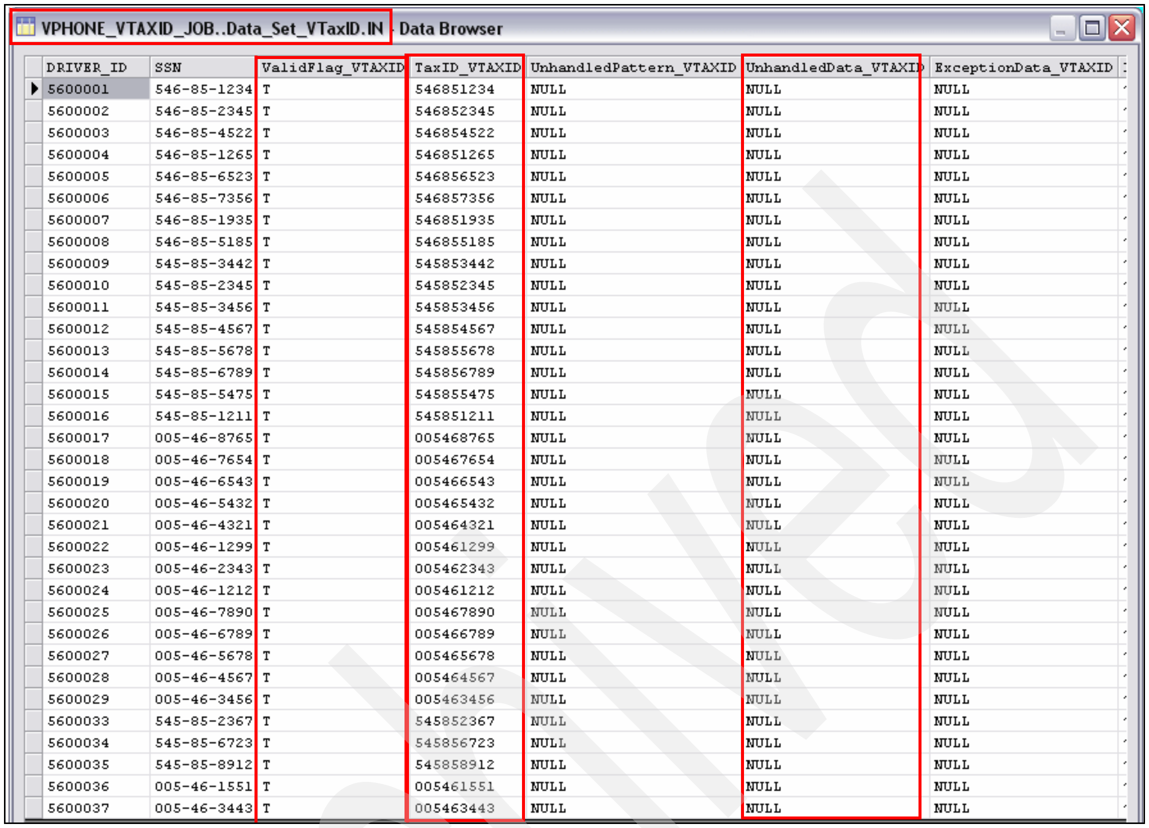
Figure 1-28   Validation rule set 8/11



VPHONE_VTAXID_JOB..Data_Set_VTaxID.IN - Data Browser

WEIGHT	ValidFlag_VTAXID	TaxID_VTAXID	UnhandledPattern_VTAXID	UnhandledData_VTAXID	ExceptionData_VTAXID	InputPattern_V
140 lbs	T	546851234	NULL	NULL	NULL	^^^
120 lbs	T	546852345	NULL	NULL	NULL	^^^
170 lbs	T	546854522	NULL	NULL	NULL	^^^
140 lbs	T	546851265	NULL	NULL	NULL	^^^
120 lbs	T	546856523	NULL	NULL	NULL	^^^
170 lbs	T	546857356	NULL	NULL	NULL	^^^
140 lbs	T	546851935	NULL	NULL	NULL	^^^
120 lbs	T	546855185	NULL	NULL	NULL	^^^
170 lbs	T	545853442	NULL	NULL	NULL	^^^
140 lbs	T	545852345	NULL	NULL	NULL	^^^
120 lbs	T	545853456	NULL	NULL	NULL	^^^
170 lbs	T	545854567	NULL	NULL	NULL	^^^
140 lbs	T	545855678	NULL	NULL	NULL	^^^
120 lbs	T	545856789	NULL	NULL	NULL	^^^
170 lbs	T	545855475	NULL	NULL	NULL	^^^
170 lbs	T	545851211	NULL	NULL	NULL	^^^
140 lbs	T	005468765	NULL	NULL	NULL	^^^
120 lbs	T	005467654	NULL	NULL	NULL	^^^
170 lbs	T	005466543	NULL	NULL	NULL	^^^
140 lbs	T	005465432	NULL	NULL	NULL	^^^
120 lbs	T	005464321	NULL	NULL	NULL	^^^
170 lbs	T	005461299	NULL	NULL	NULL	^^^
140 lbs	T	005462343	NULL	NULL	NULL	^^^
120 lbs	T	005461212	NULL	NULL	NULL	^^^
170 lbs	T	005467890	NULL	NULL	NULL	^^^
140 lbs	T	005466789	NULL	NULL	NULL	^^^
120 lbs	T	005465678	NULL	NULL	NULL	^^^
170 lbs	T	005464567	NULL	NULL	NULL	^^^
140 lbs	T	005463456	NULL	NULL	NULL	^^^
120 lbs	T	545852367	NULL	NULL	NULL	^^^
170 lbs	T	545856723	NULL	NULL	NULL	^^^
140 lbs	T	545858912	NULL	NULL	NULL	^^^
120 lbs	T	005461551	NULL	NULL	NULL	^^^
170 lbs	T	005463443	NULL	NULL	NULL	^^^

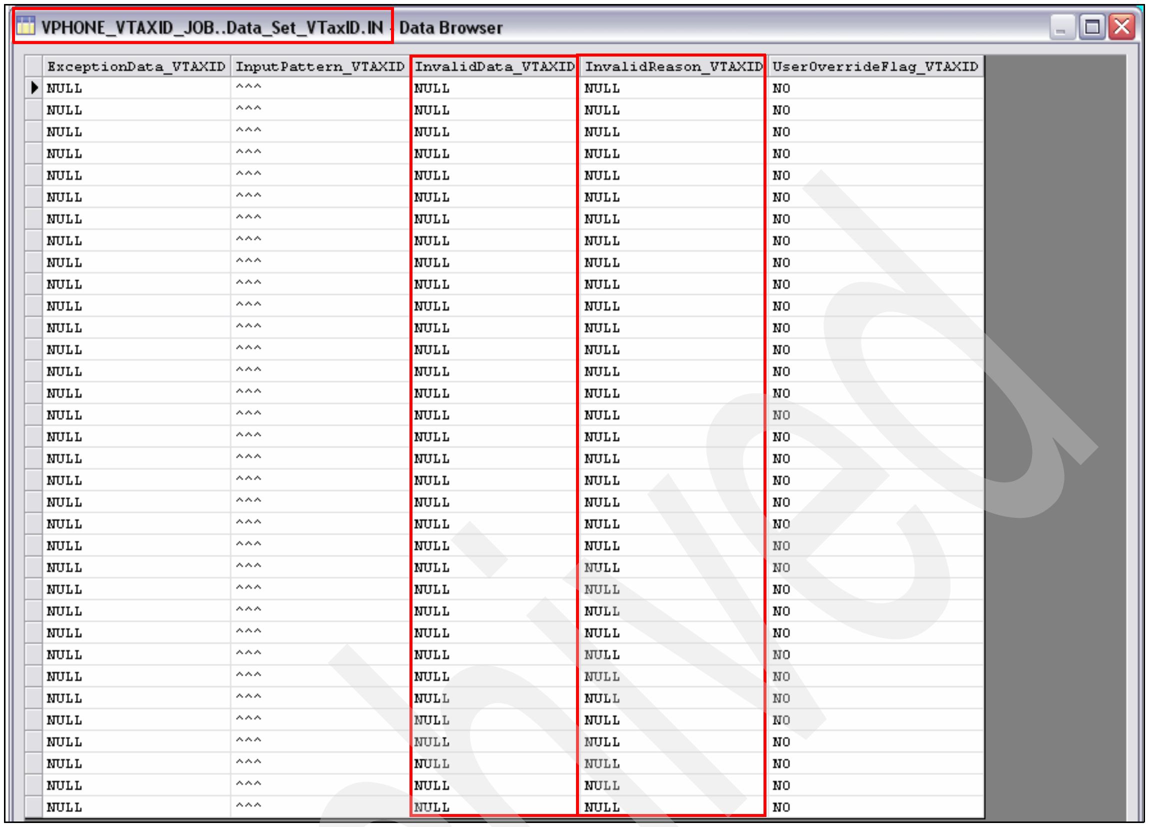
Figure 1-29 Validation rule set 9/11



VPHONE_VTAXID_JOB..Data_Set_VTaxID.IN - Data Browser

DRIVER_ID	SSN	ValidFlag_VTAXID	TaxID_VTAXID	UnhandledPattern_VTAXID	UnhandledData_VTAXID	ExceptionData_VTAXID
5600001	546-85-1234	T	546851234	NULL	NULL	NULL
5600002	546-85-2345	T	546852345	NULL	NULL	NULL
5600003	546-85-4522	T	546854522	NULL	NULL	NULL
5600004	546-85-1265	T	546851265	NULL	NULL	NULL
5600005	546-85-6523	T	546856523	NULL	NULL	NULL
5600006	546-85-7356	T	546857356	NULL	NULL	NULL
5600007	546-85-1935	T	546851935	NULL	NULL	NULL
5600008	546-85-5185	T	546855185	NULL	NULL	NULL
5600009	545-85-3442	T	545853442	NULL	NULL	NULL
5600010	545-85-2345	T	545852345	NULL	NULL	NULL
5600011	545-85-3456	T	545853456	NULL	NULL	NULL
5600012	545-85-4567	T	545854567	NULL	NULL	NULL
5600013	545-85-5678	T	545855678	NULL	NULL	NULL
5600014	545-85-6789	T	545856789	NULL	NULL	NULL
5600015	545-85-5475	T	545855475	NULL	NULL	NULL
5600016	545-85-1211	T	545851211	NULL	NULL	NULL
5600017	005-46-8765	T	005468765	NULL	NULL	NULL
5600018	005-46-7654	T	005467654	NULL	NULL	NULL
5600019	005-46-6543	T	005466543	NULL	NULL	NULL
5600020	005-46-5432	T	005465432	NULL	NULL	NULL
5600021	005-46-4321	T	005464321	NULL	NULL	NULL
5600022	005-46-1299	T	005461299	NULL	NULL	NULL
5600023	005-46-2343	T	005462343	NULL	NULL	NULL
5600024	005-46-1212	T	005461212	NULL	NULL	NULL
5600025	005-46-7890	T	005467890	NULL	NULL	NULL
5600026	005-46-6789	T	005466789	NULL	NULL	NULL
5600027	005-46-5678	T	005465678	NULL	NULL	NULL
5600028	005-46-4567	T	005464567	NULL	NULL	NULL
5600029	005-46-3456	T	005463456	NULL	NULL	NULL
5600033	545-85-2367	T	545852367	NULL	NULL	NULL
5600034	545-85-6723	T	545856723	NULL	NULL	NULL
5600035	545-85-8912	T	545858912	NULL	NULL	NULL
5600036	005-46-1551	T	005461551	NULL	NULL	NULL
5600037	005-46-3443	T	005463443	NULL	NULL	NULL

Figure 1-30 Validation rule set 10/11



VPHONE_VTAXID_JOB..Data_Set_VTaxID.IN - Data Browser

ExceptionData_VTAXID	InputPattern_VTAXID	InvalidData_VTAXID	InvalidReason_VTAXID	UserOverrideFlag_VTAXID
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO
NULL	^^^	NULL	NULL	NO

Figure 1-31 Validation rule set 11/11

Attention: The rule sets that are provided with IBM WebSphere QualityStage are designed to provide optimum results. However, if the results are not satisfactory, you can modify rule set behavior using rule override tables. You can modify both the domain-preprocessor rule sets, as well as the Domain-specific rule sets. We discuss managing rule sets briefly in “Managing rule sets” on page 60. We provide examples of rule set overrides in 1.10, “Mailing list scenario” on page 117.

You can also build custom rule sets for specific data. We do not cover that topic in this book.

Figure 1-32 shows the Standardize Stage processing flow using the COUNTRY rule set (as described in “Country Rule Set” on page 81) in the Standardize stage to first identify the country codes associated with each record from a source that contains information from multiple countries. The Investigate stage (using the same COUNTRY rule set) is used to identify the various country codes involved. You then separate the records by country codes using the Filter stage.[7] When filtered (U.S. records shown here), the Domain Pre-processor and Domain-Specific rule sets are used to standardize the U.S. records. The same workflow is representative of other countries used with the Standardize stage.

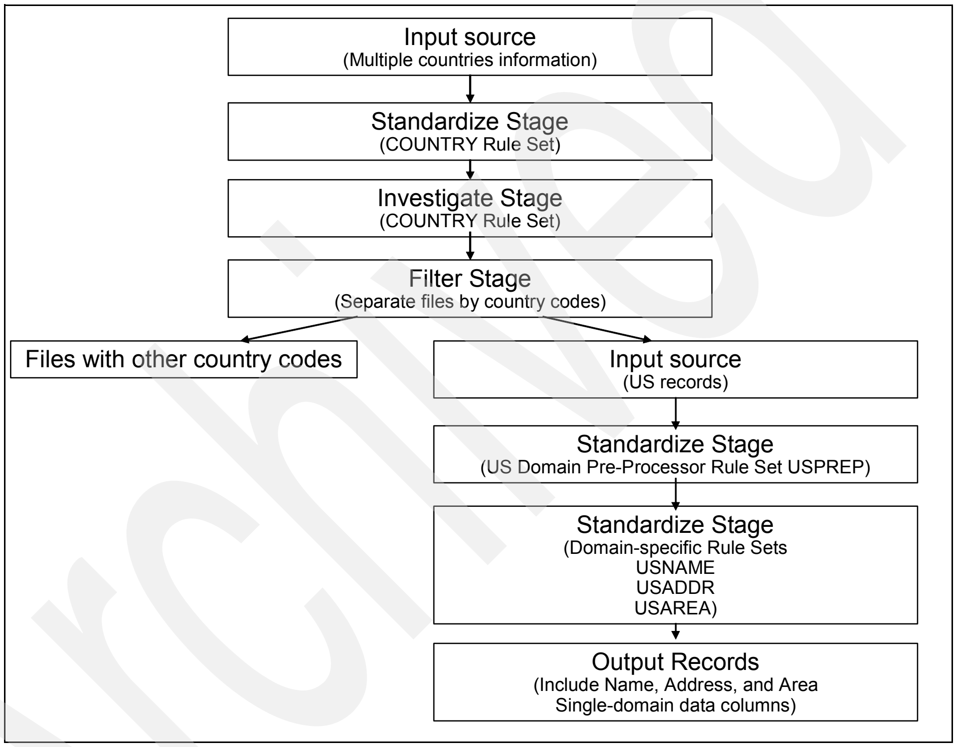

Input source
(Multiple countries information)
Standardize Stage
(COUNTRY Rule Set)
Investigate Stage
(COUNTRY Rule Set)
Filter Stage
(Separate files by country codes)
Files with other country codes
Input source
(US records)
Standardize Stage
(US Domain Pre-Processor Rule Set USPREP)
Standardize Stage
(Domain-specific Rule Sets
USNAME
USADDR
USAREA)
Output Records
(Include Name, Address, and Area
Single-domain data columns)

Figure 1-32 Standardize Stage processing flow

Managing rule sets

You apply rule sets in the Standardize stage or Investigate stage to determine how columns are parsed and classified into tokens. You can also apply rule sets for international stages such as Worldwide Address Verification and Enhancement System (WAVES) and Multinational Standardize Stage (MNS). With all of these stages, you can use rules management (that is modify existing rules and add new rules).

[7] You can also use a Transformer stage to split an input source into multiple targets.

You can modify the following standardization rule sets:

- Domain preprocessor
- Domain specific
- Validation

After you modify or add a rule set, you can test it in rules management. When you modify or add a business rule set, you can test it to ensure that it defines the data cleansing business rule that you need.

Note: The information that you enter in override tables is applied to rule sets for a given project because rule sets are project specific. Provision All is required after an override for it to take effect.

The topics that we discuss in this section are:

- Rule set files
- Rule set customization with override object types
- Selecting override object types to modify rule sets
- Standardization rule set tester
- Choosing the appropriate override method

Attention: You need to provision new, copied, or customized rule sets before you compile a job that uses them.

Rule set files

The Rule Management window in Figure 1-33 shows the different parts that make up a rule set, including Overrides.

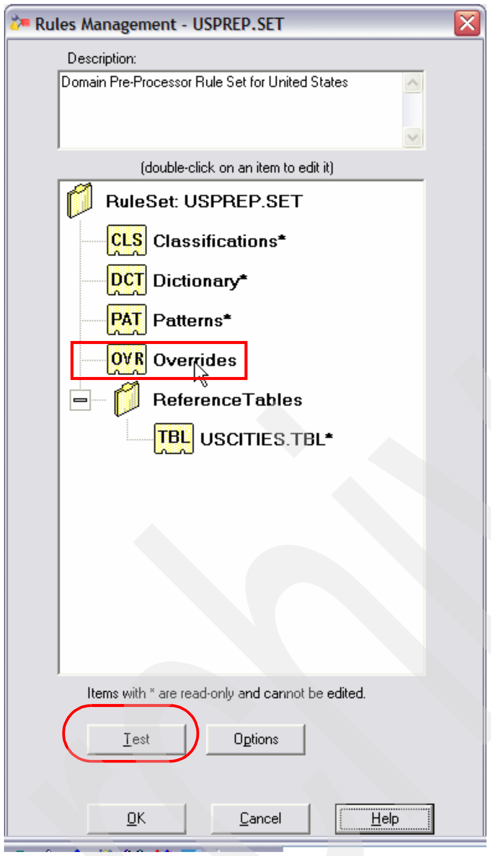

Rules Management - USPREP.SET
Description:
Domain Pre-Processor Rule Set for United States
(double-click on an item to edit it)
RuleSet: USPREP.SET
CLS Classifications*
DCT Dictionary*
PAT Patterns*
OVR Overrides
ReferenceTables
TBL USCITIES.TBL*
Items with * are read-only and cannot be edited.
Test
Options
OK
Cancel
Help

Figure 1-33 Rule Management window

Rule sets include the following files:

- Classifications

  The Classification Table is used in the standardization process to identify and classify key words such as titles, street name, street type, and directions. The Classification Table includes the name of the rule set and the classification legend. Click CLS in Figure 1-33 to view its contents. The partial contents of the Classification Table for the USPREP rule set is shown in Example 1-1 on page 33. As already mentioned, you can gain valuable insight by browsing the Classification tables to determine the classification codes, as well as the different literals supported such as ZQMIXAZQ and ZQNAMEZQ.

- Dictionary

  The Dictionary file defines the fields for the output of a particular rule set. The file contains a list of domain, matching, and reporting columns. Each column is identified by a two-character abbreviation, such as CN for City Name. Click DCT in Figure 1-33 on page 62 to view its contents. Example 1-2 shows the contents of the Dictionary Table for the USPREP rule set.

Example 1-2 Contents of the Dictionary table for the USPREP rule set

```
;;QualityStage v8.0
\FORMAT\ SORT=N
;------------------------------------------------------------------------------
; USPREP Dictionary File
;------------------------------------------------------------------------------
; Total Dictionary Length = 600
;------------------------------------------------------------------------------
; Domain Fields
;------------------------------------------------------------------------------
NameDomain  C 100 S NameDomain  ;0001-0100
AddressDomain  C 100 S AddressDomain  ;0101-0200
AreaDomain  C 100 S AreaDomain  ;0201-0300
;------------------------------------------------------------------------------
; Reporting Fields
;------------------------------------------------------------------------------
Field1Pattern  C 20 S Field1Pattern  ;0301-0320
Field2Pattern  C 20 S Field2Pattern  ;0321-0340
Field3Pattern  C 20 S Field3Pattern  ;0341-0360
Field4Pattern  C 20 S Field4Pattern  ;0361-0380
Field5Pattern  C 20 S Field5Pattern  ;0381-0400
Field6Pattern  C 20 S Field6Pattern  ;0401-0420
InputPattern  C 88 S InputPattern  ;0421-0508
OutboundPattern  C 88 S OutboundPattern  ;0509-0596
UserOverrideFlag  C 2 S UserOverrideFlag  ;0597-0598
CustomFlag  C 2 S CustomFlag  ;0599-0600
```

- Patterns

  The Pattern-Action file consists of a series of patterns and associated actions. The patterns from an input file are processed in the order as they appear in the pattern-action file. The contents of the Patterns Table for the USPREP rule set is too large to be shown here.

- Overrides

  Click **Overrides** in the Rules Management window in Figure 1-33 on page 62 to add, copy, edit, or delete overrides to rules sets.

- Lookup Tables

  Click **Reference Tables** in the Rules Management window in Figure 1-33 on page 62 to view information about the rule set.

Rule set customization with override object types

Rule sets define the way that IBM WebSphere QualityStage processes the input data. You can extend their content by using override object types.

The override object types let you do the following tasks:

- Add new rules to the repository.
- Create custom conditioning rules (that are stored in a separate folder).

The preprocessor, domain-specific, and validation override object types include the rule set file names with three characters appended indicating each override object type. For example, the WAVES and MNS override object types include eight characters. The first two characters indicate the ISO country code abbreviation, the second two characters are “MN” for multinational, the third two characters are “AD” for address, and the last two characters indicate the override object type.

When you first install IBM WebSphere QualityStage, the override object types are empty.

Selecting override object types to modify rule sets

In the Rules Management window as shown in Figure 1-33 on page 62, Overrides provides editing windows to customize rule sets for your business requirements.

IBM WebSphere QualityStage provides five methods of rule set overrides as follows:

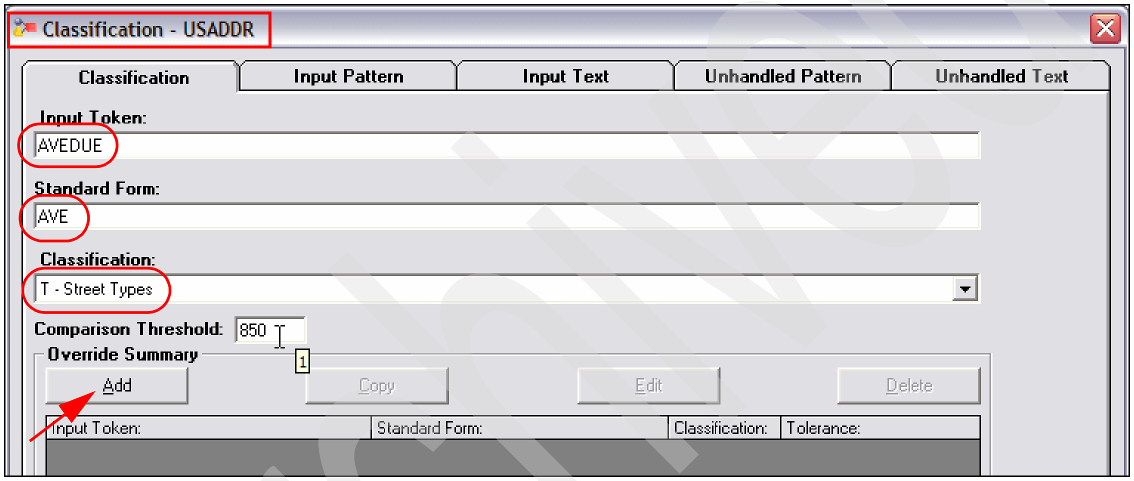
- Classification

  You can modify the classification table of any rule set using the Designer client. Figure 1-34 on page 65 and Figure 1-35 on page 66 show the classification table override for the domain-specific USADDR rule set.

In the Input Token field, type the word (AVEDUE) for which you want to override the classification as it appears in the input file. In the Standard Form field, type the standardized spelling (AVE) of the token.[8]

From the Classification menu, select the one-character tag (T- Street Types) that indicates the class of the token word. In the Comparison Threshold field, type a value (850)[9] that defines the degree of uncertainty to tolerate in the spelling of the token word. Click **Add** in Figure 1-34 to add the override to the pane at the bottom of the window as shown in Figure 1-35 on page 66.

After you create the override (and provision it), the next time you run the rule set, the word tokens are classified with the designations you specified and appear with the appropriate standard form.



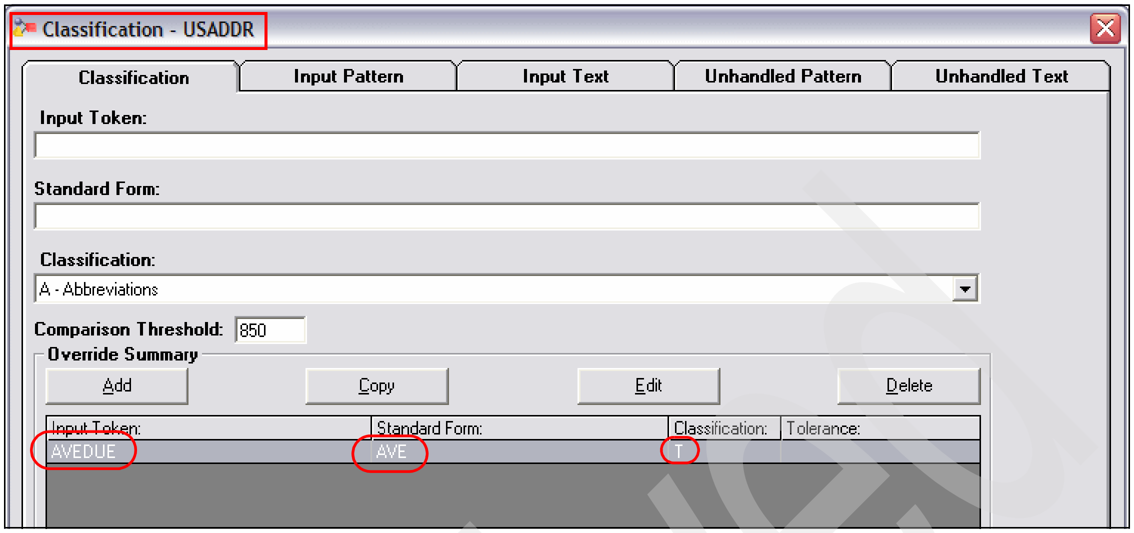
Input Token:
AVEDUE
Standard Form:
AVE
Classification:
T - Street Types
Comparison Threshold: 850

Input Token:	Standard Form:	Classification:	Tolerance:

Figure 1-34 Classification override of AVEDUE input token 1/2

8 Determined by browsing the Classification table

9 850 corresponds to allowing for one or two letter transformations depending upon the length of the word — other supported values are blanks, 700, 750 and 950



Input Token:

Standard Form:

Classification: A - Abbreviations

Comparison Threshold: 850

Override Summary

Input Token:	Standard Form:	Classification:	Tolerance:
AVEDUE	AVE	T	

Figure 1-35 Classification override of AVEDUE input token 2/2

- Input and field pattern override for domain-preprocessor rule set

  For the domain-preprocessor rule sets, the input pattern and column pattern overrides modify the override object (*.IPO) for the input pattern and the override object (*.CPO) for the column pattern.

  The pattern overrides have the following characteristics:

  – With the input pattern override, you can specify token overrides that are based on the input pattern. The input pattern overrides take precedence over the pattern-action file. Input pattern overrides are specified for the entire input pattern.

  – With the column pattern override, you can specify token overrides that are based on one column pattern. These overrides can be specified only for an entire column pattern.

  > **Note:** Column pattern overrides work on columns named field1pattern_USPREP, field2pattern_USPREP, and so forth. The input pattern comprise the collection of all the individual field patterns.

  The input pattern override process is similar to that described in Figure 1-42 on page 74 through Figure 1-44 on page 75.

- Input and field text override for domain-preprocessor rule set

  For the domain-preprocessor rule sets, the input text and column text overrides modify the override object (*.ITO) for the input text and the override object (*.CTO) for the column text.

  The input and column text objects have the following characteristics:

  – With the input text override, you can specify token overrides that are based on all the input text. These overrides take precedence over the pattern-action file. Because they are more specific, input text overrides take precedence over input pattern overrides. Input text overrides are specified for the entire input text string. Partial string matching is not allowed.

  – With the column text override, you can specify token overrides that are based on the field strings. Because they are more specific, column text overrides also take precedence over column pattern overrides. You can specify column text overrides only for an entire column text string. You cannot match a partial string within a column.

  You would add an override to the input text object if you wanted to change the domain from one kind to another. For example, assume an input text that has the following string:

  ```
  ZQNAMEZQ JAMES MASON ZQADDRZQ LOS ANGELES
  ```

  where `ZQNAMEZQ` is the name domain delimiter, and `ZQADDRZQ` is the address domain delimiter

  The domain-preprocessor override objects and their abbreviations let you specify your own custom rules. These objects are also accessible in the IBM WebSphere QualityStage interface. Table 1-5 describes the object types that override the standardize domain-preprocessor rule sets.

Table 1-5 Object types that override the standardize domain-preprocessor rule sets

Domain-preprocessor overrides	**Object type abbreviation**	**Example (U.S.)**
Classification	not applicable	USPREP.CLS
Input pattern overrides	IPO	USPREP.IPO
Input text overrides	ITO	USPREP.ITO
Column pattern overrides	CPO	USPREP.CPO
Column text overrides	CTO	USPREP.CTO

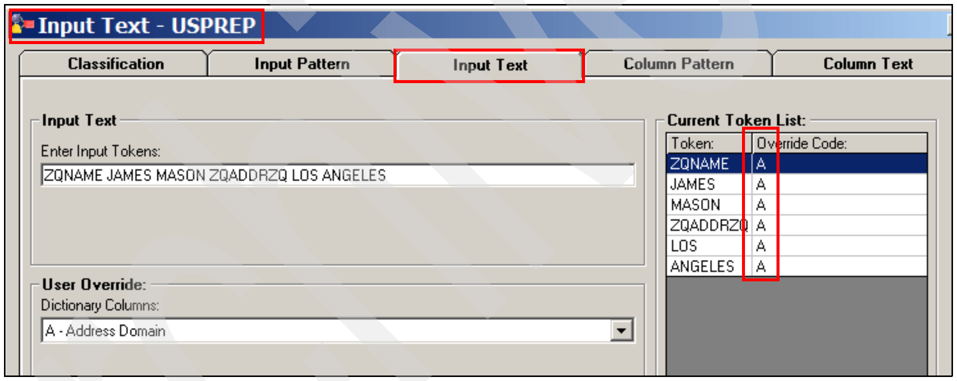
Figure 1-36 shows the override code for each token as being “A” for address domain. You can override this input text to have the tokens “LOS” and “ANGELES” assigned to the area domain (override code “R”) as shown in Figure 1-37 on page 68. Click **Add** in Figure 1-37 to create the input text override as shown in Figure 1-38 on page 69.

You can then test this override as shown in Figure 1-39 on page 70 through Figure 1-41 on page 71.

a. Select Overrides and click **Test** as shown in Figure 1-39 on page 70.

b. Select the Name delimiter and type in JAMES MASON as input string, and select Area delimiter and type LOS ANGELES as input string as shown in Figure 1-40 on page 71. Click **Test This String** to view the results.

c. Figure 1-41 on page 71 shows the results of the test. The input pattern prior to the override is NF+A++ as highlighted. Because of the override, the output pattern is NNNARR, which shows that the override is working as designed.

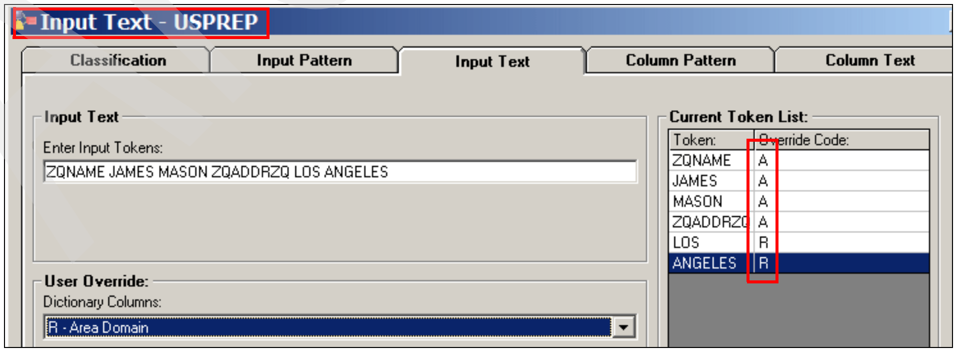


Enter Input Tokens: ZQNAME JAMES MASON ZQADDRZQ LOS ANGELES

Dictionary Columns: A - Address Domain

Token:	Override Code:
ZQNAME	A
JAMES	A
MASON	A
ZQADDRZQ	A
LOS	A
ANGELES	A

Figure 1-36 Input text override example 1/6

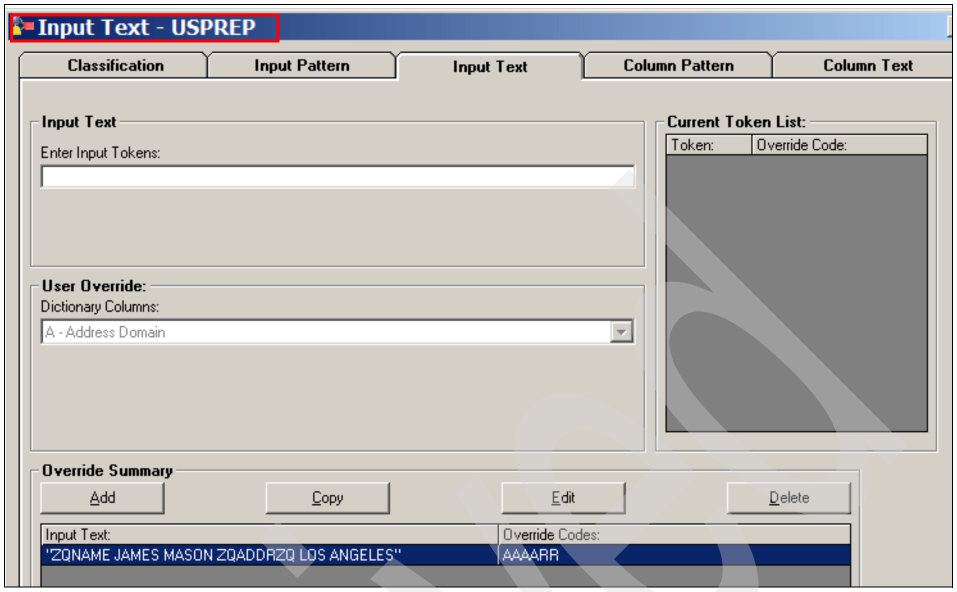


Enter Input Tokens: ZQNAME JAMES MASON ZQADDRZQ LOS ANGELES

Dictionary Columns: R - Area Domain

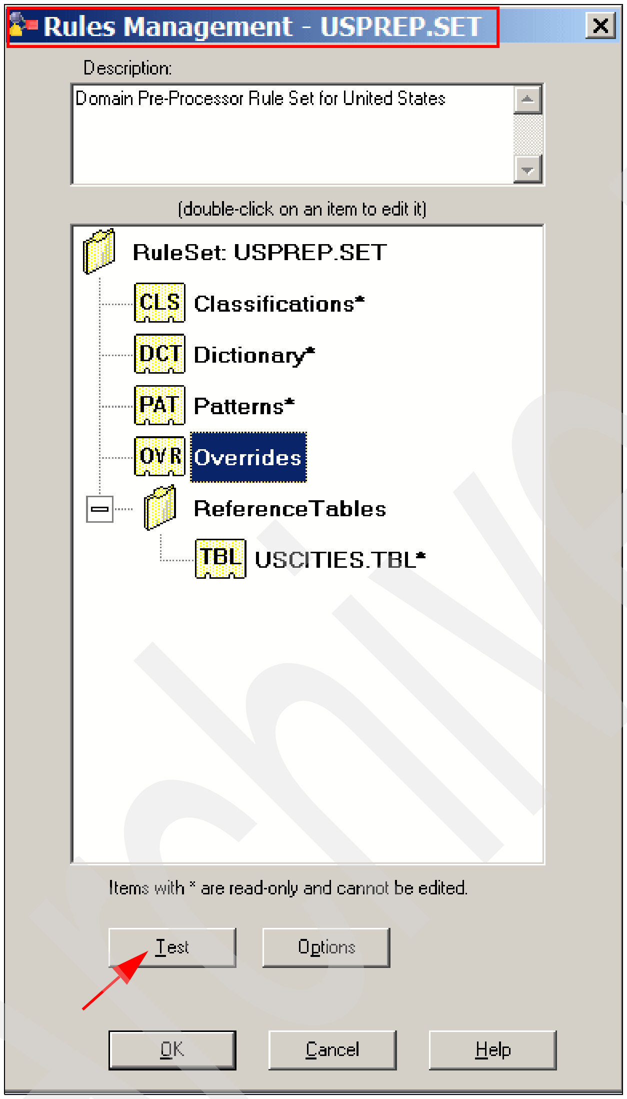
Token:	Override Code:
ZQNAME	A
JAMES	A
MASON	A
ZQADDRZQ	A
LOS	R
ANGELES	R

Figure 1-37 Input text override example 2/6



Input Text:	Override Codes:
"ZQNAME JAMES MASON ZQADDRZQ LOS ANGELES"	AAAARR

Figure 1-38 Input text override example 3/6


Rules Management - USPREP.SET

Description:

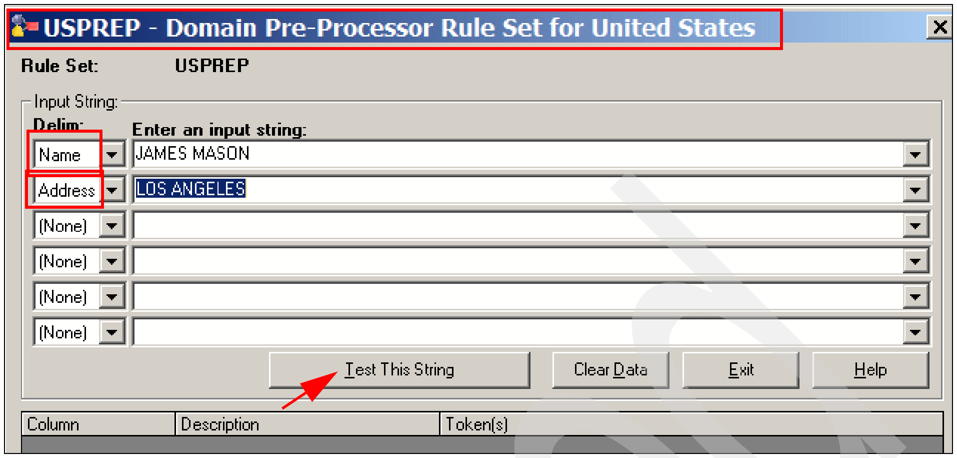
Domain Pre-Processor Rule Set for United States

(double-click on an item to edit it)

RuleSet: USPREP.SET
- Classifications*
- Dictionary*
- Patterns*
- Overrides
- ReferenceTables
  - USCITIES.TBL*

Items with * are read-only and cannot be edited.

Figure 1-39 Input text override example 4/6



USPREP - Domain Pre-Processor Rule Set for United States

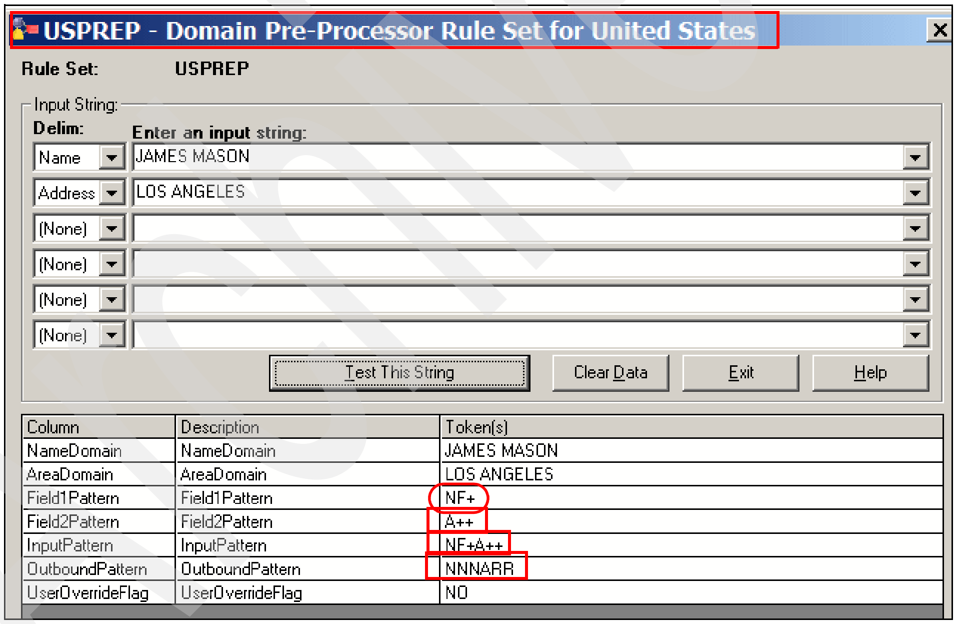
Rule Set: USPREP

Input String:

Delim:	Enter an input string:
Name	JAMES MASON
Address	LOS ANGELES
(None)	
(None)	
(None)	
(None)	

Column	Description	Token(s)

Figure 1-40 Input text override example 5/6



USPREP - Domain Pre-Processor Rule Set for United States

Rule Set: USPREP

Input String:

Delim:	Enter an input string:
Name	JAMES MASON
Address	LOS ANGELES
(None)	
(None)	
(None)	
(None)	

Column	Description	Token(s)
NameDomain	NameDomain	JAMES MASON
AreaDomain	AreaDomain	LOS ANGELES
Field1Pattern	Field1Pattern	NF+
Field2Pattern	Field2Pattern	A++
InputPattern	InputPattern	NF+A++
OutboundPattern	OutboundPattern	NNNARR
UserOverrideFlag	UserOverrideFlag	NO

Figure 1-41 Input text override example 6/6

- Input and unhandled pattern override for domain-specific rule set

  Input pattern and unhandled pattern overrides for domain-specific rule sets (such as USNAME) are used to modify the input pattern (*.IPO) and unhandled pattern (*.UPO) override objects.

  The pattern overrides have the following characteristics:

  – With the input pattern override, you can specify token overrides that are based on the input pattern. The input pattern overrides take precedence over the pattern-action file. Input pattern overrides are specified for the entire input pattern.

  – With the unhandled pattern override, you can specify token overrides that are based on the unhandled pattern. Unhandled pattern overrides work on tokens that are not totally or only partly processed by the pattern-action file. These overrides are specified only for an entire unhandled pattern. The overrides are not specified for partial pattern matching.

  For example, assume that you had a field that contained the text `1234 SNELL Avenue` that generated the pattern `^+T`. Using input pattern overrides, you can designate this pattern to have the type and values as shown in Table 1-6.

Table 1-6 Designation of pattern ^+T

Pattern token	Type	Value
^	House number	Original
+	Street name	Original
T	Street type	Standard

The domain-specific override objects and their abbreviations let you specify your own custom conditioning rules. These objects are also accessible in the QualityStage interface. Table 1-7 describes the object types used to override the domain-specific rule sets.

Table 1-7 Object types that override the standardize domain-specific rule sets

Domain-preprocessor overrides	Object type abbreviation	Example (U.S.)
Classification	not applicable	USADDR.CLS
Input pattern overrides	IPO	USADDR.IPO
Input text overrides	ITO	USADDR.ITO
Unhandled pattern overrides	UPO	USADDR.CPO
Unhandled text overrides	UTO	USADDR.CTO

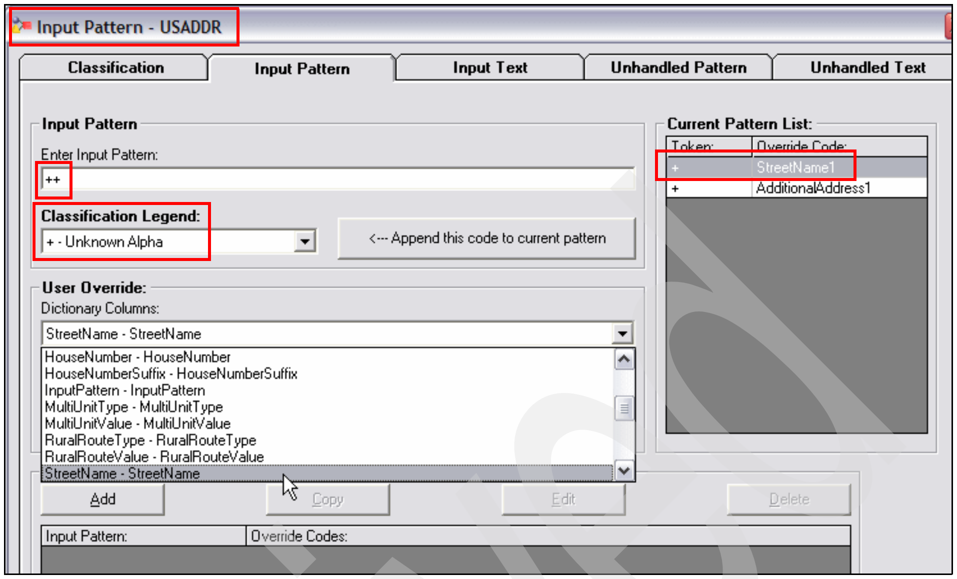
Figure 1-42 on page 74 through Figure 1-44 on page 75 show the overriding of an input pattern “++” from an Unknown Alpha to street addresses. The first “+” is overridden with the Override Code of StreetName - StreetName[10] as shown in Figure 1-42 on page 74. The second “+” is overridden with the Override Code of AddtionalAddress - AddtionalAddress[11] as shown in Figure 1-43 on page 74. All additional tokens are moved to this field.

Note: The suffix 1 in the StreetName override code (StreetName1) is an action code. The Action Codes (0 to 8) are displayed in the Current Pattern List under the Override Code column. The codes are as follows:

0	Select Drop Current Token to drop the current token
1	Select Move Current and the Original Value and Leading space for the specified data type
2	Select Move Current and the Standard Value and Leading space for the specified data type
3	Select Move Current and Original Value and No leading space for the specified data type
4	Select Move Current and Standard Value and No leading space for the specified data type
5	Select Move All Remaining and Original Values and Leading space for the specified data type
6	Select Move All Remaining and Standard Value and Leading space for the specified data type
7	Select Move All Remaining and Original Values and No leading space for the specified data type
8	Select Move All Remaining and Standard Values and No leading space for the specified data type

Click **Add** to add the input pattern overrides as seen in the Override Summary in Figure 1-44 on page 75.

[10] StreetName in the Override Code classifies this token as a street name
[11] AddtionalAddress in the Override Code classifies this token as an additional field



Input Pattern - USADDR

Input Pattern
Enter Input Pattern: ++
Classification Legend: + - Unknown Alpha

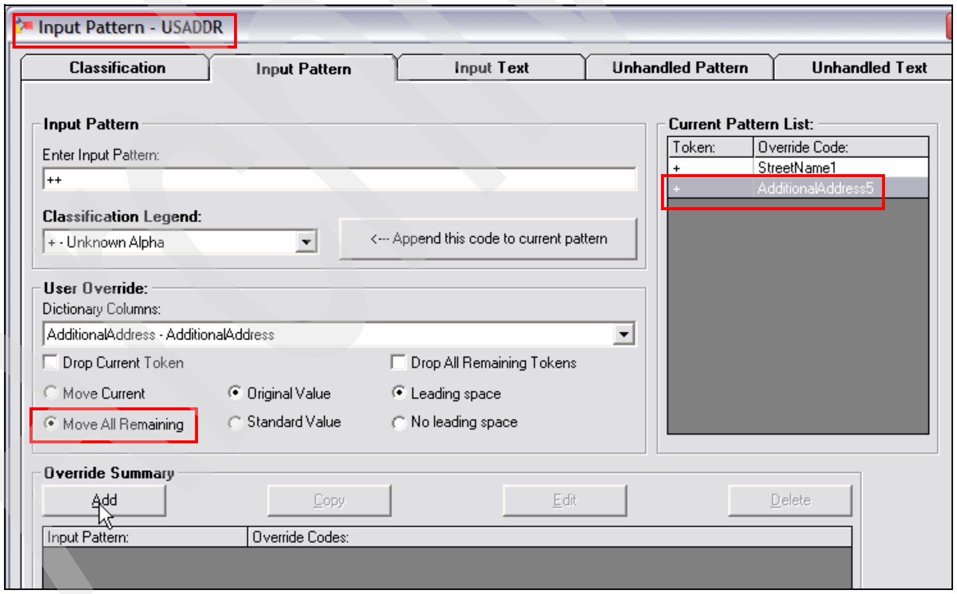
User Override:
Dictionary Columns: StreetName - StreetName

HouseNumber - HouseNumber
HouseNumberSuffix - HouseNumberSuffix
InputPattern - InputPattern
MultiUnitType - MultiUnitType
MultiUnitValue - MultiUnitValue
RuralRouteType - RuralRouteType
RuralRouteValue - RuralRouteValue
StreetName - StreetName

Current Pattern List:

Token:	Override Code:
+	StreetName1
+	AdditionalAddress1

Figure 1-42 Input pattern override for domain-specific rule set 1/3



Input Pattern - USADDR

Input Pattern
Enter Input Pattern: ++
Classification Legend: + - Unknown Alpha

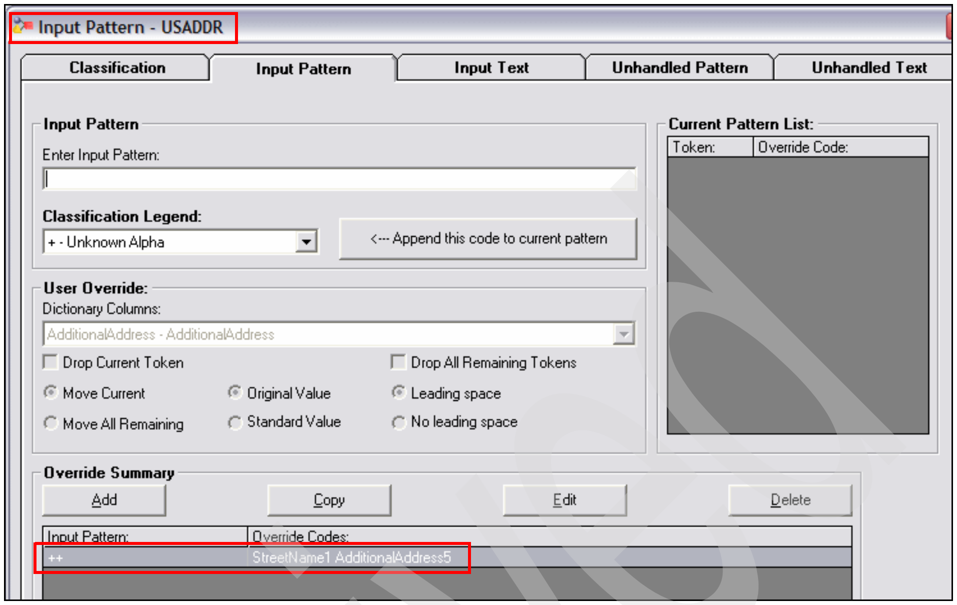
User Override:
Dictionary Columns: AdditionalAddress - AdditionalAddress

Drop Current Token
Drop All Remaining Tokens
Move Current [✓ Move All Remaining]
[✓ Original Value] Standard Value
[✓ Leading space] No leading space

Current Pattern List:

Token:	Override Code:
+	StreetName1
+	AdditionalAddress5

Figure 1-43 Input pattern override for domain-specific rule set 2/3



Classification Legend: + - Unknown Alpha

Dictionary Columns: AdditionalAddress - AdditionalAddress

[ ] Drop Current Token [ ] Drop All Remaining Tokens

[✓ Move Current] Move All Remaining

[✓ Original Value] Standard Value

[✓ Leading space] No leading space

Input Pattern:	Override Codes:
++	StreetName1 AdditionalAddress5

Figure 1-44 Input pattern override for domain-specific rule set 3/3

- Input and unhandled text override for domain-specific rule set

  For the domain-specific rule sets, the input text and unhandled text overrides modify the override object (*.ITO) for input text and the override object (*.UTO) for unhandled text.

  The input and unhandled text objects have the following characteristics:

  – With the input text override, you can specify token overrides that are based on *all* the input text. These overrides take precedence over the pattern-action file. Because they are more specific, input text overrides take precedence over input pattern overrides. Input text overrides are specified for the entire input text string. Partial string matching is not allowed.

  – With the unhandled text override, you can specify rule overrides that are based on the unhandled text string. Unhandled text overrides work on tokens that are not totally or only partly processed by the pattern-action file. Because they are more specific, unhandled text overrides take precedence over unhandled pattern overrides. Unhandled text overrides can be specified only for the entire unhandled text string. Partial string matching is not allowed.

  For example, assume that you had a field that contained the text `100 Summer Street Floor 15`, and it contains two tokens (Street and Floor) that use standard values from the classification object. The remaining tokens (100, Summer, and 15) are not associated with standard values from the classification object and use their original data values.

The input text override process is somewhat similar to that described in Figure 1-36 on page 68 through Figure 1-41 on page 71.

Standardization rule set tester

The tester lets you test quickly a selected standardize rule set against a one-line test string (a single record) before you run the rule set against an entire file.

This time-saving option is especially helpful when you plan to use a large input file. When you test the rule set, you can ensure that the resulting data file performs the way you expect.

You can test domain-preprocessor rule sets, domain-specific rule sets, and validation rule sets with the tester.

Figure 1-39 on page 70 through Figure 1-41 on page 71 show an example of the rule set tester.

Note: The tester is not available for WAVES and Multinational standardize rule sets.

Choosing the appropriate override method

Broad guidelines for choosing an appropriate override method are summarized in Table 1-8 on page 77. However, you need to evaluate these guidelines in the context of your organization's specific requirements.

The general guidelines are as follows:

- Minimize the number of overrides required to achieve the desired result, so as to avoid the possibility of negative side effects on normal processing.

  Another consideration is the impact of overrides on system performance. The more the number of overrides, the greater the impact on system performance.
- Apply classification overrides to key words to provide additional context in a pattern. These overrides change the resulting pattern of an input string such that a pattern previously unrecognized can become recognizable and handled by the standardization pattern action rules. As a rule of thumb, apply these overrides to frequently occurring key tokens or to tokens to which you want to apply a standard value on output. Apply classification overrides before Pattern Overrides.
- Do not perform column pattern and column text overrides without expert assistance.

Table 1-8 Guidelines for choosing a particular override method

Override method	**Considerations**
Classification	Can be used in the Investigate or Standardize stages. Use the classification override when you see an unclassified token that you know definitively belongs to one of the defined classes in the Classification table for that rule set. Doing so alters the input pattern and can increase the chance that the rule set can handle the input pattern. However, classifying all tokens in the input source is not the goal because this can have a detrimental effect overall. As the number of classification values increase, it might increase significantly the number of patterns necessary to handle the new classifications. Use the classification override in the Domain Pre-processor rule set in the Investigate stage to ensure that all the words go into the correct name, address, and area domains. While other overrides such as input pattern can be performed in the Domain Pre-processor rule set in the Investigate stage, we recommend that you only perform input pattern overrides in the Standardize stage.
Input pattern override	Input pattern overrides should be used if the rule set does not handle the patterns to your satisfaction. Use this if you are not satisfied with the actions (or lack of it) done by the rule set, and would like to manage the string processing yourself. Input pattern overrides target *all* records that match the pattern being overridden. It matches the entire input pattern including any literals inserted. You must ensure that the records that map to a specific input pattern (that is being overridden) actually have the same underlying data format (type and sequence of tokens such as firstname, middlename, lastname). This can be determined by using the Investigate stage with the appropriate *C*, *T*, and *X* masks. Input pattern override works well if a large number of records in the input source have the same input pattern. This input pattern can be determined through the Investigate stage using the T mask on the output of the Standardize stage.
Column pattern override	Column pattern override is the same as the input pattern override except that it operations on a portion (fields 1 through 6 in the Dictionary table of the rule set as shown in Example 1-2 on page 63) of the entire input pattern. Column pattern overrides work best if specific patterns are well contained in a segment of the data. You can use column pattern override to target a specific portion of an input string. In certain cases, this might help you perform fewer overrides. For example, a particular pattern to be overridden appears in field2 in a number of (entire) input patterns. Rather than override every the distinct input pattern in which this (subset) pattern in field2 appears, you can limit the override to field2 using Column pattern override. This is not generally recommended without expert assistance.

Override method	Considerations
Input text override	Input text override is similar to an input pattern override except that you specify the text string to be overridden, and you need to include the ZQ-delimiters in the text string. This will ensure that the override targets one and only one specific string. Input text override works well with text strings that do not fit standard classifications. Examples of strings that are good candidates for input text override include DO NOT USE and UNKNOWN ADDRESS. You can also use input text override to target specific text strings before they are handled by a general input pattern override, because input text overrides precede input pattern overrides.
Column text override	As in the case of input text override, column text override can be used to address common strings that are not easily classified. This is not generally recommended without expert assistance.

1.7.2 WAVES, CASS, DPID, SERP, MNS, Geolocator, Country rule set

International addresses can be standardized using stages such as IBM WebSphere QualityStage Worldwide Address Verification and Enhancement System (WAVES), IBM WebSphere QualityStage DPID (Australian Government Verification), IBM WebSphere QualityStage SERP (Canadian Government Verification), and Multinational Standardize stage (MNS).

These features are all separately priced features of IBM WebSphere QualityStage.

We provide a brief description of these stages in this section.

WAVES

IBM WebSphere QualityStage Worldwide Address Verification and Enhancement System (WAVES) enables users to standardize, verify, and correct extensive, multinational address data against postal reference files for better data integration and customer communication worldwide.

The WAVES stage provides the following processing to the input data:

- Corrects typographical and spelling errors in the input data.
- Matches a corrected input address against an address in a country-specific postal reference file by using probabilistic matching.
- Assigns the postal record data to the appropriate field in the output file and substitutes the erroneous and missing input data with the verified postal data. Appends the verified record to the original record.

Worldwide location data can be processed at three levels. Table 1-9 summarizes WAVES country support for standardization and verification.

Table 1-9 WAVES processing

Level	Type	Description	City Level Countries Covered	Street Level Countries Covered
1	Standardization	Identify, separate, and standardize the address	233 (WAVES)	71 (WAVES)
2	Verification	Includes Level 1 + validate, correct, and fill missing values based on matching the input to an “authoritative” reference source	215 (WAVES)	70 (WAVES)
3	Certification	Includes Level 1 + validate, correct, and format the output based on the reference data and mandated rules from a specific Government postal authority using software that has met their test criteria missing values based on matching the input to an “authoritative” reference source	► U.S. (CASS) ► CANADA (SERP) ► AUSTRALIA (DPID)	► U.S. (CASS) ► CANADA (SERP) ► AUSTRALIA (DPID)

IBM WebSphere QualityStage also has add on modules that provide certification solutions for the U.S. (CASS), Canada (SERP) and Australia (DPID):

- CASS

  IBM WebSphere QualityStage CASS (Government Certified U.S. Address Correction) enables users to make changes or append information to ensure that addresses qualify for U.S. Postal Service (USPS) mail rate discounts. Certified by the USPS, it ensures the accuracy of critical U.S. address information in corporate databases.

- SERP

  IBM WebSphere QualityStage SERP (Canadian Government Verification) enables users to correct and validate information to ensure that addresses qualify for Canada Post mail rate discounts, which is recognized by Canada Post Corporation as meeting all requirements of its Software Evaluation and Recognition Program (SERP).

- DPID

  IBM WebSphere QualityStage DPID (Australian Government Verification) enables users to make changes or append the appropriate Delivery Point Identifier (DPID) to ensure addresses qualify for Australian Post mail rate discounts, which is certified by the Australian Post as meeting all requirements of its Address Matching Approval System (AMAS).

MNS

The Multinational Standardize stage (MNS) standardizes and verifies international address data.

The MNS stage uses country names, abbreviations, or ISO country codes in the input file to apply country-appropriate standardization rules.

Thus, you can standardize all the records without having to consolidate records from each country into separate files and to standardize the records using country-specific rule sets.

For most countries, the MNS stage does the following tasks:

- Separates street-level address information from city-level information
- Assigns city, locality, province/state, postal code, and postal code add-on to separate fields
- Assigns ISO country codes (2- and 3-byte versions)
- Assigns city name phonetics with enhanced New York State Information and Intelligence Systems (NYSIIS) and reverse Soundex to match incorrectly spelled city names

MNS stage standardizes address files for city and street data in one step.

Geolocator

IBM WebSphere QualityStage GeoLocator provides multinational geocoding for spatial information management and location-based services. It provides latitude, longitude, census tract and block information about addresses that allows users to build business intelligence and Internet applications that provide highly accurate location and location-based information.

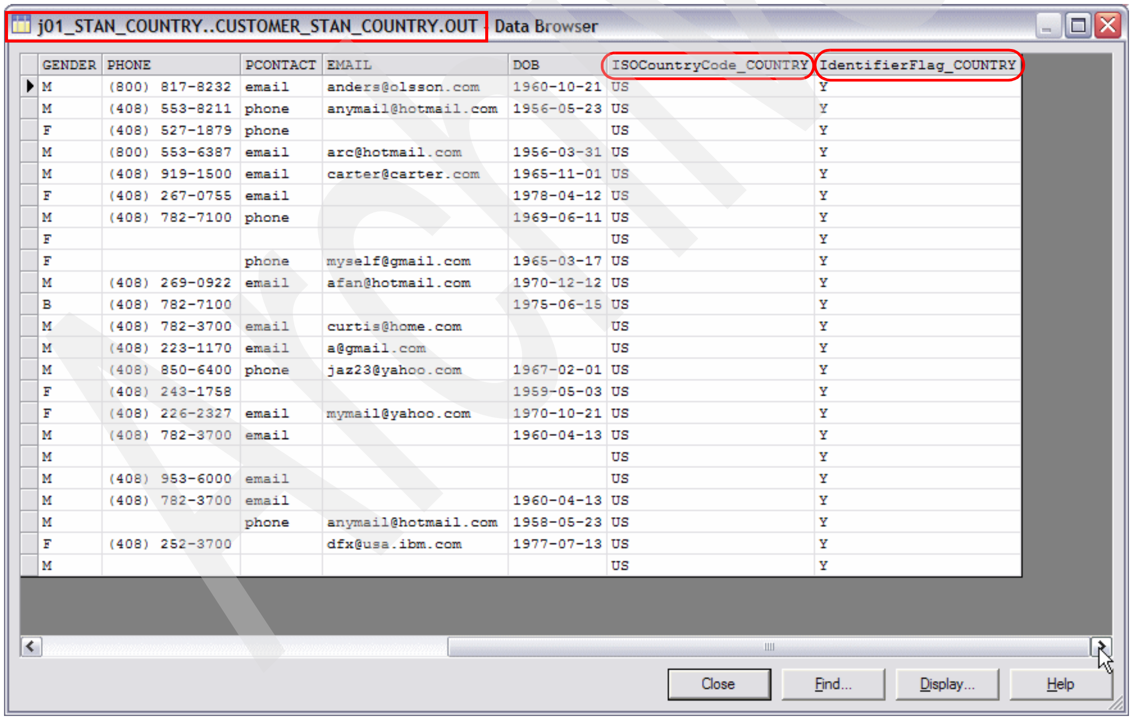
Country Rule Set

The country rule set is used to condition files for one country. The output file contains the country code and an identifier flag.

The country rule set, named Country Group, prepares international input files for standardization at the individual country level. The rule set creates an output file in which the following columns are appended to the beginning of each input record:

- A 2-byte ISO country code. The code is associated with the geographic origin of the record's address and area information.
- An identifier flag. The values are:
  - *Y* indicates that the rule set identifies the country.
  - *N* indicates that the rule set cannot identify the country and uses the value of the default country delimiter.

Figure 1-45 shows a sample output file of a Country Rule Set with the ISOCountryCode and IdentifierFlag.



GENDER	PHONE	PCONTACT	EMAIL	DOB	ISOCountryCode_COUNTRY	IdentifierFlag_COUNTRY
M	(800) 817-8232	email	anders@olsson.com	1960-10-21	US	Y
M	(408) 553-8211	phone	anymail@hotmail.com	1956-05-23	US	Y
F	(408) 527-1879	phone			US	Y
M	(800) 553-6387	email	arc@hotmail.com	1956-03-31	US	Y
M	(408) 919-1500	email	carter@carter.com	1965-11-01	US	Y
F	(408) 267-0755	email		1978-04-12	US	Y
M	(408) 782-7100	phone		1969-06-11	US	Y
F					US	Y
F		phone	myself@gmail.com	1965-03-17	US	Y
M	(408) 269-0922	email	afan@hotmail.com	1970-12-12	US	Y
B	(408) 782-7100			1975-06-15	US	Y
M	(408) 782-3700	email	curtis@home.com		US	Y
M	(408) 223-1170	email	a@gmail.com		US	Y
M	(408) 850-6400	phone	jaz23@yahoo.com	1967-02-01	US	Y
F	(408) 243-1758			1959-05-03	US	Y
F	(408) 226-2327	email	mymail@yahoo.com	1970-10-21	US	Y
M	(408) 782-3700	email		1960-04-13	US	Y
M					US	Y
M	(408) 953-6000	email			US	Y
M	(408) 782-3700	email		1960-04-13	US	Y
M		phone	anymail@hotmail.com	1958-05-23	US	Y
F	(408) 252-3700		dfx@usa.ibm.com	1977-07-13	US	Y
M					US	Y

Figure 1-45 Output of Country Rule Set

After you create this output file, you can use a Filter (or Transformer) stage to create a file that contains only one country. You can use the file with a domain-preprocessing rule set for the appropriate country.

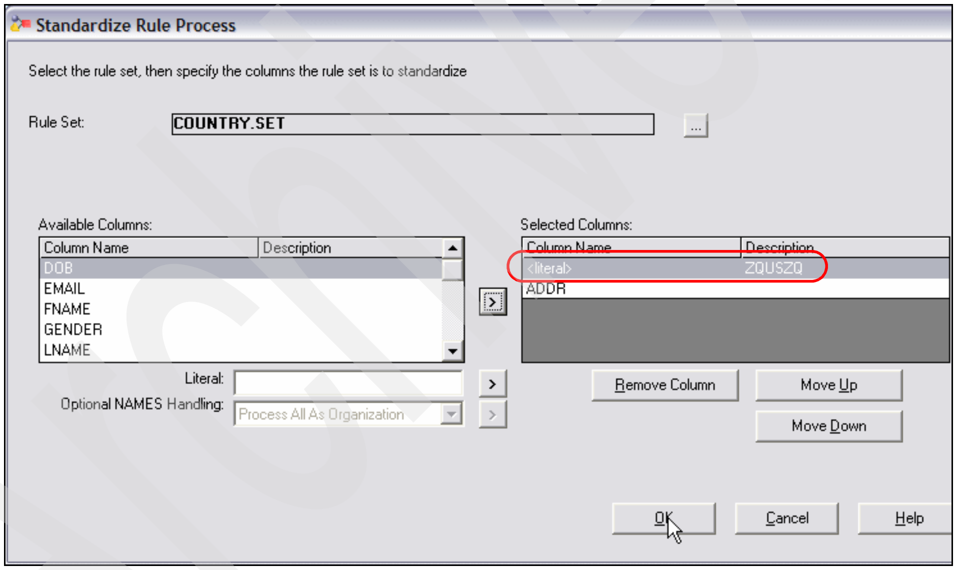
If the country rule set cannot identify the country of origin of a record, it uses the default country code value supplied as a literal. This delimiter consists of a default country code that you must define before you run the rule in a job. Use the country code that represents the majority of the records.

The delimiter name is as follows:

```
ZQ<two-character ISO code>ZQ
```

For example, you use *ZQUSZQ* as a U.S. delimiter as shown in Figure 1-46.

The ISO country codes list provides all the codes. This delimiter is inserted as a literal when you define the columns from the input file.


Standardize Rule Process

Select the rule set, then specify the columns the rule set is to standardize

Rule Set: COUNTRY.SET

Available Columns:

Column Name	Description
DOB	
EMAIL	
FNAME	
GENDER	
LNAME	

Selected Columns:

Column Name	Description
<literal>	ZQUSZQ
ADDR	

Literal:

Optional NAMES Handling: Process All As Organization

Figure 1-46 U.S. delimiter for country code

1.8 Match stage

Data matching finds records in a single data source or independent data sources that refer to the same entity (such as a person, organization, location, product, or material) even if there is no predetermined key.

To increase its usability and completeness, data can be consolidated or linked (matched) along any relationship, such as a common person, business, place, product, part, or event. You can also use matching to find duplicate entities that are caused by data entry violations or account-oriented business practices.

During the data matching stage, IBM WebSphere QualityStage takes the following actions:

- Identifies duplicate records (such as customers, suppliers, products, or parts) within one or more data sources
- Provides householding for individuals (such as a family or group of individuals at a location) and householding for commercial entities (multiple businesses in the same location or different locations)
- Enables the creation of match groups across data sources that might or might not have a predetermined key

There are two types of match stage:

- *Unduplicate match* locates and groups all similar records within a single input data source. This process identifies potential duplicate records, which might then be removed.

  An example is the need to eliminate duplicates from a consolidation of mailing lists purchased from multiple sources.

  The Unduplicate match stage accepts the following inputs:

  – Source data from any parallel database, file or processing stage. Typically, the source is standardized data from the Standardize stage.

  – Frequency information for that data as generated by the Match Frequency stage. This stage takes input from a database, file, or processing stage, and generates the frequency distribution of values for columns in the input data. Match frequency is described in detail in "Match Frequency Stage" on page 91.

    Match frequency results is used in computing "u" probabilities as described in "Matching step" on page 89.

  – Match specification to group and match the data. A Match specification provides details of blocking and matching columns for matching. Match specification is described in "Match Designer tool" on page 95.

  The Unduplicate match stage delivers up to four outputs as follows:

  – Match contains master records
  – Clerical has records that fall in the clerical range
  – Duplicate contains records that are above the match cutoff
  – Residual contains records that are not associated with any of the records

The Unduplicate match stage is covered in greater detail in "Unduplicate Match stage" on page 104.

- *Reference Match* identifies relationships among records in two data sources.

  An example of many-to-one matching is matching the ZIP codes in a customer file with the list of valid ZIP codes. More than one record in the customer file can have the same ZIP code in it.

  The Reference match stage takes up to the following inputs:

  – Source data from any parallel database, file or processing stage. Typically, the source is standardized data from the Standardize stage.

  – Reference source against which the source data is matched

  – Frequency information for that data as generated by the Match Frequency stage as described for the Unduplicate match stage.

  – Match specification to group and match the data as described for the Unduplicate match stage.

  The Reference match stage delivers up to six outputs as follows:

  – Match contains matched records for both inputs

  – Clerical has records that fall in the clerical range for both inputs

  – Data Duplicate contains duplicates in the data source

  – Reference Duplicate contains duplicates in the reference source

  – Data Residual contains records that are non-matches from the data input

  – Reference Residual contains records that are non-matches from the reference input

  The Reference match stage is covered in greater detail in "Reference Match stage" on page 108.

Matching is a two-step process: first you block records and then you match them.

Blocking step

Blocking provides a method of limiting the number of pairs to examine. When you partition data sources into mutually-exclusive and exhaustive subsets and only search for matches within a subset, the process of matching becomes manageable.

Basic blocking concepts include:

- Blocking partitions the sources into subsets that make computation feasible.
- Block size is the single most important factor in match performance.
- Blocks should be as small as possible without causing block overflows. Smaller blocks are more efficient than larger blocks during matching.

To understand the concept of blocking, consider a column that contains age data. If there are 100 possible ages, blocking partitions the source data into 100 subsets. The first subset is all people with an age of zero, the next is people with an age of 1, and so on. These subsets are called *blocks*.

If the age values are uniformly distributed, 10 records out of the 1000-record source contain data for people of age 0 on each source, 10 records for people of age 1, and so on.

The pairs of records to be compared are taken from records in the same block:

- The first block consists of all people of age 0 on each data source. This value is 10 times 10 or 100 record pairs.
- The second block consists of all people on each data source with an age of 1.

When the process is complete, you compared 100 (blocks) x 100 (pairs in a block) = 10 000 pairs, rather than the 1 000 000 record pairs that are required without blocking.

You can also combine multiple blocking variables into a single block for a single pass. For example, blocking on age and gender divides the sources into sets of 0-year-old males, 0-year-old females, 1-year-old males, 1-year-old females, and so on.

Blocking creates a set of records that all have the same values in the blocking variables. Records in different blocks (which means they do not have the same values in the blocking variables) are classified as unmatched automatically. If you believe that the unmatched records might still include matches, then a second pass can be defined with the new blocking variables to look for matches in the newly constituted blocks.

For example, if you run a match where age is the blocking variable, you can rematch any records that do not match using another blocking scheme, such as residential postal code. If a record does not match on age in the first pass, it has the possibility to match on the postal code in the second pass. Thus, only those cases that have errors in both the age and postal code columns are unmatched. You can then run a third pass with different blocking variables if you believe there are still matches that have not been identified. Errors on all three blocking variables are unlikely.

Blocking strategies

Blocking strategies are used to limit the number of records to compare.

Smaller blocks are many times more efficient than larger blocks. Use restrictive blocking schemes in the first pass with blocking variables (such as SSN) that are expected to match records definitively. Unmatched records can then be subject to less restrictive blocking variables such as first and last name. Thus progressively, fewer and fewer unmatched records can be subject to matching in subsequent passes that have less and less restrictive blocking variables. The reason for having multiple passes is to provide for cases where the data in the blocking variables is error-prone.[12]

The matcher computes composite weights for all pairs in the set. IBM WebSphere QualityStage then finds the mathematically optimal sets of matches and duplicates on each source.

To do this, the matcher maintains a matrix of weights that can be optimized. This matrix is size limited so the number of records in a set is limited. If the matrix size is exceeded, all records in the set are skipped and must be handled by a subsequent matching pass.

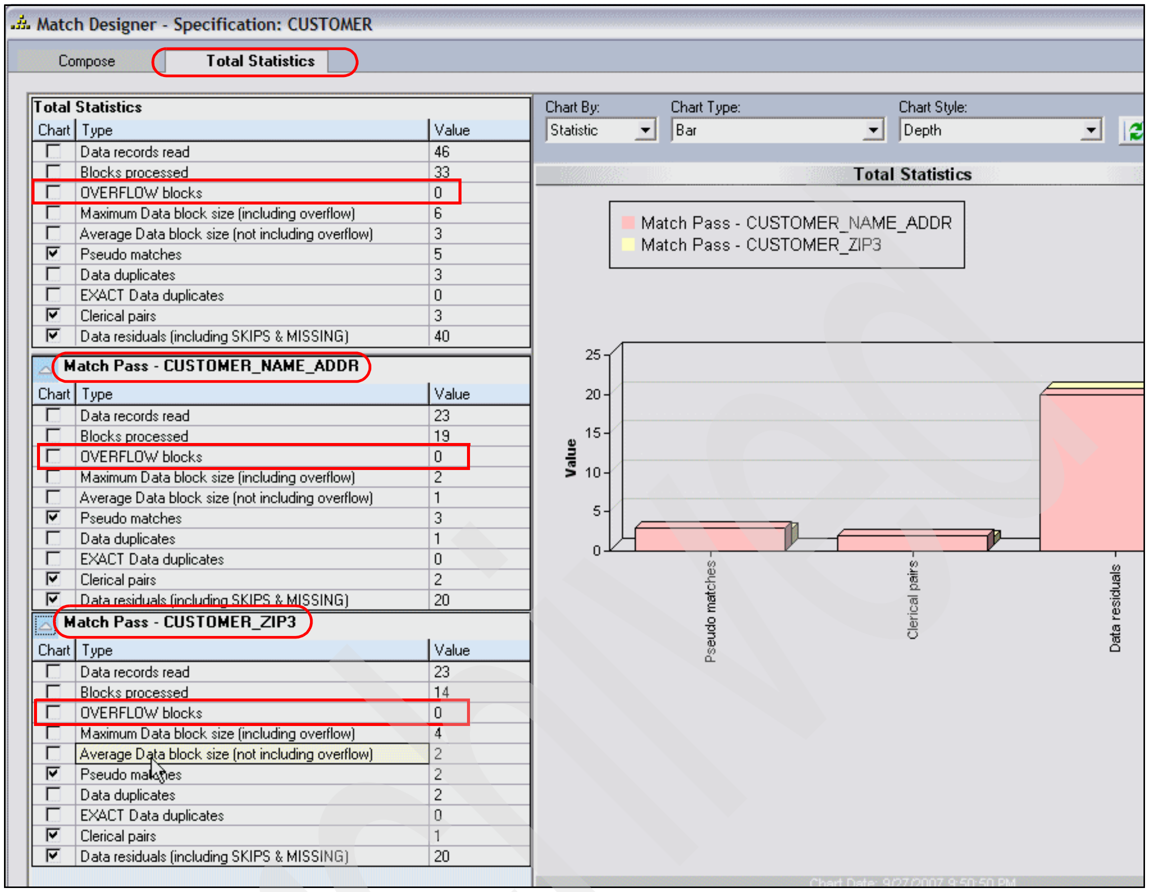
- Avoiding block overflow

  To avoid block overflow, the number of records to be compared in one block must not exceed system memory on the server and the total number of records on either source in one block must not exceed the block overflow setting.

  You can determine whether block overflow has occurred in the Match Designer tool's Total Statistics panel as shown in Figure 1-47 on page 87. This information is provided for each match pass.

  Note: When large volumes of data are involved, a representative sample of data is typically used as input to Match Designer when developing the match specification. In such cases, there is no guarantee that block overflow will not occur when the full volume of data is input to Match Designer.

[12] In other words, the data has a low degree of integrity.



Total Statistics

Chart	Type	Value
☐	Data records read	46
☐	Blocks processed	33
☐	OVERFLOW blocks	0
☐	Maximum Data block size (including overflow)	6
☐	Average Data block size (not including overflow)	3
☑	Pseudo matches	5
☐	Data duplicates	3
☐	EXACT Data duplicates	0
☑	Clerical pairs	3
☑	Data residuals (including SKIPS & MISSING)	40

Match Pass - CUSTOMER_NAME_ADDR

Chart	Type	Value
☐	Data records read	23
☐	Blocks processed	19
☐	OVERFLOW blocks	0
☐	Maximum Data block size (including overflow)	2
☐	Average Data block size (not including overflow)	1
☑	Pseudo matches	3
☐	Data duplicates	1
☐	EXACT Data duplicates	0
☑	Clerical pairs	2
☑	Data residuals (including SKIPS & MISSING)	20

Match Pass - CUSTOMER_ZIP3

Chart	Type	Value
☐	Data records read	23
☐	Blocks processed	14
☐	OVERFLOW blocks	0
☐	Maximum Data block size (including overflow)	4
☐	Average Data block size (not including overflow)	2
☑	Pseudo matches	2
☐	Data duplicates	2
☐	EXACT Data duplicates	0
☑	Clerical pairs	1
☑	Data residuals (including SKIPS & MISSING)	20

Figure 1-47 Block overflow

For many-to-one runs, an overflow occurs only when more than 10 000 reference source records are read in one block.

For large sets, the matcher performs many more comparisons which reduces efficiency. Try to keep the block sizes as small as possible and compensate for errors in blocking by running multiple passes using different blocking variables.

- Applying blocking variables

  The best blocking variables are those with the largest number of values possible and the highest reliability.[13] Base a blocking strategy on the following principles:

  – Make blocking sets as small as possible. A good size is 10 to 20 records per source. Efficiency becomes quite poor when blocks exceed 100 records per source.

  – Use multiple passes to define different sets.

  – Find easy matches quickly, then widen the net in your subsequent passes.

  The following examples provide some considerations about setting up useful blocking strategies. For example:

  – Two data sources containing a date, given name, family name and gender:

    - Pass 1: date and gender
    - Pass 2: Soundex of family name and first two characters of given name
    - Pass 3: Soundex of given name, year and month (from date)

  – Two data sources containing a date, last name, city, and postal code:

    - Pass 1: date and gender
    - Pass 2: postal code
    - Pass 3: Soundex of family name and first two characters of city

  – Two data sources containing national identity number, last name, first name, and birth date:

    - Pass 1: national identity number
    - Pass 2: birth date
    - Pass 3: Soundex of family name, birth year

  Each succeeding pass has diminishing returns. Three passes might not be worthwhile in terms of additional matches found. Notice how sometimes variables are taken whole (birth date) or divided into parts (birth year). Where necessary, this is accomplished by manufacturing the additional columns.

  For sources with limited information, a reverse Soundex code is often useful for blocking. The reverse Soundex is formed by looking at the name backwards and computing a Soundex. For example, the reverse of JONES would be SENOJ. Because the Soundex algorithm preserves the first letter, running a reverse Soundex accounts for errors at the beginning of names.

  Use your imagination to design good strategies. Remember to keep the first passes more restrictive than the later passes.

[13] Reliability is defined as information whose validity is associated with a high degree of confidence.

Important: Convert all missing values to nulls. Failure to do this creates large blocks. For example, if a national identity number were present in only half the records, but the missing values were reported as spaces instead of nulls, all of the blank numbers would form one large block, causing incorrect results and possible block overflow. Similarly, remove bogus values such as UNKNOWN from any variables that are used as blocking variables.

Matching step

The match process only looks for matches in records that have the same values in the blocking columns.

After you create a block of records, Match stage compares columns that you specified as matching columns (in the Match specification) to determine the best match for a record.

To determine whether a record is a match, the Match stage performs the following steps:

1. Calculates a weight for each comparison, according to the probability associated with each column. The Match stage uses two probabilities for each column:
   - The *m* probability reflects the error rate for the column.

     You set the probability that a column agrees provided that the record pair is a match.

     The *m* probability is one minus the error rate of the column. If a column in a sample of matched records disagrees 10% of the time, the *m* probability for this column is 1 - 0.1, or 0.9. The closer the *m* probability is to one, the more critical is a disagreement on the column. You can use a very high *m* probability to force columns that are very important to have a high penalty for disagreeing.
   - The *u* probability is the likelihood that a column agrees at random

     If a *u* probability is high, the weight is low. The *u* probability is the probability that the column agrees, provided that the record pair does not match. The *u* probability is the probability that the column agrees at random.

     For example, the probability that gender agrees at random is 50% of the time. With a uniform distribution, you have four combinations in which gender agrees in two of the four, or a 0.5 *u* probability. If you assign a high *u* probability, a low weight is calculated when a column agrees.

During the match run, the Match stage calculates a u probability for each comparison using the frequency distribution statistics for the column, described in “Match Frequency Stage” on page 91. Usually your concern for u probability is to provide a base point between a rare event (1 out of the total # of records) and a common event (1 out of 2). A value of 0.1 or 0.01 is a typical starting point. If you need to control the u probability (such as columns for an individual identification number, a national ID number, or a Patient Identification Number), specify the vartype to be NOFREQ which indicates that frequency distribution information should not be used by the Match stage.

2. Weights are used in the Match stage to determine by what percentage a column matches and to verify that the matches have a high degree of agreement. For each matching column, the Match stage computes a weight using the following equations:

   – An agreement weight is computed when the comparison between a pair of columns agrees.

     ```
     log2(m probability / u probability)
     ```

   – A disagreement weight is computed when the comparison between the pair of columns disagrees.

     ```
     log2((1 - m probability)/(1 - u probability))
     ```

   The higher the m-probability, the higher the disagreement weight will be for the field not matching because errors are relatively rare events; the lower the u probability the greater the potential weight for the comparison. Additionally, the calculated weight or penalty represents a maximum or minimum range of scoring. Specific match comparisons can generate a score within the range depending on the comparison used.

   The Match stage adds the weights assigned to each column comparison and obtains a composite weight for the record.

   The Match stage then computes the composite weight as the sum of the agreement weight and the disagreement weight of each column. The higher the composite weight, the greater the agreement.

3. In the Match specification, you can specify cutoff threshold as follows:

   – Match Cutoff threshold

     When a record pair receives a composite weight greater than or equal to this weight, the pair is declared a match.

   – Clerical Cutoff threshold

     When a record pair receives a composite weight greater than or equal to this weight and less than the match cutoff, the pair is marked for clerical review. This weight must be equal to or less than the match cutoff weight.

  If you do not want a clerical review, set the clerical cutoff equal to the match cutoff.

- Duplicate threshold

  The lowest weight that a record pair can have to be considered a duplicate. This cutoff weight is optional and must be higher than the match cutoff weight.

  This cutoff is only used with Reference Match and not with Unduplicate Match.

These thresholds are used to send the results of matching to the appropriate outputs.

You use the Match Designer tool to set which columns are to be compared for the matching process. When the match specification is completed, you can use it in the Unduplicate Match stage, Reference Match stage, and Match Frequency stages. You can then apply the results from these match stages to the next stage of your project, such as Survive or for loading into a database.

The Match Frequency stage, Match Designer tool, Unduplicate Match stage, and Reference Match stage are described briefly in the following sections.

Match Frequency Stage

The Match Frequency stage gives you direct control over the disposition of generated frequency data. This stage provides results that can be used by the Match Designer tool and match stages, but enables you to generate the frequency data independent of running the matches. You can generate frequency information by using any data that provides the fields that are needed by a match. Then you can let the generated frequency data flow into a match stage, store it for later use, or both.

Attention: While you could conceivably take input directly from a Standardize stage and output it directly to a Match stage in the same IBM WebSphere QualityStage job, there are some potential problems with this approach as follows:

- Performance issues: Generating frequencies is time consuming. Instead, run the frequencies periodically (monthly), to support a nightly match run.
- Not real time: Because the Match Frequency stage does not run in real-time mode, this stage is not real-time compatible.
- Unrepresentative data: The data for any particular match run might not be representative. You want to build the frequencies on a representative data set.

It is better to build modular jobs.

As mentioned earlier, the Match Frequency stage takes one input data file. This data file can be from one of the following places:

- A link from a sample data set file for use in the Match Designer tool
- A link from any parallel database, file or processing stage that carries information you need for the following stages:
  - An unduplicate match
  - The data source input of a reference match
  - The reference source input of a reference match

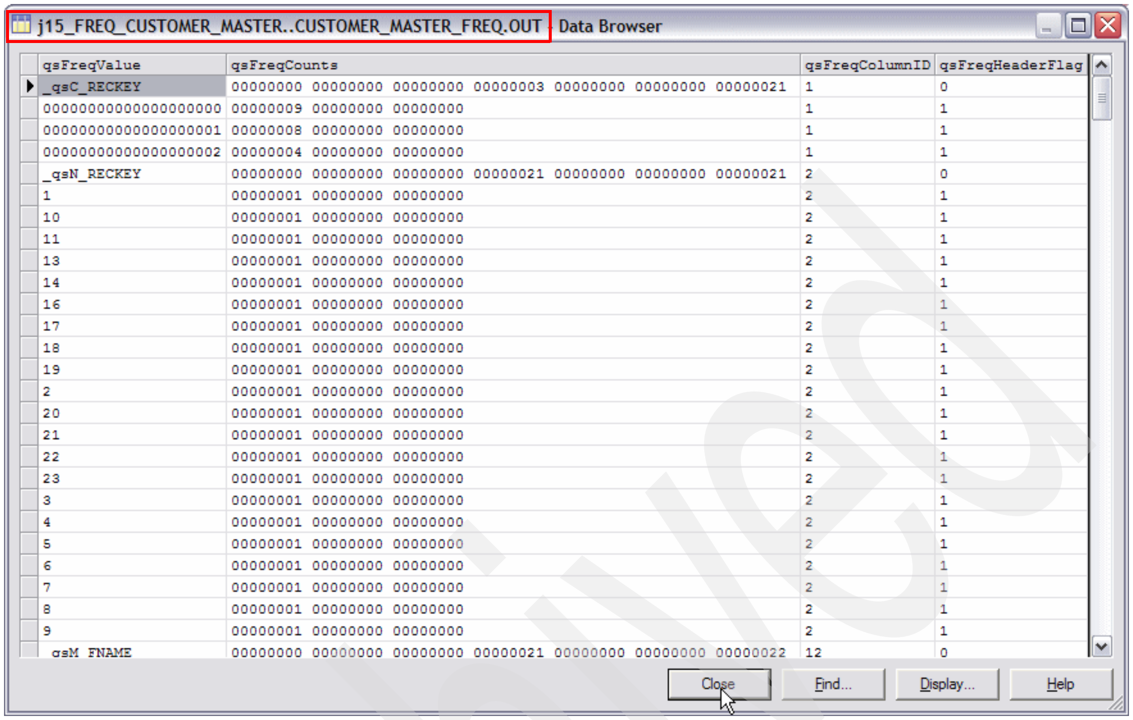
Attention: We recommend that you standardize the input data before you use it with the Match Designer tool and the match stages.

The Match Frequency stage takes a single output link that carries four columns:

- qsFreqVal
- qsFreqCounts
- qsFreqColumnID
- qsFreqHeaderFlag

The data (shown in Figure 1-48 on page 93 is not easily decipherable) in these columns provides the necessary information to give input to the Match Designer tool and to the Unduplication Match and Reference Match stages.

Note: You can link to the input or output of the Match Frequency stage from any parallel database and file stages, and most processing stages.


j15_FREQ_CUSTOMER_MASTER..CUSTOMER_MASTER_FREQ.OUT - Data Browser

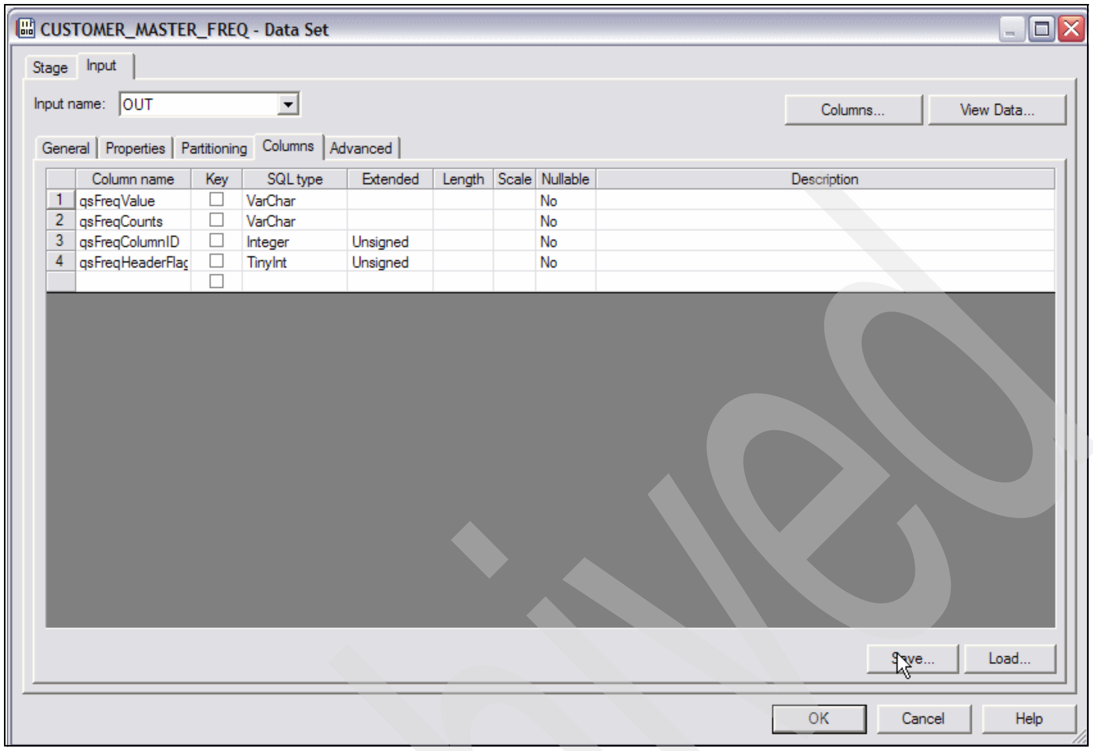
qsFreqValue	qsFreqCounts	qsFreqColumnID	qsFreqHeaderFlag
_qsC_RECKEY	00000000 00000000 00000000 00000003 00000000 00000000 00000021	1	0
00000000000000000000	00000009 00000000 00000000	1	1
00000000000000000001	00000008 00000000 00000000	1	1
00000000000000000002	00000004 00000000 00000000	1	1
_qsN_RECKEY	00000000 00000000 00000000 00000021 00000000 00000000 00000021	2	0
1	00000001 00000000 00000000	2	1
10	00000001 00000000 00000000	2	1
11	00000001 00000000 00000000	2	1
13	00000001 00000000 00000000	2	1
14	00000001 00000000 00000000	2	1
16	00000001 00000000 00000000	2	1
17	00000001 00000000 00000000	2	1
18	00000001 00000000 00000000	2	1
19	00000001 00000000 00000000	2	1
2	00000001 00000000 00000000	2	1
20	00000001 00000000 00000000	2	1
21	00000001 00000000 00000000	2	1
22	00000001 00000000 00000000	2	1
23	00000001 00000000 00000000	2	1
3	00000001 00000000 00000000	2	1
4	00000001 00000000 00000000	2	1
5	00000001 00000000 00000000	2	1
6	00000001 00000000 00000000	2	1
7	00000001 00000000 00000000	2	1
8	00000001 00000000 00000000	2	1
9	00000001 00000000 00000000	2	1
_qsM_FNAME	00000000 00000000 00000000 00000021 00000000 00000000 00000022	12	0

Figure 1-48 Match Frequency stage output

When you are configuring the Match Frequency stage, you can choose from the following options:

- Designate a match specification from the repository, so that frequency information is generated only for the match columns in the specification.
- Do not designate a match specification, thereby generating frequency information for *all* columns of the input.

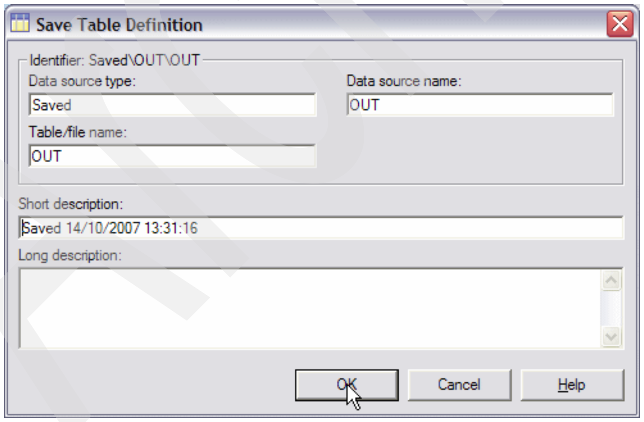
  When multiple match specifications are defined for the same table, it is more efficient to generate a single match frequency output for all the columns in the table, and then use it in the different match stages.
- Increase the number of frequencies reported in the output. (The default is 100.) You should increase this number if you are processing a large number of records.


CUSTOMER_MASTER_FREQ - Data Set

Input name: OUT

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	Description
1	qsFreqValue	☐	VarChar				No	
2	qsFreqCounts	☐	VarChar				No	
3	qsFreqColumnID	☐	Integer	Unsigned			No	
4	qsFreqHeaderFlag	☐	TinyInt	Unsigned			No	
		☐						

Figure 1-49   Saving table definition of the Match Frequency stage 1/3


Save Table Definition

Identifier: Saved\OUT\OUT

Data source type: Saved

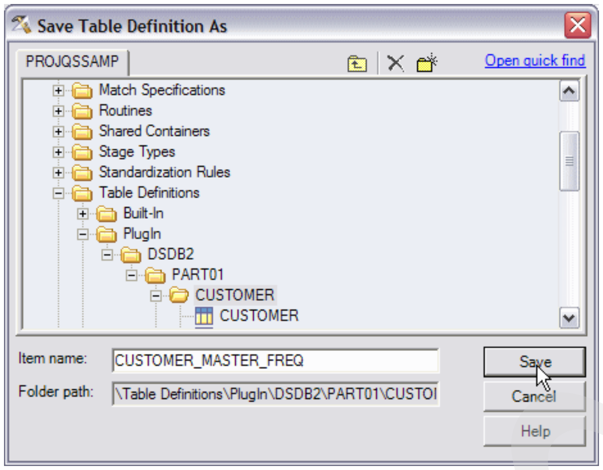
Data source name: OUT

Table/file name: OUT

Short description: Saved 14/10/2007 13:31:16

Long description:

Figure 1-50   Saving table definition of the Match Frequency stage 2/3



Item name: CUSTOMER_MASTER_FREQ

Folder path: \Table Definitions\PlugIn\DSDB2\PART01\CUSTOI

Figure 1-51 Saving table definition of the Match Frequency stage 3/3

Match Designer tool

The Match Designer is a tool for creating a match specification.

When you create the match specification, you can select from more than 25 types of comparisons such as the such as CHAR (character comparisons), and UNCERT (character uncertainty comparisons). The comparisons are algorithms that are based on the type of data in the columns. Table 1-10 shows most the most common comparisons and where they are used.

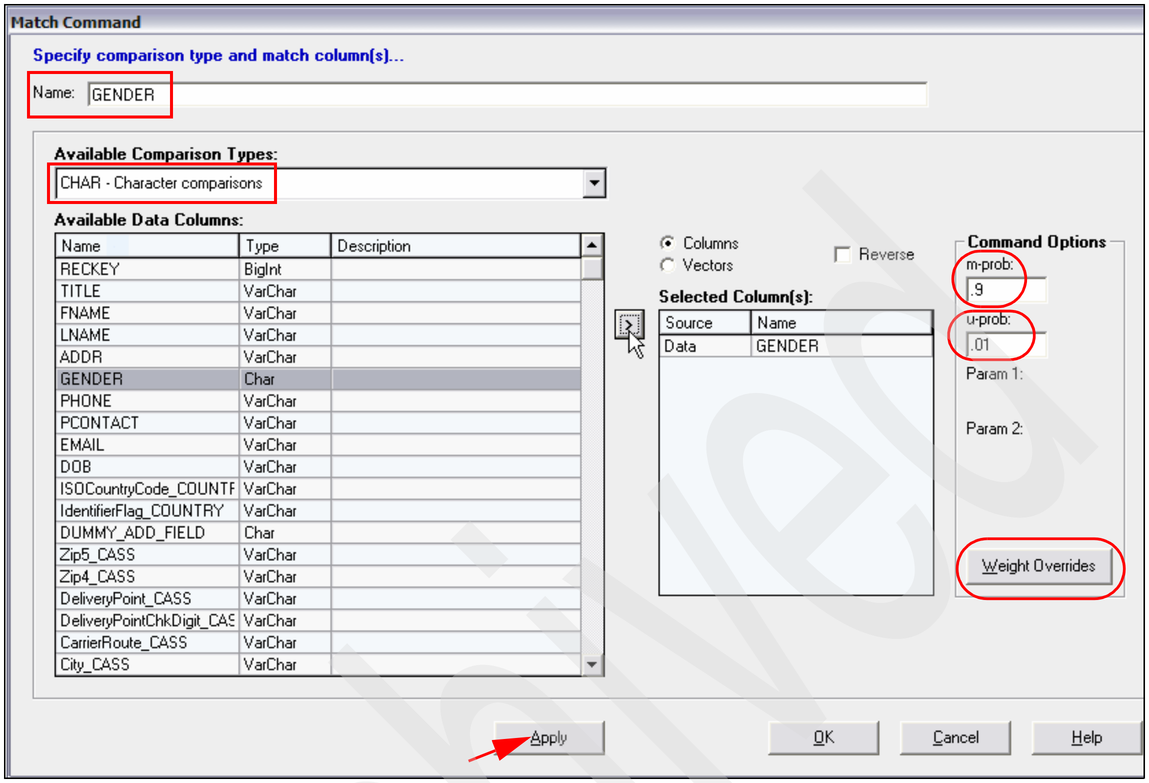
Table 1-10 Common comparisons and where they are used

Common comparisons	Where mostly used
CHAR (character comparison) character by character, left to right.	This is most useful to ensure a direct comparison, particular on code related fields such as Gender.
CNT_DIFF (count differences) tolerates a set number of key stroke errors (typically 1-2) on a prorated basis.	This is most commonly used with data such as telephone number or SSN where there is higher probability of key stroke issues.
DATE8 (basic tolerance of date errors) tolerates a set number of variations in days on a prorated basis.	This is most commonly used with data such as date of birth, date of encounter, and so forth where there is possibility of slight error in the day entered.

Common comparisons	Where mostly used
MULT_UNCERT - (word uncertainty in single field) tolerates phonetic errors, transpositions, random insertion, deletion, and replacement of characters and words using a comparison threshold.	This is the most common comparison for business names, arrays of identifiers, or other descriptive data where multiple words exist that can vary in position.
NAME_UNCERT - (shortest value character uncertainty) tolerates phonetic errors, transpositions, random insertion, deletion, and replacement of characters using a comparison threshold but only to a maximum of the shortest value length.	This is the standard comparison for personal names.
PREFIX - (shortest value character comparison) character by character, left to right to a maximum of the shortest value length	This is most useful to ensure matching of truncated data such as middle initial.
UNCERT - (character uncertainty) tolerates phonetic errors, transpositions, random insertion, deletion, and replacement of characters using a comparison threshold.	This is the standard comparison for street names or other basic text data.

You can use the Match Designer to create multiple match specifications that include one or more passes. Each pass is separately defined and is stored in the repository to be reused.

Attention: Within each pass of a match specification, you define the blocking fields, match commands, matching columns, match/clerical cutoff values, and m and u probabilities. Figure 1-52 on page 97 shows a panel that specifies the match column (GENDER), comparison type (CHAR - Character comparisons), and m (.9) and u (.01) probabilities chosen.


Match Command

Specify comparison type and match column(s)...

Name: GENDER

Available Comparison Types: CHAR - Character comparisons

Available Data Columns:

Name	Type	Description
RECKEY	BigInt	
TITLE	VarChar	
FNAME	VarChar	
LNAME	VarChar	
ADDR	VarChar	
GENDER	Char	
PHONE	VarChar	
PCONTACT	VarChar	
EMAIL	VarChar	
DOB	VarChar	
ISOCountryCode_COUNTF	VarChar	
IdentifierFlag_COUNTRY	VarChar	
DUMMY_ADD_FIELD	Char	
Zip5_CASS	VarChar	
Zip4_CASS	VarChar	
DeliveryPoint_CASS	VarChar	
DeliveryPointChkDigit_CAS	VarChar	
CarrierRoute_CASS	VarChar	
City_CASS	VarChar	

[✓ Columns] Vectors

Reverse

Selected Column(s):

Source	Name
Data	GENDER

Command Options

m-prob: .9

u-prob: .01

Param 1:

Param 2:

Figure 1-52 Match command example

You can run each pass on the complete source data or test data that is created from a representative subset of your production data and view the results in a variety of graphic displays. You can also use frequency information that is generated by the Match Frequency stage to help create your match specifications.

Important: While you can use the actual data source as input to this utility, we recommend that you use a representative subset of the data reference source instead. This subset allows you to test your passes in the Match Designer (as described in “J10_Undup_MatchSpec_STAN_CUSTOMER” on page 269) against a small sample of the data the match ultimately runs against.

You create separate match specification for each type of match as follows:

- For Unduplicate Match, you can create a match specification that is one of the following types:
  - Unduplicate (dependent) removes duplicates from match consideration in subsequent passes
  - Unduplicate Independent includes all duplicates in subsequent passes
- For Reference Match, you can create a match specification that is one of the following types:
  - One-to-One to create a two-file match specification based on a one-to-one relationship. In this match type, a record on the reference source can match only one data source record.
  - Reference to create a two-file match specification based on a many-to-one relationship.

    There are multiple flavors of this type as follows:

    - Many-to-One in which a record on the reference source can match many data source records. However, any one data source record can only match one reference source record.

      For example, 101 Main St. on the data source matches to two records on the reference source: 101-199 Main St SW and 101-199 Main St SE. The first reference source record is the matched record and the other is disregarded.
    - Many-to-one Multiple in which each reference source record having the same weight as the matched pair when it is scored against the data record is flagged as a duplicate record. Any one data source record might match more than one reference source record.

      For example, 101 Main St. on the data source matches to two records on the reference source: 101-199 Main St SW and 101-199 Main St SE. One reference source record is the matched record and the other is the duplicate.
    - Many-to-one Duplicates is similar to the Many-to-one Multiple option, except that additional reference source records that match to a level above the duplicate cutoff value are flagged as duplicates. Thus, records with lower weights than the match weight can be flagged as duplicates.

      For example, 101 Main St on the data source matches to three records on the reference source: 101-199 Main St SW, 101-199 Main St SE, and 101 Main Rd. One record gets 101-199 Main St SW as the match and both of the other addresses could be duplicates.

Note: You can assign up to six variables to a column or columns (in the **Configure Specification** → **Variables Special Handling** window) to judge the column according to the definition of the variable that you select.

For any match specification, you can assign a column a special treatment such as specifying that a disagreement on the column would cause the record pair automatically to be considered a non-match. You can assign a column more than one special treatment. A treatment applies to all passes for this match job.

The list of possible actions for a given column are:

- CRITICAL: A disagreement on the column causes the record pair automatically to be considered a non-match.
- CRITICAL MISSINGOK: Missing values on one or both columns is acceptable; that is, the record pair is rejected automatically if there is a non-missing disagreement.
- CLERICAL: A disagreement on the column causes the record pair automatically to be considered a clerical review case regardless of the weight.
- CLERICAL MISSINGOK: A missing value should not cause the record pair being considered to be forced into clerical review, but a disagreement would.
- NOFREQ: A frequency analysis is not run on the column. Used when a column has unique values, such as SSN.
- CONCAT: Concatenates up to four columns to form one frequency count.

The inputs to the Match Designer tool are:

- Input data source which is usually a representative subset of the data or reference source
- Frequency information from the Match Frequency stage as described in “Match Frequency Stage” on page 91.

The output of the Match Designer tool is a test results database—you must provide the ODBC Data Source Name (DSN) of this test results database (on the client computer), and the user name and password for accessing it.

The main area of the Match Designer is made up of two tabs:

- Compose tab

  On the Compose tab, you design and fine-tune the match passes. You design the Match passes and add them to the Match job. You can add, delete, and modify Match passes in this section. For each Match pass, you can specify blocking fields, matching fields, cutoff weights, and view the weight histogram, data results, and Match pass statistics.

– The Match Type area is a kind of sandbox for designing jobs that displays the current Match job. In this area, you can rearrange the order in which the Match passes run in the Match job, add or remove passes, and create new passes.

– The Match Pass Holding area is used to keep iterations of a particular pass definition or alternate approaches to a pass. The passes in the holding area do not run as part of the match job, but can be tested in isolation. You can add any of the Match passes in the holding area to the Match job by moving the Match pass to the Match Type area. Also, you can remove any of the Match passes from the Match job by moving it from the type area into the Match Pass Holding Area. Any pass, whether in the type or holding areas, can be test run. This approach lets you perform trial runs of different pass definitions without needing to lose alternate definitions.

– The Blocking Columns area designates the fields that must match exactly for records to be in the same processing group for the match. The right pane shows the histogram and data sections when the run is complete. You can sort and search the data columns from the match results.

– Cutoff values area identifies the match cutoff and clerical cutoff.

  Match cutoff verifies that any record pairs above the cutoff have a high probability of matching and those below the cutoff have a high probability of not being a match.

  The composite weights assigned to each record pair create a distribution of scores that range from very high positive to very high negative.

  - Within the distribution of positive values, define a value or cutoff at which any record pair receiving a weight equal to or greater than this cutoff is considered a match.

    This is referred to as the *match cutoff*.

  - Conversely, define a cutoff at which any record pair receiving a weight equal to or less than this cutoff is considered a non-match. Record pairs with weights that fall between these two cutoff values are considered clerical review cases.

    This is referred to as the *clerical cutoff*.

Note: Set the clerical cutoff to less than zero and run the pass. Look at the resulting graph and adjust the settings according to your requirements.

If more than one record pair receives a composite weight higher than the match cutoff weight, those records are declared duplicates.

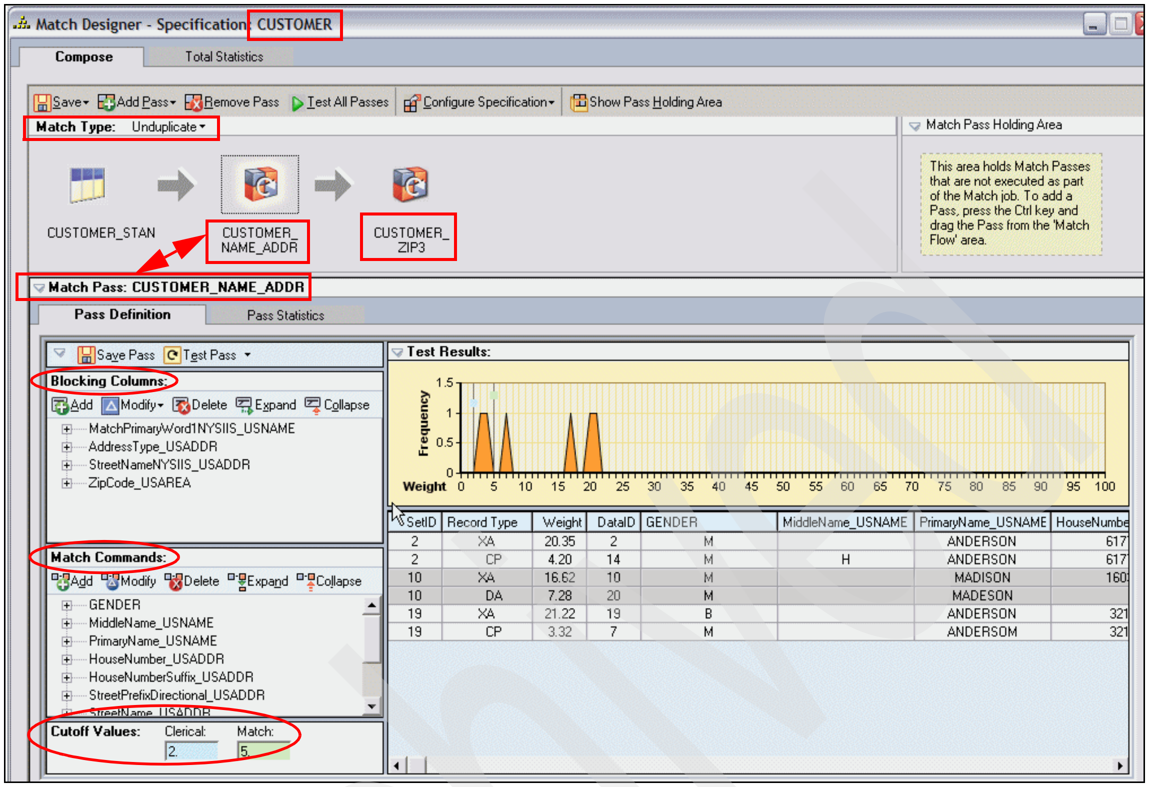
The *duplicate cutoff* weight is optional and must be higher than the match cutoff weight. This cutoff is not used with the Unduplicate Match stage. The way in which duplicate records are handled is based on what type of matching you selected.

Any record pair that falls below the clerical cutoff becomes a residual and is eligible for the next matching pass.

Note: If you want the stage to do exact matching only, specify the blocking columns and not the matching columns. This results in all record pairs that agree on the blocking columns being declared matches or duplicates.

Figure 1-53 on page 102 and Figure 1-54 on page 103 show the options in the Compose tab. The data source is the output of the Standardize stage on CUSTOMER data. Two passes, CUSTOMER_NAME_ADDR and CUSTOMER_ZIP3 are defined in this CUSTOMER specification. This specification is an Unduplicate Match type.

– Figure 1-53 on page 102 shows the configuration of the first pass CUSTOMER_NAME_ADDR as follows:
  - Blocking columns are MatchPrimaryWord1NYSIIS_USNAME, AddressType_USADDR, StreetNameNYSIIS_USADDR, and ZipCode_USAREA.
  - Match columns include GENDER, MiddleName_USNAME, PrimaryName_USNAME, HouseNumberSuffix_USADDR, StreetPrefixDirectional_USADDR, and StreetName_USADDR.
  - Clerical cutoff value is 2, while Match cutoff is 5.
  - The test results in the right pane show the histogram of composite weights and the records in the individual blocks. Rows with the same SetID belong to the same block and is a unique id for each set of matched rows. Record Type identifies the type of match for the record within each SetID—XA identifies the master record in a set of records, DA is a duplicate record, CP is a clerical review record, and RA is a residual record.



Match Type: Undulicate

Match Pass Holding Area

This area holds Match Passes that are not executed as part of the Match job. To add a Pass, press the Ctrl key and drag the Pass from the 'Match Flow' area.

Match Pass: CUSTOMER_NAME_ADDR

Blocking Columns:

- MatchPrimaryWord1NYSIIS_USNAME
- AddressType_USADDR
- StreetNameNYSIIS_USADDR
- ZipCode_USAREA

Match Commands:

- GENDER
- MiddleName_USNAME
- PrimaryName_USNAME
- HouseNumber_USADDR
- HouseNumberSuffix_USADDR
- StreetPrefixDirectional_USADDR
- StreetName_USADDR

Cutoff Values: Clerical: 2. Match: 5.

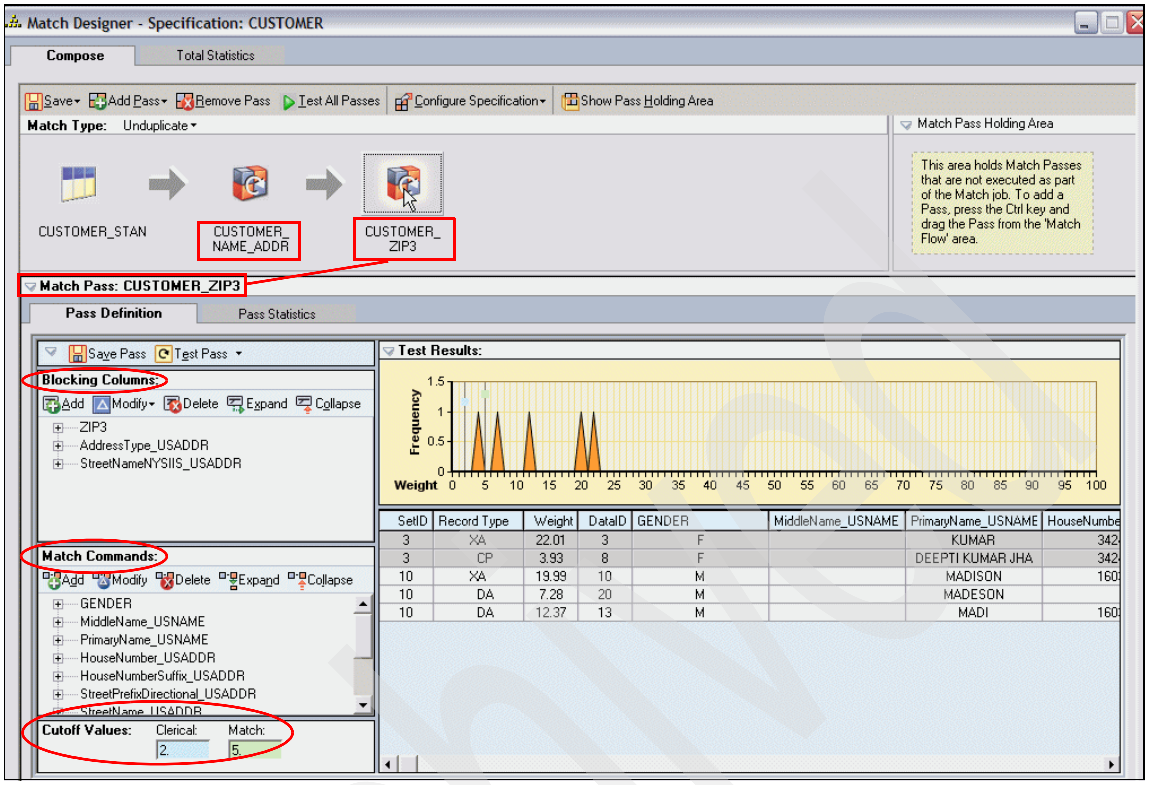
Test Results:

SetID	Record Type	Weight	DataID	GENDER	MiddleName_USNAME	PrimaryName_USNAME	HouseNumbe
2	XA	20.35	2	M		ANDERSON	617
2	CP	4.20	14	M	H	ANDERSON	617
10	XA	16.62	10	M		MADISON	160
10	DA	7.28	20	M		MADESON	
19	XA	21.22	19	B		ANDERSON	321
19	CP	3.32	7	M		ANDERSOM	321

Figure 1-53 Compose tab showing CUSTOMER_NAME_ADDR pass

– Figure 1-54 on page 103 shows the configuration of the first pass CUSTOMER_ZIP3 as follows:

- Blocking columns are ZIP3, AddressType_USADDR, and StreetNameNYSIIS_USADDR.
- Match columns include GENDER, MiddleName_USNAME, PrimaryName_USNAME, HouseNumberSuffix_USADDR, StreetPrefixDirectional_USADDR, and StreetName_USADDR.
- Clerical cutoff value is 2, while Match cutoff is 5.
- The test results in the right pane show the histogram of composite weights and the records in the individual blocks.



This area holds Match Passes that are not executed as part of the Match job. To add a Pass, press the Ctrl key and drag the Pass from the 'Match Flow' area.

Blocking Columns:

- ZIP3
- AddressType_USADDR
- StreetNameNYSIIS_USADDR

Match Commands:

- GENDER
- MiddleName_USNAME
- PrimaryName_USNAME
- HouseNumber_USADDR
- HouseNumberSuffix_USADDR
- StreetPrefixDirectional_USADDR
- StreetName_USADDR

Cutoff Values: Clerical: 2. Match: 5.

SetID	Record Type	Weight	DataID	GENDER	MiddleName_USNAME	PrimaryName_USNAME	HouseNumbe
3	XA	22.01	3	F		KUMAR	342
3	CP	3.93	8	F		DEEPTI KUMAR JHA	342
10	XA	19.99	10	M		MADISON	160
10	DA	7.28	20	M		MADESON	
10	DA	12.37	13	M		MADI	160

Figure 1-54 Compose tab showing CUSTOMER_ZIP3 pass

- Total statistics tab

  This tab provides statistical data in a graphical format for all the passes that run.

  The cumulative statistics are of value only if you test multiple passes consecutively, in the order that they appear in the match specification. The Total Statistics page displays the following information:

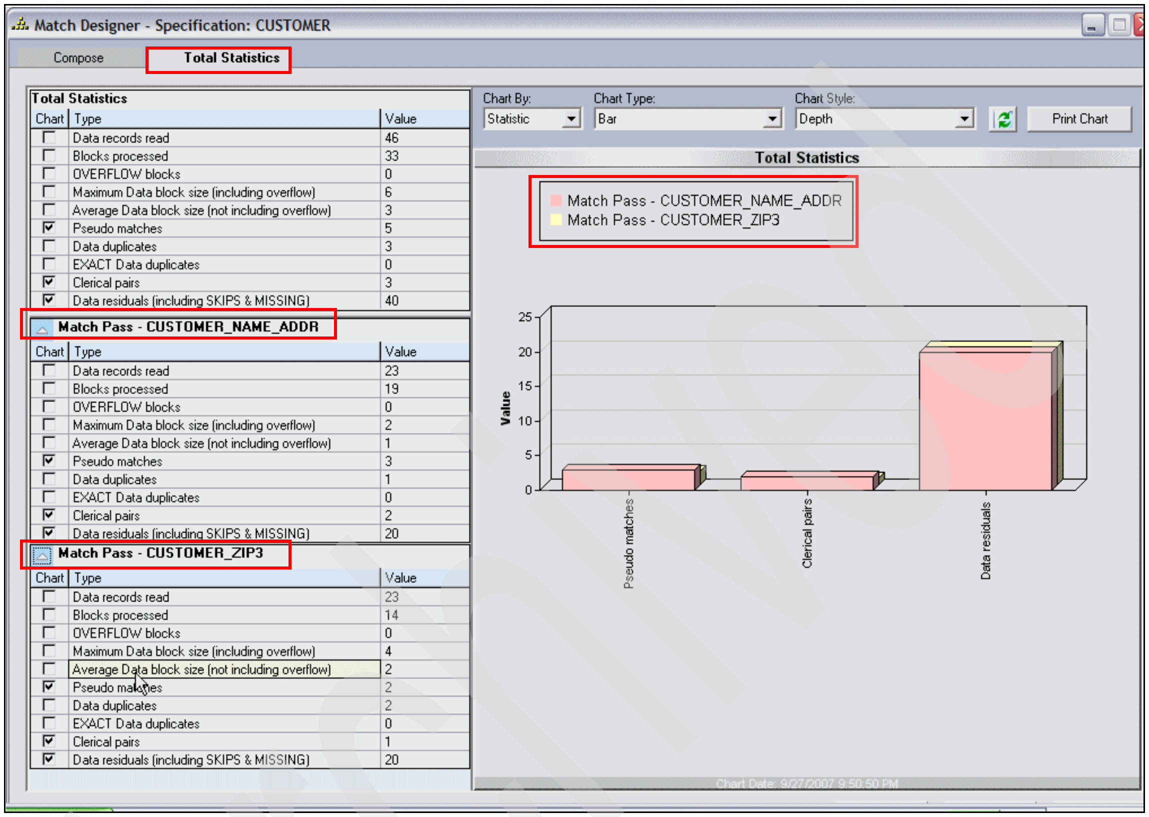
  – Cumulative statistics for the current runs of all passes in the match specification.

  – Individual statistics for the current run of each pass.

  – Charts that compare the statistics for the current run of all passes.

Note: The Total Statistics page does not display information about passes in the Match Pass Holding Area.

Figure 1-55 shows a bar chart that was built by using the results for pseudo matches, clerical pairs, and data residuals for each of the match passes CUSTOMER_NAME_ADDR and CUSTOMER_ZIP3.



🟥 Match Pass - CUSTOMER_NAME_ADDR
🟨 Match Pass - CUSTOMER_ZIP3

Total Statistics

Chart	Type	Value
☐	Data records read	46
☐	Blocks processed	33
☐	OVERFLOW blocks	0
☐	Maximum Data block size (including overflow)	6
☐	Average Data block size (not including overflow)	3
✓	Pseudo matches	5
☐	Data duplicates	3
☐	EXACT Data duplicates	0
✓	Clerical pairs	3
✓	Data residuals (including SKIPS & MISSING)	40

Match Pass - CUSTOMER_NAME_ADDR

Chart	Type	Value
☐	Data records read	23
☐	Blocks processed	19
☐	OVERFLOW blocks	0
☐	Maximum Data block size (including overflow)	2
☐	Average Data block size (not including overflow)	1
✓	Pseudo matches	3
☐	Data duplicates	1
☐	EXACT Data duplicates	0
✓	Clerical pairs	2
✓	Data residuals (including SKIPS & MISSING)	20

Match Pass - CUSTOMER_ZIP3

Chart	Type	Value
☐	Data records read	23
☐	Blocks processed	14
☐	OVERFLOW blocks	0
☐	Maximum Data block size (including overflow)	4
☐	Average Data block size (not including overflow)	2
✓	Pseudo matches	2
☐	Data duplicates	2
☐	EXACT Data duplicates	0
✓	Clerical pairs	1
✓	Data residuals (including SKIPS & MISSING)	20

Figure 1-55 Total Statistics tab

Unduplicate Match stage

This stage locates and groups all similar records within a single input data source. This process identifies potential duplicate records, which might then be removed.

An example is the need to eliminate duplicates from a consolidation of mailing lists purchased from multiple sources.

The Unduplicate Match stage accomplishes the following actions:

- Categorizes all records with weights above the match cutoff as a set of duplicates.
- Identifies a master record by selecting the record within the set that matches to itself with the highest weight. The master record is associated with its set of duplicates.
- Determines that records not part of a set of duplicates are residuals. The residuals and the master records are generally made available for the next pass.
- Excludes duplicates in subsequent passes. However, you can choose the Independent match type (more later) if you want duplicates to be included in subsequent passes.

The output of the Unduplicate Match stage can include master records, duplicates above the match cutoff, clerical duplicates, and residuals. You can use this output as input to the Survive stage.

Unduplicate Match stage uses two match types as follows:

- Dependent: This match type is the default choice. With this match type, after the first pass, duplicates are removed from match consideration in subsequent passes.
- Independent: With this match type, duplicates are included in subsequent passes.

  Across all passes, all records that match to any given record are also considered matched to each other. For example, if records A and B match in pass 1, records B and C match in pass 2, and records C and D match in pass 3, then records A, B, C and D are matched to each other.

Note: Master records are always considered in subsequent passes.

Tip: Usually, you would choose Dependent match type, because you want duplicates removed from consideration so that they do not match to other records in subsequent passes.

However, the Independent option is useful in circumstances where you want to link people or organizations regardless of address. For example, you can link together all the locations where a doctor practices.

Table 1-11 shows sample data to describe how to use the Independent match type option with the Unduplicate Match stage. The table shows four records that describe the same person. You require that all records concerning the same person match without regard to address.

Table 1-11 Sample data to demonstrate Dependent and Independent Match

Record	Name	Address	Tax Identifier
1	William Nickson	123 Rodeo Drive	123456789
2	Bill Nixon	123 Rodeo Drive	
3	B Nickson	978 Sunset Blvd.	123456789
4	Nickson	456 Western Ave.	123456789

The matching process using this data can yield different results depending on whether you choose the Dependent or Independent match type:

- Dependent Match

  The first pass blocks and matches on Name and Address. Records 1 and 2 are considered a matched pair, while records 3 and 4 are considered residuals.

  If Record 2 (without the TaxID) is selected as the master, and Record 1 is considered a duplicate, then Record 1 is not available for the second pass.

  If the second pass blocks and matches on Name and TaxID, then only Records 3 and 4 match. The result is two groups of matched records: Records 1 and 2, and Records 3 and 4.

- Independent Match

  The first pass results are the same as the Dependent Match. Records 1 and 2 are considered a matched pair, and records 3 and 4 are considered residuals.

  If Record 2 (without the TaxID) is selected as the master record in the second pass, the duplicate record, Record 1, is also compared to the rest of the records. When you block on Name and TaxID, records 1, 3, and 4 match. Because Record 1 matched Record 2 in the first pass, the output is one group with all four records linked.

As mentioned earlier, the Unduplicate Match stage accepts the following inputs:

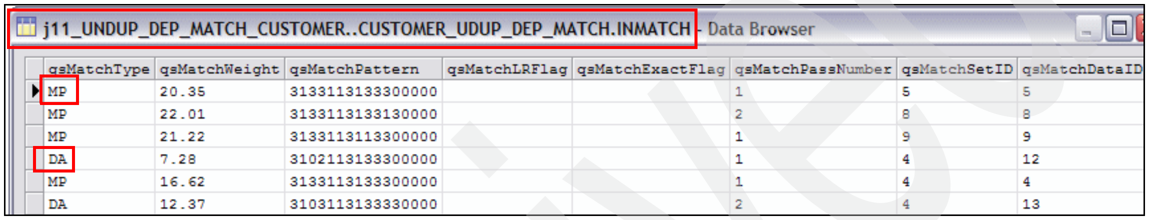
- Source data from any parallel database, file or processing stage. Typically, the source is standardized data from the Standardize stage.
- Frequency information for that data as generated by the Match Frequency stage as described in “Match Frequency Stage” on page 91. This stage takes

input from a database, file, or processing stage, and generates the frequency distribution of values for columns in the input data.

- Match specification to group and match the data. A Match specification provides details of blocking and matching columns for matching as described in "Match Designer tool" on page 95.

The Unduplicate match stage delivers up to four outputs as follows:

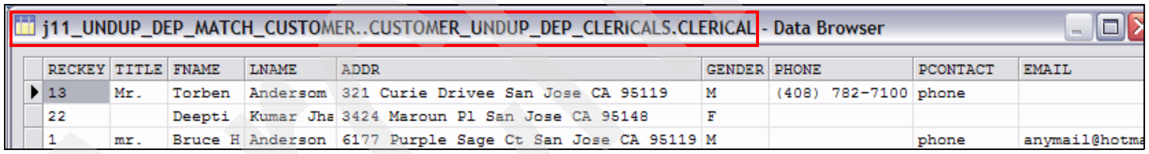
- Match contains master records. Figure 1-56 shows a report containing the merged match and duplicate records.
- Duplicate contains records that are above the match cutoff. Figure 1-56 shows a report containing the merged match and duplicate records


j11_UNDUP_DEP_MATCH_CUSTOMER..CUSTOMER_UDUP_DEP_MATCH.INMATCH - Data Browser

qsMatchType	qsMatchWeight	qsMatchPattern	qsMatchLRFlag	qsMatchExactFlag	qsMatchPassNumber	qsMatchSetID	qsMatchDataID
MP	20.35	3133113133300000			1	5	5
MP	22.01	3133113133130000			2	8	8
MP	21.22	3133113113300000			1	9	9
DA	7.28	3102113133300000			1	4	12
MP	16.62	3133113133300000			1	4	4
DA	12.37	3103113133330000			2	4	13

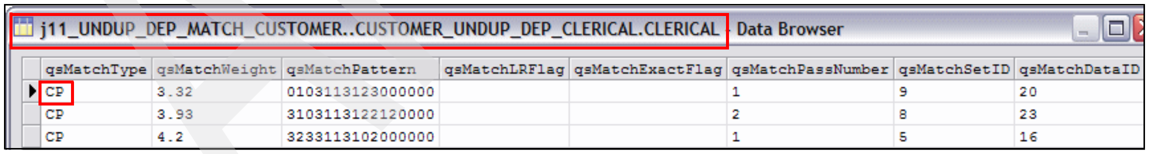
Figure 1-56 Merged match and duplicate records

- Clerical has records that fall in the clerical range as shown in Figure 1-57 and Figure 1-58.


j11_UNDUP_DEP_MATCH_CUSTOMER..CUSTOMER_UNDUP_DEP_CLERICALS.CLERICAL - Data Browser

RECKEY	TITLE	FNAME	LNAME	ADDR	GENDER	PHONE	PCONTACT	EMAIL
13	Mr.	Torben	Andersom	321 Curie Drivee San Jose CA 95119	M	(408) 782-7100	phone	
22		Deepti	Kumar Jha	3424 Maroun Pl San Jose CA 95148	F			
1	mr.	Bruce H	Anderson	6177 Purple Sage Ct San Jose CA 95119	M		phone	anymail@hotma

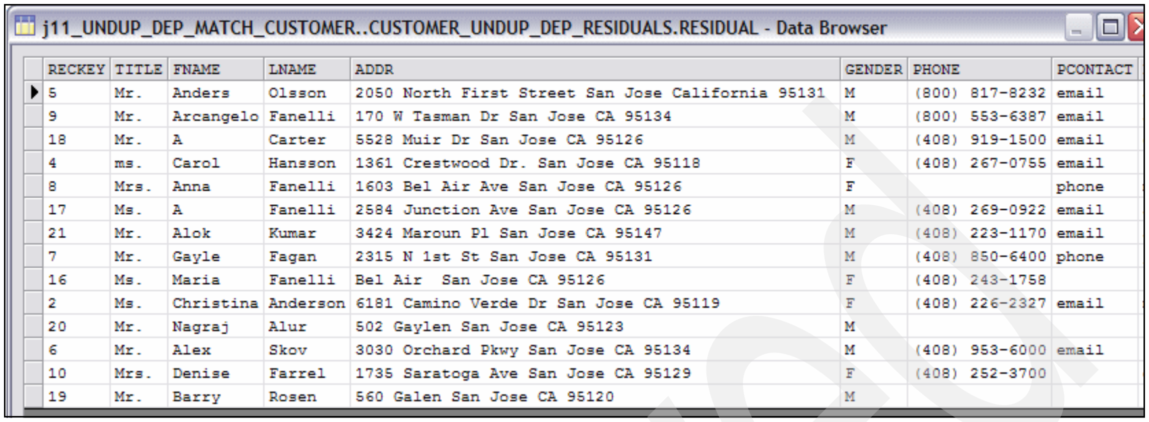
Figure 1-57 Clerical review records


j11_UNDUP_DEP_MATCH_CUSTOMER..CUSTOMER_UNDUP_DEP_CLERICAL.CLERICAL - Data Browser

qsMatchType	qsMatchWeight	qsMatchPattern	qsMatchLRFlag	qsMatchExactFlag	qsMatchPassNumber	qsMatchSetID	qsMatchDataID
CP	3.32	0103113123000000			1	9	20
CP	3.93	3103113122120000			2	8	23
CP	4.2	3233113102000000			1	5	16

Figure 1-58 Clerical review records

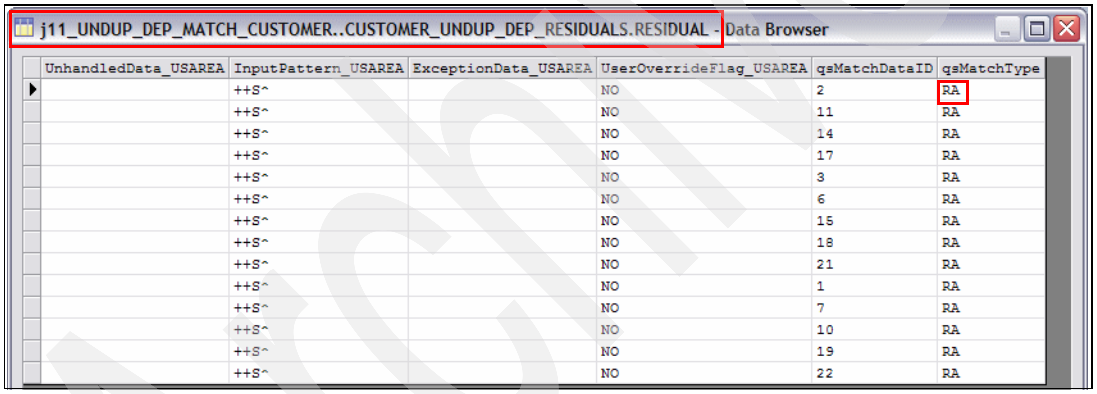
- ► Residual contains records that are not associated with any of the records as shown in Figure 1-59 and Figure 1-60.



j11_UNDUP_DEP_MATCH_CUSTOMER..CUSTOMER_UNDUP_DEP_RESIDUALS.RESIDUAL - Data Browser

RECKEY	TITLE	FNAME	LNAME	ADDR	GENDER	PHONE	PCONTACT
5	Mr.	Anders	Olsson	2050 North First Street San Jose California 95131	M	(800) 817-8232	email
9	Mr.	Arcangelo	Fanelli	170 W Tasman Dr San Jose CA 95134	M	(800) 553-6387	email
18	Mr.	A	Carter	5528 Muir Dr San Jose CA 95126	M	(408) 919-1500	email
4	ms.	Carol	Hansson	1361 Crestwood Dr. San Jose CA 95118	F	(408) 267-0755	email
8	Mrs.	Anna	Fanelli	1603 Bel Air Ave San Jose CA 95126	F		phone
17	Ms.	A	Fanelli	2584 Junction Ave San Jose CA 95126	M	(408) 269-0922	email
21	Mr.	Alok	Kumar	3424 Maroun Pl San Jose CA 95147	M	(408) 223-1170	email
7	Mr.	Gayle	Fagan	2315 N 1st St San Jose CA 95131	M	(408) 850-6400	phone
16	Ms.	Maria	Fanelli	Bel Air San Jose CA 95126	F	(408) 243-1758	
2	Ms.	Christina	Anderson	6181 Camino Verde Dr San Jose CA 95119	F	(408) 226-2327	email
20	Mr.	Nagraj	Alur	502 Gaylen San Jose CA 95123	M		
6	Mr.	Alex	Skov	3030 Orchard Pkwy San Jose CA 95134	M	(408) 953-6000	email
10	Mrs.	Denise	Farrel	1735 Saratoga Ave San Jose CA 95129	F	(408) 252-3700	
19	Mr.	Barry	Rosen	560 Galen San Jose CA 95120	M		

Figure 1-59 Residual records 1/2



j11_UNDUP_DEP_MATCH_CUSTOMER..CUSTOMER_UNDUP_DEP_RESIDUALS.RESIDUAL - Data Browser

UnhandledData_USAREA	InputPattern_USAREA	ExceptionData_USAREA	UserOverrideFlag_USAREA	qsMatchDataID	qsMatchType
	++S^		NO	2	RA
	++S^		NO	11	RA
	++S^		NO	14	RA
	++S^		NO	17	RA
	++S^		NO	3	RA
	++S^		NO	6	RA
	++S^		NO	15	RA
	++S^		NO	18	RA
	++S^		NO	21	RA
	++S^		NO	1	RA
	++S^		NO	7	RA
	++S^		NO	10	RA
	++S^		NO	19	RA
	++S^		NO	22	RA

Figure 1-60 Residual records 2/2

Reference Match stage

The Reference Match stage identifies relationships among records. This match can group records that are being compared in different ways as follows:

- ► One-to-many matching

  Identifies all records in one data source that correspond to a record for the same individual, event, household, or street address in a second data source. Only one record in the reference source can match one record in the data source because the matching applies to individual events.

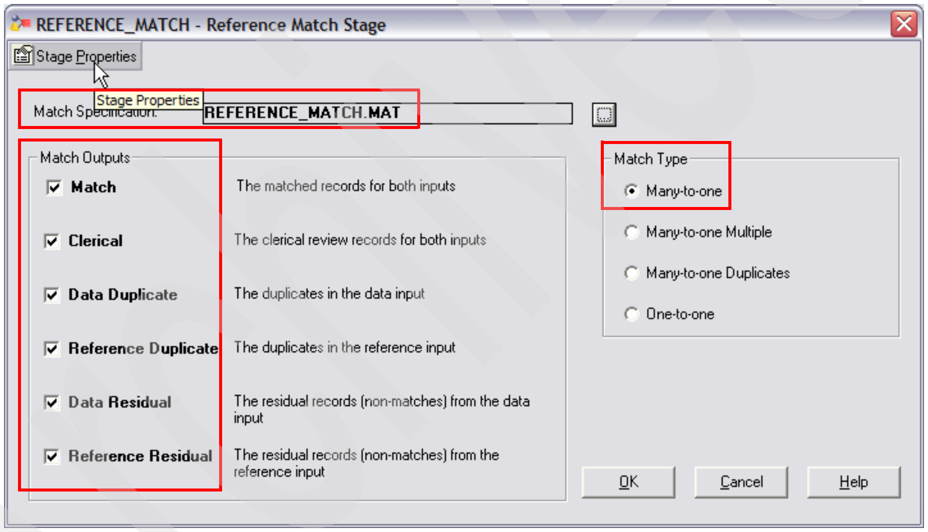
  An example of one-to-many matching is finding the same individual (based on comparing social security numbers) in two different sources such as a department of motor vehicles (DMV) list and a voter registration list. There should be at most only one record matching on social security number in each list.

- Many-to-one matching

  Multiple records in the data file can match a single record in the reference file. For example, matching a transaction data source to a master data source allows many transactions for one person in the master data source.

  An example of many-to-one matching is matching the ZIP codes in a customer file with the list of valid ZIP codes. More than one record in the customer file can have the same ZIP code in it.

When you configure the Reference Match stage, you select an existing Reference-type Match specification stored in the repository, select the match type, and select the match outputs as shown in Figure 1-61.



Match Specification: REFERENCE_MATCH.MAT

Match Outputs

[✓] Match	The matched records for both inputs
[✓] Clerical	The clerical review records for both inputs
[✓] Data Duplicate	The duplicates in the data input
[✓] Reference Duplicate	The duplicates in the reference input
[✓] Data Residual	The residual records (non-matches) from the data input
[✓] Reference Residual	The residual records (non-matches) from the reference input

Match Type

[✓ Many-to-one]
Many-to-one Multiple
Many-to-one Duplicates
One-to-one

Figure 1-61 Configure Reference Match stage

The Reference Match stage requires standardized data and reference data as source data, a reference match specification, and frequency information for both sources. Briefly, these inputs are:

- Source data from any parallel database, file or processing stage. Typically, the source is standardized data from the Standardize stage.
- Reference source against which the source data is matched.
- Frequency information for that data as generated by the Match Frequency stage as described “Match Frequency Stage” on page 91.

- Match specification to group and match the data as described for the “Match Designer tool” on page 95.

The output of the Reference Match stage includes master records, clerical review records, duplicates, and residuals. You can use this output as input to the Survive stage. Briefly, the Reference Match stage delivers up to six outputs as follows:

- Match contains matched records for both inputs
- Clerical has records that fall in the clerical range for both inputs
- Data Duplicate contains duplicates in the data source
- Reference Duplicate contains duplicates in the reference source
- Data Residual contains records that are non-matches from the data input
- Reference Residual contains records that are non-matches from the reference input

These reports are similar to the corresponding ones in Figure 1-56 on page 107 through Figure 1-60 on page 108.

1.9 Survive stage

The Survive stage consolidates duplicate records, which creates a best-of-breed representation of the matched data. Survive consolidates duplicate records, creating the best representation of the match data so companies can use it to load a master data record, cross-populate all data sources, or both.

During the Survive stage, IBM WebSphere QualityStage takes the following actions:

- Replaces existing data with “better” data from other records based on user specified rules
- Supplies missing values in one record with values from other records on the same entity
- Populates missing values in one record with values from corresponding records which have been identified as a group in the matching stage
- Enriches existing data with external data

The Survive stage constructs column values from groups of related or duplicate records and stores the column values in the survived record (the best result) from each group.

The Survive job is the last job in the IBM WebSphere QualityStage workflow and is usually run after the Unduplicate Match stage job. The output from the

Unduplicate Match stage, and in some cases the Reference Match stage, becomes the source data that you use for the Survive stage.

The Survive stage requires one input source, which can be sourced from a database connector, flat file, data set, or another processing stage. If your input is the result of a match stage, you need to set up another stage (for example, a Funnel stage) to combine the master, duplicate records, and clerical reviewed duplicates into one input source.

While it is not necessary to process the data through the match stages before you use the Survive stage, the source data must include related or duplicate groups of rows. Also, the data must be able to be sorted on one or more columns that identify each group. These columns are referred to as *group keys*.

- To order the records, you sort on the group key or keys so that all records in a group are contiguous. The Survive stage sorts records automatically if the “Don’t Pre-sort Input” option is not selected in the Survive Stage window. However, the automatic sort provides no control over the order of the records within each group. To control the order within groups, you can presort the input by using the Sort stage.
- The Survive stage accepts all basic data types (non-vector, non-aggregate) other than binary.

The Survive stage can have only one output link. This link can send output to any other stage. You specify which columns and column values from each group create the output record for the group. The output record can include an entire input record, or selected columns from the record, or selected columns from different records in the group.

- You select column values based on rules for testing the columns.

  A rule contains a set of conditions and a list of one or more target columns. The rows are tested one by one in the order they appear within the match set. For each row tested, the values of the “test” row is updated.

  Rules are described in 1.9.1, “Survive rules” on page 112.
- To select a best candidate match, you can specify multiple columns such as the record creation date, data source from which the record originated, length of data in the column, and frequency of data in a group

You set up and configure the Survive stage to create a survived record. You need to configure the Survive stage before you run a Survive job. Before you can add source data to the Survive stage, all input records must be combined into one input source.

When you configure the Survive stage, you choose simple rules that are provided in the New Rules window or you select the Complex Survive Expression® to create your own custom rules. You use some or all of the columns from the source file, add a rule to each column, and apply the data.

After the Survive stage processes the records to select the best record, this information is sent to the target file.

1.9.1 Survive rules

To consider a target as the best candidate for the output record requires a rule that comprises one or more targets and a TRUE conditional expression. A condition is made up of:

- Column names
- Constant or literal values
- Operators that specify comparison or arithmetic operations

> **Note:** You can create more than one rule for a target. In addition, you can use any input column for testing the condition. The column can be the target or another column that is not associated with the target. The last rule tested "true" determines the final value.

The Survive stage evaluates the columns to the rule (simple or complex) and selects those that meet the conditions of the rule as best columns. The Survive stage reads the first record and evaluates the record according to any rule that you select. The evaluation process uses the following method:

- If the first record has no best columns then the selected rule for the target record is evaluated against all the columns in the record. If a target record passes the test, its columns become best columns and a "b" appears in front of the column names—this is only visible when building the rule and not part of the evaluation process.
- Each subsequent record in the group is evaluated in relation to the current record. If a target record passes the test then its columns become the best columns and replace any existing best columns. If none of the current columns meets the conditions, the best columns remain unchanged.
- After all records in the group are evaluated, the values that are designated as the best values are combined in the output record. Survive continues the process with the next records in the next group.

To apply survival rules to the input columns, the Survive stage includes:

- A set of predefined Techniques (packaged survive expressions) shown in Table 1-12 from which you can select the desired technique. These techniques are sometimes referred to as *simple survive rules*.

Table 1-12 Techniques (commonly used survive rules) available

Technique	**Pattern**
Shortest Field	SIZEOF(TRIM(c,"column")) <= SIZEOF(TRIM(b."column"))
Longest Field	SIZEOF(TRIM(c,"column")) >= SIZEOF(TRIM(b."column"))
Most Frequent	FREQUENCY
Most Frequent (non blank)	FREQUENCY (Skips missing values when counting most frequent)
Equals	c."column" = "DATA"
Not Equals	c."column" <> "DATA"
Greater Than	c."column" >= "DATA"
Less Than	c."column" <= "DATA"
At Least One	1 (At least one record survives, regardless of other rules)

You define a simple rule by specifying each of the following elements:

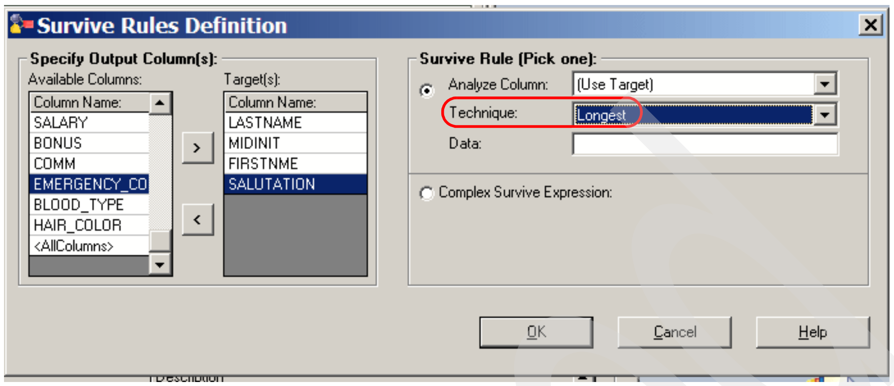
– Target column or columns

– Column to analyze

– Technique to apply to the column that is being analyzed

The rule is defined for you depending on the Technique that you select.

Figure 1-62 shows an example of a simple rule involving the Longest field technique.


Survive Rules Definition

Specify Output Column(s):

Available Columns:

Column Name:
SALARY
BONUS
COMM
EMERGENCY_CO
BLOOD_TYPE
HAIR_COLOR
<AllColumns>

Target(s):

Column Name:
LASTNAME
MIDINIT
FIRSTNME
SALUTATION

Survive Rule (Pick one):

(●) Analyze Column: (Use Target)

Technique: Longest

Data:

( ) Complex Survive Expression:

Figure 1-62 Simple rule example

For example, assume that the Unduplicate Match stage identified three records shown in Table 1-13 as representing the same person using different variations of the name.

Table 1-13 Duplicates representing the same individual

Column Name qsMatchSetID	SALUTATION	FIRSTNME	MIDINIT	LASTNAME
9	MR	JON		SMITH
9		J		SMITHE
9		JOHN	E	SMITH

The Survive stage constructs the best record using length analysis (Longest Field technique which corresponds to the pattern SIZEOF(TRIM(c,"column")) >= SIZEOF(TRIM(b."column")) on the columns SALUTATION, FIRSTNME, MIDINIT, and LASTNAME, and using frequency analysis on the column Family Name, with the result as shown in Table 1-13.

Table 1-14 Best record output of the Survive Stage

Column Name qsMatchSetID	SALUTATION	FIRSTNME	MIDINIT	LASTNAME
9	MR	JOHN	E	SMITHE

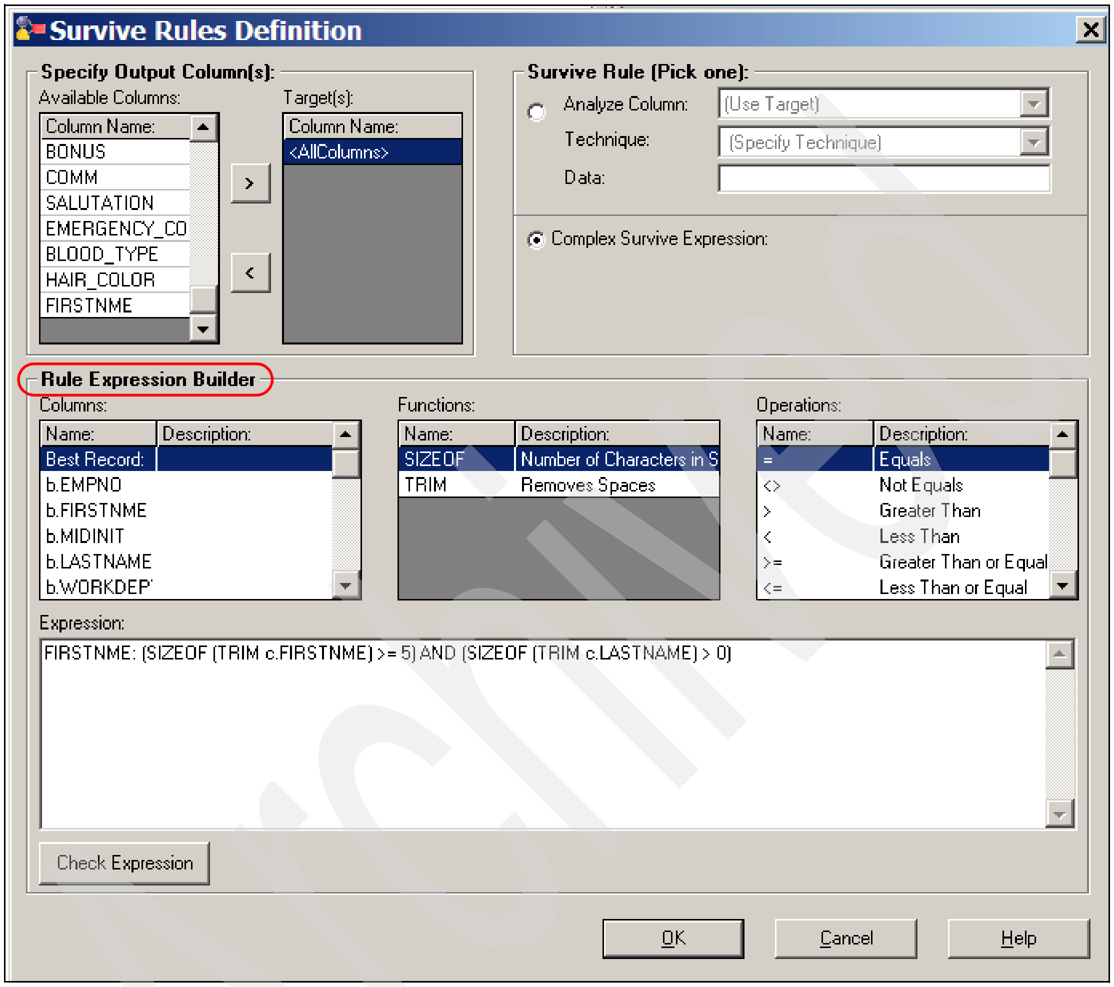
- The Rule Expression Builder for creating your own complex expressions. The rules created by the Rule Expression Builder are also referred to as complex survive rules.

  In the Rule Expression Builder, you define a rule by specifying each of the following elements:

  – A current record from the Columns list
  – One or both functions (SIZEOF and TRIM)
  – An operator such as =, <>, <, >, +, -, *, /, %, AND, OR and NOT
  – A best record from the Columns list
  – One or both functions

  In a complex rule, you can use more than one column (other than the target itself) to define the rule for a target.

Figure 1-63 shows a sample complex expression rule using the Rule Expression Builder.


Survive Rules Definition

Specify Output Column(s):

Available Columns:

Column Name:
BONUS
COMM
SALUTATION
EMERGENCY_CO
BLOOD_TYPE
HAIR_COLOR
FIRSTNME

Target(s):

Column Name:
<AllColumns>

Survive Rule (Pick one):

( ) Analyze Column: (Use Target)

Technique: (Specify Technique)

Data:

[✓ Complex Survive Expression:]

Rule Expression Builder

Columns:

Name:	Description:
Best Record:	
b.EMPNO	
b.FIRSTNME	
b.MIDINIT	
b.LASTNAME	
b.WORKDEP'	

Functions:

Name:	Description:
SIZEOF	Number of Characters in S
TRIM	Removes Spaces

Operations:

Name:	Description:
=	Equals
<>	Not Equals
>	Greater Than
<	Less Than
>=	Greater Than or Equal
<=	Less Than or Equal

Expression:

```
FIRSTNME: (SIZEOF (TRIM c.FIRSTNME) >= 5) AND (SIZEOF (TRIM c.LASTNAME) > 0)
```

Figure 1-63 Rule Expression Builder

An example of a complex survive expression built by the Rule Expression Builder is shown in the Expression pane as follows:

```
FIRSTNME: (SIZEOF (TRIM c.FIRSTNME) >= 5) AND (SIZEOF (TRIM c.LASTNAME) > 0);
```

This rule states to retain FIRSTNME of the current record if the column contains five or more characters and if LASTNAME has any contents.

This rule is created by selecting FIRSTNME as the target and using the Rule Expression Builder to create the complex expression. Table 1-15 shows the number of characters in the three records in the first record group.

Table 1-15 Number of characters in each column in a record group

Record	**LASTNAME**	FIRSTNME
1	3	2
2	5	7
3	7	5

Table 1-15 shows the following:

– Record 1 has two characters in FIRSTNME and three characters in LASTNAME. This record fails the test, because FIRSTNME has less than five characters.

– Record 2 has seven characters in FIRSTNME and five in LASTNAME. This record passes the conditions for the rule. The current FIRSTNME (from the second record) becomes the best column.

– Record 3 has five characters in FIRSTNME and seven in LASTNAME and also passes the conditions. FIRSTNME from this record replaces the best value as the new best value.

When you define multiple rules for the same target, the rule that appears later in the list of rules has precedence. For example, if you define two rules for the target LASTNAME, the record value that meets listed conditions for the second rule becomes the best value. If no target passes the second rule, the best values for the first rule become part of the output record.

1.10 Mailing list scenario

In this scenario, we use a real-world scenario to demonstrate a typical data cleansing processing flow using IBM WebSphere QualityStage. You can then extrapolate or customize this process flow to address the unique data cleansing requirements of your organization.

Our scenario assumes a hypothetical national department store named *GOOD ALL*. The store has a customer list of its credit card holders and wants to expand its customer base through promotional marketing of its products and GOOD ALL. It also wants to cleanse and enhance data of its existing customers. GOOD ALL wants to identify households (customers having the same home address) to reduce mailing costs by sending only one set of promotional or other materials to a single household.

To achieve this end, GOOD ALL purchases multiple mailing lists from different sources that includes demographic information such as household incomes and number of members in the household.

To achieve the business objectives for GOOD ALL, the following tasks need to be performed:

1. Cleanse the credit card customer file (standardize, match, and remove duplicates). Also identify households.
2. Cleanse the acquired mailing lists after merging them into a single file (standardize, match, and remove duplicates).
3. Match the cleansed mailing list with the cleansed credit card customer file to achieve the following objectives:
   - Add demographic information to the credit card customers from the mailing list for matched customers to enhance it.
   - Generate a mailing list that is a merge of GOOD ALL's credit card customers and prospects (names and addresses in the mailing list that do not overlap with credit card customers).

After consulting with IBM WebSphere QualityStage specialists, GOOD ALL designs a series of steps to perform these tasks as follows:

- Figure 1-88 on page 141 shows the processing flow that is used for cleansing GOOD ALL's credit card customer list and for determining household information.
- Figure 1-398 on page 341 shows the processing flow that is used for cleansing the mailing lists acquired from multiple data sources, enhancing GOOD ALL's credit card customer list with information from the mailing list, and generating a consolidated mailing list for promotional mailing of its products and services.

IBM Information Server is a project-based development environment as shown in Figure 1-64.

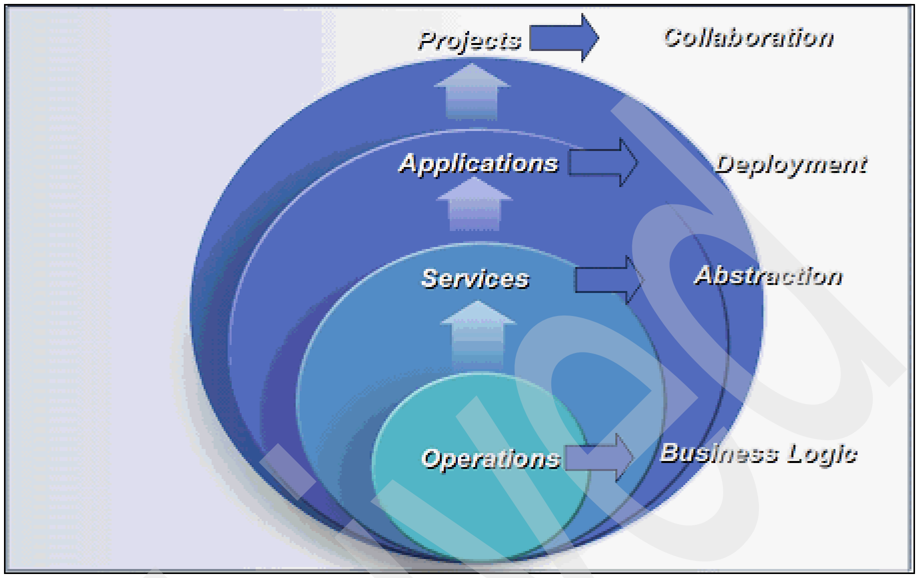


Figure 1-64 IBM Information Server development paradigm

All applications, operations and services are associated with a project as shown in Figure 1-64. Therefore, you first need to create a project before you can define any applications, operations, or services. A project is a collaborative environment that you use to design applications, services, and operations. All project information that you create is saved in the common metadata repository so that it can easily be shared among other IBM Information Server components. For our mailing list scenario, we created a project named PROJQSSAMP as described in 1.10.1, “Create a project” on page 121.

Jobs define the sequence of steps that determine how IBM Information Server performs its work. After they are designed, jobs are compiled and run on the parallel processing engine. The engine runs functions such as connectivity, extraction, cleansing, transformation, and data loading based on the design of the job. The individual steps that make up a job are called stages. IBM Information Server offers dozens of prebuilt stages for performing most common data integration tasks such as sort, merge, join, filter, transform, lookup, aggregate, investigate, standardize, match, and survive. The stages include powerful components for high-performance access to relational databases for reading and loading, including parallel relational databases. IBM Information Server also provides a number of stage types for building and integrating custom stages.

Using IBM WebSphere QualityStage involves designing, executing, managing and deploying, and administering IBM WebSphere DataStage jobs. Each time that a IBM WebSphere DataStage job is validated, run, or scheduled, you can set options to change parameters, override default limits for row processing, assign invocation IDs, and set tracing options. When you have a large number of jobs that run with the same parameters, it is more efficient to create a parameter set object once and have it reused by all the jobs. We created such an object named QSPARAMETERSET as described in 1.10.4, “Create a parameter set object” on page 130.

The jobs created for the mailing list scenario need to be stored in its own “object,” but the folder you choose to save it in is entirely up to you. The same applies to the table definitions that are required for the mailing list scenario. We describe this process in 1.10.2, “Create additional folders” on page 124.

Finally, you need to import table definitions for the credit card customer (CUSTOMER table in the QSSAMPLE database) and mailing list (MAILINGLIST table in the QSSAMPLE database) into the Designer Client as described in 1.10.3, “Import table definitions” on page 126.

After creating the PROJQSSAMP project, creating the required folders, importing the table definitions, and creating the QSPARAMETERSET parameter set object, we designed and executed the tasks for implementing the GOOD ALL mailing list scenario as described in the following sections:

- 1.10.5, “Credit Card Customer cleansing” on page 135
- 1.10.6, “Mailing list cleansing” on page 334
- 1.10.7, “Enhance credit card customers” on page 437
- 1.10.8, “Generate mailing master with household for promotion mailing” on page 472.

Attention: In the following sections, to avoid overburdening you with excessive screen captures, we do not include all the panels through which you typically navigate to perform the desired function. Instead, we include select screen captures (and in some cases, just portions of these screen captures) that highlight the key items of interest, thereby skipping initial screen captures, as well as some intervening screen captures, in the process.

1.10.1 Create a project

Figure 1-65 through Figure 1-70 on page 123 describe the steps to create the PROJQSSAMP project in IBM Information Server as follows:

1. Launch the IBM WebSphere DataStage and QualityStage Administrator program by clicking **Start** → **All Programs** → **IBM Information Server** → **IBM WebSphere DataStage and QualityStage Administrator** as shown in Figure 1-65.
2. Attach to the DataStage server KAZAN.ITSOSJ.SANJOSE.IBM.COM at domain 9.43.86.77:9080 with user name (admin) and the appropriate password as shown in Figure 1-66 on page 122. Click **OK**.
3. Under the Projects tab, click **Add** to add a new project as shown in Figure 1-67 on page 122.
4. Provide the Name (PROJQSSAMP) in Figure 1-68 on page 123 and click **OK**.

   Figure 1-69 on page 123 and Figure 1-70 on page 123 show the successful creation of the PROJQSSAMP project.

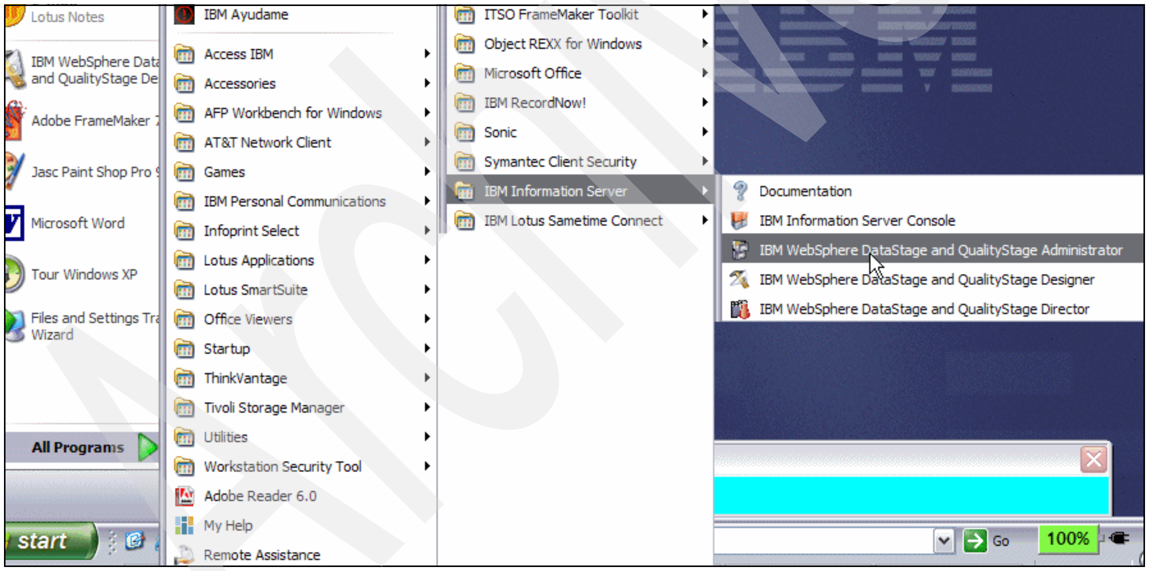
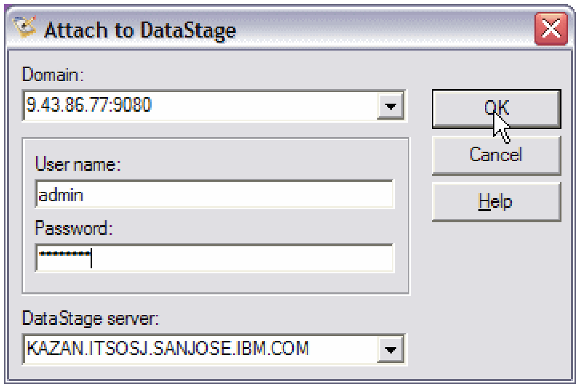


Figure 1-65 Create the PROJQSSAMP project 1/6


Attach to DataStage

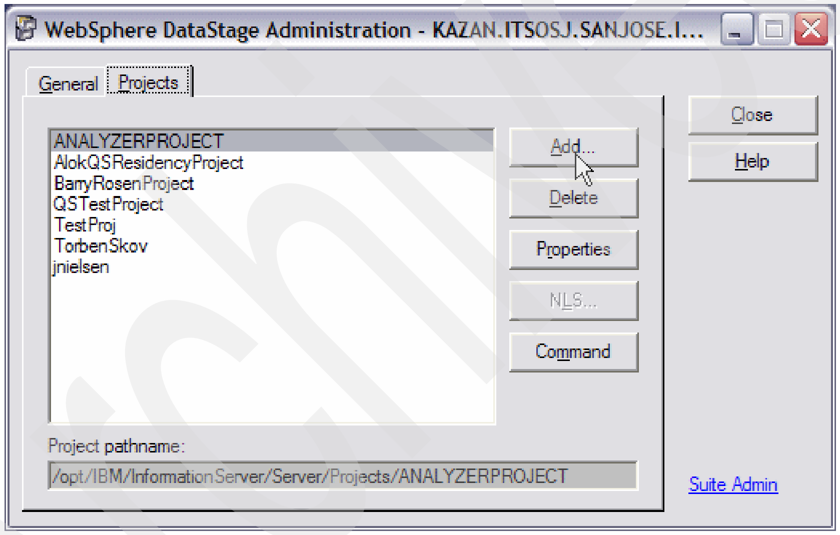
Domain: 9.43.86.77:9080

User name: admin

Password: ********

DataStage server: KAZAN.ITSOSJ.SANJOSE.IBM.COM

Figure 1-66 Create the PROJQSSAMP project 2/6

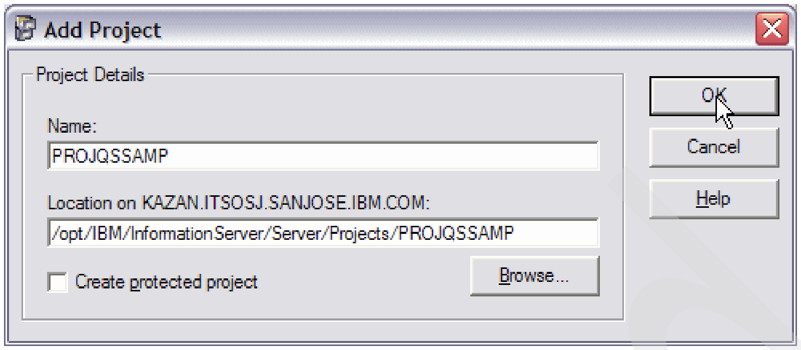

WebSphere DataStage Administration - KAZAN.ITSOSJ.SANJOSE.I...

General | Projects

ANALYZERPROJECT
AlokQSResidencyProject
BarryRosenProject
QSTestProject
TestProj
TorbenSkov
jnielsen

Project pathname: /opt/IBM/InformationServer/Server/Projects/ANALYZERPROJECT

Figure 1-67 Create the PROJQSSAMP project 3/6


Add Project

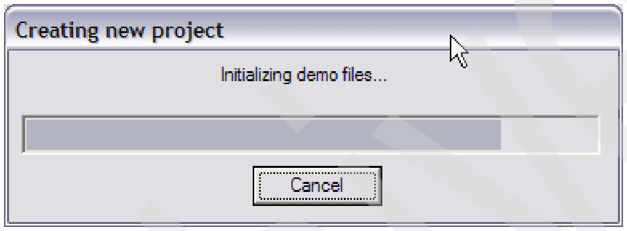
Project Details

Name: PROJQSSAMP

Location on KAZAN.ITSOSJ.SANJOSE.IBM.COM: /opt/IBM/InformationServer/Server/Projects/PROJQSSAMP

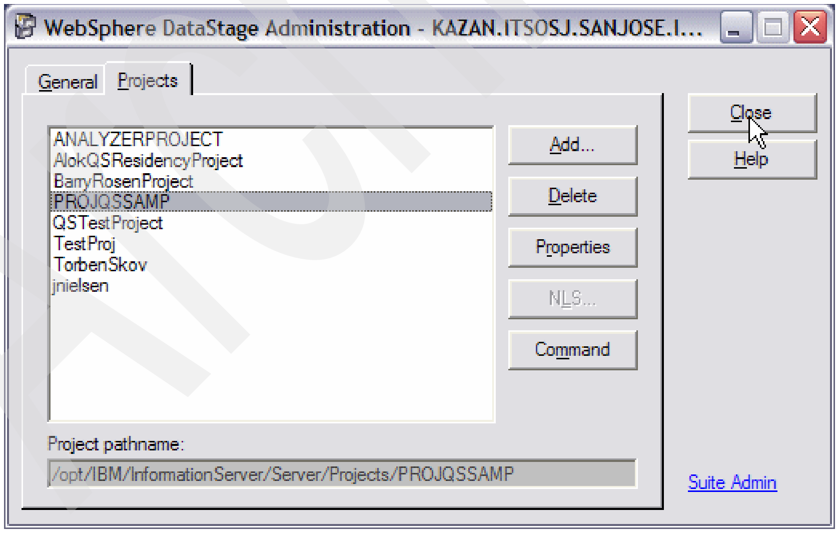
☐ Create protected project

Figure 1-68   Create the PROJQSSAMP project 4/6


Creating new project

Initializing demo files...

Figure 1-69   Create the PROJQSSAMP project 5/6


WebSphere DataStage Administration - KAZAN.ITSOSJ.SANJOSE.I...

ANALYZERPROJECT
AlokQSResidencyProject
BarryRosenProject
PROJQSSAMP
QSTestProject
TestProj
TorbenSkov
jnielsen

Project pathname: /opt/IBM/InformationServer/Server/Projects/PROJQSSAMP

Figure 1-70   Create the PROJQSSAMP project 6/6

1.10.2 Create additional folders

In this section, we create folders in the Jobs and Table Definitions categories in the Designer Client repository tree in which we store jobs and imported table definitions that are required for the mailing list scenario. Figure 1-71 on page 125 through Figure 1-74 on page 126 illustrate this process.

1. Launch the IBM WebSphere DataStage and QualityStage Designer[14] program by clicking **Start** → **All Programs** → **IBM Information Server** → **IBM WebSphere DataStage and QualityStage Designer** as shown in Figure 1-71 on page 125.
2. Attach to project PROJQSSAMP at domain 9.43.86.77:9080 with the user name (admin) and the appropriate password as shown in Figure 1-72 on page 125. Click **OK**.
3. Right-click the Jobs folder in the repository tree, select **New** → **Folder** as shown in Figure 1-73 on page 126.
4. Create a folder PART01 and the subfolders CUSTOMER, MAILING_LIST, and PARAMETER_SET. Figure 1-74 on page 126 shows the resulting tree structure under the Jobs folder.
5. Additional folders (CUSTOMER and MAILING_LIST in the PlugIn folder and folder DSDB2) were created in the Table Definitions folder. We do not show this step here.

[14] Referred to *Designer Client* for the remainder of this book.

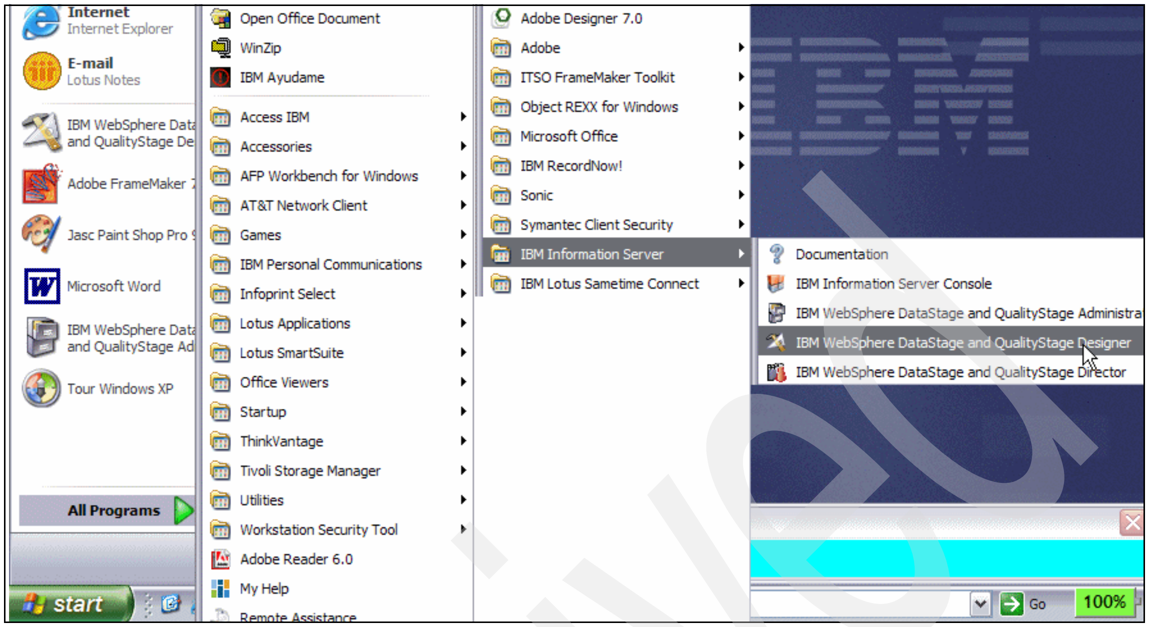
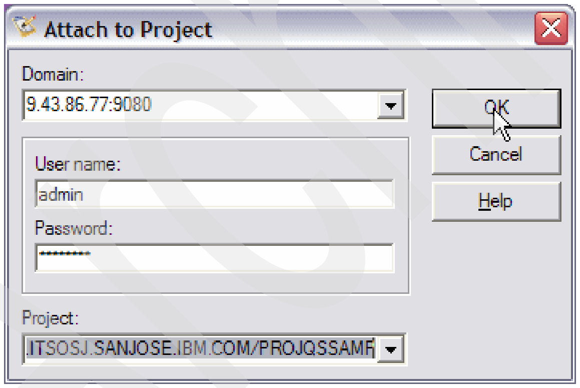


Figure 1-71 Create mailing list scenario folders 1/4



Attach to Project

Domain: 9.43.86.77:9080

User name: admin

Password: ********

Project: ITSOSJ.SANJOSE.IBM.COM/PROJQSSAMP

Figure 1-72 Create mailing list scenario folders 2/4

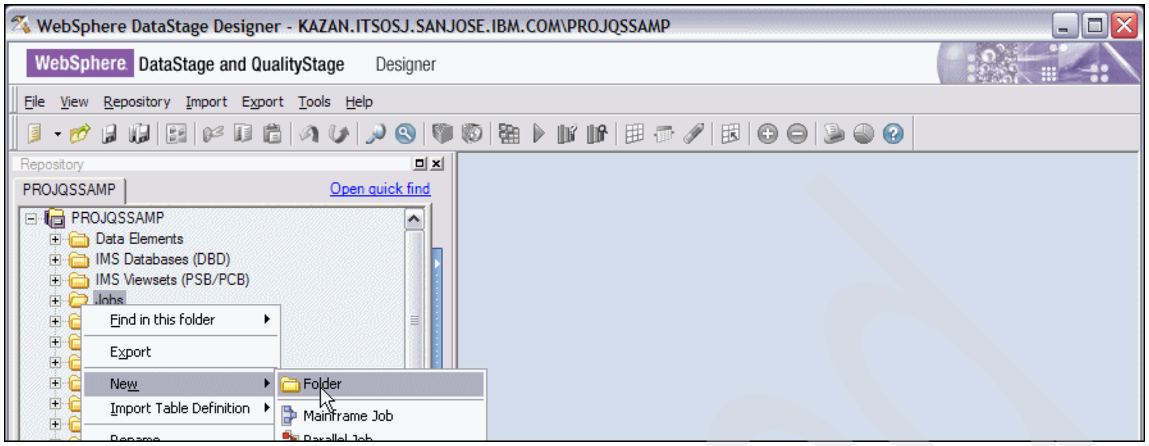


Figure 1-73 Create mailing list scenario folders 3/4

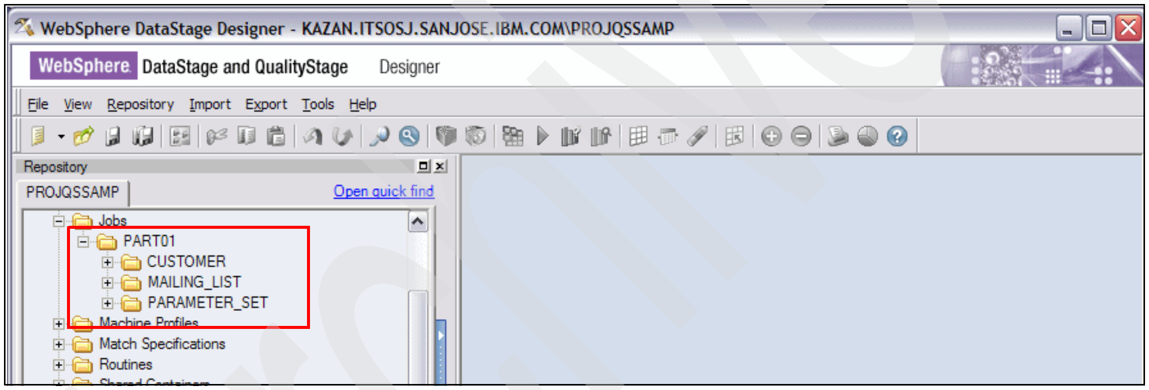

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP
WebSphere DataStage and QualityStage Designer
File View Repository Import Export Tools Help
Repository
PROJQSSAMP
Open quick find
Jobs
PART01
CUSTOMER
MAILING_LIST
PARAMETER_SET
Machine Profiles
Match Specifications
Routines

Figure 1-74 Create mailing list scenario folders 4/4

1.10.3 Import table definitions

In this section, we import the plug-in table definitions for the credit card customer (CUSTOMER table in the QSSAMPLE database) and mailing list (MAILING_LIST table in the QSSAMPLE database) into the Designer Client. This is shown in Figure 1-75 on page 127 through Figure 1-80 on page 130.

1. In the repository tree in the Designer Client, right-click the **Table Definitions** folder, select **Import Table Definition** → **Plug-in Meta Data Definitions** as shown in Figure 1-75 on page 127.
2. Select the plug-in corresponding to IBM DB2 UDB access (DSDB2) and click **OK** as shown in Figure 1-76 on page 128.

3. Provide the Server Name (QSSAMPLE) and user name and password to access it for the import details. Check the boxes Tables and Fully Qualified Table Names and click **Next** as shown in Figure 1-77 on page 128.

4. Select the DB2INST1.CUSTOMER and click **Import** as shown in Figure 1-78 on page 129. The imported metadata definition of the DB2INST1.CUSTOMER table is reflected in the repository tree as shown in Figure 1-79 on page 129.

   We do not show the import of the DB2INST1.MAILING_LIST metadata definition.

   Figure 1-80 on page 130 shows the resulting tree structure under the Table Definitions folder.

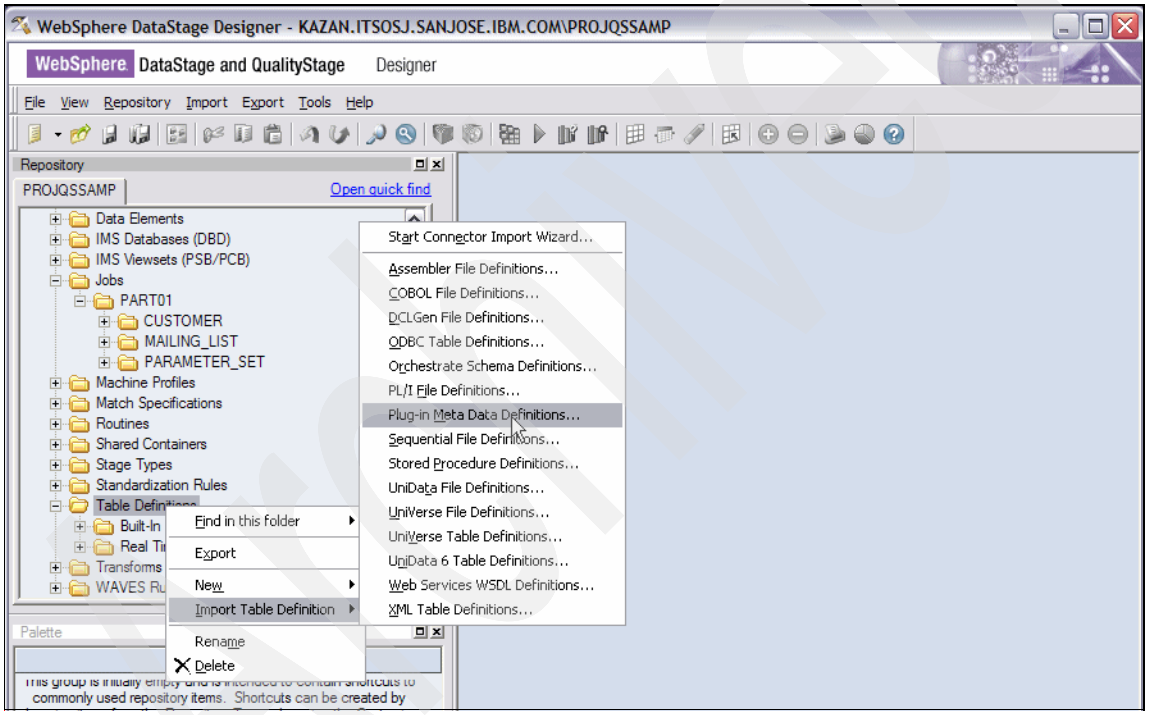
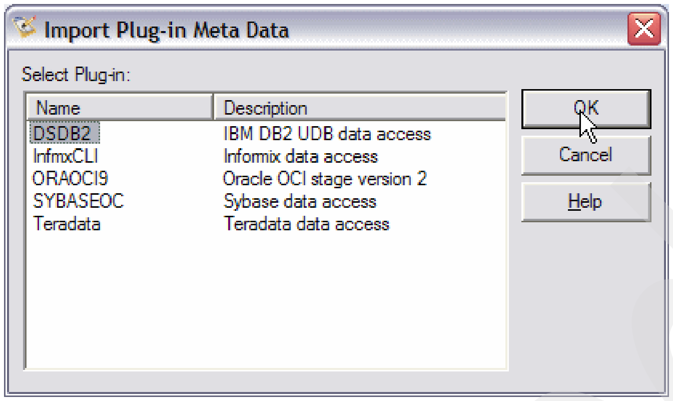

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP
WebSphere DataStage and QualityStage Designer
File View Repository Import Export Tools Help
Repository
PROJQSSAMP
Open quick find
Data Elements
IMS Databases (DBD)
IMS Viewsets (PSB/PCB)
Jobs
PART01
CUSTOMER
MAILING_LIST
PARAMETER_SET
Machine Profiles
Match Specifications
Routines
Shared Containers
Stage Types
Standardization Rules
Table Definitions
Built-In
Real Ti
Transforms
WAVES Ru
Palette
Find in this folder
Export
New
Import Table Definition
Rename
Delete
Start Connector Import Wizard...
Assembler File Definitions...
COBOL File Definitions...
DCLGen File Definitions...
ODBC Table Definitions...
Orchestrate Schema Definitions...
PL/I File Definitions...
Plug-in Meta Data Definitions...
Sequential File Definitions...
Stored Procedure Definitions...
UniData File Definitions...
UniVerse File Definitions...
UniVerse Table Definitions...
UniData 6 Table Definitions...
Web Services WSDL Definitions...
XML Table Definitions...
This group is initially empty and is intended to contain shortcuts to commonly used repository items. Shortcuts can be created by

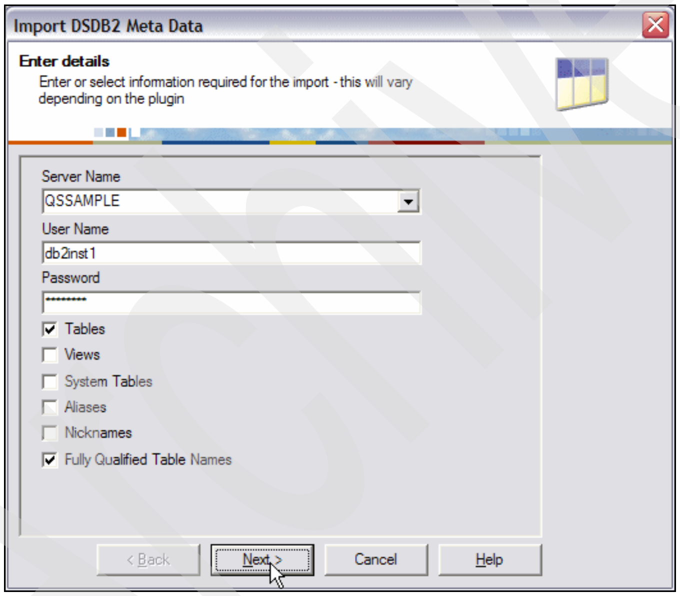
Figure 1-75 Import plug-in metadata definitions 1/6


Import Plug-in Meta Data

Select Plug-in:

Name	Description
DSDB2	IBM DB2 UDB data access
InfmxCLI	Informix data access
ORAOCI9	Oracle OCI stage version 2
SYBASEOC	Sybase data access
Teradata	Teradata data access

Figure 1-76   Import plug-in metadata definitions 2/6


Import DSDB2 Meta Data

Enter details

Enter or select information required for the import - this will vary depending on the plugin

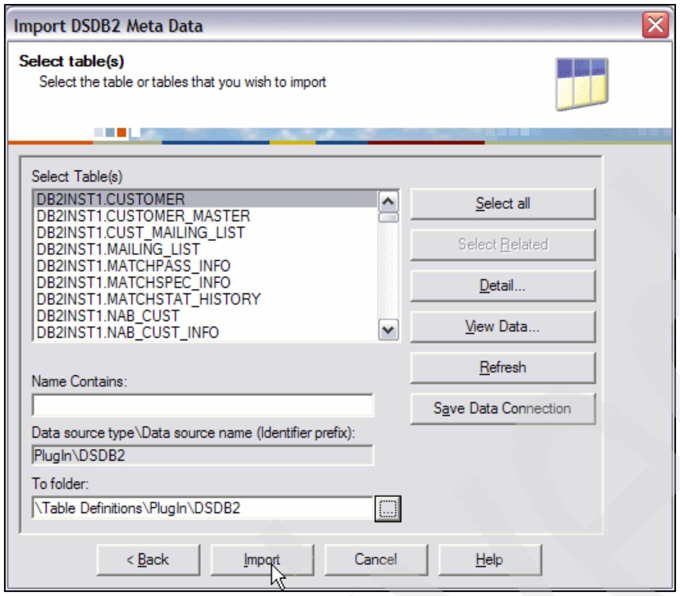
Server Name: QSSAMPLE

User Name: db2inst1

Password: ********

- [✓] Tables
- [ ] Views
- [ ] System Tables
- [ ] Aliases
- [ ] Nicknames
- [✓] Fully Qualified Table Names

Figure 1-77   Import plug-in metadata definitions 3/6


Import DSDB2 Meta Data

Select table(s)

Select the table or tables that you wish to import

Select Table(s)

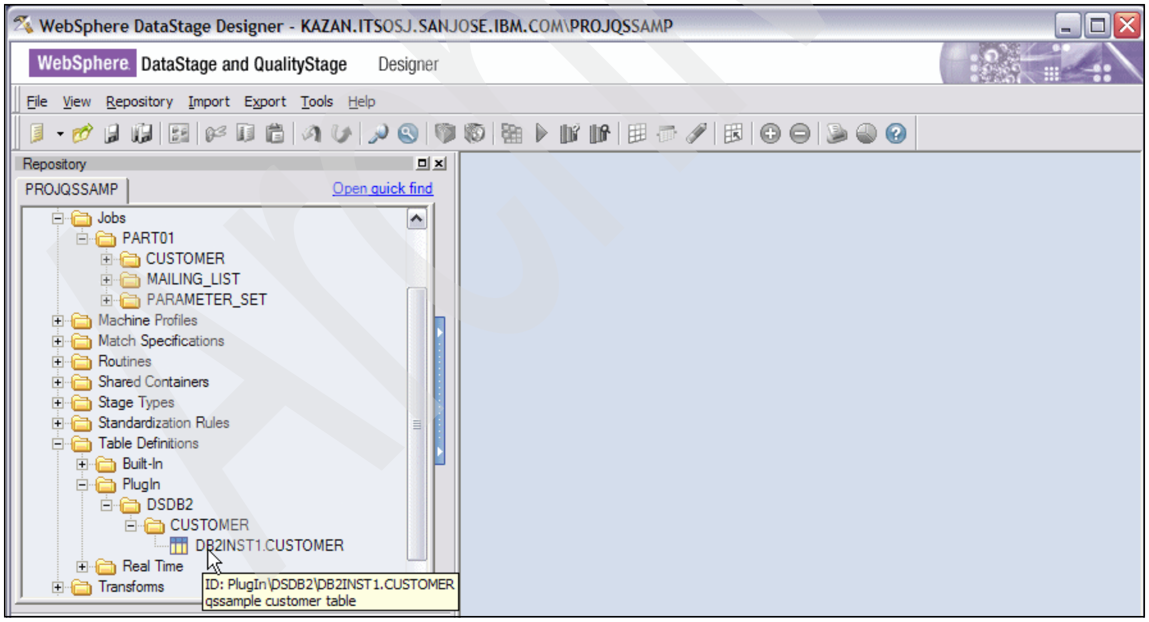
DB2INST1.CUSTOMER
DB2INST1.CUSTOMER_MASTER
DB2INST1.CUST_MAILING_LIST
DB2INST1.MAILING_LIST
DB2INST1.MATCHPASS_INFO
DB2INST1.MATCHSPEC_INFO
DB2INST1.MATCHSTAT_HISTORY
DB2INST1.NAB_CUST
DB2INST1.NAB_CUST_INFO

Name Contains:

Data source type\Data source name (Identifier prefix): PlugIn\DSDB2

To folder: \Table Definitions\PlugIn\DSDB2

Figure 1-78 Import plug-in metadata definitions 4/6


WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP

WebSphere DataStage and QualityStage Designer
File View Repository Import Export Tools Help
Repository
PROJQSSAMP
Open quick find

Jobs
PART01
CUSTOMER
MAILING_LIST
PARAMETER_SET
Machine Profiles
Match Specifications
Routines
Shared Containers
Stage Types
Standardization Rules
Table Definitions
Built-In
PlugIn
DSDB2
CUSTOMER
DB2INST1.CUSTOMER
Real Time
Transforms

ID: PlugIn\DSDB2\DB2INST1.CUSTOMER
qssample customer table

Figure 1-79 Import plug-in metadata definitions 5/6

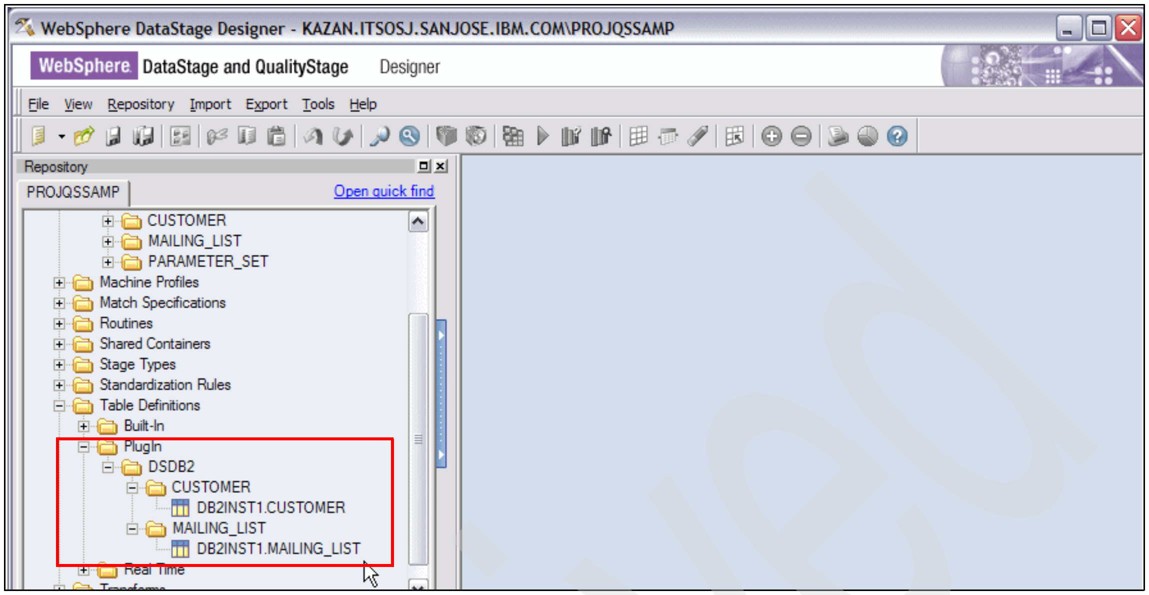


Figure 1-80 Import plug-in metadata definitions 6/6

1.10.4 Create a parameter set object

In this section, we explain how to create a parameter set object named *QSPARAMETERSET* to be reused in all the jobs used in GOOD ALL's mailing list scenario. A parameter set contains the names and values of job parameters and can be referenced by one or more jobs. An environment variable can also be added to the parameter set. The benefits of using a parameter set object is that you can share job parameters across jobs, more easily deploy jobs across machines, more easily propagate a changed job parameter value, and use impact analysis to determine which jobs are using a parameter set. You can override the job parameters in a parameter set at run time.

Figure 1-81 on page 131 through Figure 1-87 on page 134 describe some of the steps involved. To create a parameter set object, we followed these steps:

1. From the **File** menu, select **New** as shown in Figure 1-81 on page 131.
2. Select **Parameter Set** in the New window and click **OK** as shown in Figure 1-82 on page 132 to create such an object.
3. Provide the Parameter Set Name (QSPARAMETERSET) and a Short Description (Parameters shared across all jobs), and then select the Parameters tab as shown in Figure 1-83 on page 132.

4. Key in the Parameter name, Prompt, Type and Default Value parameters and click **OK** as shown in Figure 1-84 on page 133 and Figure 1-85 on page 133. In this window, we added path parameters and the CASS stage related parameters.
5. Save the parameter set object QSPARAMETERSET in the PARAMETER_SET in the Jobs folder as shown in Figure 1-86 on page 134.

   The saved parameter set object QSPARAMETERSET is displayed in the repository tree as shown in Figure 1-87 on page 134.

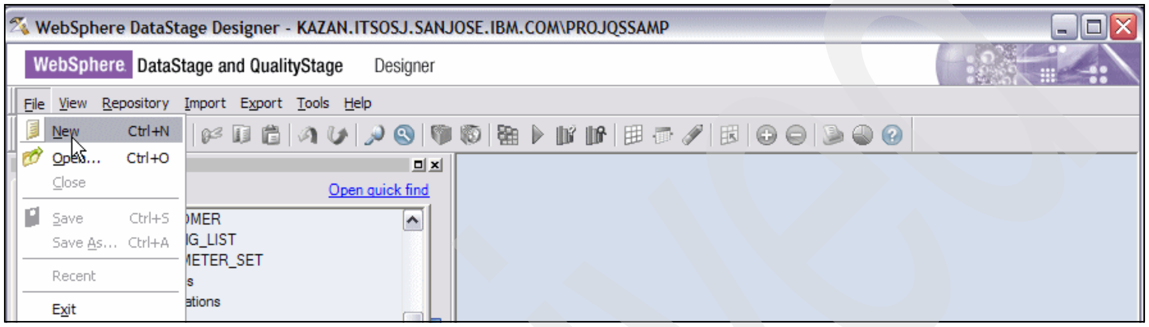
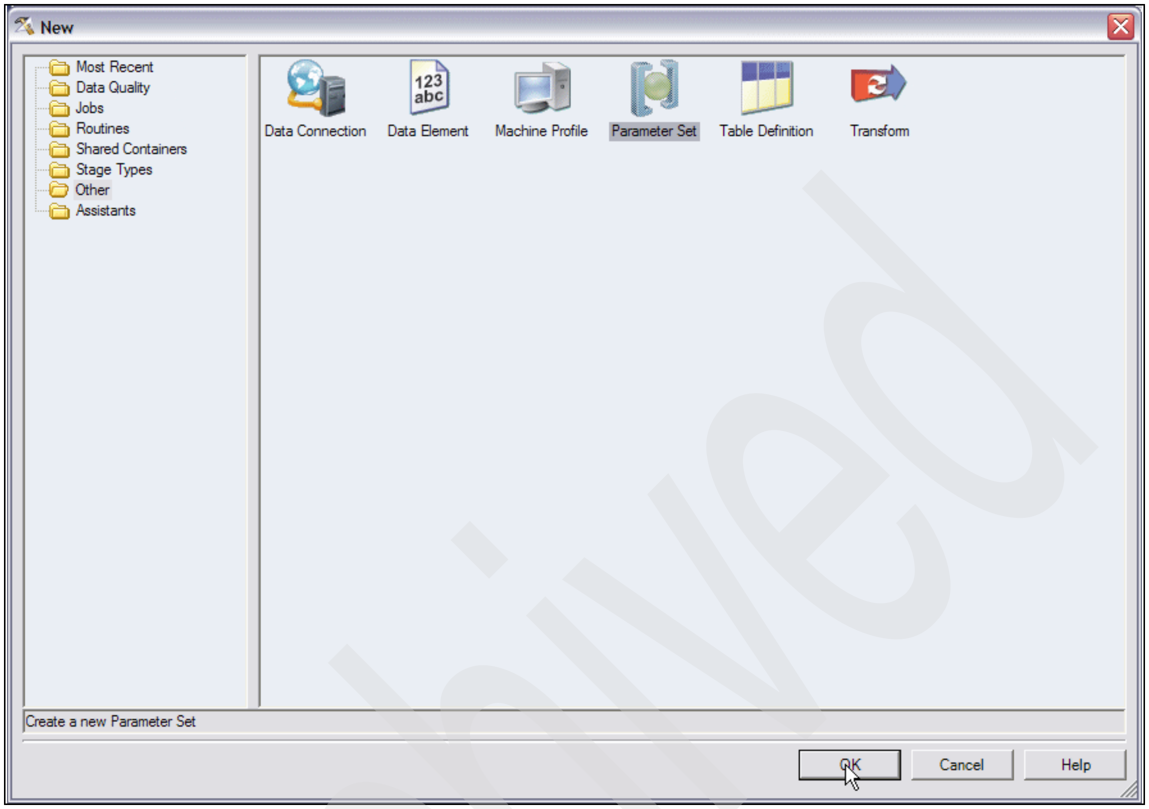
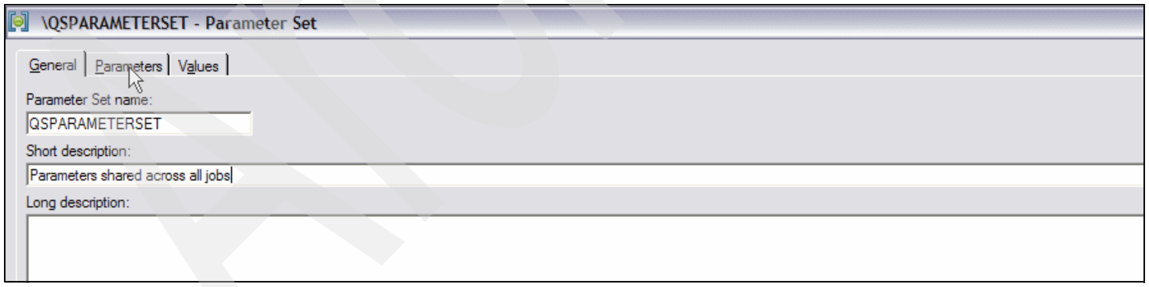

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP
WebSphere DataStage and QualityStage Designer
File View Repository Import Export Tools Help
New Ctrl+N
Open... Ctrl+O
Close
Save Ctrl+S
Save As... Ctrl+A
Recent
Exit
Open quick find
)MER
IG_LIST
IETER_SET
s
ations

Figure 1-81   Create a parameter set specification 1/7



Create a new Parameter Set

Figure 1-82 Create a parameter set specification 2/7

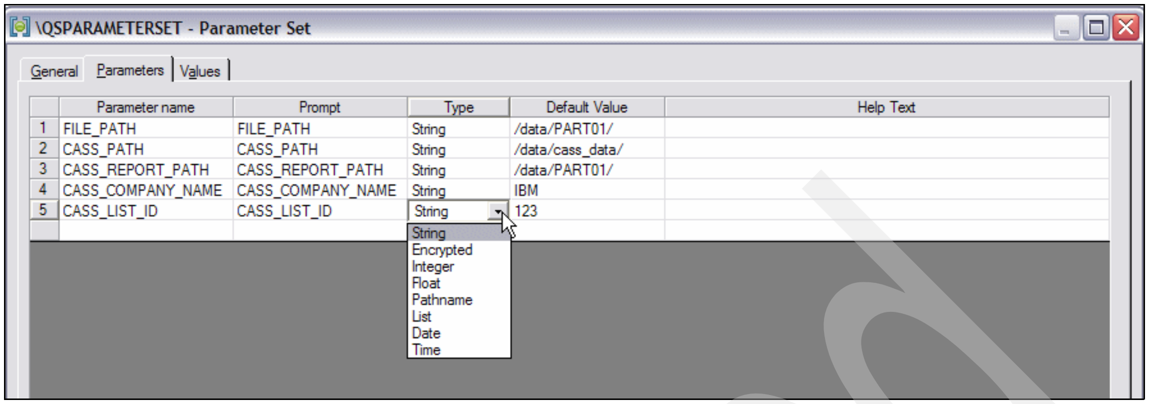


Parameter Set name: QSPARAMETERSET

Short description: Parameters shared across all jobs

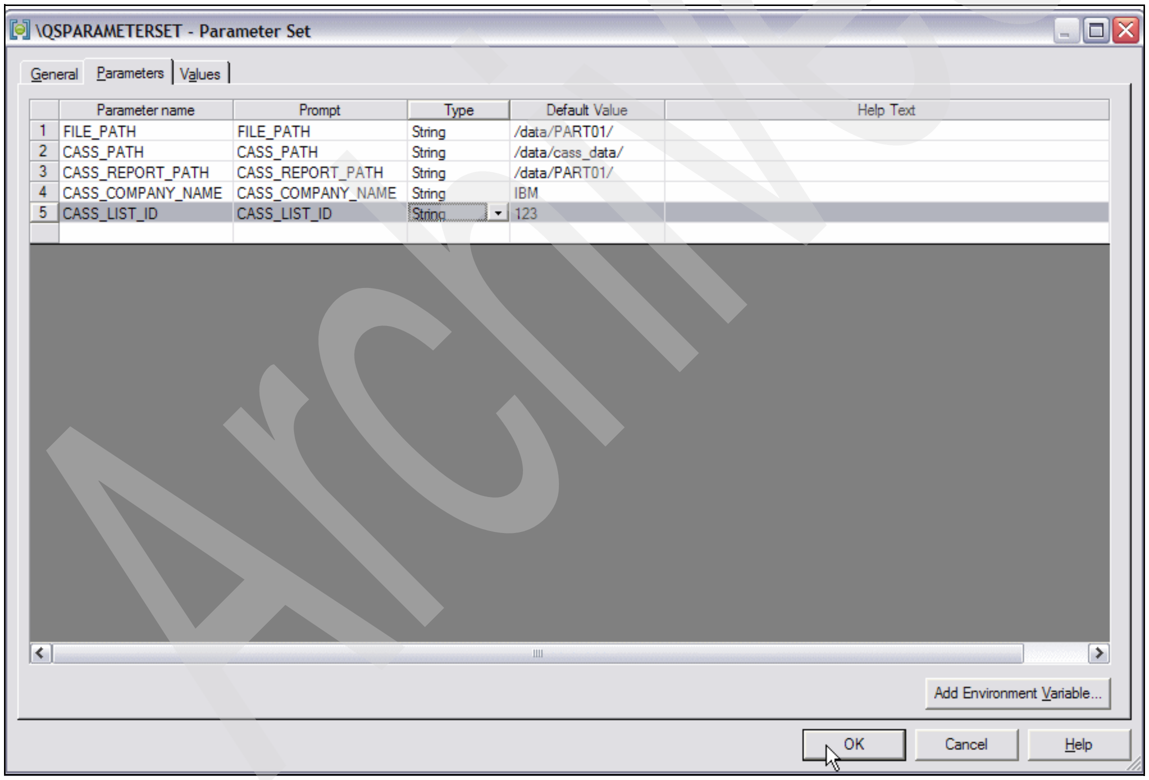
Long description:

Figure 1-83 Create a parameter set specification 3/7



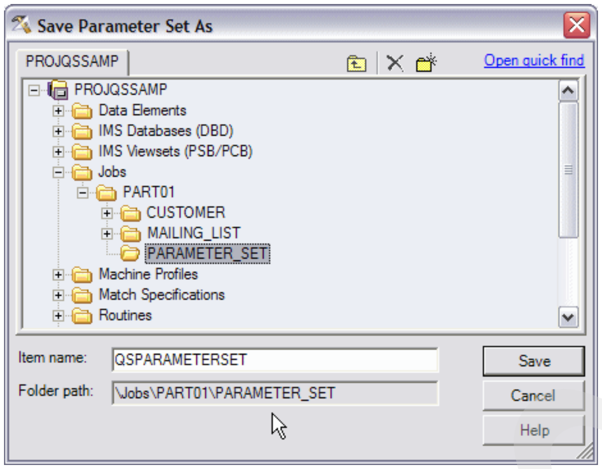
	Parameter name	Prompt	Type	Default Value	Help Text
1	FILE_PATH	FILE_PATH	String	/data/PART01/	
2	CASS_PATH	CASS_PATH	String	/data/cass_data/	
3	CASS_REPORT_PATH	CASS_REPORT_PATH	String	/data/PART01/	
4	CASS_COMPANY_NAME	CASS_COMPANY_NAME	String	IBM	
5	CASS_LIST_ID	CASS_LIST_ID	String	123	

Figure 1-84 Create a parameter set specification 4/7



	Parameter name	Prompt	Type	Default Value	Help Text
1	FILE_PATH	FILE_PATH	String	/data/PART01/	
2	CASS_PATH	CASS_PATH	String	/data/cass_data/	
3	CASS_REPORT_PATH	CASS_REPORT_PATH	String	/data/PART01/	
4	CASS_COMPANY_NAME	CASS_COMPANY_NAME	String	IBM	
5	CASS_LIST_ID	CASS_LIST_ID	String	123	

Figure 1-85 Create a parameter set specification 5/7



Item name: QSPARAMETERSET
Folder path: \Jobs\PART01\PARAMETER_SET

Figure 1-86 Create a parameter set specification 6/7

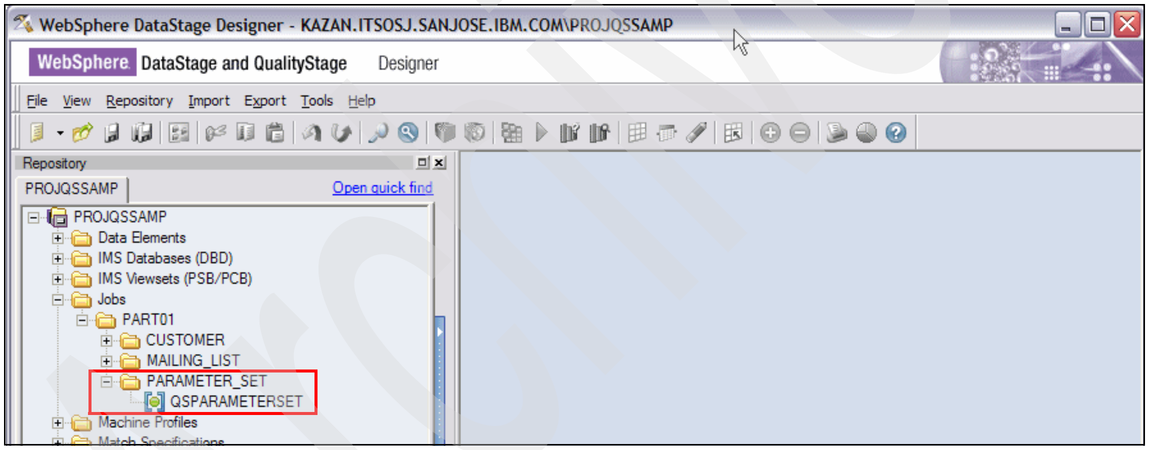


Figure 1-87 Create a parameter set specification 7/7

1.10.5 Credit Card Customer cleansing

Figure 1-88 on page 141 shows the processing flow and jobs that are used for cleansing GOOD ALL's credit card customer list and for determining household information.

We describe the steps briefly here:

1. Extract all the credit card customer data from the DB2 database and load it into a data set so as to isolate it from changes during analysis. Also pre-process it for analysis, which involves changing the default values to nulls for columns such as the telephone number.[15]

   Job "J00_SRC_CUSTOMER" on page 142 performs this step.

2. Split the customer records into two files—one with name/address fields and another for single domain fields such as telephone numbers and e-mail addresses. You split the customer records to allow processing of the single domain and text data in parallel by the QualityStage administrators and appropriate subject matter experts.

   > **Note:** In this scenario, for convenience we chose to use all the columns in the input data for the both streams. Job "J01_STAN_COUNTRY" on page 168 for the name/address fields, and job "J00A_INV_CUSTOMER" on page 162 for the telephone numbers, e-mail, and preferred method contact.
   >
   > In "J00A_INV_CUSTOMER" on page 162 job, we used the Investigate stage on non-text columns such as telephone number, e-mail, and preferred method of contact. We used character concatenate and character discrete with combinations of *C*, *T*, and *X* masks. The objective of this step was to validate some of the main non-text columns. Any errors detected can be resolved by modifying the source directly or by using IBM WebSphere QualityStage jobs to cleanse and modify the targets.

3. Analyze the credit card customer records' addresses to determine the (ISO code) country using the COUNTRY rule set in the Standardize stage.

   Job "J01_STAN_COUNTRY" on page 168 performs this step.

[15] If the content was (999) 999-9999 (user default), then insert a null value in the output.

4. Analyze the ISO codes that were generated by the previous step with the Investigate stage using character discrete with the *C* mask to obtain frequency distribution. This step identified whether the addresses in the credit card customer file belonged to more than one country and identified the codes of the countries in the addresses. In this case, all the addresses were U.S. addresses.

   Job “J02_INVCC_ISCODE” on page 186 performs this step.

5. Use the Standardize stage with the domain-preprocessor rule set USPREP to move name and address data into Name, Address, and Area domains.

   Job “J03_STAN_USPREP” on page 196 performs this step.

6. Use the Investigate stage using word investigate on the Name, Address, and Area domains to determine whether the domain-preprocessor USPREP rule set successfully parsed the tokens in the name/address fields into the correct domains.

   Job “J04_INVW_USPREP” on page 203 performs this step.

7. Perform a visual analysis of the token and pattern reports of the J04_INVW_USPREP step to determine if the parsing was successful.

   In our scenario, we found an error with the parsing, and so chose to perform a classification override for the USPREP rule set to fix the errors.

   Job “J03_Z_Override_And_After” on page 215 performs this step.

8. Repeat the steps in the “J03_STAN_USPREP” on page 196, “J04_INVW_USPREP” on page 203, and “J03_Z_Override_And_After” on page 215 steps until the name and address data is moved into the correct domain columns.

   Attention: You need to generally limit the overrides in the job “J03_Z_Override_And_After” on page 215 to simple classification overrides that move name and address data to the correct name/address/area domain buckets, rather than more complex pattern overrides.

9. After all the name and address data is moved to the correct domain buckets, use the CASS stage to validate, correct, and standardize the U.S. addresses in the Address domain. In our scenario, we also include a Transformer stage to add a second address line column to customer file because CASS requires two address lines as input for its processing.

   Job “J05_CASS_USPREP” on page 219 performs this step.

Note: CASS is a separately priced module that requires installation of the CASS module.

10. Then, run the Investigate stage with character concatenate on the (address related columns) results of the job “J05_CASS_USPREP” on page 219 step to determine addresses not recognized by CASS (delivery point verification or DPV).

    Note: Due to a bug with handling nulls, we introduced a Transformer stage in our scenario to convert nulls to a space using column derivation.

    We also investigated the output of CASS using character concatenate (on CASS generated columns DPVMATCHFLAG_CASS[16] and DPVCODE1_CASS[17]) using a *C* mask. A value of “A1” in the DPVCODE1_CASS field indicated a potential problem.

    Job “J06_INVCC_CASS” on page 228 performs this step.

11. Standardize the name and address contents of the output of job “J05_CASS_USPREP” on page 219 using the domain-preprocessor USPREP rule set. Also add a column to the output that only had the first three characters of the ZIP code using a Transformer stage. In our scenario, this new column (ZIP3) is used as a blocking variable in the next matching stage.

    Job “J07_STAN_CUSTOMER_Domain_Preprocessor” on page 234 preforms this step.

12. Standardize the name and address contents of the output of job “J07_STAN_CUSTOMER_Domain_Preprocessor” on page 234 using the domain-specific USNAME (with column NameDomain_USPREP), USADDR (with column AddressDomain_USPREP) and USAREA (with column AreaDomain_USPREP) rule sets. Three separate processes were defined—one for each rule set.

    Note: Here again, we had to introduce a null handling Transformer stage to circumvent a bug.

[16] Can have one of four values: *Y*, *S*, *D*, and *N*. A value of *Y* indicates that the complete address matches the ZIP code + 4 file and the delivery point is validated. A value of *S* indicates that the address has both primary and secondary components, but only the primary address matches (for example 23 MAIN ST, APT 10 matches to 23 MAIN ST but not to APT 10). A value of *D* indicates that the address is a default (for example, a high-rise building with no apartment indicated or a rural route without a box number). A value of *N* indicates that the address does not match.

[17] A value of *AA* indicates that the input address matches the ZIP code + 4 table, and DPV processing is initiated. If the value is *A1*, the input address does not match the ZIP code + 4 table, and DPV processing is not initiated. All DPV indicators are spaces.

Job “J08_STAN_CUSTOMER_Domain_Specific” on page 241 performs this step.

13. Identify unhandled patterns and classifications in the “J08_STAN_CUSTOMER_Domain_Specific” on page 241. In our scenario, we ran the Investigate stage with character concatenate using the *C* mask on the unhandled pattern column from the results of “J08_STAN_CUSTOMER_Domain_Specific” on page 241. The columns investigated corresponded to the name, address, and area domains.

    Job “J09_INVCC_STAN_CUSTOMER” on page 253 performs this step.

14. Because there are unhandled patterns (such as “++” and “+++”) in the address domain, create pattern overrides in the domain-specific USADDR rule set and re-run the “J08_STAN_CUSTOMER_Domain_Specific” on page 241 step.

    Job “J09_Z_Override_And_After” on page 262 performs this step.

15. After the unhandled patterns are handled, proceed to generate frequency distribution on all the columns in the credit card customer file using the Match Frequency stage. The idea is to generate match frequency for all the columns so that it can be used with any match specification.

    Job “J10_MATCHFREQ_STAN_CUSTOMER” on page 266 performs this step.

16. Generate a match specification for an Unduplicate match stage using as input the match frequency data created in the “J10_Undup_MatchSpec_STAN_CUSTOMER” on page 269 step. The specification includes two passes: The first one blocking on name, address, and area and the second pass on the first three digits of the ZIP code.

    Job “J10_Undup_MatchSpec_STAN_CUSTOMER” on page 269 performs this step.

17. Determine if there were any duplicates in the credit card customer file by running the Unduplicate stage with the match specification and match frequency information that was created in steps “J10_Undup_MatchSpec_STAN_CUSTOMER” on page 269 and “J10_MATCHFREQ_STAN_CUSTOMER” on page 266 respectively. The output is matched (merge of master and duplicates using a Funnel stage) records, records for clerical review, and residuals (records that do not match).

    Job “J11_UNDUP_DEP_MATCH_CUSTOMER” on page 282 performs this step.

18. Then create a single report of the matched credit card customers and the records in the clerical review records to enable a manual review of potential match records.

    Job "J12_CLERICAL_REVIEW_CUSTOMER" on page 290 performs this step.

19. If a manual review of the report finds duplicates, then they have to be considered as part of the file that contains matched records. We do not show this step here.

20. Survive the best information from the set of duplicates using the SURVIVE stage.

    Job "J13_SURVIVE_CUSTOMER" on page 295 performs this step.

21. Create a clean master of credit card customers by merging the master, clerical review (with duplicates removed), and residual records into a sequential file using the FUNNEL stage.

    Job "J14_CUSTOMER_MASTER" on page 303 performs this step.

22. Copy the contents of the sequential file that was created in the "J14_CUSTOMER_MASTER" on page 303 step into a data set, because a subsequent match specification that reads this data only accepts a data set as input.

    Job "J14A_CUSTOMER_MASTER" on page 316 performs this step.

    Note: In our scenario, we could not directly create the merged credit card customer file into a data set in the "J14_CUSTOMER_MASTER" on page 303 job due to a bug. Therefore, we had to circumvent the bug by first creating the sequential file and then copying its contents into a data set in a separate step.

23. Generate frequency distribution on all the columns in the cleansed credit card customer file of the "J14A_CUSTOMER_MASTER" on page 316 step using the Match Frequency stage. Here again, the idea is to generate match frequency for all the columns so that it can be used with any match specification.

    Job "J15_FREQ_CUSTOMER_MASTER" on page 316 performed this step.

24. Generate a match specification for an Unduplicate match stage using as input the match frequency data that was created in the "J15_FREQ_CUSTOMER_MASTER" on page 316 job. The specification included two passes: The first pass blocks on the primary (last) name and a phonetic encoding (NYSIIS) of the address, and the second pass blocks on the five digit ZIP code and a phonetic encoding (NYSIIS) of the address.

Note: We repeated the phonetic encoding (NYSIIS) of the address as a blocking variable in the second pass to reduce the potentially large block size that could arise out of choosing a blocking variable only involving the ZIP code. While it was not an issue in our data, it is likely to be an issue in a real-world environment.

Job “J15_Undup_MatchSpec_CUSTOMER” on page 318 performs this step.

25. Determine whether there were duplicates in the credit card customers file using the Unduplicate stage with the match specification and match frequency information that was created in steps “J15_Undup_MatchSpec_CUSTOMER” on page 318 and “J15_FREQ_CUSTOMER_MASTER” on page 316 respectively. As before, the output is matched records (merge of master and duplicates using a Funnel stage), records for clerical review, and residuals (records that do not match).

    Job “J16_UNDUP_IND_MATCH_CUSTOMER” on page 322 performs this step.

26. Create a clean master of credit card customers by merging the master and residual records into a data set using the FUNNEL stage. We added a household ID (using a Transformer stage) to these records in the output, with a value (qsMatchSetID) when a record belonged to a household and a zero when it did not belong to a household.

    Job “J17_CUSTOMER_MASTER_WITH_HOUSEHOLD” on page 327 performs this step.

Note: At this point, we now have a cleansed credit card customer file with household information.

27. Generate frequency distribution on all the columns in the cleansed credit card customer file including household information of the “J17_CUSTOMER_MASTER_WITH_HOUSEHOLD” on page 327 step using the Match Frequency stage. Here again, the idea is to generate match frequency for all the columns so that it can be used with any match specification. This is used in a later matching stage with mailing lists as described in 1.10.6, “Mailing list cleansing” on page 334.

    Job “J18_MATCHFREQ_CUSTOMER_WITH_HOUSEHOLD” on page 332 performs this step.

We describe these jobs in more detail in the following sections.

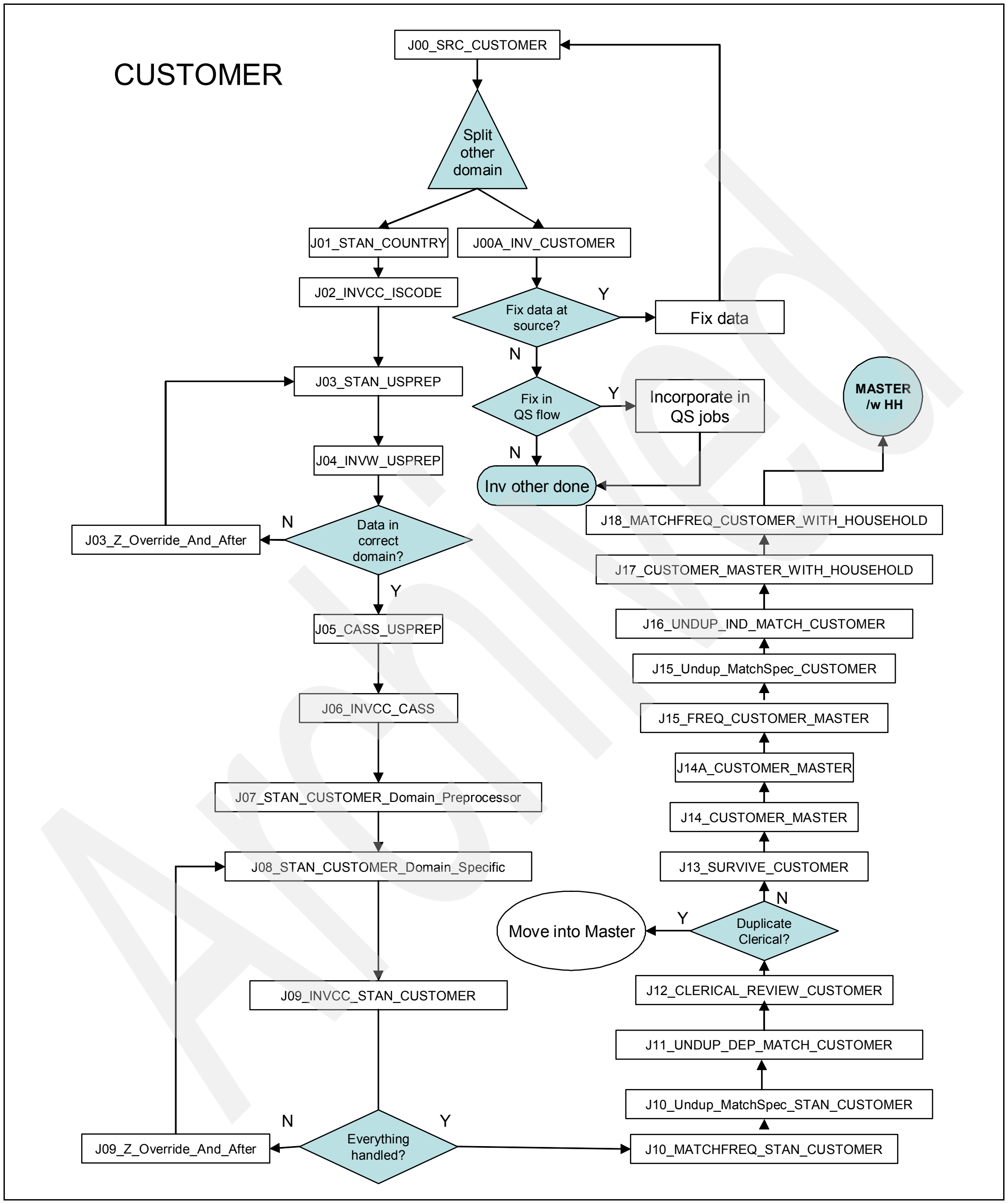


Figure 1-88 Customer information cleanup process flow

J00_SRC_CUSTOMER

As mentioned earlier, this step involves extracting credit card customer data from the DB2 database and loading it into a data set to isolate it from changes during analysis.

Figure 1-89 on page 145 through Figure 1-127 on page 162 describe the steps using Designer Client to build and execute the DataStage job to perform this task. A DB2 UDB API stage is used to access the data in the CUSTOMER table. Its rows are pre-processed to convert default values (such as (999) 999-9999) in the PHONE column to nulls using the Transformer stage. The transformed data is written to a data set.

The steps are as follows:

1. From the **File** menu of the Designer Client, select **New** as shown in Figure 1-89 on page 145.
2. From the New window, select the Jobs folder in the left pane and then select the **Parallel Job** icon in the right pane. Click **OK** as shown in Figure 1-90 on page 145.
3. Click **View** → **Pallete** to open the Pallete pane. Then from the Database section, select and drag the **DB2 UDB** stage icon onto the Designer canvas and drop it to the left of the canvas as shown in Figure 1-91 on page 146 and Figure 1-92 on page 146.
4. Drag a **Transformer** stage icon from the Processing section, and drop it near the middle of the canvas as shown in Figure 1-93 on page 147.
5. Add the **Data Set** icon from the **File** section to the right of the canvas as shown in Figure 1-94 on page 147.
6. Right-click the **DB2 UDB API** stage icon and drag a link to the **Transformer** stage icon and another from the **Transformer** stage icon to the **Data Set** stage icon as shown in Figure 1-95 on page 148 and Figure 1-96 on page 148.
7. Change the name of the stages to CUSTOMER_DATA (from DB2_UDB_API_0), RECODE_PHONE (from Transformer _1), and CUSTOMER (from Data_Set_2) by clicking the stage icon until a highlighted box displays around the name. Then type in the new name, and click outside the box to deselect it. We do not show this process here.

   Also change the name of the links between the stages to IN (from DSLink3) and OUT (from DSLink4). To rename a link, right-click the particular generic link name (DSLinknn) and select **Rename** from the shortcut menu. A highlighted box displays around the default name. Type in the new name and then click outside the box to deselect it. We do not show this process here.

8. To configure the DB2 UDB stage, double click the **CUSTOMER_DATA** stage icon to view the Properties window as shown in Figure 1-97 on page 149.
9. In the CUSTOMER_DATA - DSDB2PX stage window, provide details of the Server name (QSSAMPLE), User ID (db2inst1) and Password to connect to it. Let the Transaction Isolation default to Cursor Stability as shown in Figure 1-98 on page 149.
10. Select the Output tab in Figure 1-98 on page 149 to view Figure 1-99 on page 149. For the Output name (IN) link, in the General tab, select Use SQL Builder tool from the Query Type drop-down list, and click **SQL Builder** to build the query to access the CUSTOMER table.

    Figure 1-100 on page 150 through Figure 1-104 on page 152 describe the building of a query that retrieves all the columns in the CUSTOMER table. We do not intend to describe the navigation of the screens to build the SQL query.
11. Configure the Transformer stage to replace default values with nulls as shown in Figure 1-105 on page 153. From the Transformer stage window, right-click the **Input** columns and choose **Select All** to highlight all the columns from the **DB2 UDB API** stage. Drag the selected columns to the Output link. You have now populated the Output pane and the Output metadata pane. We do not show this process here.

    To add derivations to the PHONE column, double click the **Derivation** area for the PHONE column to open the Expression Editor and type the following derivative:

    ```
    If IN.PHONE ='(999) 999-9999' Then '' Else IN.PHONE
    ```

    This derivative directs that a value of `(999) 999-9999` be replaced with a null. If not, leave the value as is.
12. To configure the data set CUSTOMER object, right-click the **Data Set** stage icon and select **Properties** as shown in Figure 1-106 on page 153.
13. In the CUSTOMER - Data Set window, expand Target under the **Properties** tab, select **File = ?**. You could directly type the path and file name for this object, but we chose to use the same path where the QSPARAMETERSET is stored. This is achieved by selecting **Insert job parameter** from the File drop-down list and then by typing **QSPARAMETERSET.FILE_PATH** that was defined in 1.10.4, "Create a parameter set object" on page 130. Then, enter the name of the file CUSTOMER.ds. This sequence of steps is shown in Figure 1-107 on page 154 through Figure 1-109 on page 154.
14. Select the Columns tab in Figure 1-109 on page 154 to view all the columns associated with the OUT link under the Properties tab as shown in Figure 1-110 on page 155.

    You can save this set of columns as a table definition by clicking **Save** in Figure 1-110 on page 155.

Figure 1-111 on page 155 through Figure 1-113 on page 156 show the saving of this table definition as CUSTOMER_KEY in the folder path \Table Definitions\ PlugIn\DSDB2\CUSTOMER.

15. When all the stages have been configured, save this parallel job from the **File** menu by selecting **Save As**. Figure 1-186 on page 200 and Figure 1-115 on page 157 show saving this job as j00_SRC_CUSTOMER in the folder path \Jobs\Part01\CUSTOMER.
16. After the j00_SRC_CUSTOMER job is saved, you can associate job parameters object QSPARAMETERSET with it as shown in Figure 1-116 on page 158 through Figure 1-118 on page 159.
    a. Click the **Job Properties** icon as shown in Figure 1-116 on page 158.
    b. In the \Jobs\PART01\CUSTOMER\j00_SRC_CUSTOMER - Job Properties window under the Parameters tab, click **Add Parameter Set** as shown in Figure 1-117 on page 158.
    c. In the Open window, select the QSPARAMETERSET object that was created in 1.10.4, “Create a parameter set object” on page 130 and click **OK**.
17. We can now compile the job by clicking the **Compile** icon as shown in Figure 1-119 on page 159. A successful compile status is shown in Figure 1-120 on page 159.
18. To run the compiled job, click the **Run** icon as shown in Figure 1-121 on page 160. In the following j00_SRC_CUSTOMER - Job Run Options dialog window, you can choose to modify the job parameters for this execution. Select QSPARAMETERSET and click **Run** as shown in Figure 1-122 on page 160. When the execution is complete, the links between the stages turn green (if it ran without errors), and statistics about rows processed is displayed as shown in Figure 1-123 on page 160.
19. You can now view the rows in the CUSTOMER data set object by right-clicking that icon and selecting **Properties** as shown in Figure 1-124 on page 161. In the following CUSTOMER - Data Set window, select **File=#QSPARAMETERSET.FILE_PATH#CUSTOMER.ds** in the **Target** folder, and click **View Data** as shown in Figure 1-125 on page 161. At the prompt in the j00_SRC_CUSTOMER...CUSTOMER.OUT - Data Browser window, alter the Rows to display, or Skip count and click OK as shown in Figure 1-126 on page 161. The contents of the CUSTOMER Data Set object are shown in Figure 1-127 on page 162.

Now, proceed to “J01_STAN_COUNTRY” on page 168.

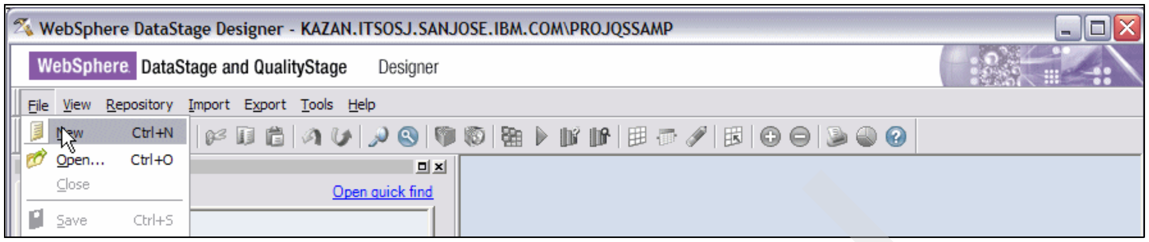

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP
WebSphere DataStage and QualityStage Designer
File View Repository Import Export Tools Help
New Ctrl+N
Open... Ctrl+O
Close
Save Ctrl+S
Open quick find

Figure 1-89   Create the J00_SRC_CUSTOMER job 1/39

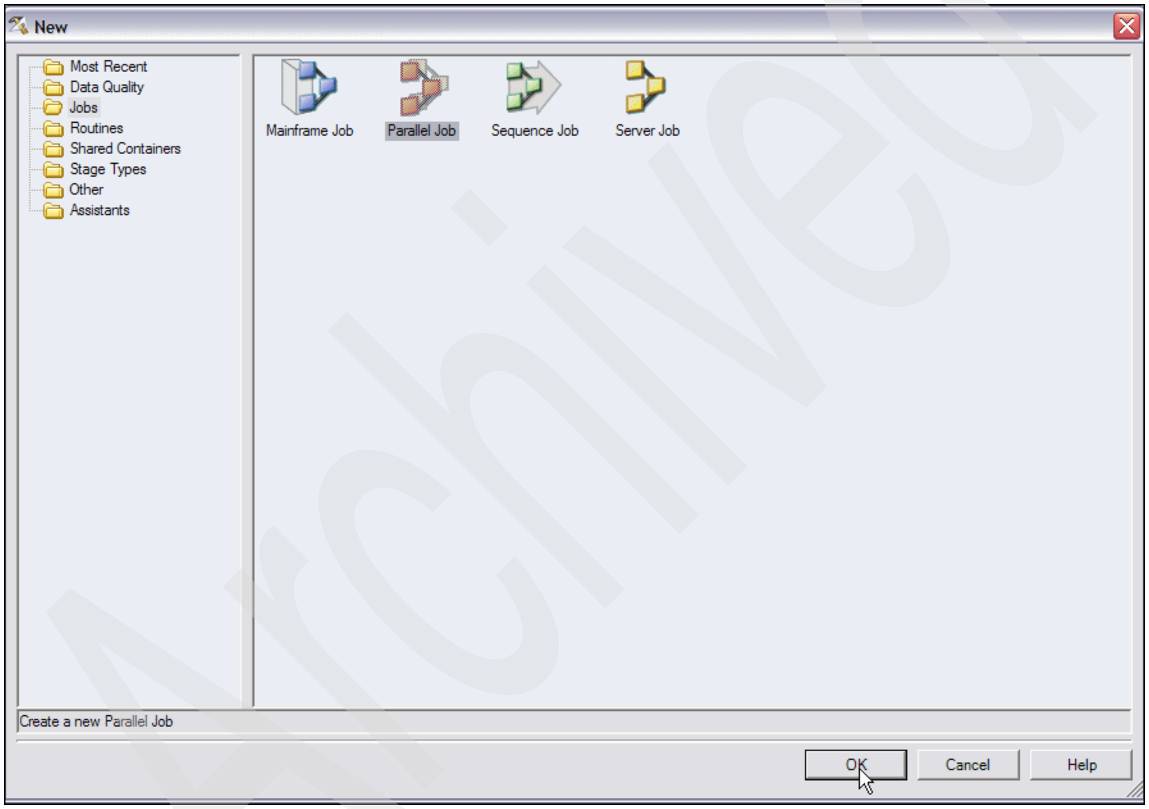


Figure 1-90   Create the J00_SRC_CUSTOMER job 2/39

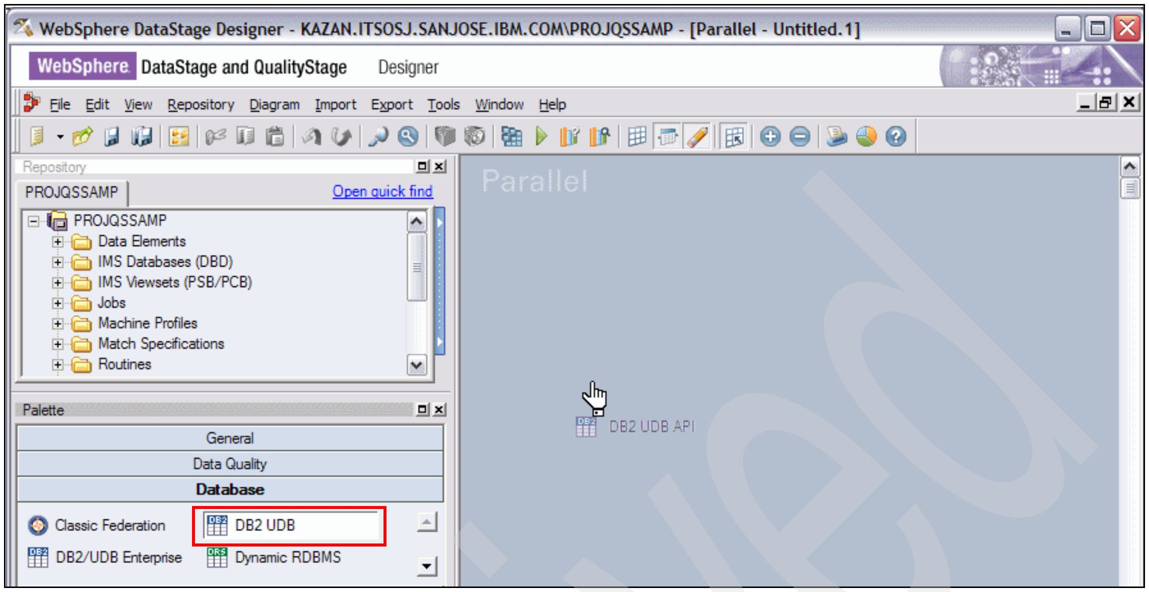


Figure 1-91   Create the J00_SRC_CUSTOMER job 3/39

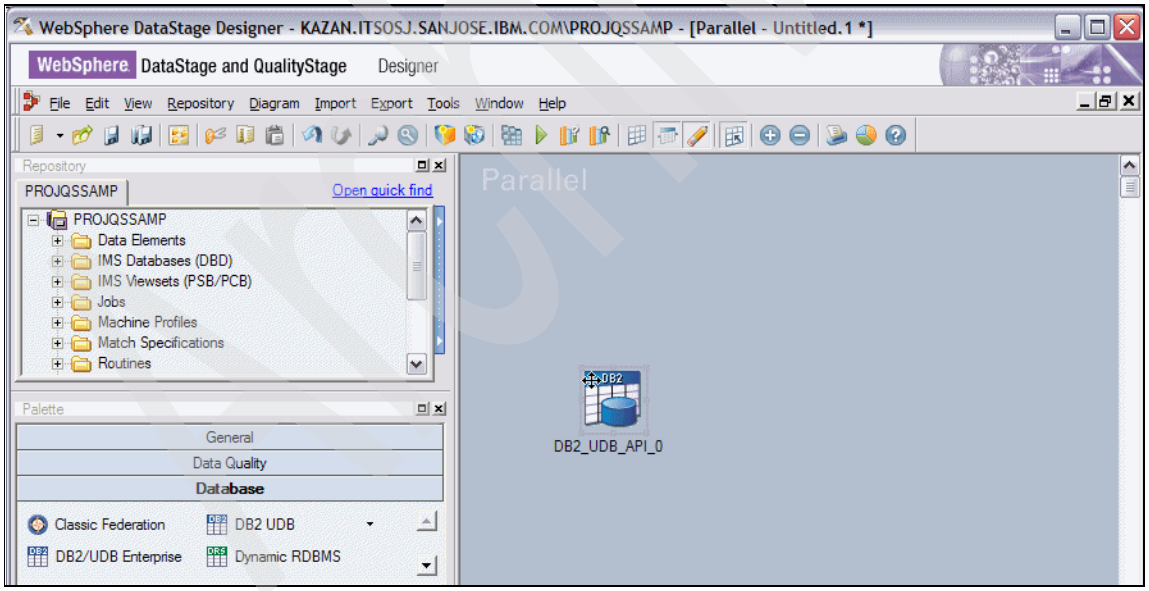


Figure 1-92   Create the J00_SRC_CUSTOMER job 4/39

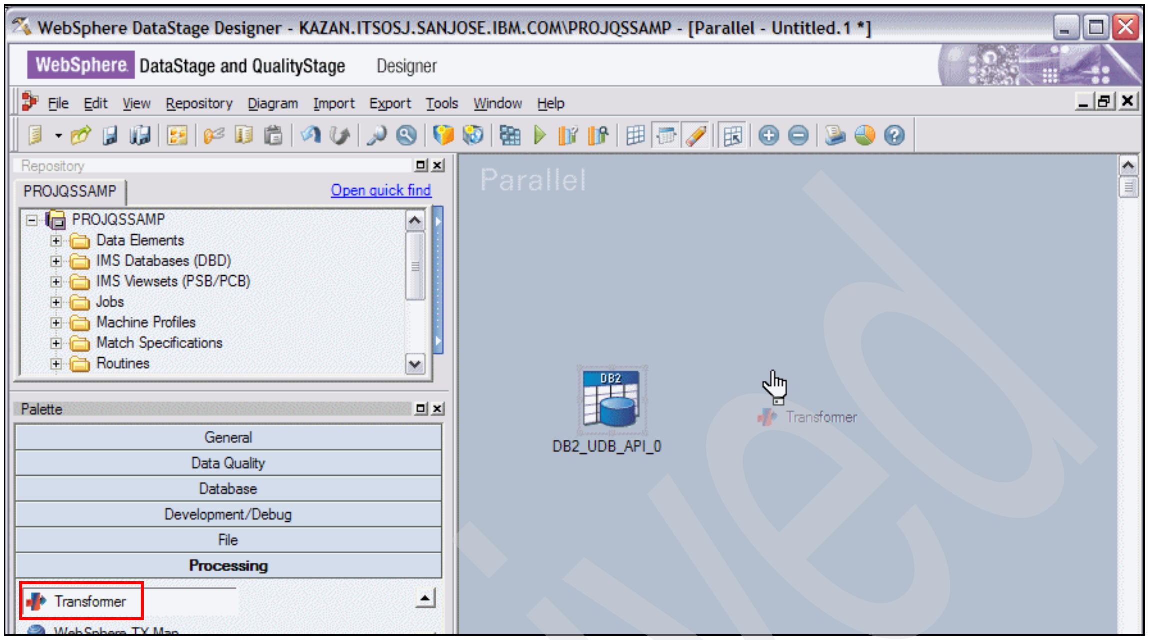


Figure 1-93 Create the J00_SRC_CUSTOMER job 5/39

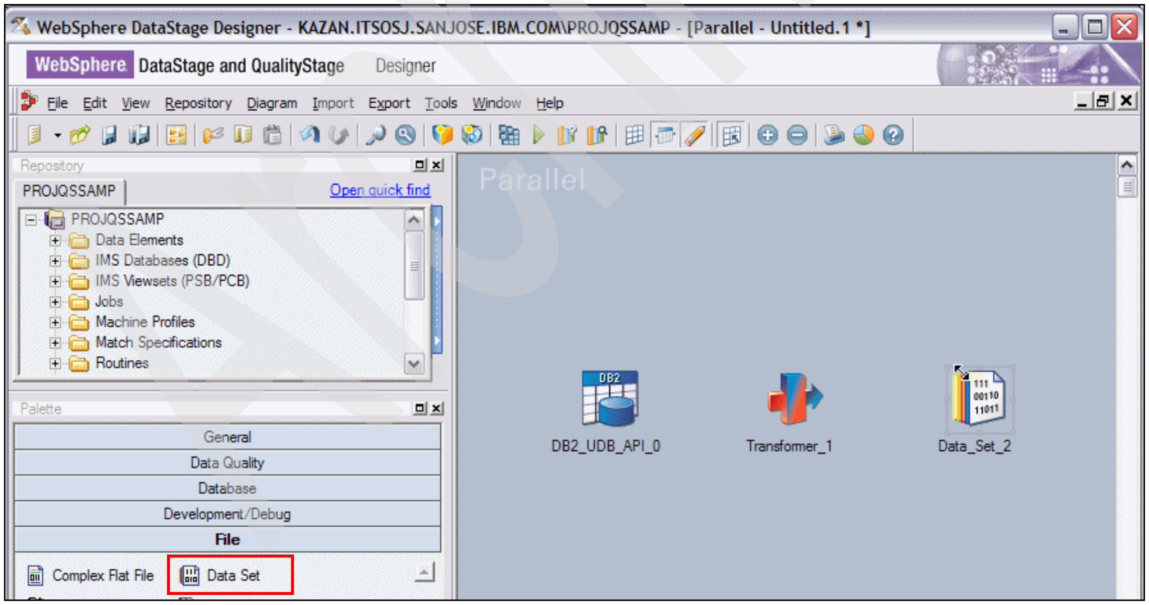


Figure 1-94 Create the J00_SRC_CUSTOMER job 6/39

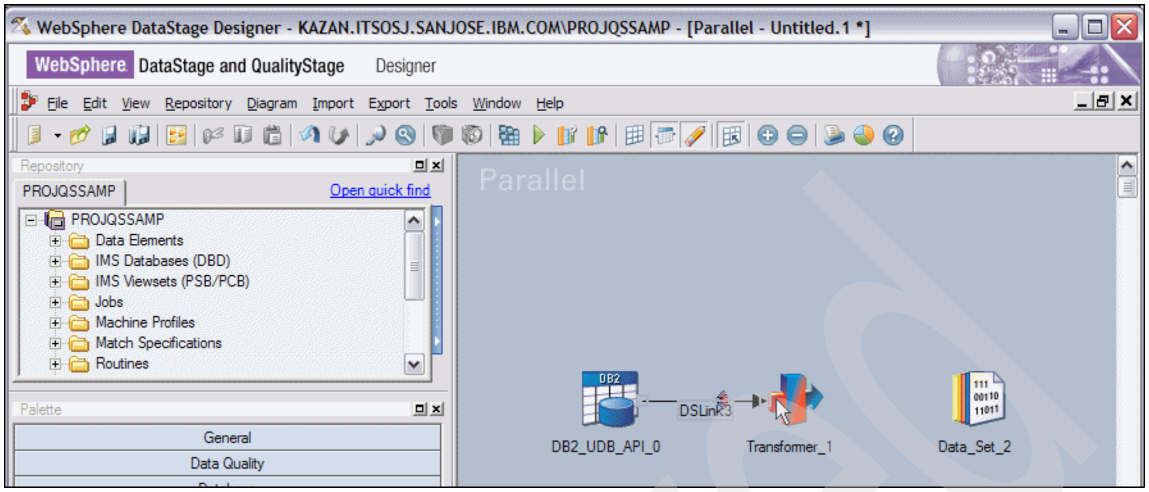


Figure 1-95 Create the J00_SRC_CUSTOMER job 7/39

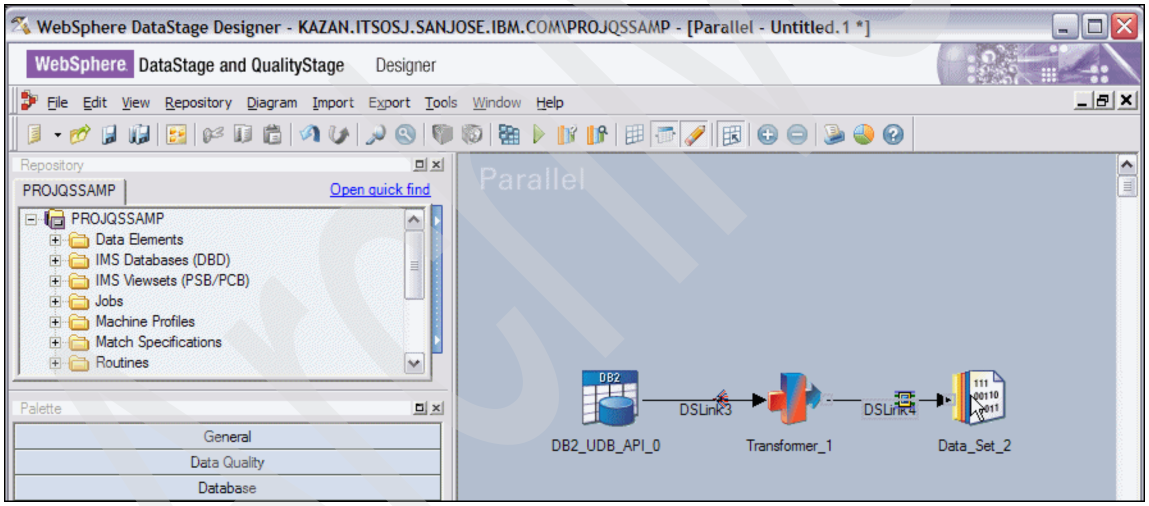


Figure 1-96 Create the J00_SRC_CUSTOMER job 8/39

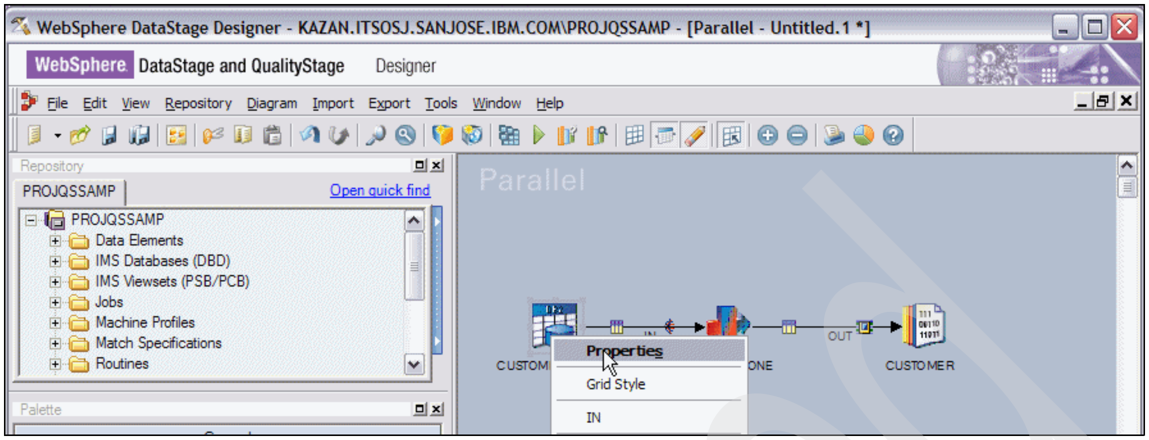
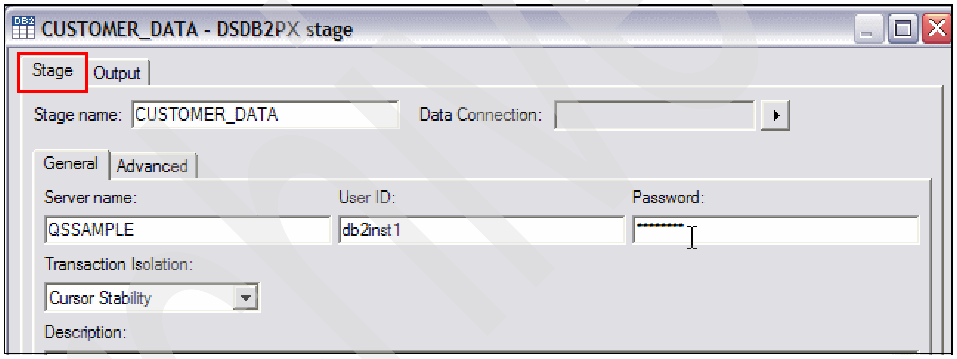
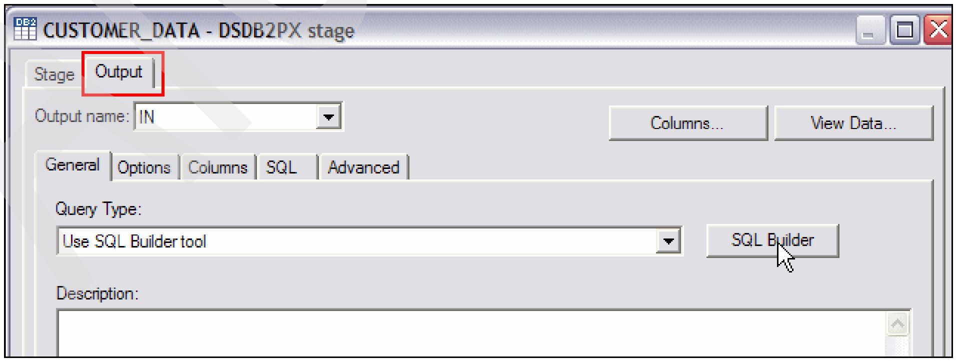

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - Untitled.1 *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Repository
PROJQSSAMP
Open quick find
PROJQSSAMP
Data Elements
IMS Databases (DBD)
IMS Viewsets (PSB/PCB)
Jobs
Machine Profiles
Match Specifications
Routines
Palette
Parallel
CUSTOM
ONE
OUT
CUSTOMER
Properties
Grid Style
IN

Figure 1-97 Create the J00_SRC_CUSTOMER job 9/39



Stage name: CUSTOMER_DATA
Data Connection:
Server name: QSSAMPLE
User ID: db2inst1
Password: *********
Transaction Isolation: Cursor Stability
Description:

Figure 1-98 Create the J00_SRC_CUSTOMER job 10/39



Output name: IN
Query Type: Use SQL Builder tool
Description:

Figure 1-99 Create the J00_SRC_CUSTOMER job 11/39

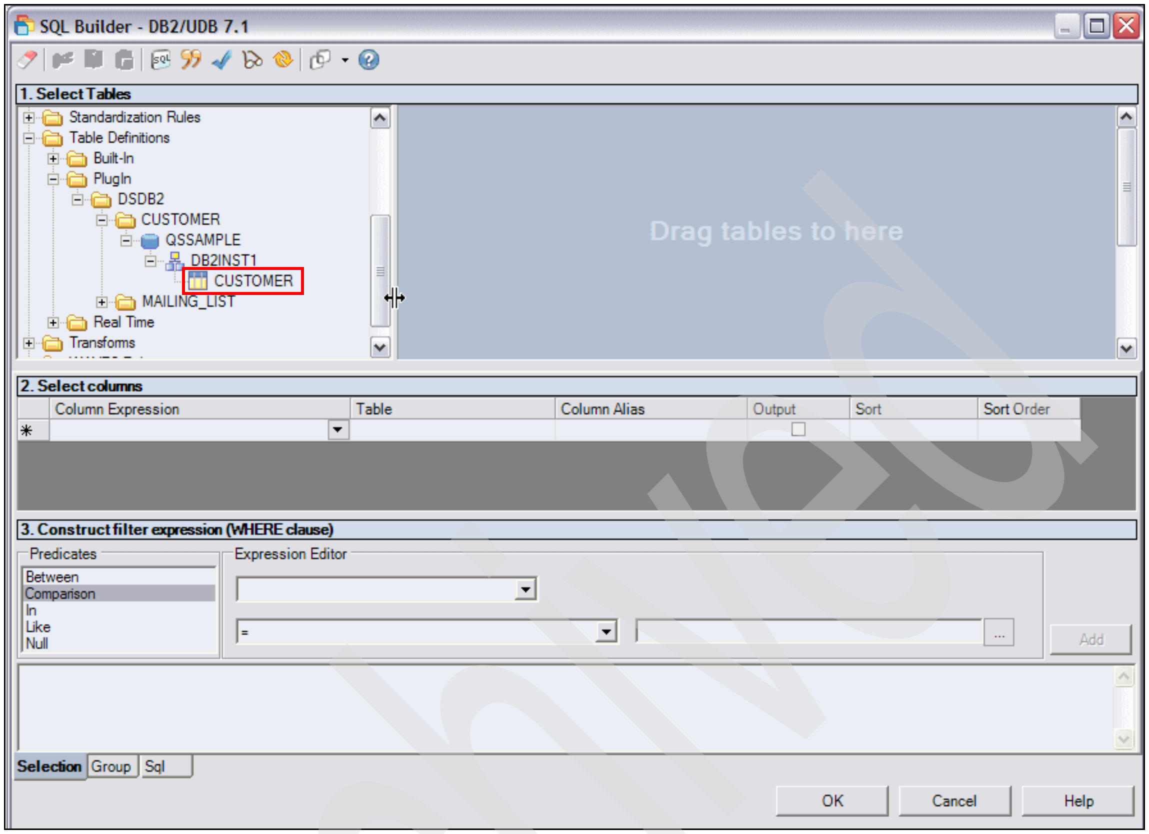
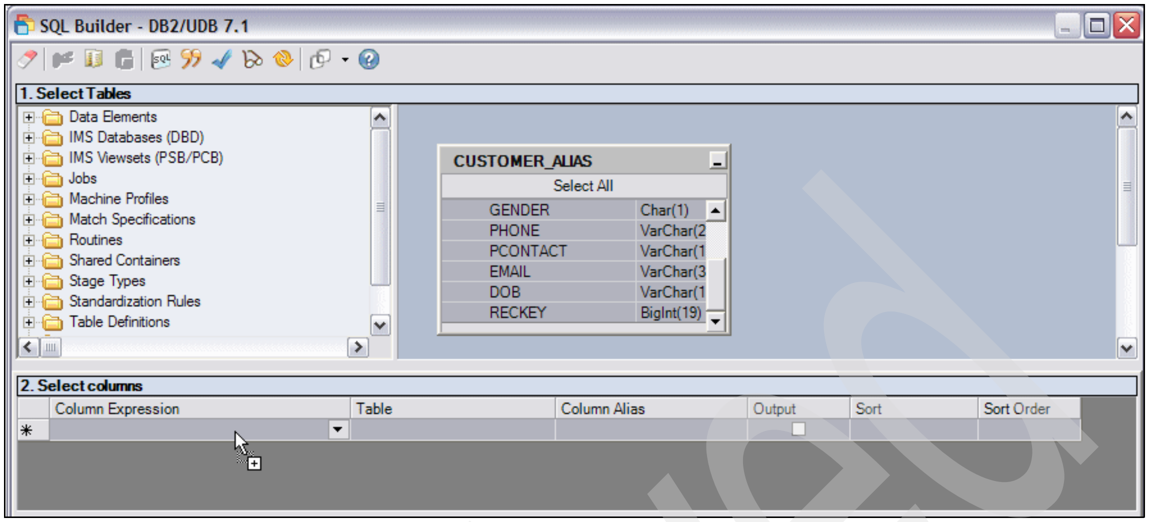
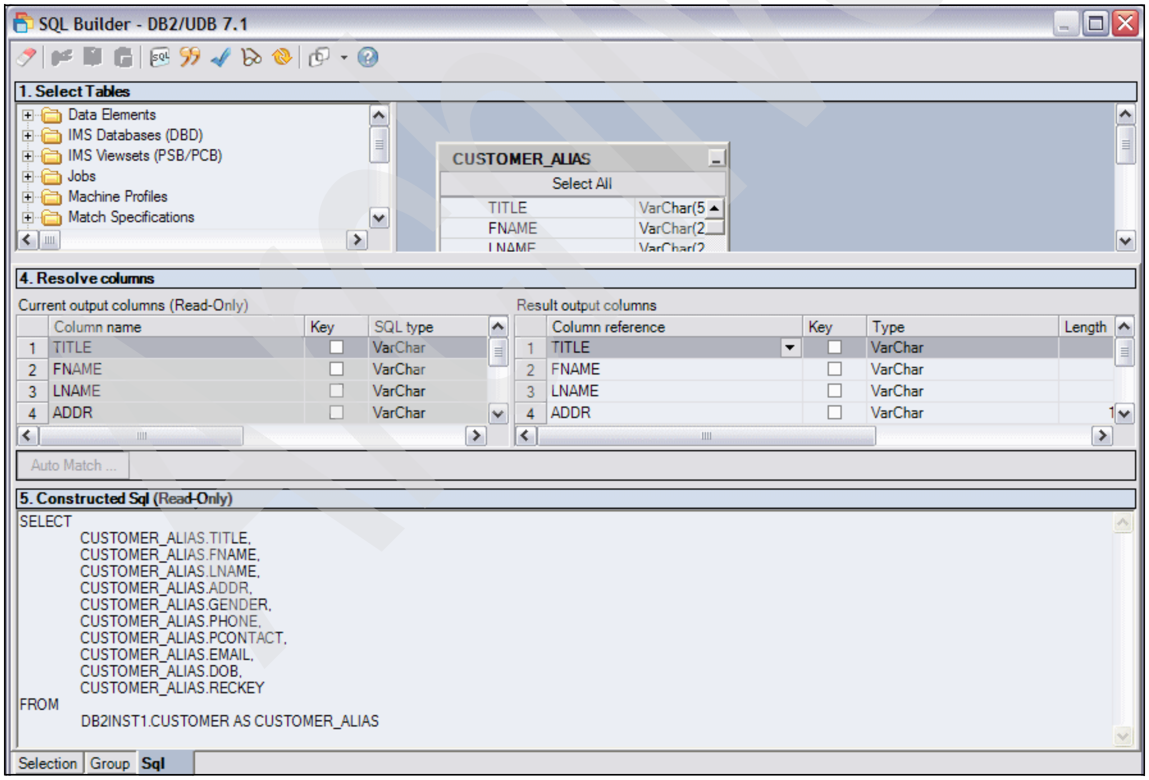


Figure 1-100 Create the J00_SRC_CUSTOMER job 12/39



GENDER	Char(1)
PHONE	VarChar(2
PCONTACT	VarChar(1
EMAIL	VarChar(3
DOB	VarChar(1
RECKEY	BigInt(19)

Figure 1-101   Create the J00_SRC_CUSTOMER job 13/39



Current output columns (Read-Only)

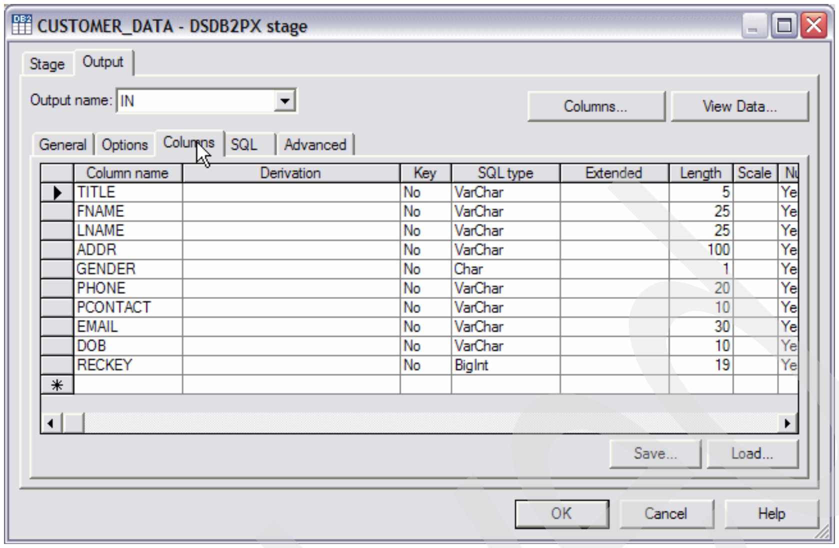
	Column name	Key	SQL type
1	TITLE	☐	VarChar
2	FNAME	☐	VarChar
3	LNAME	☐	VarChar
4	ADDR	☐	VarChar

Result output columns

	Column reference	Key	Type	Length
1	TITLE	☐	VarChar	
2	FNAME	☐	VarChar	
3	LNAME	☐	VarChar	
4	ADDR	☐	VarChar	1

```
SELECT
          CUSTOMER_ALIAS.TITLE,
          CUSTOMER_ALIAS.FNAME,
          CUSTOMER_ALIAS.LNAME,
          CUSTOMER_ALIAS.ADDR,
          CUSTOMER_ALIAS.GENDER,
          CUSTOMER_ALIAS.PHONE,
          CUSTOMER_ALIAS.PCONTACT,
          CUSTOMER_ALIAS.EMAIL,
          CUSTOMER_ALIAS.DOB,
          CUSTOMER_ALIAS.RECKEY
FROM
          DB2INST1.CUSTOMER AS CUSTOMER_ALIAS
```

Figure 1-102   Create the J00_SRC_CUSTOMER job 14/39

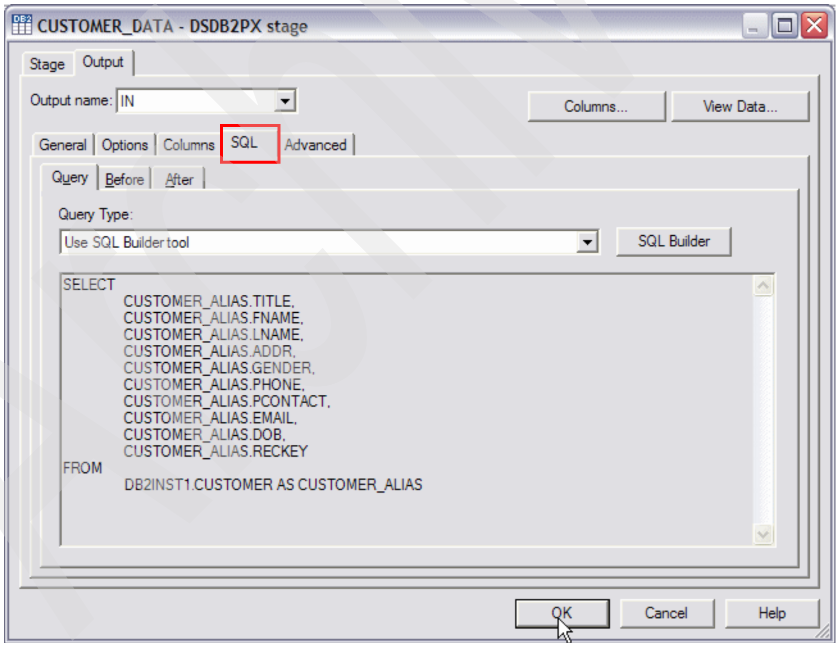


CUSTOMER_DATA - DSDB2PX stage

Output name: IN

	Column name	Derivation	Key	SQL type	Extended	Length	Scale	Nu
▶	TITLE		No	VarChar		5		Ye
	FNAME		No	VarChar		25		Ye
	LNAME		No	VarChar		25		Ye
	ADDR		No	VarChar		100		Ye
	GENDER		No	Char		1		Ye
	PHONE		No	VarChar		20		Ye
	PCONTACT		No	VarChar		10		Ye
	EMAIL		No	VarChar		30		Ye
	DOB		No	VarChar		10		Ye
	RECKEY		No	BigInt		19		Ye
✳								

Figure 1-103 Create the J00_SRC_CUSTOMER job 15/39



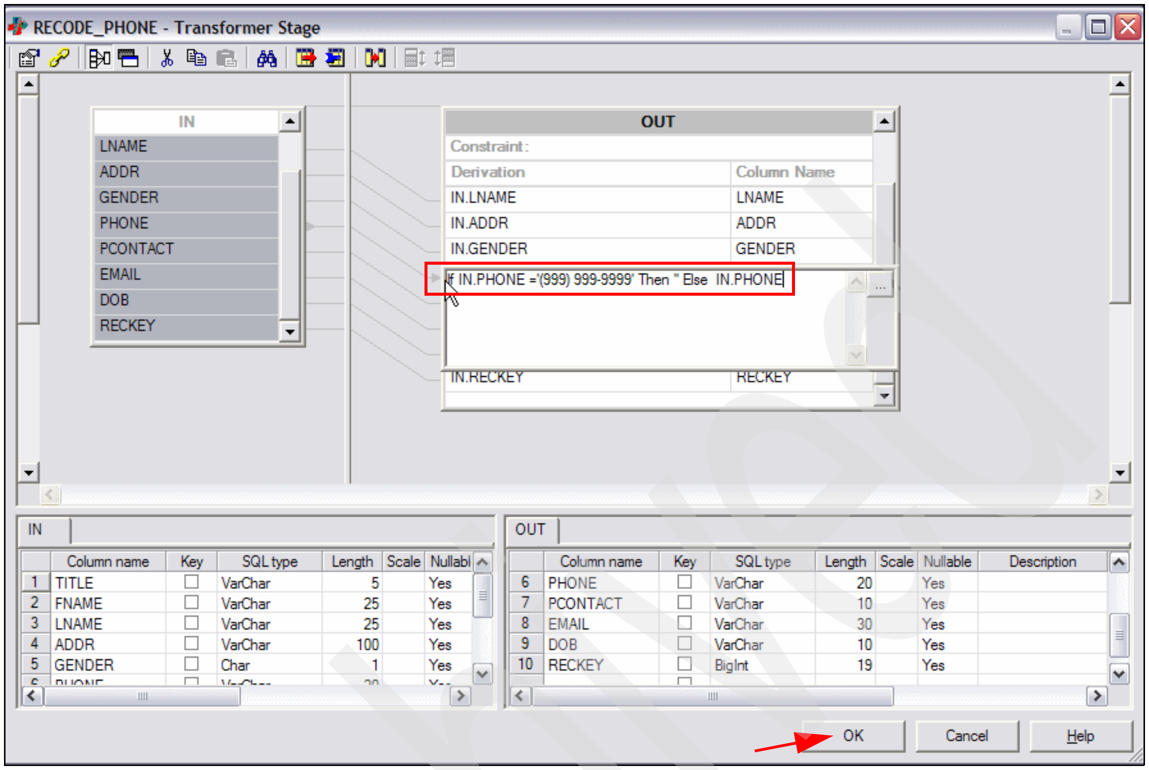
CUSTOMER_DATA - DSDB2PX stage

Output name: IN

Query Type: Use SQL Builder tool

```
SELECT
        CUSTOMER_ALIAS.TITLE,
        CUSTOMER_ALIAS.FNAME,
        CUSTOMER_ALIAS.LNAME,
        CUSTOMER_ALIAS.ADDR,
        CUSTOMER_ALIAS.GENDER,
        CUSTOMER_ALIAS.PHONE,
        CUSTOMER_ALIAS.PCONTACT,
        CUSTOMER_ALIAS.EMAIL,
        CUSTOMER_ALIAS.DOB,
        CUSTOMER_ALIAS.RECKEY
FROM
        DB2INST1.CUSTOMER AS CUSTOMER_ALIAS
```

Figure 1-104 Create the J00_SRC_CUSTOMER job 16/39



RECODE_PHONE - Transformer Stage

IN
LNAME
ADDR
GENDER
PHONE
PCONTACT
EMAIL
DOB
RECKEY

OUT

Constraint:

Derivation	Column Name
IN.LNAME	LNAME
IN.ADDR	ADDR
IN.GENDER	GENDER
If IN.PHONE ='(999) 999-9999' Then '' Else IN.PHONE	
IN.RECKEY	RECKEY

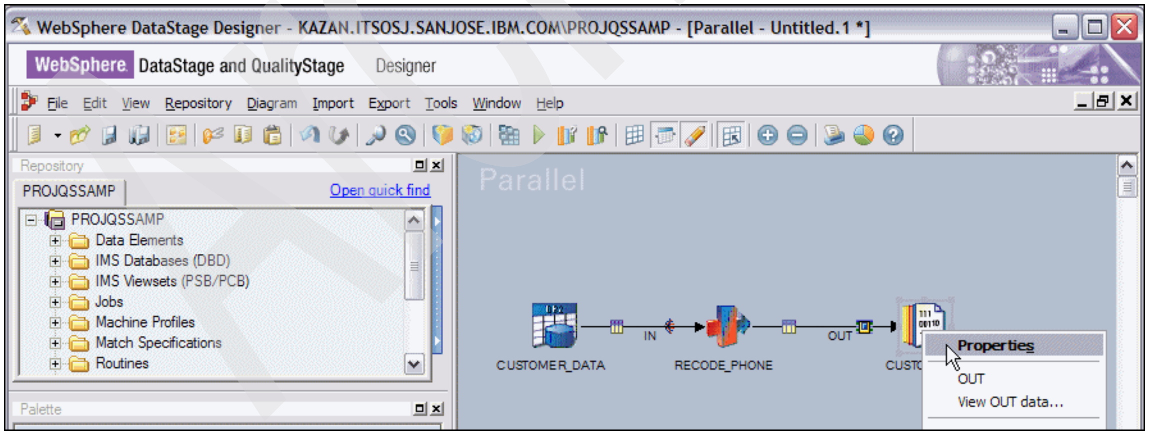
IN

	Column name	Key	SQL type	Length	Scale	Nullabl
1	TITLE	☐	VarChar	5		Yes
2	FNAME	☐	VarChar	25		Yes
3	LNAME	☐	VarChar	25		Yes
4	ADDR	☐	VarChar	100		Yes
5	GENDER	☐	Char	1		Yes

OUT

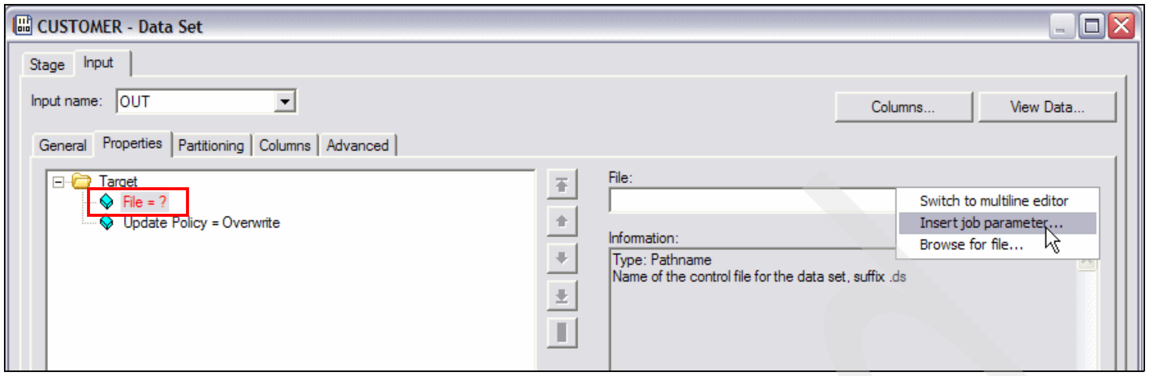
	Column name	Key	SQL type	Length	Scale	Nullable	Description
6	PHONE	☐	VarChar	20		Yes	
7	PCONTACT	☐	VarChar	10		Yes	
8	EMAIL	☐	VarChar	30		Yes	
9	DOB	☐	VarChar	10		Yes	
10	RECKEY	☐	BigInt	19		Yes	

Figure 1-105   Create the J00_SRC_CUSTOMER job 17/39



WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - Untitled.1 *]

Figure 1-106   Create the J00_SRC_CUSTOMER job 18/39


CUSTOMER - Data Set

Input name: OUT

Target
File = ?
Update Policy = Overwrite

File:

Information:
Type: Pathname
Name of the control file for the data set, suffix .ds

Figure 1-107   Create the J00_SRC_CUSTOMER job 19/39

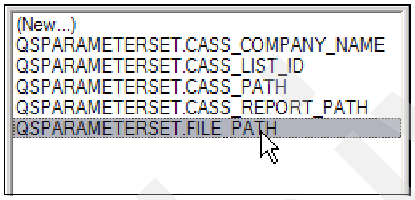
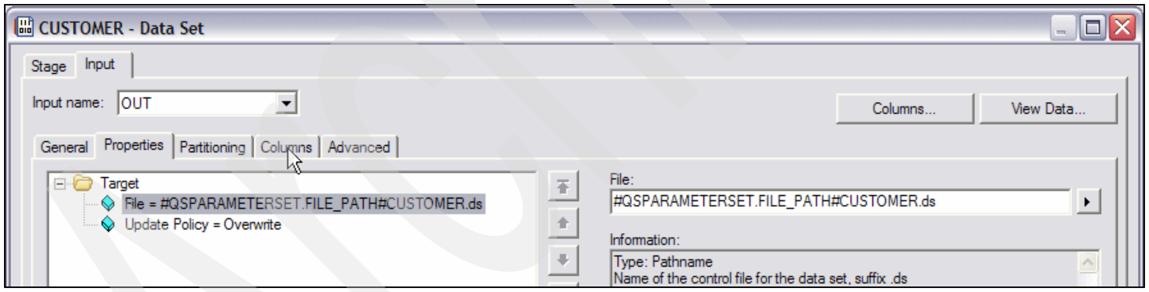

```
(New...)
QSPARAMETERSET.CASS_COMPANY_NAME
QSPARAMETERSET.CASS_LIST_ID
QSPARAMETERSET.CASS_PATH
QSPARAMETERSET.CASS_REPORT_PATH
QSPARAMETERSET.FILE_PATH
```

Figure 1-108   Create the J00_SRC_CUSTOMER job 20/39


CUSTOMER - Data Set

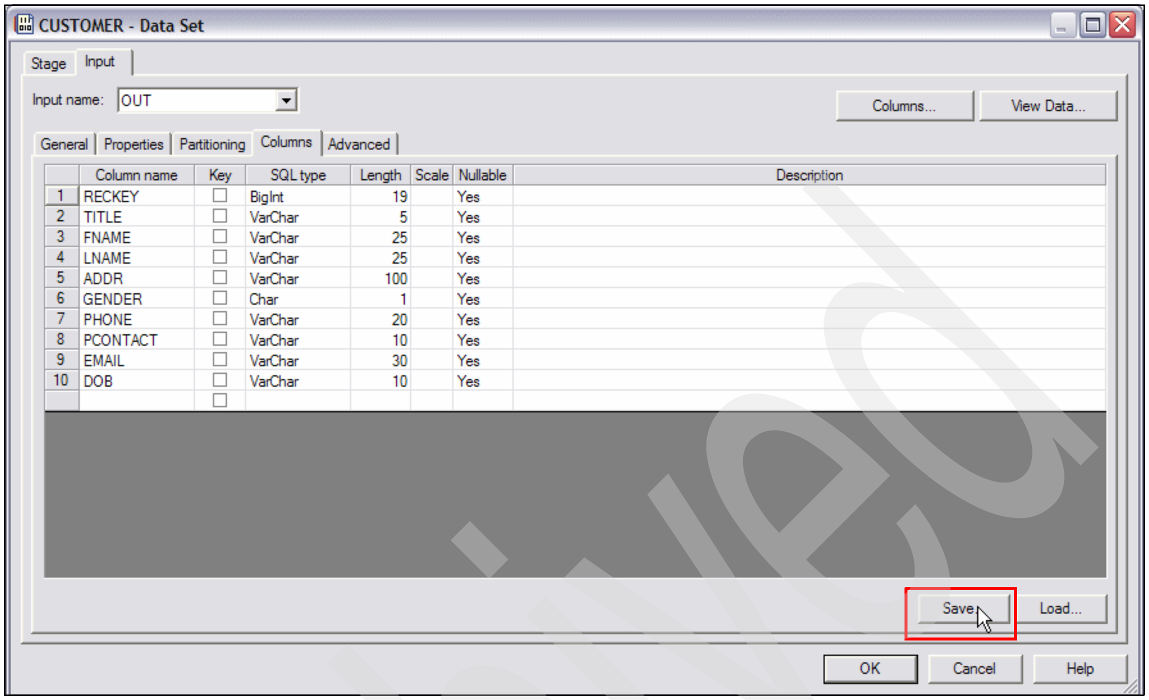
Input name: OUT

Target
File = #QSPARAMETERSET.FILE_PATH#CUSTOMER.ds
Update Policy = Overwrite

File: #QSPARAMETERSET.FILE_PATH#CUSTOMER.ds

Information:
Type: Pathname
Name of the control file for the data set, suffix .ds

Figure 1-109   Create the J00_SRC_CUSTOMER job 21/39


CUSTOMER - Data Set

Stage | Input

Input name: OUT

Columns... | View Data...

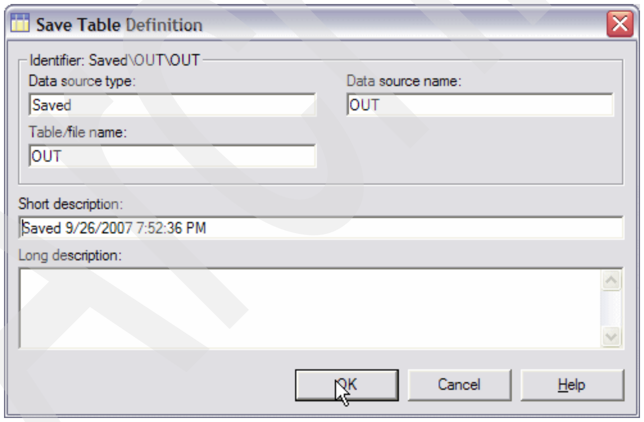
General | Properties | Partitioning | Columns | Advanced

	Column name	Key	SQL type	Length	Scale	Nullable	Description
1	RECKEY	☐	BigInt	19		Yes	
2	TITLE	☐	VarChar	5		Yes	
3	FNAME	☐	VarChar	25		Yes	
4	LNAME	☐	VarChar	25		Yes	
5	ADDR	☐	VarChar	100		Yes	
6	GENDER	☐	Char	1		Yes	
7	PHONE	☐	VarChar	20		Yes	
8	PCONTACT	☐	VarChar	10		Yes	
9	EMAIL	☐	VarChar	30		Yes	
10	DOB	☐	VarChar	10		Yes	
		☐					

Save... | Load...

OK | Cancel | Help

Figure 1-110 Create the J00_SRC_CUSTOMER job 22/39


Save Table Definition

Identifier: Saved\OUT\OUT

Data source type: Saved

Data source name: OUT

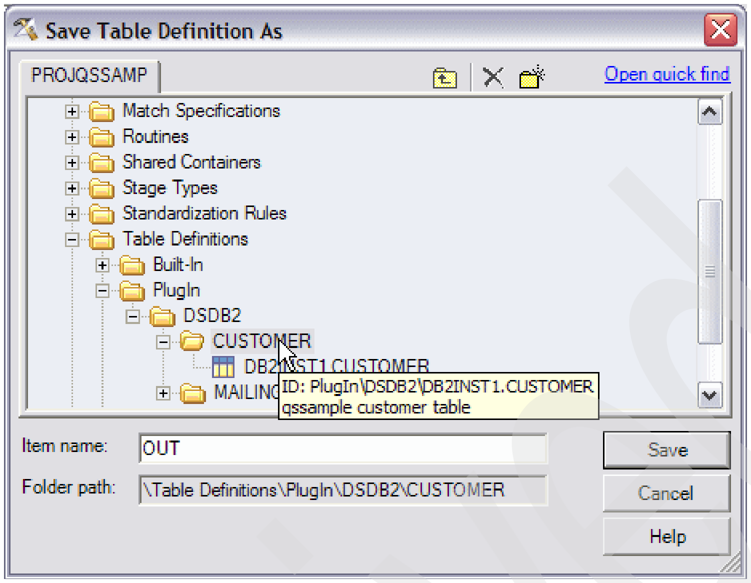
Table/file name: OUT

Short description: Saved 9/26/2007 7:52:36 PM

Long description:

OK | Cancel | Help

Figure 1-111 Create the J00_SRC_CUSTOMER job 23/39


Save Table Definition As

PROJQSSAMP

Open quick find

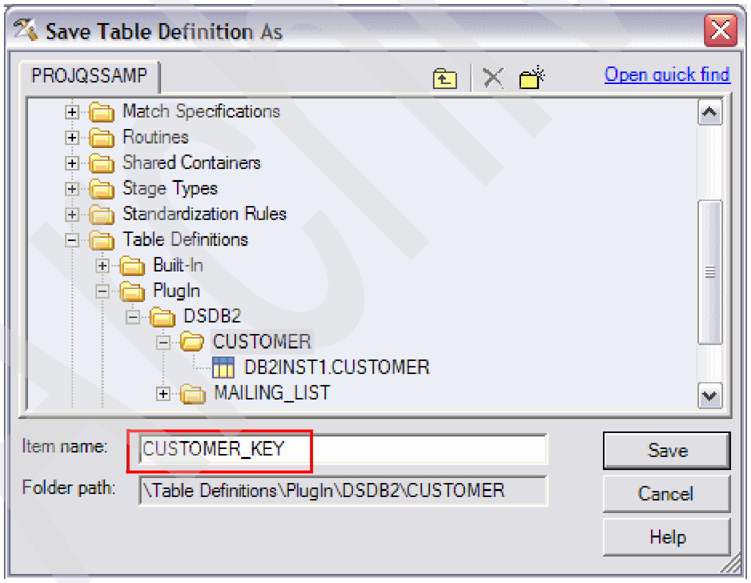
Match Specifications
Routines
Shared Containers
Stage Types
Standardization Rules
Table Definitions
Built-In
PlugIn
DSDB2
CUSTOMER
DB2INST1.CUSTOMER
MAILING

ID: PlugIn\DSDB2\DB2INST1.CUSTOMER
qssample customer table

Item name: OUT

Folder path: \Table Definitions\PlugIn\DSDB2\CUSTOMER

Figure 1-112 Create the J00_SRC_CUSTOMER job 24/39


Save Table Definition As

PROJQSSAMP

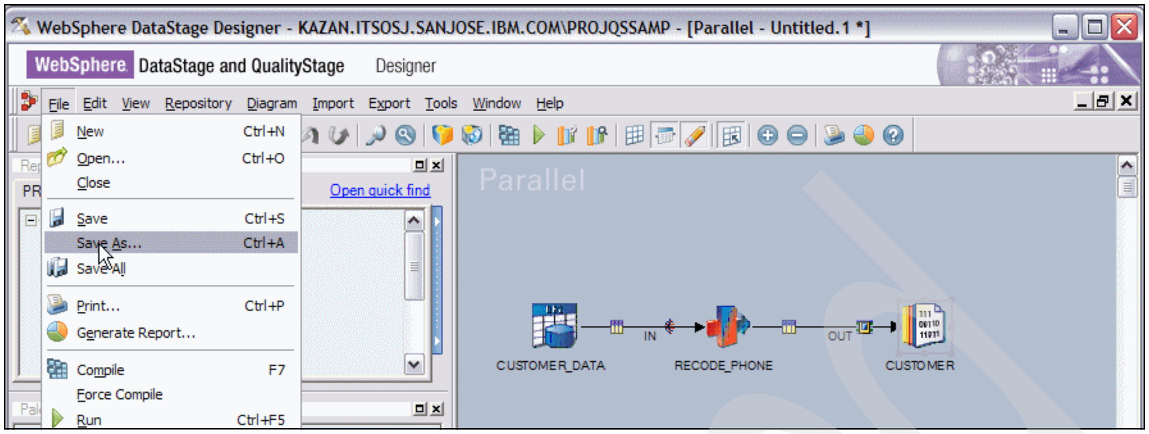
Open quick find

Match Specifications
Routines
Shared Containers
Stage Types
Standardization Rules
Table Definitions
Built-In
PlugIn
DSDB2
CUSTOMER
DB2INST1.CUSTOMER
MAILING_LIST

Item name: CUSTOMER_KEY

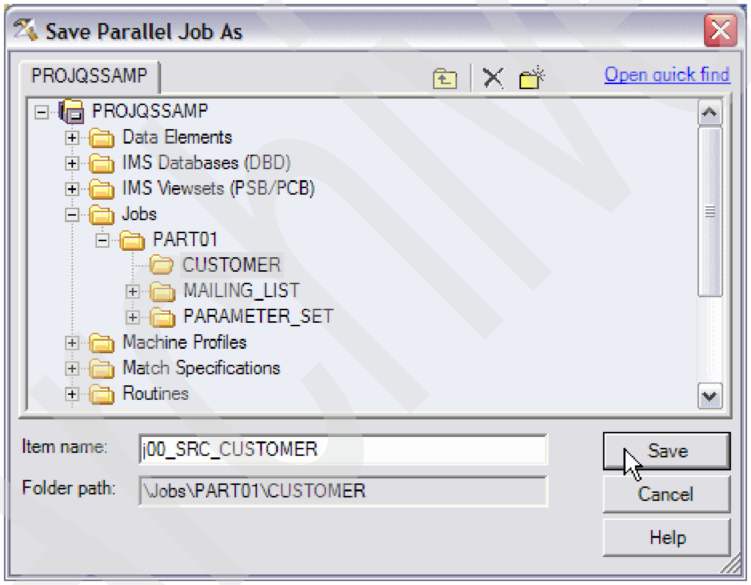
Folder path: \Table Definitions\PlugIn\DSDB2\CUSTOMER

Figure 1-113 Create the J00_SRC_CUSTOMER job 25/39



WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - Untitled.1 *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
New Ctrl+N
Open... Ctrl+O
Close
Save Ctrl+S
Save As... Ctrl+A
Save All
Print... Ctrl+P
Generate Report...
Compile F7
Force Compile
Run Ctrl+F5
Open quick find
Parallel
CUSTOMER_DATA
IN
RECODE_PHONE
OUT
CUSTOMER

Figure 1-114 Create the J00_SRC_CUSTOMER job 26/39



Item name: j00_SRC_CUSTOMER

Folder path: \Jobs\PART01\CUSTOMER

Figure 1-115 Create the J00_SRC_CUSTOMER job 27/39

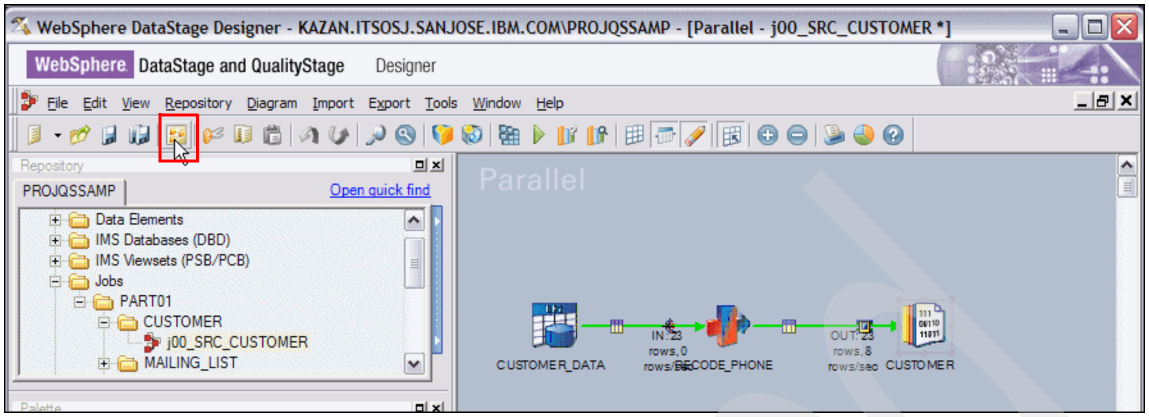

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j00_SRC_CUSTOMER *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Repository
PROJQSSAMP
Open quick find
Data Elements
IMS Databases (DBD)
IMS Viewsets (PSB/PCB)
Jobs
PART01
CUSTOMER
j00_SRC_CUSTOMER
MAILING_LIST
Palette
Parallel
CUSTOMER_DATA
RECODE_PHONE
CUSTOMER

Figure 1-116   Create the J00_SRC_CUSTOMER job 28/39

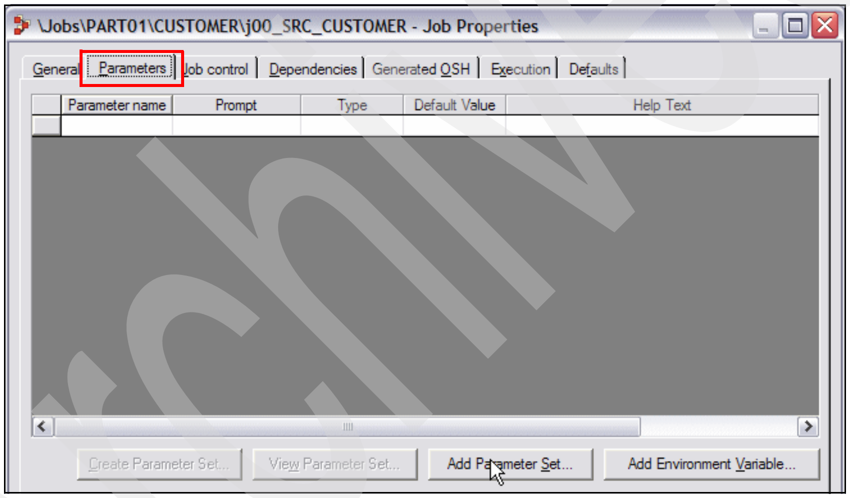
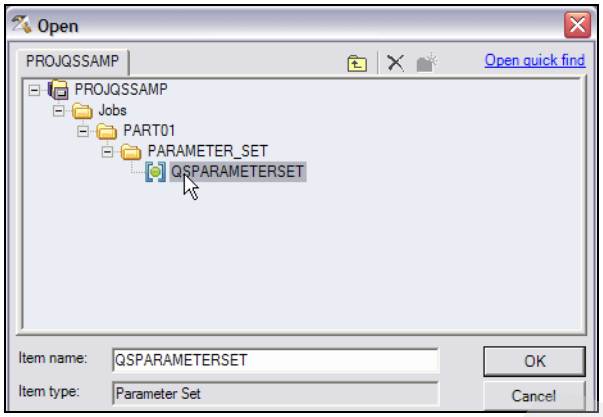


Figure 1-117   Create the J00_SRC_CUSTOMER job 29/39



Item name: QSPARAMETERSET

Item type: Parameter Set

Figure 1-118   Create the J00_SRC_CUSTOMER job 30/39

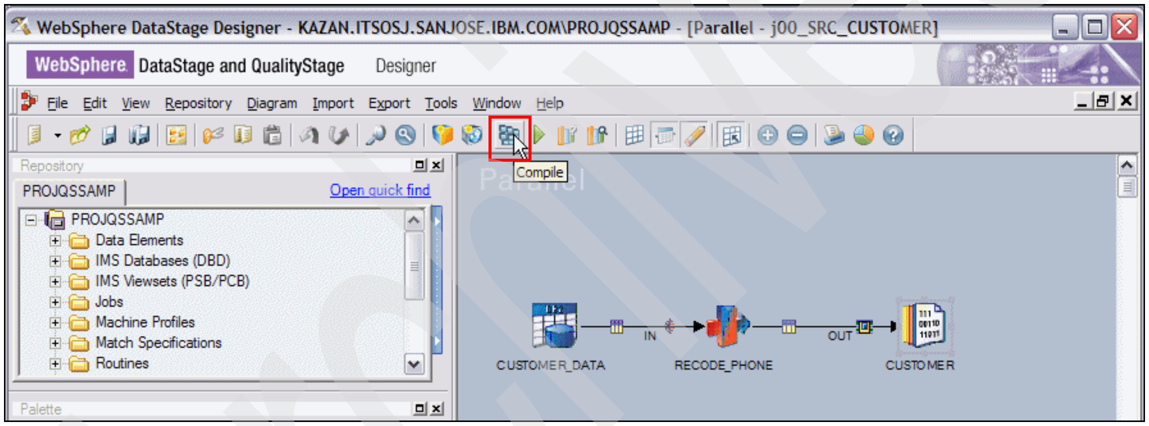
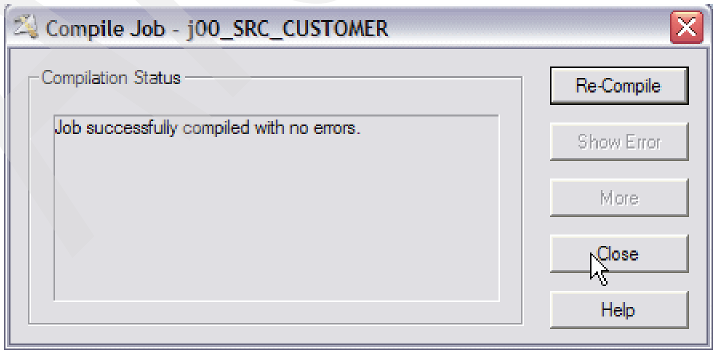

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j00_SRC_CUSTOMER]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Compile
Repository
PROJQSSAMP
Open quick find
PROJQSSAMP
Data Elements
IMS Databases (DBD)
IMS Viewsets (PSB/PCB)
Jobs
Machine Profiles
Match Specifications
Routines
Palette
Parallel
CUSTOMER_DATA
IN
RECODE_PHONE
OUT
CUSTOMER

Figure 1-119   Create the J00_SRC_CUSTOMER job 31/39



Job successfully compiled with no errors.

Figure 1-120   Create the J00_SRC_CUSTOMER job 32/39

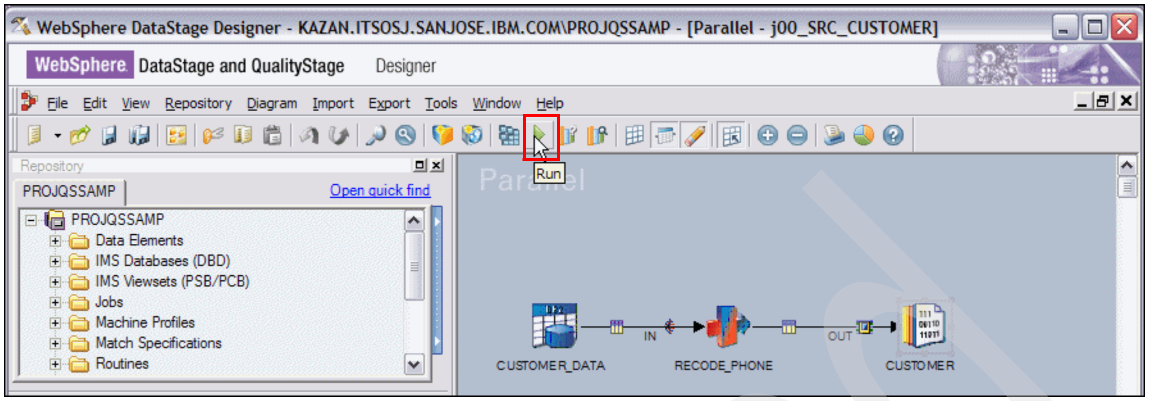
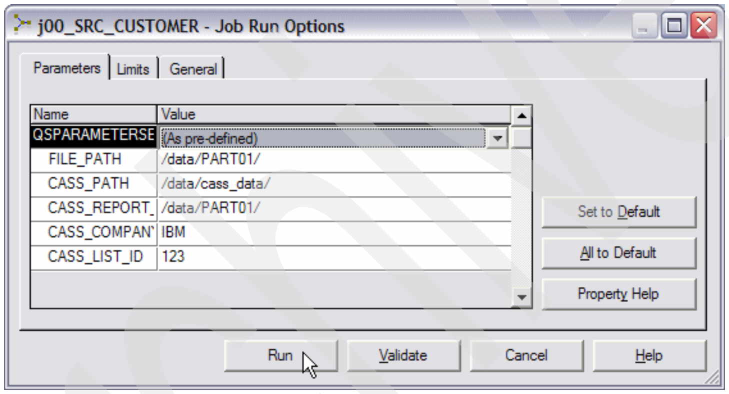

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j00_SRC_CUSTOMER]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Run
Repository
PROJQSSAMP
Open quick find
PROJQSSAMP
Data Elements
IMS Databases (DBD)
IMS Viewsets (PSB/PCB)
Jobs
Machine Profiles
Match Specifications
Routines
Parallel
CUSTOMER_DATA
IN
RECODE_PHONE
OUT
CUSTOMER

Figure 1-121 Create the J00_SRC_CUSTOMER job 33/39



Name	Value
QSPARAMETERSE	(As pre-defined)
FILE_PATH	/data/PART01/
CASS_PATH	/data/cass_data/
CASS_REPORT_	/data/PART01/
CASS_COMPAN'	IBM
CASS_LIST_ID	123

Figure 1-122 Create the J00_SRC_CUSTOMER job 34/39

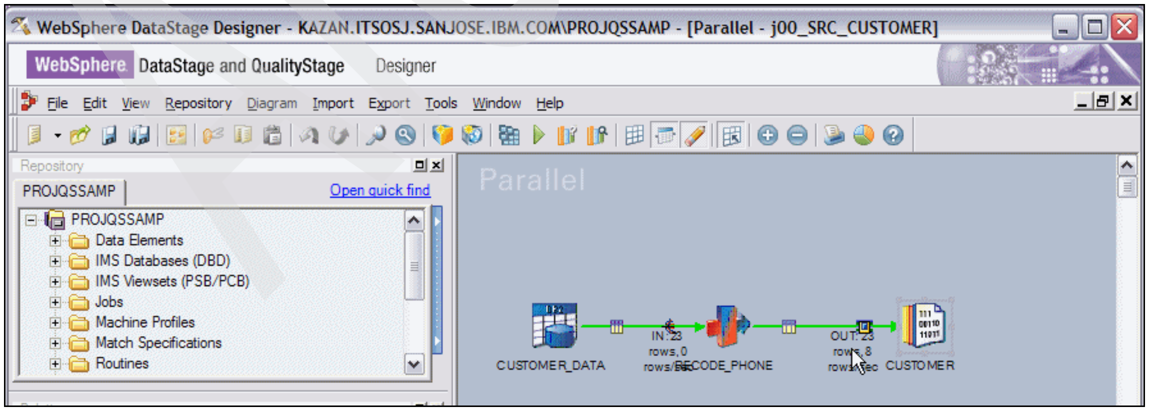


Figure 1-123 Create the J00_SRC_CUSTOMER job 35/39

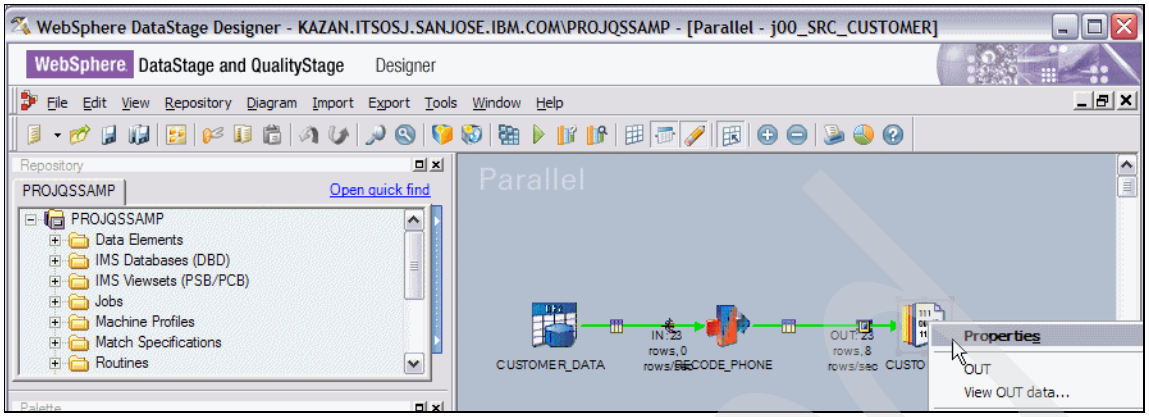
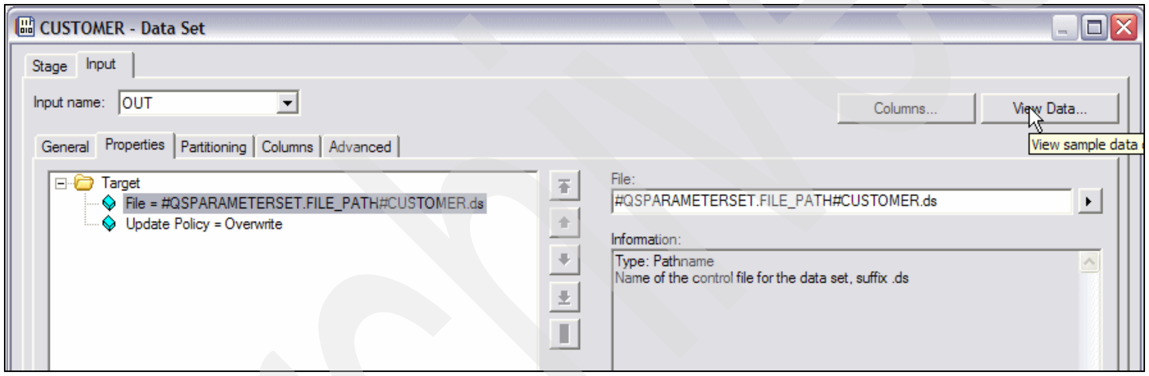


Figure 1-124 Create the J00_SRC_CUSTOMER job 36/39



Input name: OUT

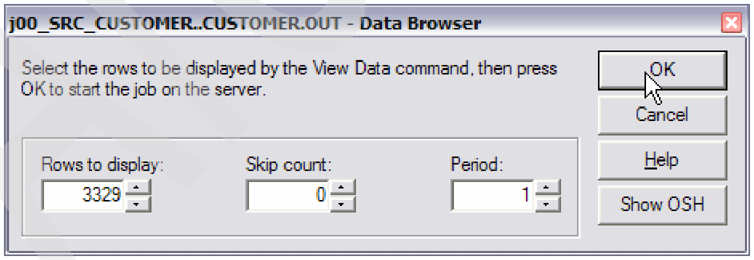
File = #QSPARAMETERSET.FILE_PATH#CUSTOMER.ds

Update Policy = Overwrite

File: #QSPARAMETERSET.FILE_PATH#CUSTOMER.ds

Information:
Type: Pathname
Name of the control file for the data set, suffix .ds

Figure 1-125 Create the J00_SRC_CUSTOMER job 37/39



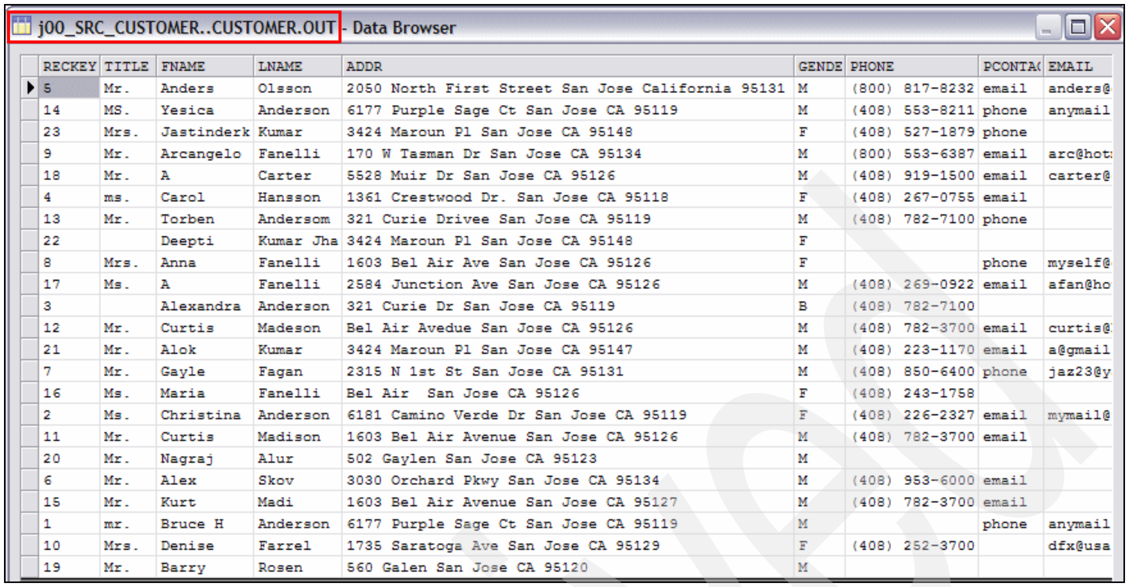
Select the rows to be displayed by the View Data command, then press OK to start the job on the server.

Rows to display: 3329

Skip count: 0

Period: 1

Figure 1-126 Create the J00_SRC_CUSTOMER job 38/39


j00_SRC_CUSTOMER..CUSTOMER.OUT - Data Browser

RECKEY	TITLE	FNAME	LNAME	ADDR	GENDE	PHONE	PCONTA(	EMAIL
5	Mr.	Anders	Olsson	2050 North First Street San Jose California 95131	M	(800) 817-8232	email	anders@
14	MS.	Yesica	Anderson	6177 Purple Sage Ct San Jose CA 95119	M	(408) 553-8211	phone	anymail
23	Mrs.	Jastinderk	Kumar	3424 Maroun Pl San Jose CA 95148	F	(408) 527-1879	phone	
9	Mr.	Arcangelo	Fanelli	170 W Tasman Dr San Jose CA 95134	M	(800) 553-6387	email	arc@hot:
18	Mr.	A	Carter	5528 Muir Dr San Jose CA 95126	M	(408) 919-1500	email	carter@
4	ms.	Carol	Hansson	1361 Crestwood Dr. San Jose CA 95118	F	(408) 267-0755	email	
13	Mr.	Torben	Andersom	321 Curie Drivee San Jose CA 95119	M	(408) 782-7100	phone	
22		Deepti	Kumar Jha	3424 Maroun Pl San Jose CA 95148	F			
8	Mrs.	Anna	Fanelli	1603 Bel Air Ave San Jose CA 95126	F		phone	myself@
17	Ms.	A	Fanelli	2584 Junction Ave San Jose CA 95126	M	(408) 269-0922	email	afan@ho
3		Alexandra	Anderson	321 Curie Dr San Jose CA 95119	B	(408) 782-7100		
12	Mr.	Curtis	Madeson	Bel Air Avedue San Jose CA 95126	M	(408) 782-3700	email	curtis@
21	Mr.	Alok	Kumar	3424 Maroun Pl San Jose CA 95147	M	(408) 223-1170	email	a@gmail
7	Mr.	Gayle	Fagan	2315 N 1st St San Jose CA 95131	M	(408) 850-6400	phone	jaz23@y
16	Ms.	Maria	Fanelli	Bel Air San Jose CA 95126	F	(408) 243-1758		
2	Ms.	Christina	Anderson	6181 Camino Verde Dr San Jose CA 95119	F	(408) 226-2327	email	mymail@
11	Mr.	Curtis	Madison	1603 Bel Air Avenue San Jose CA 95126	M	(408) 782-3700	email	
20	Mr.	Nagraj	Alur	502 Gaylen San Jose CA 95123	M			
6	Mr.	Alex	Skov	3030 Orchard Pkwy San Jose CA 95134	M	(408) 953-6000	email	
15	Mr.	Kurt	Madi	1603 Bel Air Avenue San Jose CA 95127	M	(408) 782-3700	email	
1	mr.	Bruce H	Anderson	6177 Purple Sage Ct San Jose CA 95119	M		phone	anymail
10	Mrs.	Denise	Farrel	1735 Saratoga Ave San Jose CA 95129	F	(408) 252-3700		dfx@usa
19	Mr.	Barry	Rosen	560 Galen San Jose CA 95120	M			

Figure 1-127 Create the J00_SRC_CUSTOMER job 39/39

J00A_INV_CUSTOMER

In this step, we use the Investigate stage on non-text columns such as telephone number, e-mail, and preferred method of contact. We used character concatenate and character discrete with combinations of *C*, *T*, and *X* masks. The objective of this step is to validate some of the main non-text columns. Any errors detected are resolved by modifying the source directly or by using IBM WebSphere QualityStage jobs to cleanse and modify the targets.

Figure 1-128 on page 164 shows the various stages that are used in this job, including the data set that was created in “J00_SRC_CUSTOMER” on page 142, one COPY stage, three Investigate stages, and one Sequential File stage for each Investigate stage. We modified the names of the stages as shown, and this job used the same QSPARAMETERSET object that was created earlier.

Because the configuration of this job is very similar to that described in “J02_INVCC_ISCODE” on page 186, we do not repeat it here. However, some of the configurations of interest are as follows:

- Figure 1-129 on page 164 shows the contents of the input file to the COPY stage and there on to the Investigate stages.
- Figure 1-130 on page 165 shows the character discrete investigate columns EMAIL, PHONE, and PCONTACT configured with the *C* mask.

Figure 1-134 on page 167 shows the report that was generated for this Investigate stage after compiling and running (Figure 1-133 on page 166) this job.

- Figure 1-131 on page 165 shows the character discrete investigate columns EMAIL, PHONE, and PCONTACT configured with the *T* mask.

  Figure 1-135 on page 168 shows the report that was generated for this Investigate stage after compiling and running this job.

- Figure 1-132 on page 166 shows the character concatenate investigate columns EMAIL, PHONE, and PCONTACT configured with a combination of *C*, *T*, and *X* masks. This Investigate stage reveals whether the business rule that states that the preferred method of contact (PCONTACT) needs a valid value (for example, if the preferred method of contact is through telephone, then a telephone number must be present).

  Figure 1-136 on page 168 shows the report that was generated for this Investigate stage (sample size of 1) after compiling and running this job. The highlighted boxes show the rows that violate the business rule of having a value in the column that corresponds to the preferred method of contact.

Note: You need to review these reports for validity and then correct them using data cleansing techniques. Time constraints during our testing prevented us from creating data cleansing jobs to do so.

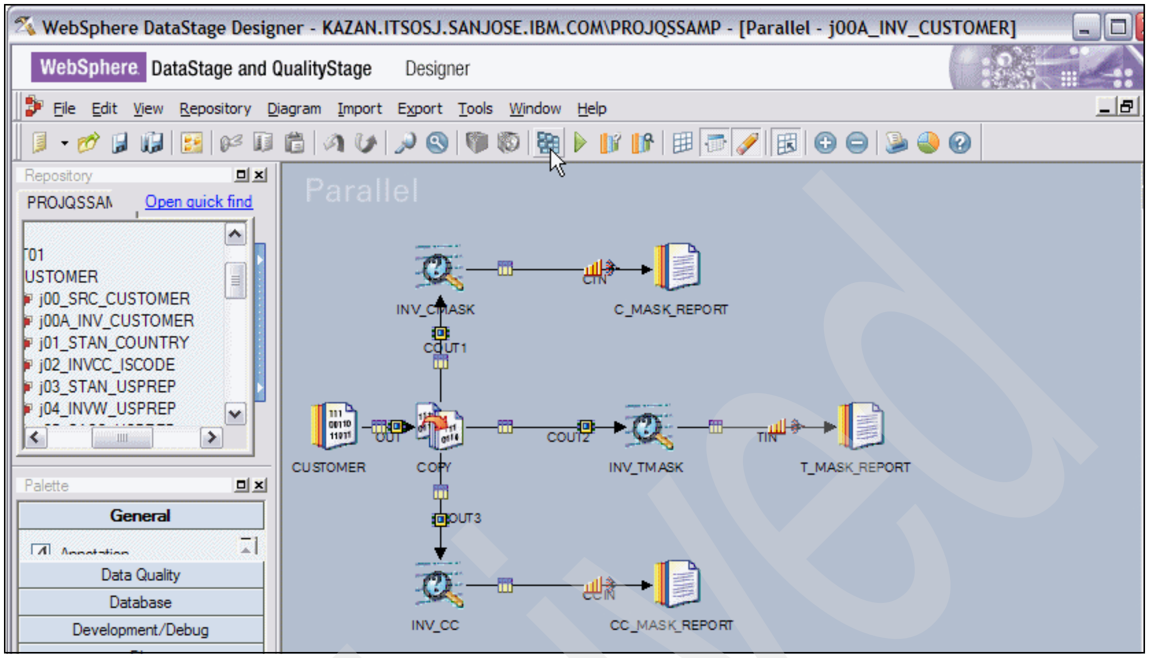
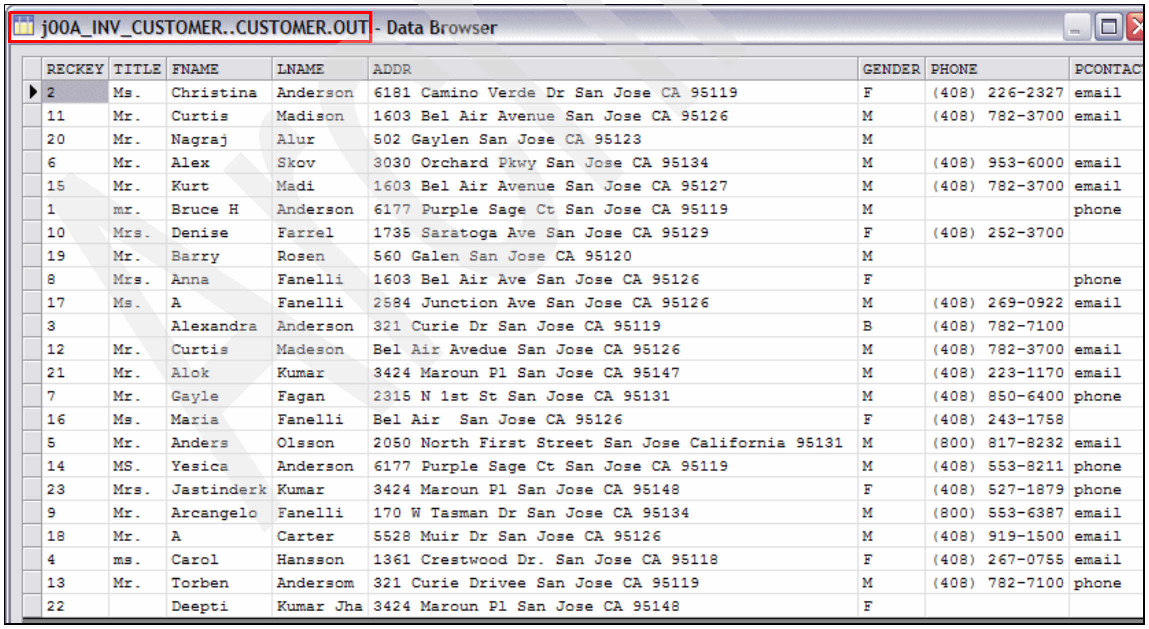
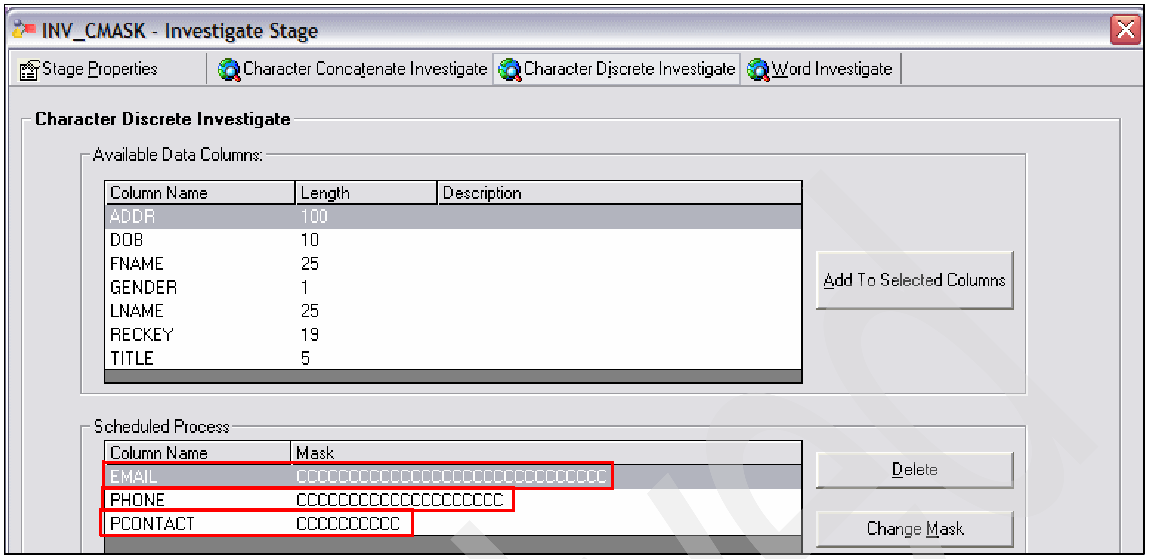


Figure 1-128 Create J00A_INV_CUSTOMER job 1/9


j00A_INV_CUSTOMER..CUSTOMER.OUT - Data Browser

RECKEY	TITLE	FNAME	LNAME	ADDR	GENDER	PHONE	PCONTAC
2	Ms.	Christina	Anderson	6181 Camino Verde Dr San Jose CA 95119	F	(408) 226-2327	email
11	Mr.	Curtis	Madison	1603 Bel Air Avenue San Jose CA 95126	M	(408) 782-3700	email
20	Mr.	Nagraj	Alur	502 Gaylen San Jose CA 95123	M		
6	Mr.	Alex	Skov	3030 Orchard Pkwy San Jose CA 95134	M	(408) 953-6000	email
15	Mr.	Kurt	Madi	1603 Bel Air Avenue San Jose CA 95127	M	(408) 782-3700	email
1	mr.	Bruce H	Anderson	6177 Purple Sage Ct San Jose CA 95119	M		phone
10	Mrs.	Denise	Farrel	1735 Saratoga Ave San Jose CA 95129	F	(408) 252-3700	
19	Mr.	Barry	Rosen	560 Galen San Jose CA 95120	M		
8	Mrs.	Anna	Fanelli	1603 Bel Air Ave San Jose CA 95126	F		phone
17	Ms.	A	Fanelli	2584 Junction Ave San Jose CA 95126	M	(408) 269-0922	email
3		Alexandra	Anderson	321 Curie Dr San Jose CA 95119	B	(408) 782-7100	
12	Mr.	Curtis	Madeson	Bel Air Avedue San Jose CA 95126	M	(408) 782-3700	email
21	Mr.	Alok	Kumar	3424 Maroun Pl San Jose CA 95147	M	(408) 223-1170	email
7	Mr.	Gayle	Fagan	2315 N 1st St San Jose CA 95131	M	(408) 850-6400	phone
16	Ms.	Maria	Fanelli	Bel Air San Jose CA 95126	F	(408) 243-1758	
5	Mr.	Anders	Olsson	2050 North First Street San Jose California 95131	M	(800) 817-8232	email
14	MS.	Yesica	Anderson	6177 Purple Sage Ct San Jose CA 95119	M	(408) 553-8211	phone
23	Mrs.	Jastinderk	Kumar	3424 Maroun Pl San Jose CA 95148	F	(408) 527-1879	phone
9	Mr.	Arcangelo	Fanelli	170 W Tasman Dr San Jose CA 95134	M	(800) 553-6387	email
18	Mr.	A	Carter	5528 Muir Dr San Jose CA 95126	M	(408) 919-1500	email
4	ms.	Carol	Hansson	1361 Crestwood Dr. San Jose CA 95118	F	(408) 267-0755	email
13	Mr.	Torben	Andersom	321 Curie Drivee San Jose CA 95119	M	(408) 782-7100	phone
22		Deepti	Kumar Jha	3424 Maroun Pl San Jose CA 95148	F		

Figure 1-129 Create J00A_INV_CUSTOMER job 2/9



INV_CMASK - Investigate Stage

Stage Properties | Character Concatenate Investigate | Character Discrete Investigate | Word Investigate

Character Discrete Investigate

Available Data Columns:

Column Name	Length	Description
ADDR	100	
DOB	10	
FNAME	25	
GENDER	1	
LNAME	25	
RECKEY	19	
TITLE	5	

Add To Selected Columns

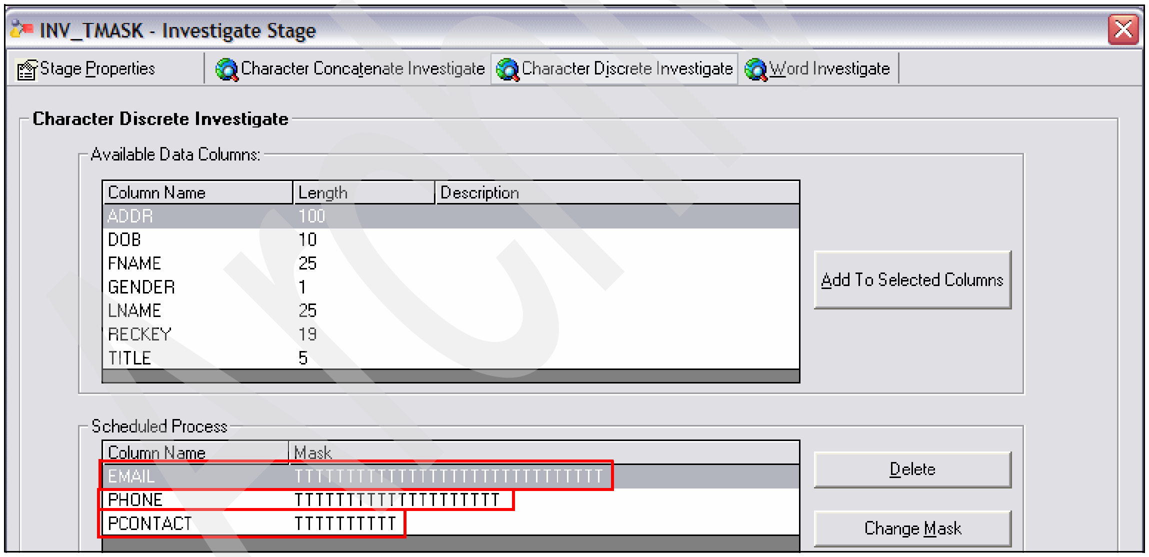
Scheduled Process

Column Name	Mask
EMAIL	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
PHONE	CCCCCCCCCCCCCCCCCCCCC
PCONTACT	CCCCCCCCCC

Delete

Change Mask

Figure 1-130 Create J00A_INV_CUSTOMER job 3/9



INV_TMASK - Investigate Stage

Stage Properties | Character Concatenate Investigate | Character Discrete Investigate | Word Investigate

Character Discrete Investigate

Available Data Columns:

Column Name	Length	Description
ADDR	100	
DOB	10	
FNAME	25	
GENDER	1	
LNAME	25	
RECKEY	19	
TITLE	5	

Add To Selected Columns

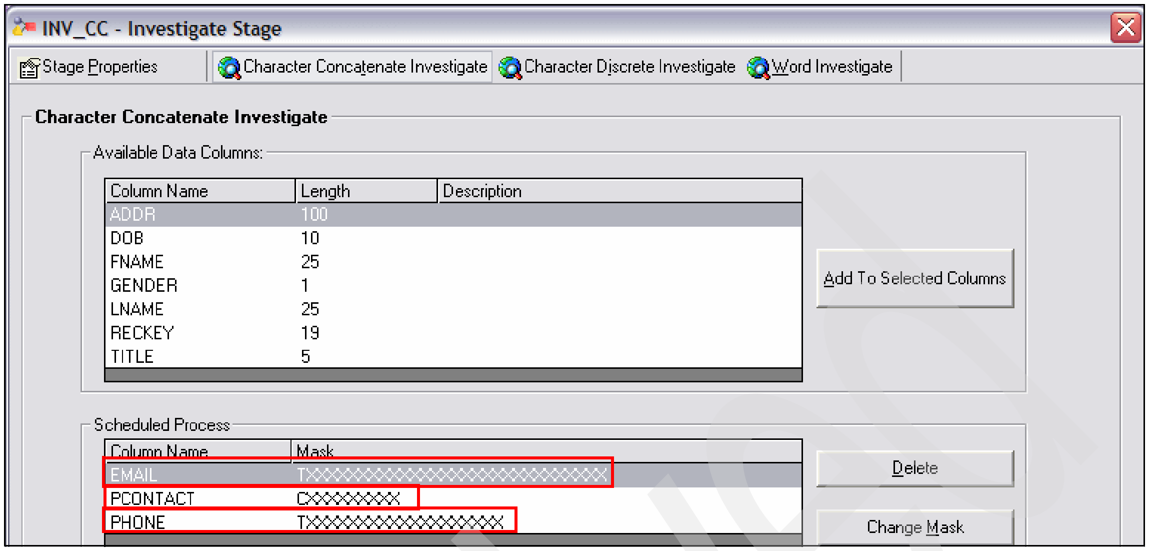
Scheduled Process

Column Name	Mask
EMAIL	TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
PHONE	TTTTTTTTTTTTTTTTTTTTT
PCONTACT	TTTTTTTTTT

Delete

Change Mask

Figure 1-131 Create J00A_INV_CUSTOMER job 4/9


INV_CC - Investigate Stage

Stage Properties
Character Concatenate Investigate
Character Discrete Investigate
Word Investigate

Character Concatenate Investigate

Available Data Columns:

Column Name	Length	Description
ADDR	100	
DOB	10	
FNAME	25	
GENDER	1	
LNAME	25	
RECKEY	19	
TITLE	5	

Scheduled Process

Column Name	Mask
EMAIL	TXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
PCONTACT	CXXXXXXXXX
PHONE	TXXXXXXXXXXXXXXXXXXXX

Figure 1-132 Create J00A_INV_CUSTOMER job 5/9

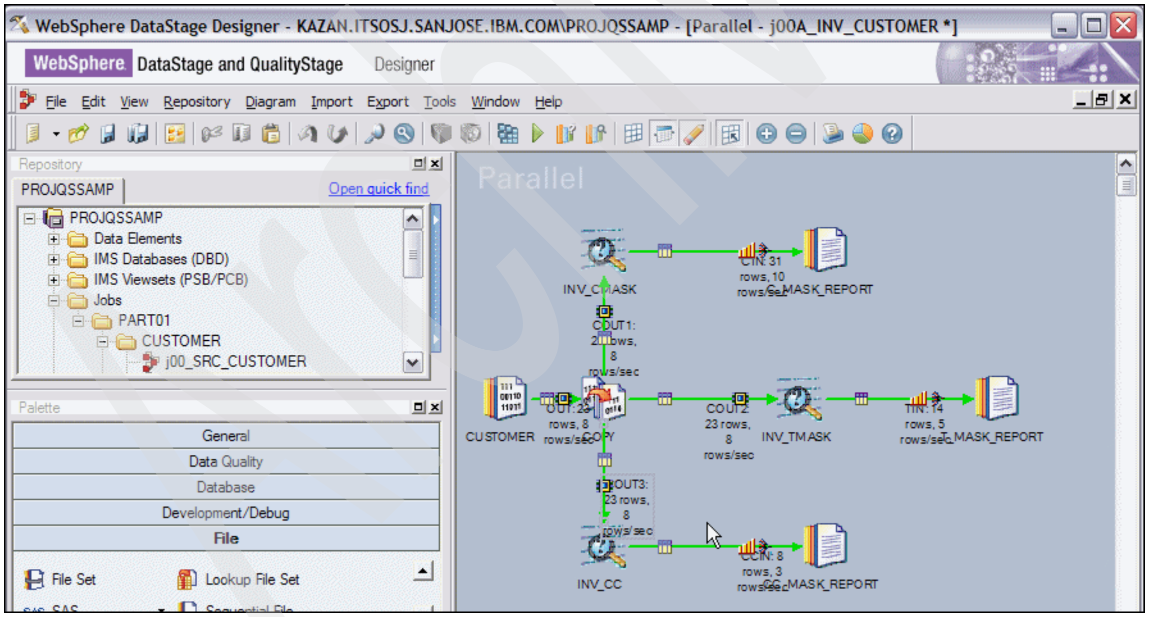
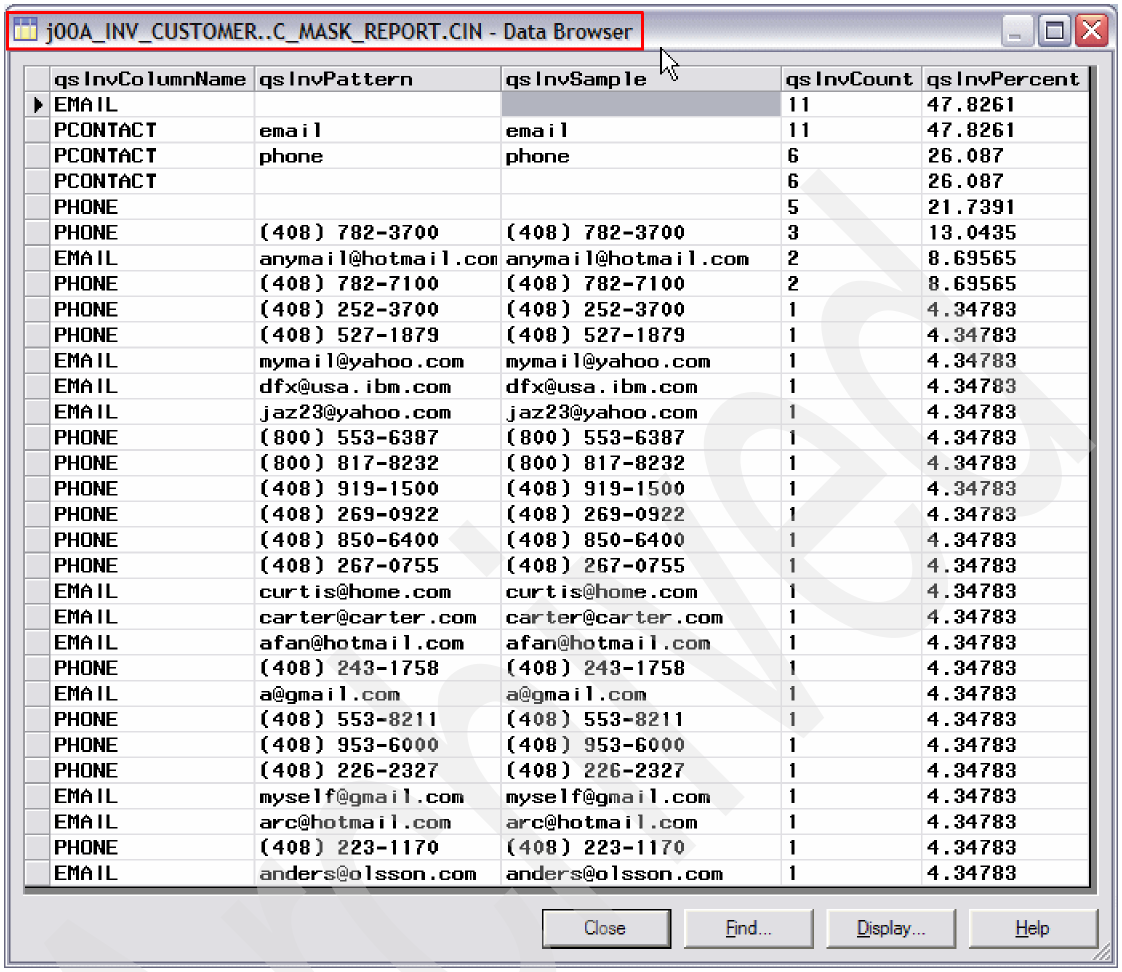
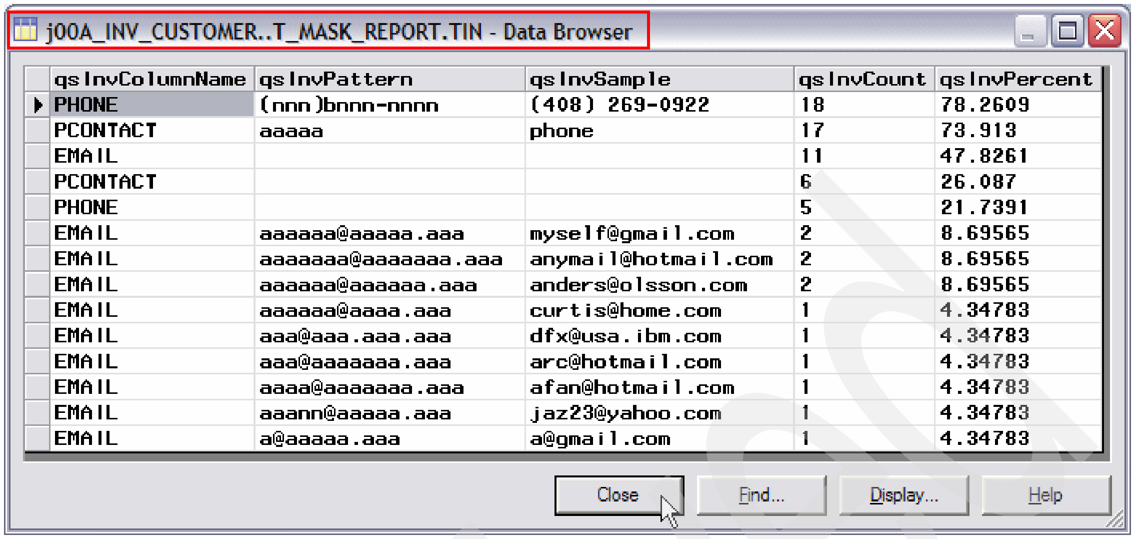


Figure 1-133 Create J00A_INV_CUSTOMER job 6/9



qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
EMAIL			11	47.8261
PCONTACT	email	email	11	47.8261
PCONTACT	phone	phone	6	26.087
PCONTACT			6	26.087
PHONE			5	21.7391
PHONE	(408) 782-3700	(408) 782-3700	3	13.0435
EMAIL	anymail@hotmail.com	anymail@hotmail.com	2	8.69565
PHONE	(408) 782-7100	(408) 782-7100	2	8.69565
PHONE	(408) 252-3700	(408) 252-3700	1	4.34783
PHONE	(408) 527-1879	(408) 527-1879	1	4.34783
EMAIL	mymail@yahoo.com	mymail@yahoo.com	1	4.34783
EMAIL	dfx@usa.ibm.com	dfx@usa.ibm.com	1	4.34783
EMAIL	jaz23@yahoo.com	jaz23@yahoo.com	1	4.34783
PHONE	(800) 553-6387	(800) 553-6387	1	4.34783
PHONE	(800) 817-8232	(800) 817-8232	1	4.34783
PHONE	(408) 919-1500	(408) 919-1500	1	4.34783
PHONE	(408) 269-0922	(408) 269-0922	1	4.34783
PHONE	(408) 850-6400	(408) 850-6400	1	4.34783
PHONE	(408) 267-0755	(408) 267-0755	1	4.34783
EMAIL	curtis@home.com	curtis@home.com	1	4.34783
EMAIL	carter@carter.com	carter@carter.com	1	4.34783
EMAIL	afan@hotmail.com	afan@hotmail.com	1	4.34783
PHONE	(408) 243-1758	(408) 243-1758	1	4.34783
EMAIL	a@gmail.com	a@gmail.com	1	4.34783
PHONE	(408) 553-8211	(408) 553-8211	1	4.34783
PHONE	(408) 953-6000	(408) 953-6000	1	4.34783
PHONE	(408) 226-2327	(408) 226-2327	1	4.34783
EMAIL	myself@gmail.com	myself@gmail.com	1	4.34783
EMAIL	arc@hotmail.com	arc@hotmail.com	1	4.34783
PHONE	(408) 223-1170	(408) 223-1170	1	4.34783
EMAIL	anders@olsson.com	anders@olsson.com	1	4.34783

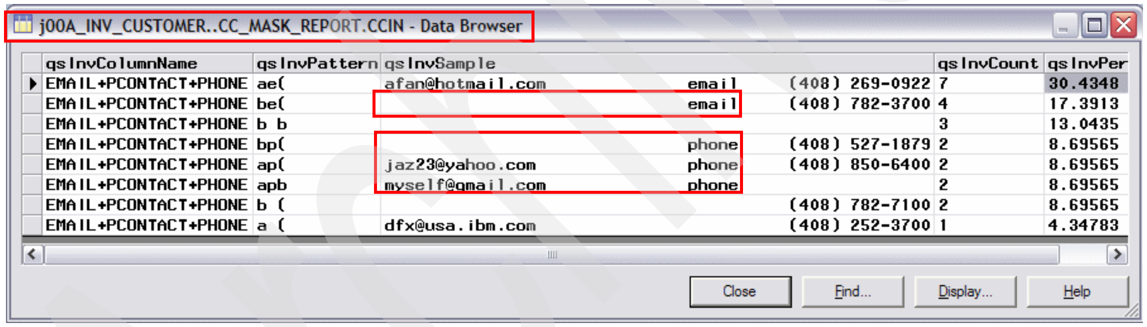
Figure 1-134 Create J00A_INV_CUSTOMER job 7/9



j00A_INV_CUSTOMER..T_MASK_REPORT.TIN - Data Browser

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
PHONE	(nnn)bnnn-nnnn	(408) 269-0922	18	78.2609
PCONTACT	aaaaa	phone	17	73.913
EMAIL			11	47.8261
PCONTACT			6	26.087
PHONE			5	21.7391
EMAIL	aaaaaa@aaaaa.aaa	myself@gmail.com	2	8.69565
EMAIL	aaaaaaa@aaaaaaa.aaa	anymail@hotmail.com	2	8.69565
EMAIL	aaaaaa@aaaaaa.aaa	anders@olsson.com	2	8.69565
EMAIL	aaaaaa@aaaa.aaa	curtis@home.com	1	4.34783
EMAIL	aaa@aaa.aaa.aaa	dfx@usa.ibm.com	1	4.34783
EMAIL	aaa@aaaaaaa.aaa	arc@hotmail.com	1	4.34783
EMAIL	aaaa@aaaaaaa.aaa	afan@hotmail.com	1	4.34783
EMAIL	aaann@aaaaa.aaa	jaz23@yahoo.com	1	4.34783
EMAIL	a@aaaaa.aaa	a@gmail.com	1	4.34783

Figure 1-135 Create J00A_INV_CUSTOMER job 8/9



j00A_INV_CUSTOMER..CC_MASK_REPORT.CCIN - Data Browser

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPer
EMAIL+PCONTACT+PHONE	ae(	afan@hotmail.com email (408) 269-0922	7	30.4348
EMAIL+PCONTACT+PHONE	be(	email (408) 782-3700	4	17.3913
EMAIL+PCONTACT+PHONE	b b		3	13.0435
EMAIL+PCONTACT+PHONE	bp(	phone (408) 527-1879	2	8.69565
EMAIL+PCONTACT+PHONE	ap(	jaz23@yahoo.com phone (408) 850-6400	2	8.69565
EMAIL+PCONTACT+PHONE	apb	myself@gmail.com phone	2	8.69565
EMAIL+PCONTACT+PHONE	b (	(408) 782-7100	2	8.69565
EMAIL+PCONTACT+PHONE	a (	dfx@usa.ibm.com (408) 252-3700	1	4.34783

Figure 1-136 Create J00A_INV_CUSTOMER job 9/9

J01_STAN_COUNTRY

This step involves analyzing the addresses on the credit card customer records to determine the (ISO code) country using the COUNTRY rule set in the Standardize stage.

Figure 1-137 on page 171 through Figure 1-167 on page 186 describe the steps using Designer Client to build and execute the DataStage job to perform this task.

The steps are as follows:

1. Figure 1-137 on page 171 shows the various stages that are used in this job, including the data set that was created in “J00_SRC_CUSTOMER” on page 142, a Standardize stage, and an output Data Set stage. This job uses the same QSPARAMETERSET object that was created earlier.
2. Attach the input credit card customer data and load the metadata as follows:
   a. Double-click the input **Data_Set_0** stage and enter the location of the data set that was created in “J00_SRC_CUSTOMER” on page 142 as shown in Figure 1-138 on page 172.
   b. Select the Columns tab to define the column metadata for the selected output link (DSLINK3). You can enter column definitions by typing them in this tab, or you can load predefined columns definitions from the repository. We chose to load by clicking **Load** to load a table definition from the repository and populate the Columns tab as shown in Figure 1-139 on page 172.
   c. In the Table Definition window that opens, browse the repository tree for the table definition that you want to load. Select the table definition (CUSTOMER) in the tree, and click **OK** as shown in Figure 1-140 on page 173.
   d. In the Select Columns window that opens, use the arrow buttons to move the columns that you want to load from the Available columns list to the Selected column list. In our case, we chose to load all the columns as shown in Figure 1-141 on page 173. Click **OK** to load the selected column definition into the Columns tab as shown in Figure 1-142 on page 174. Click **OK** to close the Data_Set_0 - Data Set window.
3. Then, modify the stage object names as follows:
   – Data_Set_0 to CUSTOMER
   – Standardize3 to STAN_COUNTRY
   – Data_Set_1 to CUSTOMER_STAN_COUNTRY
   – DSLink3 to IN
   – DSLink4 to OUT

   We do not show these steps here.
4. Next, configure the Standardize stage with the COUNTRY rule set as follows:
   a. Right-click the **STAN_COUNTRY** Standardize stage object and select **Properties** as shown in Figure 1-143 on page 174.
   b. In the Standardize Stage window, click **New Process** to open the Standardize Rule Process window as shown in Figure 1-144 on page 175.
   c. From the Standardize Rule Process window, click the ... button to browse the available rule sets as shown in Figure 1-145 on page 175.

d. In the Rule Sets window, expand the Standardization Rules folder (Figure 1-146 on page 176), and select the COUNTRY folder and COUNTRY (rule set) under the Other folder as shown in Figure 1-147 on page 176. Click **OK**.

e. Right-click **COUNTRY** (rule set) and select **Provision All** as shown in Figure 1-148 on page 177.

> **Note:** You need to provision new, copied, or customized rule sets in the Designer client before you can compile a job that uses them.

f. Figure 1-149 on page 177 and Figure 1-150 on page 177 show the progress and successful completion of the provisioning. Click **OK** in Figure 1-151 on page 178 to proceed to the Standardize Rule Process window.

g. In the Standardize Rule Process window that opens, key in the literal ZQUSZQ[18] in the Literal field and move it to the Selected Columns list, and do the same for the ADDR column as shown in Figure 1-152 on page 178 and Figure 1-153 on page 179. Click **OK**.

h. Click **Stage Properties** in the STAN_COUNTRY - Standardize Stage window to map the Standardize stage output columns as shown in Figure 1-154 on page 179.

i. Click the Mapping tab under Output in Figure 1-155 on page 180, and select all the columns in the left pane and copy them to the right pane. Click **OK**.

j. Click **OK** in Figure 1-156 on page 180 to complete the configuration of the Standardize stage.

5. Finally, configure the output CUSTOMER_STAN_COUNTRY data set object by right-clicking the icon and selecting **Properties** as shown in Figure 1-157 on page 181.

   You need to map the Standardize stage output columns and save the table definition to the Table Definitions folder. This definition is used in a subsequent stage. Click the Columns tab in the STAN_CUSTOMER - Data Set window (Figure 1-158 on page 181) to view the Standardize stage output columns in Figure 1-159 on page 182. You can save this set of columns as a table definition by clicking **Save** in Figure 1-159 on page 182.

   Figure 1-160 on page 182 and Figure 1-161 on page 183 show the saving of this table definition as CUSTOMER_STAN_CNTRY in the folder path \Table Definitions\ PlugIn\DSDB2\CUSTOMER.

[18] This literal indicates that the following columns mainly have U.S. addresses.

6. After saving, compiling, and running the job j01_STAN_COUNTRY, you view the results as shown in Figure 1-162 on page 183.
7. You can view the contents of the input data by right-clicking the CUSTOMER Data Set icon and selecting **View IN data** as shown in Figure 1-163 on page 184. Figure 1-164 on page 184 through Figure 1-165 on page 185 shows the contents of this data set.
8. Similarly, you can view the contents of the CUSTOMER_STAN_COUNTRY data set object as shown in Figure 1-166 on page 185 through Figure 1-167 on page 186. The report shows the two columns ISOCountryCode_COUNTRY and IdentifierFlag_COUNTRY added by the Standardize stage. It shows US for each row with the identifier flag of Y.

Now, proceed to “J02_INVCC_ISCODE” on page 186.

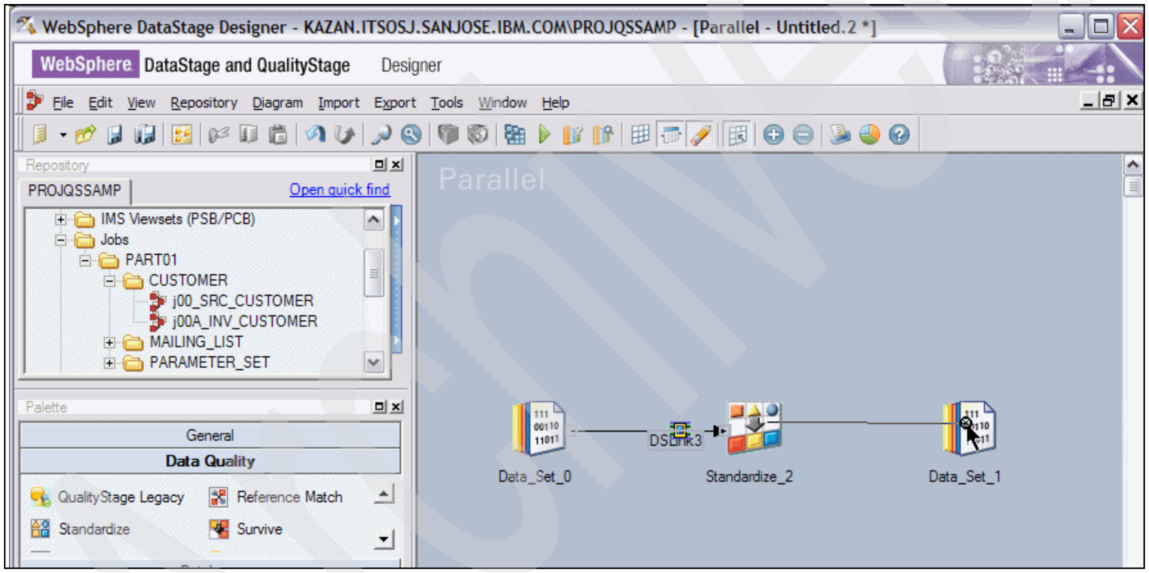


Figure 1-137 Create J01_STAN_COUNTRY job 1/31

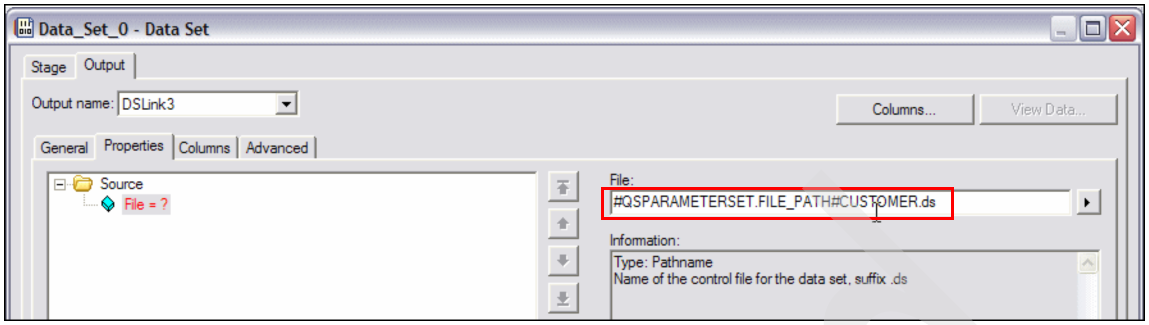


Figure 1-138 Create J01_STAN_COUNTRY job 2/31

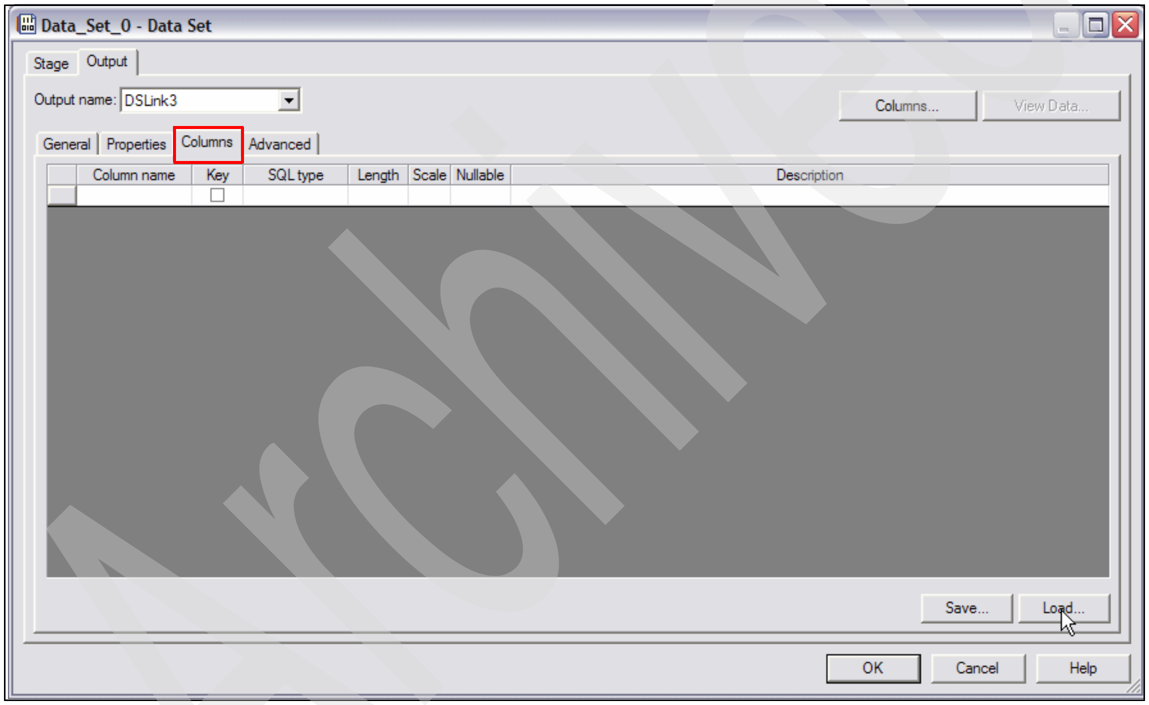


Figure 1-139 Create J01_STAN_COUNTRY job 3/31

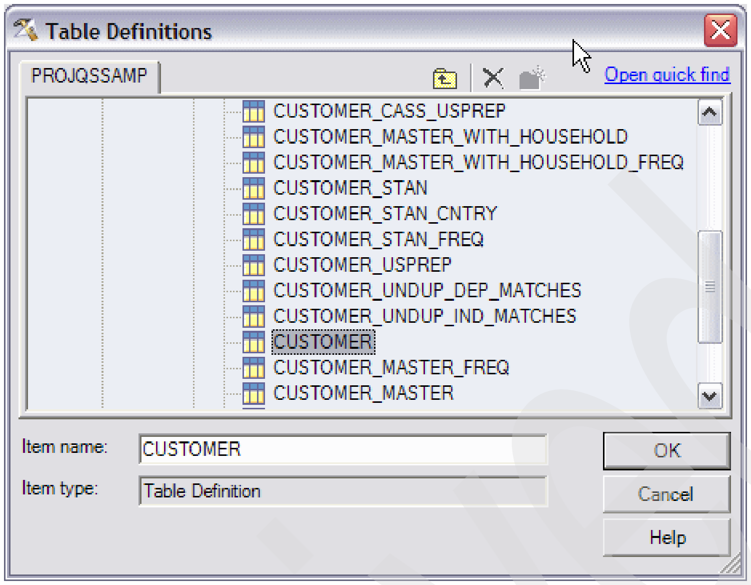


Table Definitions

PROJQSSAMP

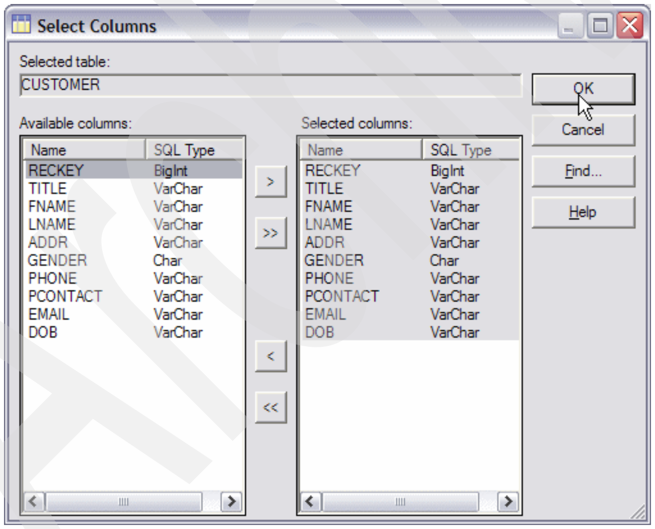
Open quick find

- CUSTOMER_CASS_USPREP
- CUSTOMER_MASTER_WITH_HOUSEHOLD
- CUSTOMER_MASTER_WITH_HOUSEHOLD_FREQ
- CUSTOMER_STAN
- CUSTOMER_STAN_CNTRY
- CUSTOMER_STAN_FREQ
- CUSTOMER_USPREP
- CUSTOMER_UNDUP_DEP_MATCHES
- CUSTOMER_UNDUP_IND_MATCHES
- CUSTOMER
- CUSTOMER_MASTER_FREQ
- CUSTOMER_MASTER

Item name: CUSTOMER

Item type: Table Definition

Figure 1-140 Create J01_STAN_COUNTRY job 4/31


Select Columns

Selected table: CUSTOMER

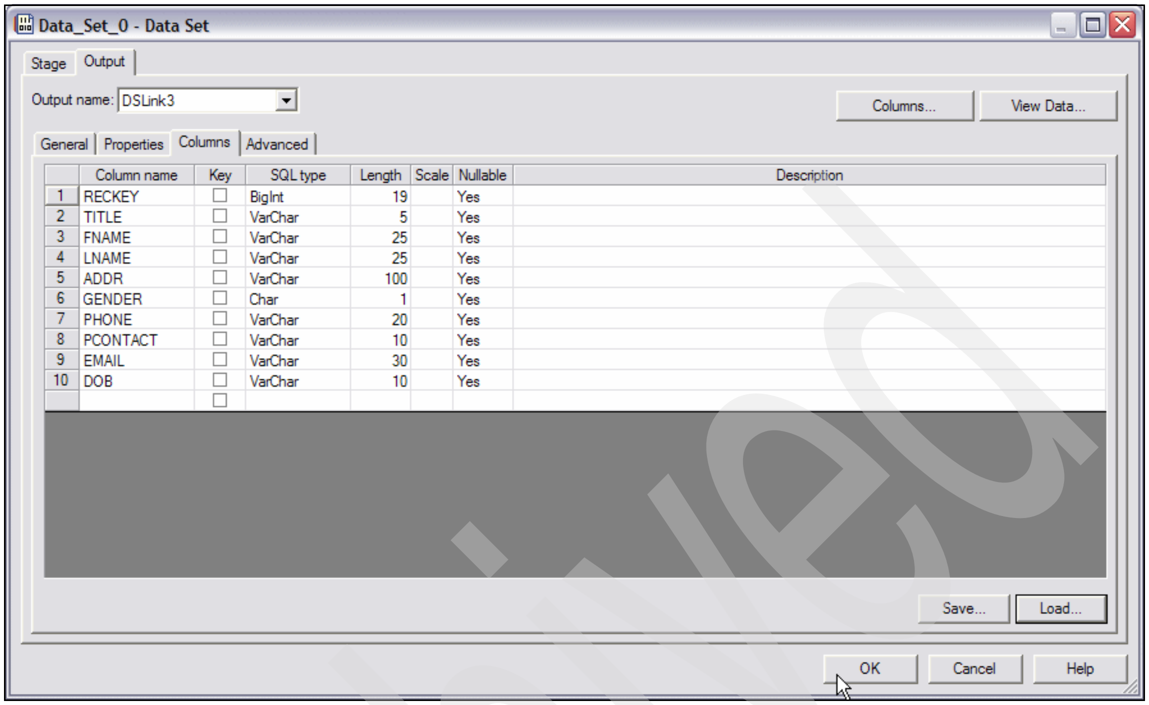
Available columns:

Name	SQL Type
RECKEY	BigInt
TITLE	VarChar
FNAME	VarChar
LNAME	VarChar
ADDR	VarChar
GENDER	Char
PHONE	VarChar
PCONTACT	VarChar
EMAIL	VarChar
DOB	VarChar

Selected columns:

Name	SQL Type
RECKEY	BigInt
TITLE	VarChar
FNAME	VarChar
LNAME	VarChar
ADDR	VarChar
GENDER	Char
PHONE	VarChar
PCONTACT	VarChar
EMAIL	VarChar
DOB	VarChar

Figure 1-141 Create J01_STAN_COUNTRY job 5/31



	Column name	Key	SQL type	Length	Scale	Nullable	Description
1	RECKEY	☐	BigInt	19		Yes	
2	TITLE	☐	VarChar	5		Yes	
3	FNAME	☐	VarChar	25		Yes	
4	LNAME	☐	VarChar	25		Yes	
5	ADDR	☐	VarChar	100		Yes	
6	GENDER	☐	Char	1		Yes	
7	PHONE	☐	VarChar	20		Yes	
8	PCONTACT	☐	VarChar	10		Yes	
9	EMAIL	☐	VarChar	30		Yes	
10	DOB	☐	VarChar	10		Yes	
		☐					

Figure 1-142 Create J01_STAN_COUNTRY job 6/31

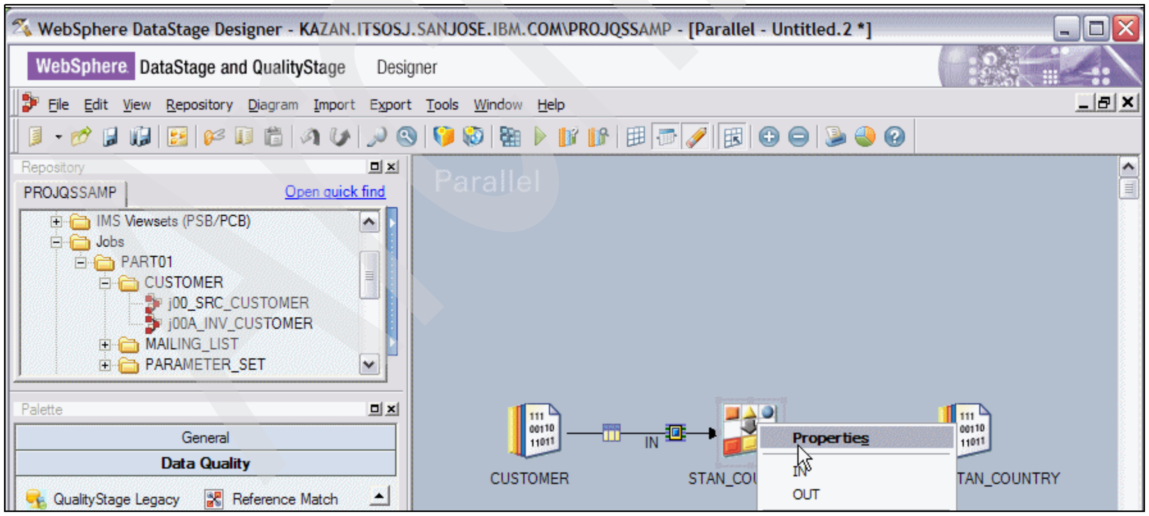
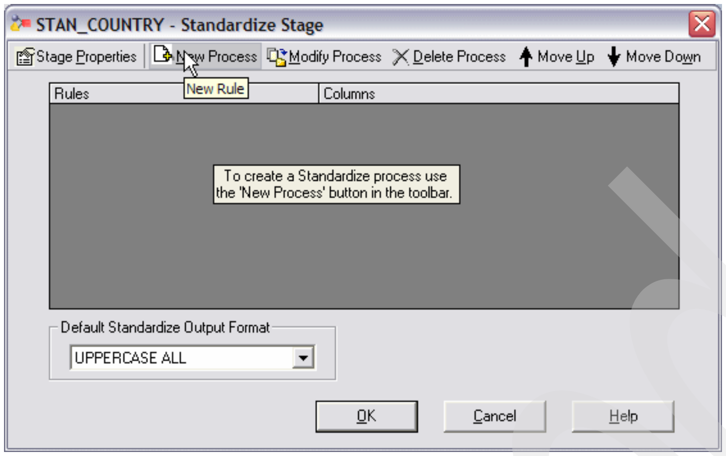
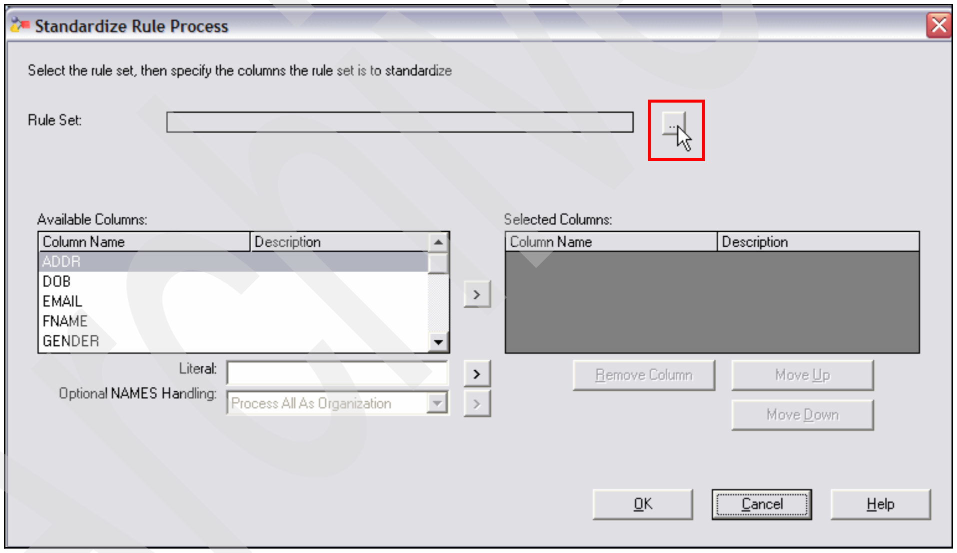


Figure 1-143 Create J01_STAN_COUNTRY job 7/31



To create a Standardize process use the 'New Process' button in the toolbar.

Figure 1-144   Create J01_STAN_COUNTRY job 8/31



Select the rule set, then specify the columns the rule set is to standardize

Figure 1-145   Create J01_STAN_COUNTRY job 9/31

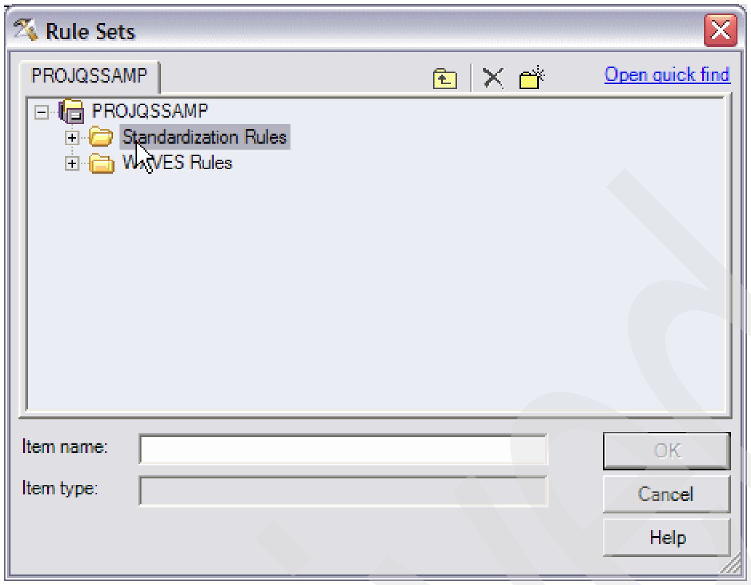
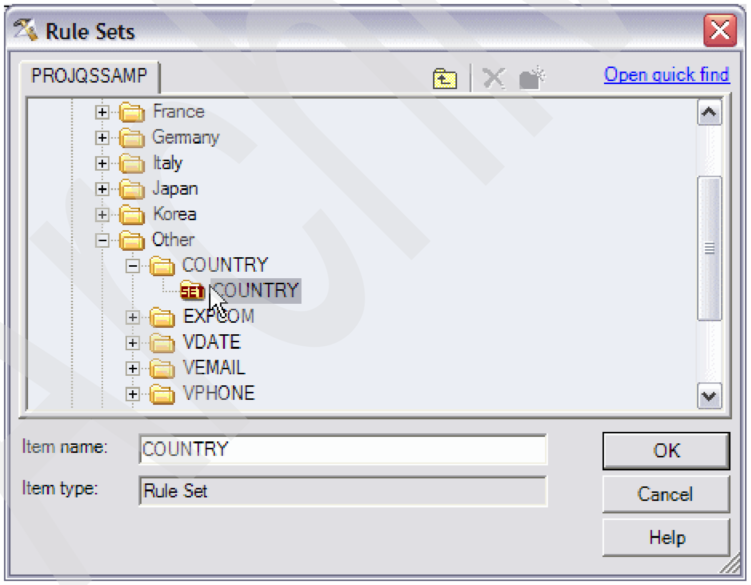


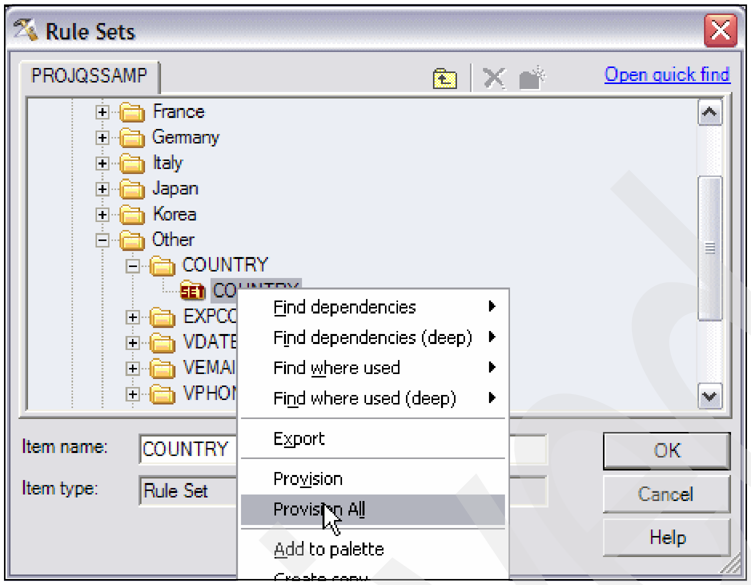
Figure 1-146   Create J01_STAN_COUNTRY job 10/31



Item name: COUNTRY

Item type: Rule Set

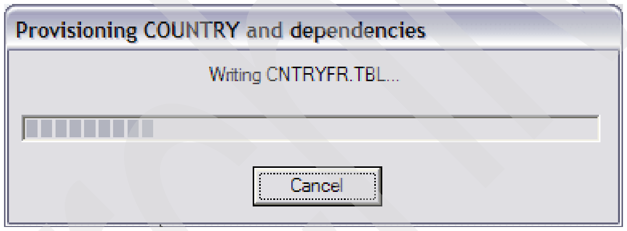
Figure 1-147   Create J01_STAN_COUNTRY job 11/31



Item name: COUNTRY

Item type: Rule Set

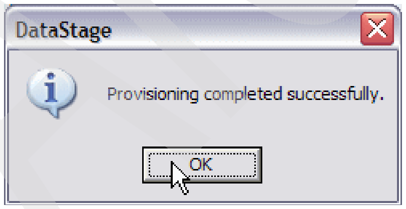
Figure 1-148   Create J01_STAN_COUNTRY job 12/31



Provisioning COUNTRY and dependencies

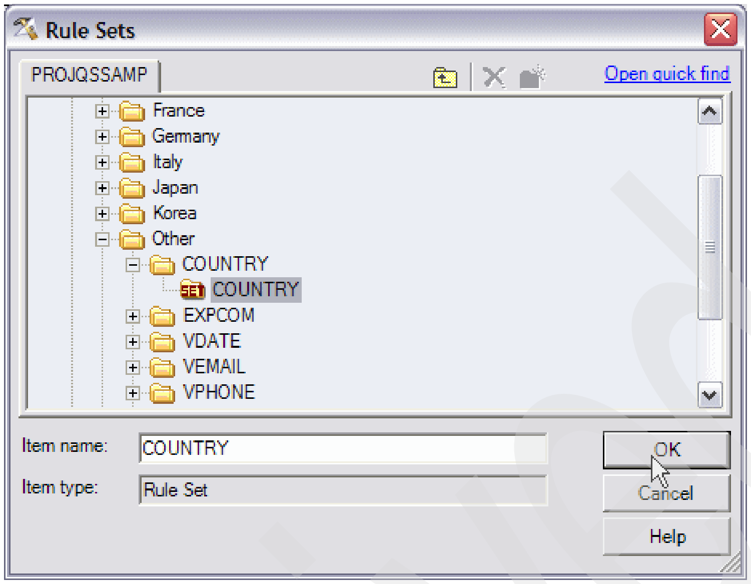
Writing CNTRYFR.TBL...

Figure 1-149   Create J01_STAN_COUNTRY job 13/31



Provisioning completed successfully.

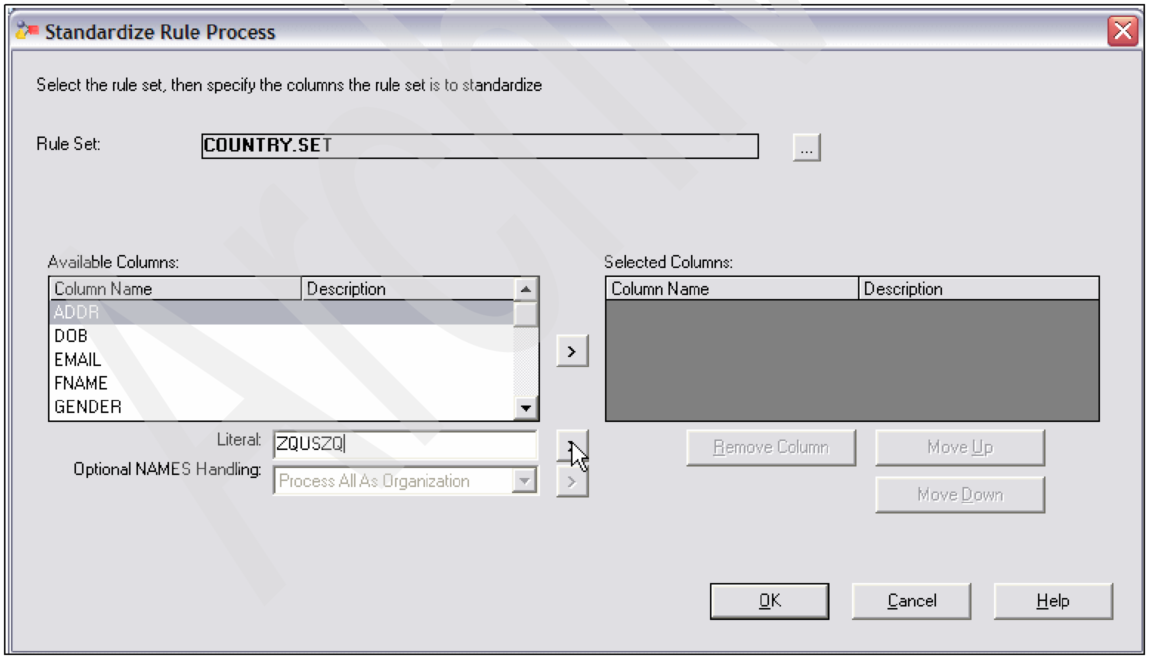
Figure 1-150   Create J01_STAN_COUNTRY job 14/31



Item name: COUNTRY

Item type: Rule Set

Figure 1-151   Create J01_STAN_COUNTRY job 15/31



Select the rule set, then specify the columns the rule set is to standardize

Rule Set: COUNTRY.SET

Available Columns:

Column Name	Description
ADDR	
DOB	
EMAIL	
FNAME	
GENDER	

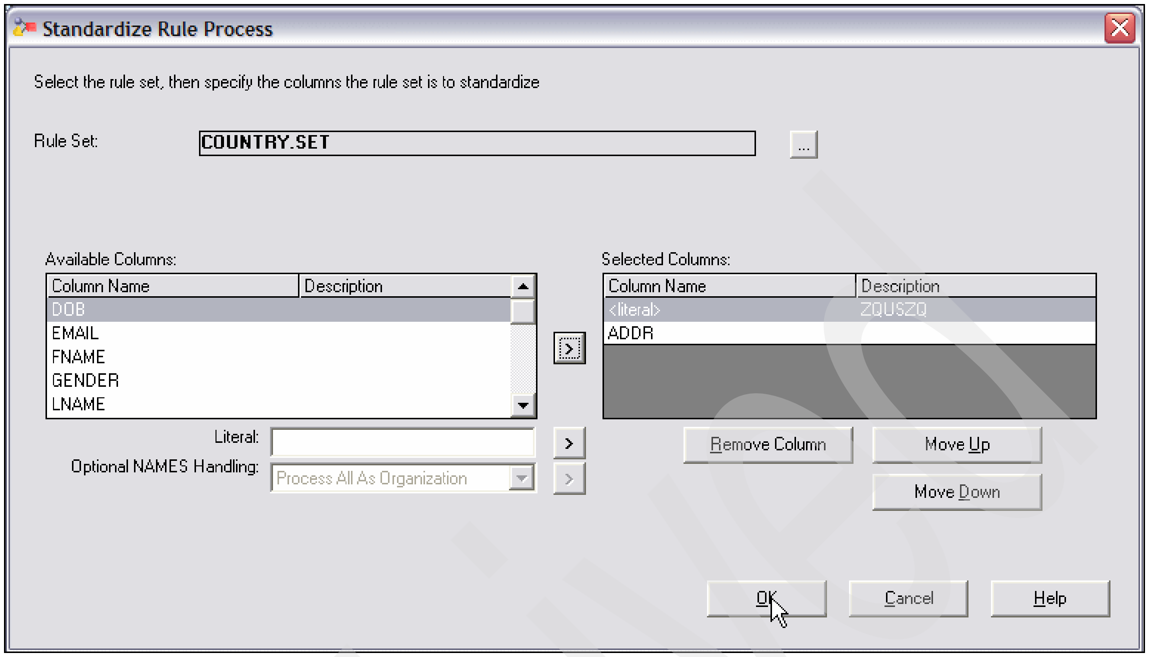
Selected Columns:

Column Name	Description

Literal: ZQUSZQ

Optional NAMES Handling: Process All As Organization

Figure 1-152   Create J01_STAN_COUNTRY job 16/31



Standardize Rule Process

Select the rule set, then specify the columns the rule set is to standardize

Rule Set: COUNTRY.SET

Available Columns:

Column Name	Description
DOB	
EMAIL	
FNAME	
GENDER	
LNAME	

Selected Columns:

Column Name	Description
<literal>	ZQUSZQ
ADDR	

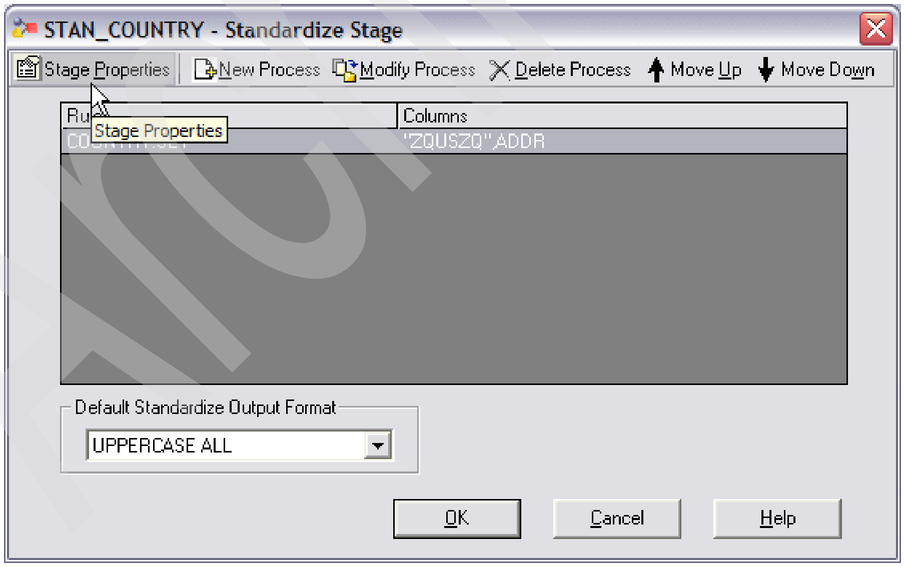
Literal:

Optional NAMES Handling: Process All As Organization

Remove Column Move Up Move Down

OK Cancel Help

Figure 1-153 Create J01_STAN_COUNTRY job 17/31



STAN_COUNTRY - Standardize Stage

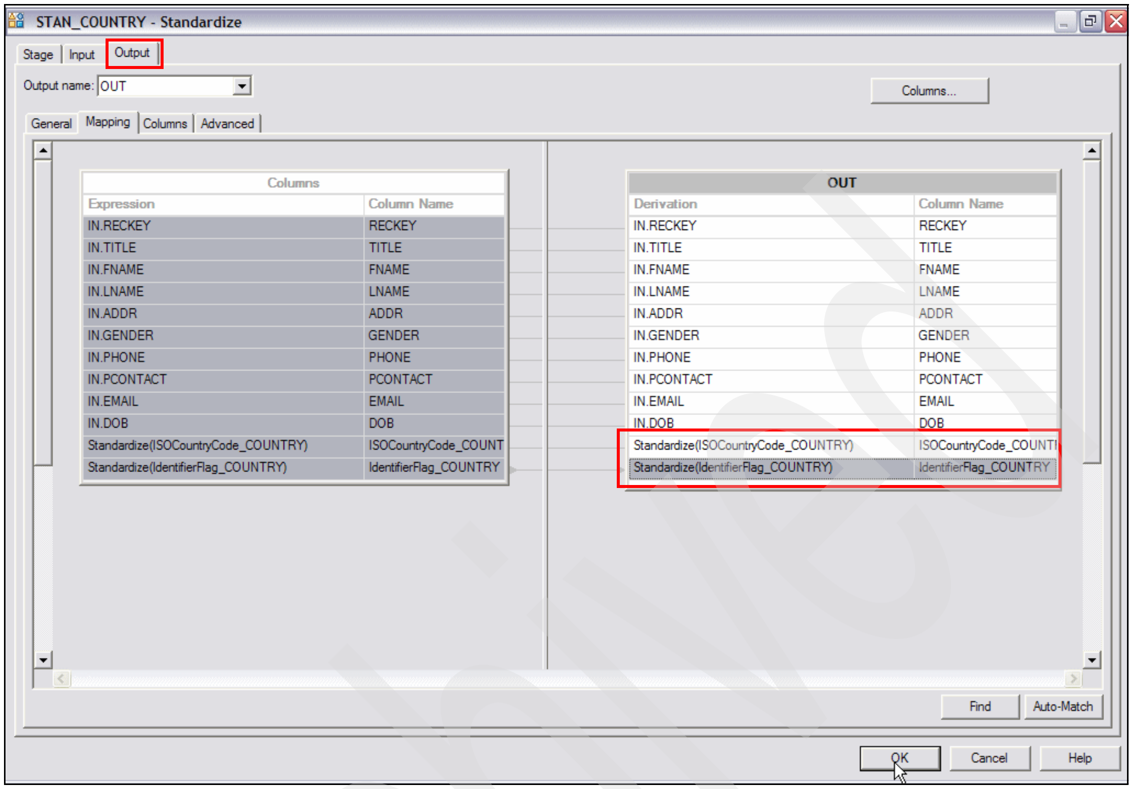
Stage Properties New Process Modify Process Delete Process Move Up Move Down

Ru	Columns
CO	"ZQUSZQ",ADDR

Default Standardize Output Format: UPPERCASE ALL

OK Cancel Help

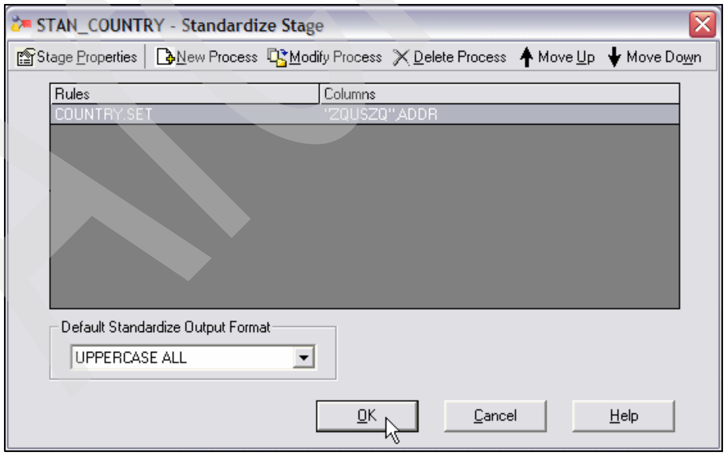
Figure 1-154 Create J01_STAN_COUNTRY job 18/31



Columns	
Expression	Column Name
IN.RECKEY	RECKEY
IN.TITLE	TITLE
IN.FNAME	FNAME
IN.LNAME	LNAME
IN.ADDR	ADDR
IN.GENDER	GENDER
IN.PHONE	PHONE
IN.PCONTACT	PCONTACT
IN.EMAIL	EMAIL
IN.DOB	DOB
Standardize(ISOCountryCode_COUNTRY)	ISOCountryCode_COUNT
Standardize(IdentifierFlag_COUNTRY)	IdentifierFlag_COUNTRY

OUT	
Derivation	Column Name
IN.RECKEY	RECKEY
IN.TITLE	TITLE
IN.FNAME	FNAME
IN.LNAME	LNAME
IN.ADDR	ADDR
IN.GENDER	GENDER
IN.PHONE	PHONE
IN.PCONTACT	PCONTACT
IN.EMAIL	EMAIL
IN.DOB	DOB
Standardize(ISOCountryCode_COUNTRY)	ISOCountryCode_COUNTI
Standardize(IdentifierFlag_COUNTRY)	IdentifierFlag_COUNTRY

Figure 1-155 Create J01_STAN_COUNTRY job 19/31



Rules	Columns
COUNTRY.SET	"ZQUSZQ" ADDR

Default Standardize Output Format: UPPERCASE ALL

Figure 1-156 Create J01_STAN_COUNTRY job 20/31

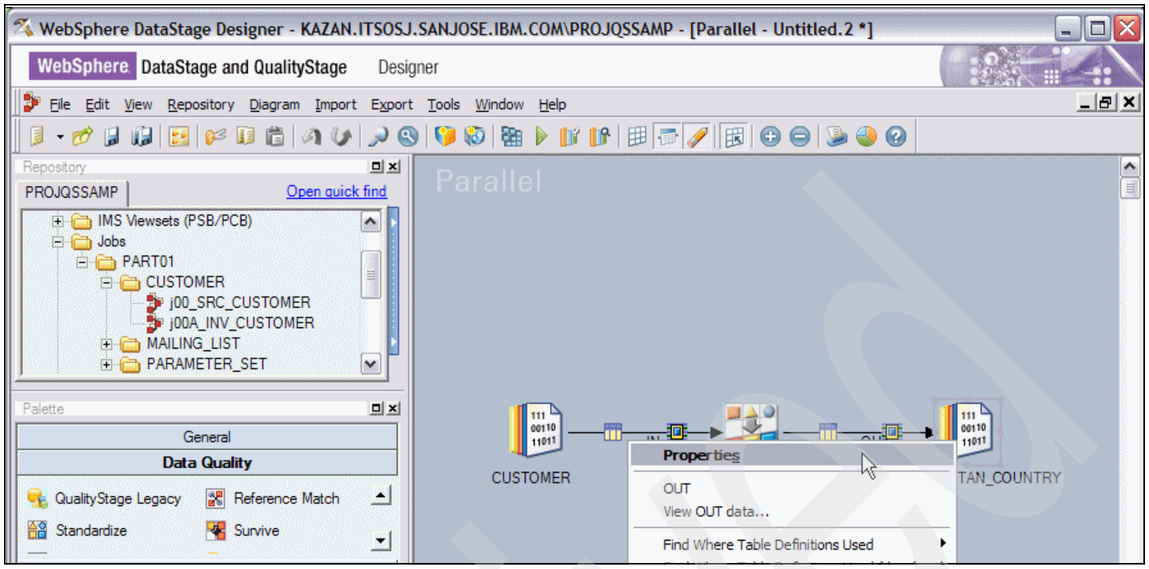


Figure 1-157 Create J01_STAN_COUNTRY job 21/31

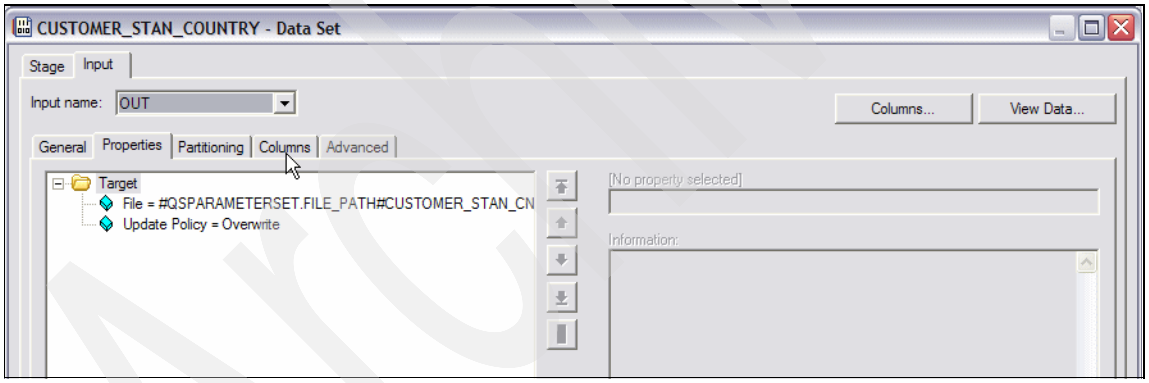
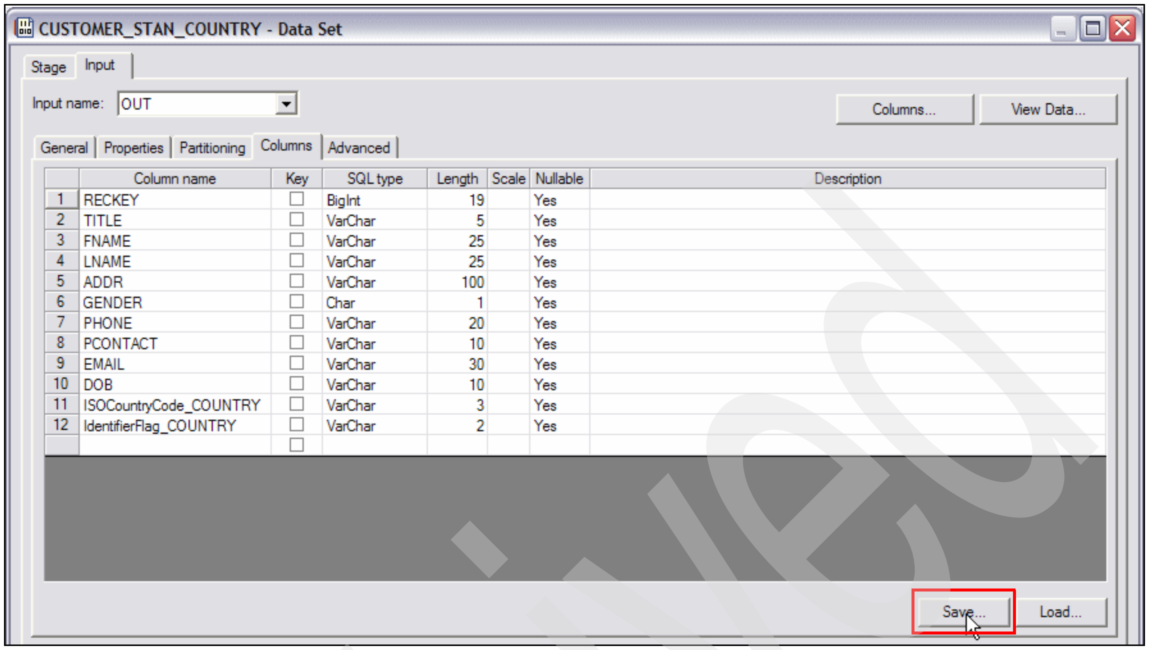


Figure 1-158 Create J01_STAN_COUNTRY job 22/31

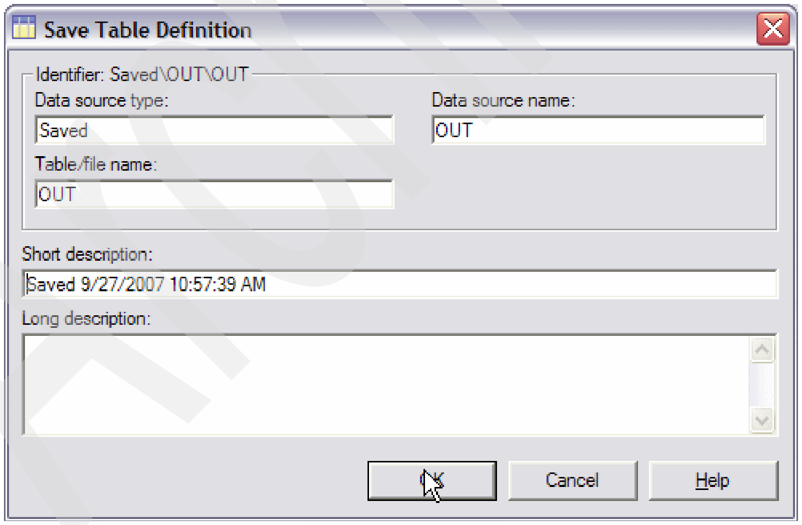


CUSTOMER_STAN_COUNTRY - Data Set

Input name: OUT

	Column name	Key	SQL type	Length	Scale	Nullable	Description
1	RECKEY	☐	BigInt	19		Yes	
2	TITLE	☐	VarChar	5		Yes	
3	FNAME	☐	VarChar	25		Yes	
4	LNAME	☐	VarChar	25		Yes	
5	ADDR	☐	VarChar	100		Yes	
6	GENDER	☐	Char	1		Yes	
7	PHONE	☐	VarChar	20		Yes	
8	PCONTACT	☐	VarChar	10		Yes	
9	EMAIL	☐	VarChar	30		Yes	
10	DOB	☐	VarChar	10		Yes	
11	ISOCountryCode_COUNTRY	☐	VarChar	3		Yes	
12	IdentifierFlag_COUNTRY	☐	VarChar	2		Yes	

Figure 1-159 Create J01_STAN_COUNTRY job 23/31



Save Table Definition

Identifier: Saved\OUT\OUT

Data source type: Saved

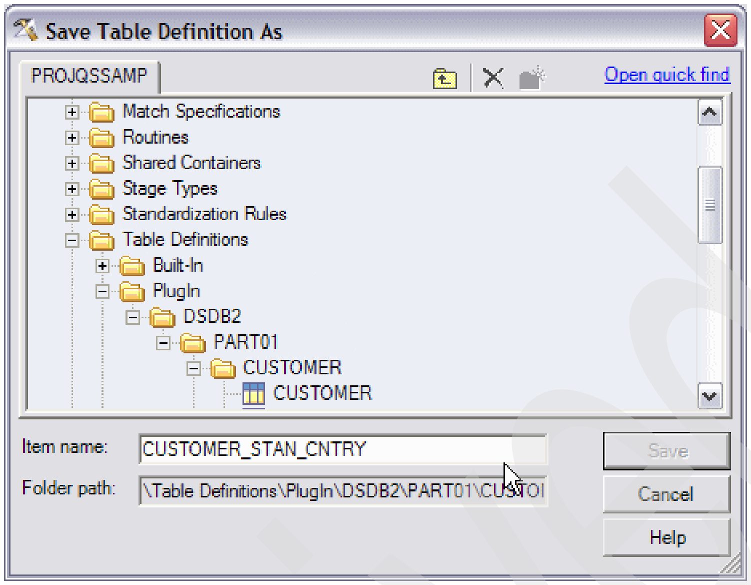
Data source name: OUT

Table/file name: OUT

Short description: Saved 9/27/2007 10:57:39 AM

Long description:

Figure 1-160 Create J01_STAN_COUNTRY job 24/31


Save Table Definition As

PROJQSSAMP

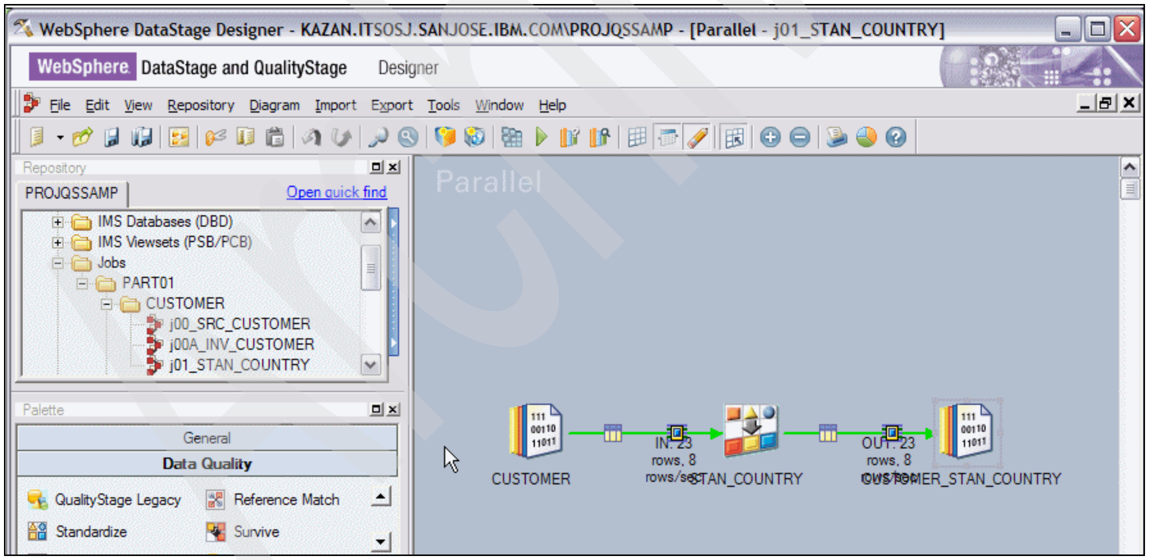
Open quick find

- Match Specifications
- Routines
- Shared Containers
- Stage Types
- Standardization Rules
- Table Definitions
  - Built-In
  - PlugIn
    - DSDB2
      - PART01
        - CUSTOMER
          - CUSTOMER

Item name: CUSTOMER_STAN_CNTRY

Folder path: \Table Definitions\PlugIn\DSDB2\PART01\CUSTOI

Figure 1-161 Create J01_STAN_COUNTRY job 25/31


WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j01_STAN_COUNTRY]

WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Repository
PROJQSSAMP
Open quick find

- IMS Databases (DBD)
- IMS Viewsets (PSB/PCB)
- Jobs
  - PART01
    - CUSTOMER
      - j00_SRC_CUSTOMER
      - j00A_INV_CUSTOMER
      - j01_STAN_COUNTRY

Figure 1-162 Create J01_STAN_COUNTRY job 26/31

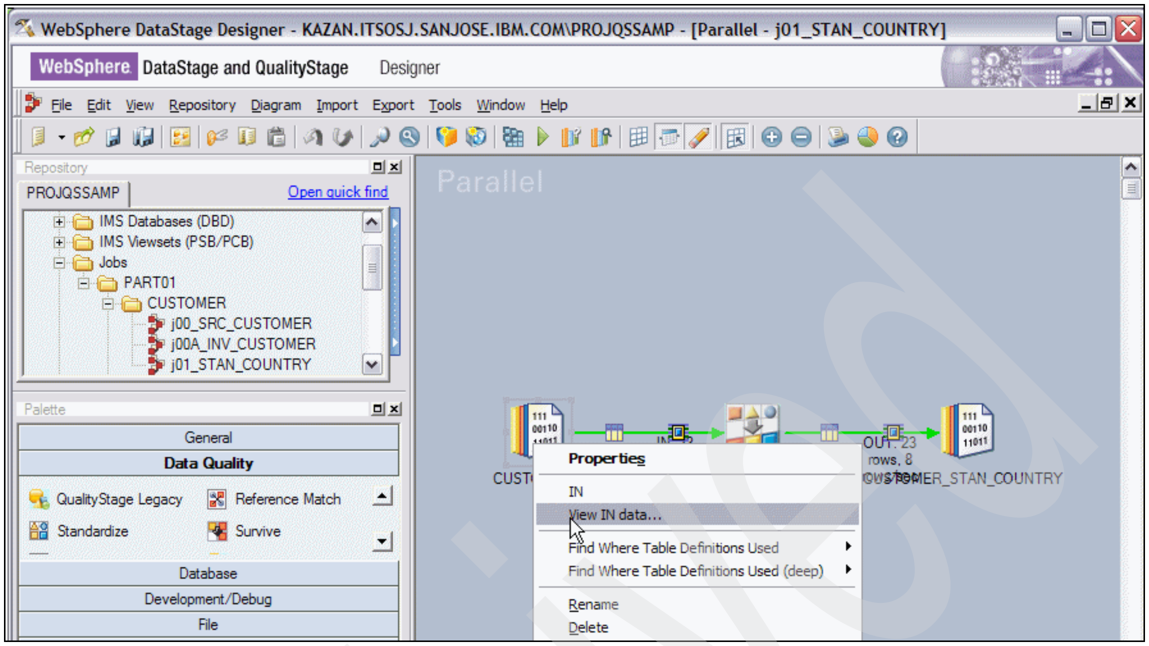
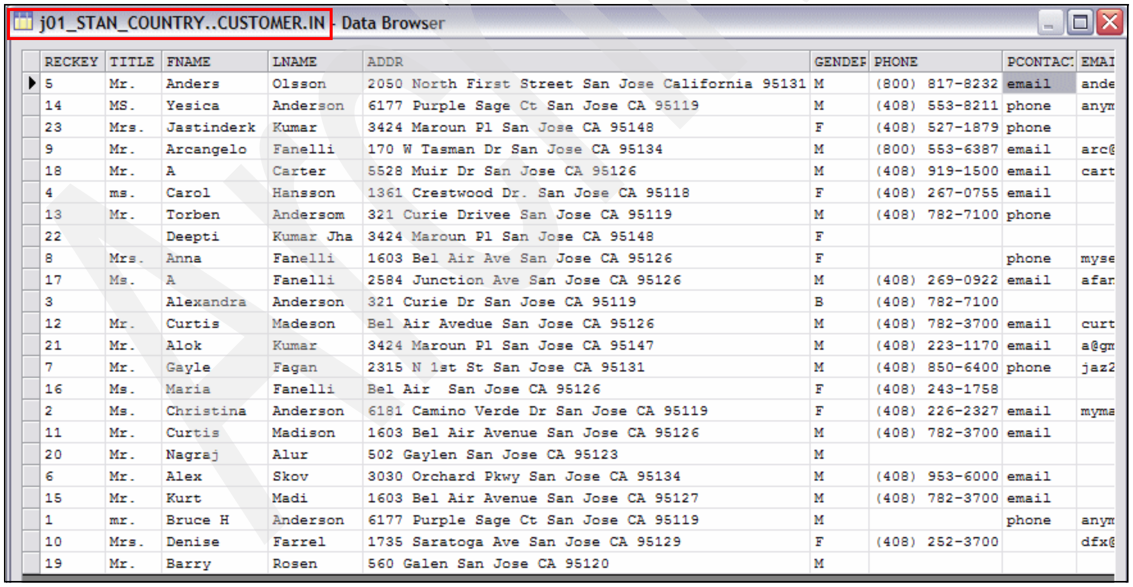
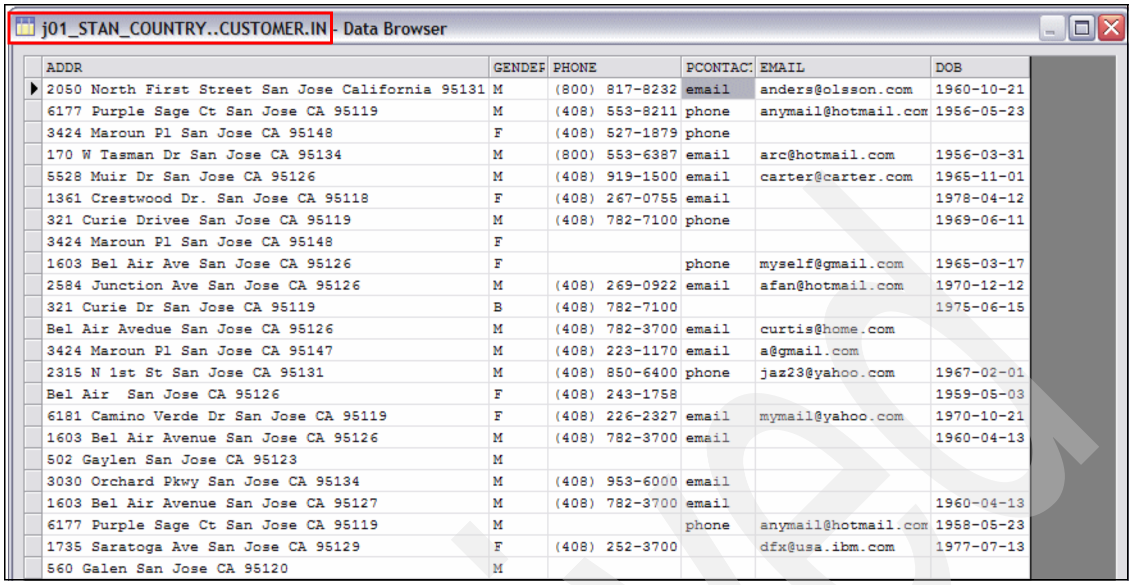


Figure 1-163   Create J01_STAN_COUNTRY job 27/31



RECKEY	TITLE	FNAME	LNAME	ADDR	GENDEF	PHONE	PCONTAC	EMAI
5	Mr.	Anders	Olsson	2050 North First Street San Jose California 95131	M	(800) 817-8232	email	ande
14	MS.	Yesica	Anderson	6177 Purple Sage Ct San Jose CA 95119	M	(408) 553-8211	phone	anym
23	Mrs.	Jastinderk	Kumar	3424 Maroun Pl San Jose CA 95148	F	(408) 527-1879	phone	
9	Mr.	Arcangelo	Fanelli	170 W Tasman Dr San Jose CA 95134	M	(800) 553-6387	email	arc@
18	Mr.	A	Carter	5528 Muir Dr San Jose CA 95126	M	(408) 919-1500	email	cart
4	ms.	Carol	Hansson	1361 Crestwood Dr. San Jose CA 95118	F	(408) 267-0755	email	
13	Mr.	Torben	Andersom	321 Curie Drivee San Jose CA 95119	M	(408) 782-7100	phone	
22		Deepti	Kumar Jha	3424 Maroun Pl San Jose CA 95148	F			
8	Mrs.	Anna	Fanelli	1603 Bel Air Ave San Jose CA 95126	F		phone	myse
17	Ms.	A	Fanelli	2584 Junction Ave San Jose CA 95126	M	(408) 269-0922	email	afar
3		Alexandra	Anderson	321 Curie Dr San Jose CA 95119	B	(408) 782-7100		
12	Mr.	Curtis	Madeson	Bel Air Avedue San Jose CA 95126	M	(408) 782-3700	email	curt
21	Mr.	Alok	Kumar	3424 Maroun Pl San Jose CA 95147	M	(408) 223-1170	email	a@gm
7	Mr.	Gayle	Fagan	2315 N 1st St San Jose CA 95131	M	(408) 850-6400	phone	jaz2
16	Ms.	Maria	Fanelli	Bel Air San Jose CA 95126	F	(408) 243-1758		
2	Ms.	Christina	Anderson	6181 Camino Verde Dr San Jose CA 95119	F	(408) 226-2327	email	myma
11	Mr.	Curtis	Madison	1603 Bel Air Avenue San Jose CA 95126	M	(408) 782-3700	email	
20	Mr.	Nagraj	Alur	502 Gaylen San Jose CA 95123	M			
6	Mr.	Alex	Skov	3030 Orchard Pkwy San Jose CA 95134	M	(408) 953-6000	email	
15	Mr.	Kurt	Madi	1603 Bel Air Avenue San Jose CA 95127	M	(408) 782-3700	email	
1	mr.	Bruce H	Anderson	6177 Purple Sage Ct San Jose CA 95119	M		phone	anym
10	Mrs.	Denise	Farrel	1735 Saratoga Ave San Jose CA 95129	F	(408) 252-3700		dfx@
19	Mr.	Barry	Rosen	560 Galen San Jose CA 95120	M			

Figure 1-164   Create J01_STAN_COUNTRY job 28/31



j01_STAN_COUNTRY..CUSTOMER.IN - Data Browser

ADDR	GENDEF	PHONE	PCONTAC:	EMAIL	DOB
2050 North First Street San Jose California 95131	M	(800) 817-8232	email	anders@olsson.com	1960-10-21
6177 Purple Sage Ct San Jose CA 95119	M	(408) 553-8211	phone	anymail@hotmail.com	1956-05-23
3424 Maroun Pl San Jose CA 95148	F	(408) 527-1879	phone		
170 W Tasman Dr San Jose CA 95134	M	(800) 553-6387	email	arc@hotmail.com	1956-03-31
5528 Muir Dr San Jose CA 95126	M	(408) 919-1500	email	carter@carter.com	1965-11-01
1361 Crestwood Dr. San Jose CA 95118	F	(408) 267-0755	email		1978-04-12
321 Curie Drivee San Jose CA 95119	M	(408) 782-7100	phone		1969-06-11
3424 Maroun Pl San Jose CA 95148	F				
1603 Bel Air Ave San Jose CA 95126	F		phone	myself@gmail.com	1965-03-17
2584 Junction Ave San Jose CA 95126	M	(408) 269-0922	email	afan@hotmail.com	1970-12-12
321 Curie Dr San Jose CA 95119	B	(408) 782-7100			1975-06-15
Bel Air Avedue San Jose CA 95126	M	(408) 782-3700	email	curtis@home.com	
3424 Maroun Pl San Jose CA 95147	M	(408) 223-1170	email	a@gmail.com	
2315 N 1st St San Jose CA 95131	M	(408) 850-6400	phone	jaz23@yahoo.com	1967-02-01
Bel Air San Jose CA 95126	F	(408) 243-1758			1959-05-03
6181 Camino Verde Dr San Jose CA 95119	F	(408) 226-2327	email	mymail@yahoo.com	1970-10-21
1603 Bel Air Avenue San Jose CA 95126	M	(408) 782-3700	email		1960-04-13
502 Gaylen San Jose CA 95123	M				
3030 Orchard Pkwy San Jose CA 95134	M	(408) 953-6000	email		
1603 Bel Air Avenue San Jose CA 95127	M	(408) 782-3700	email		1960-04-13
6177 Purple Sage Ct San Jose CA 95119	M		phone	anymail@hotmail.com	1958-05-23
1735 Saratoga Ave San Jose CA 95129	F	(408) 252-3700		dfx@usa.ibm.com	1977-07-13
560 Galen San Jose CA 95120	M				

Figure 1-165   Create J01_STAN_COUNTRY job 29/31

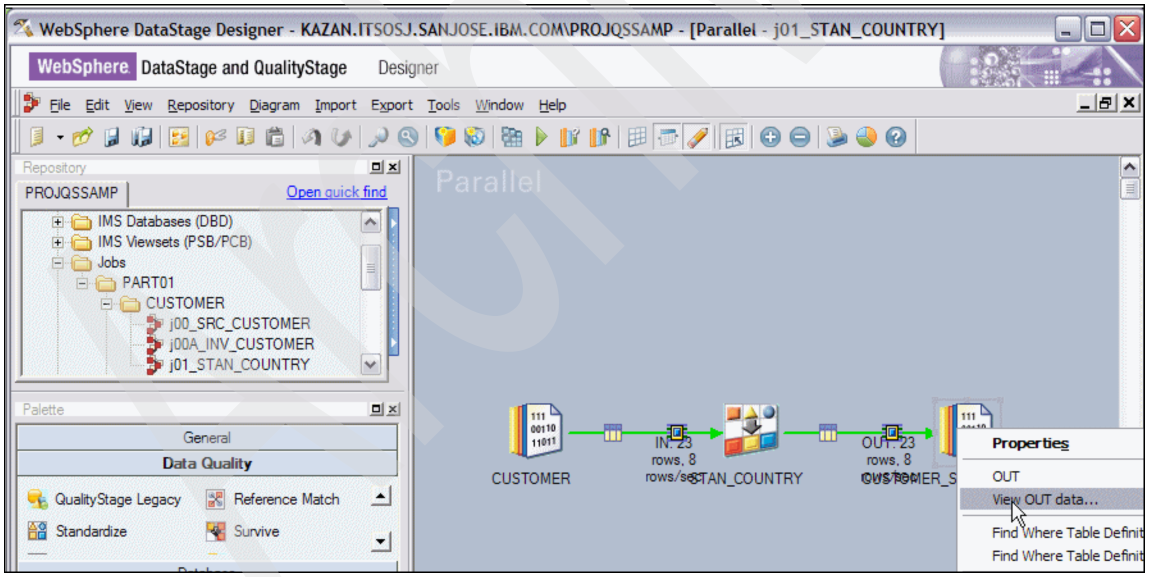
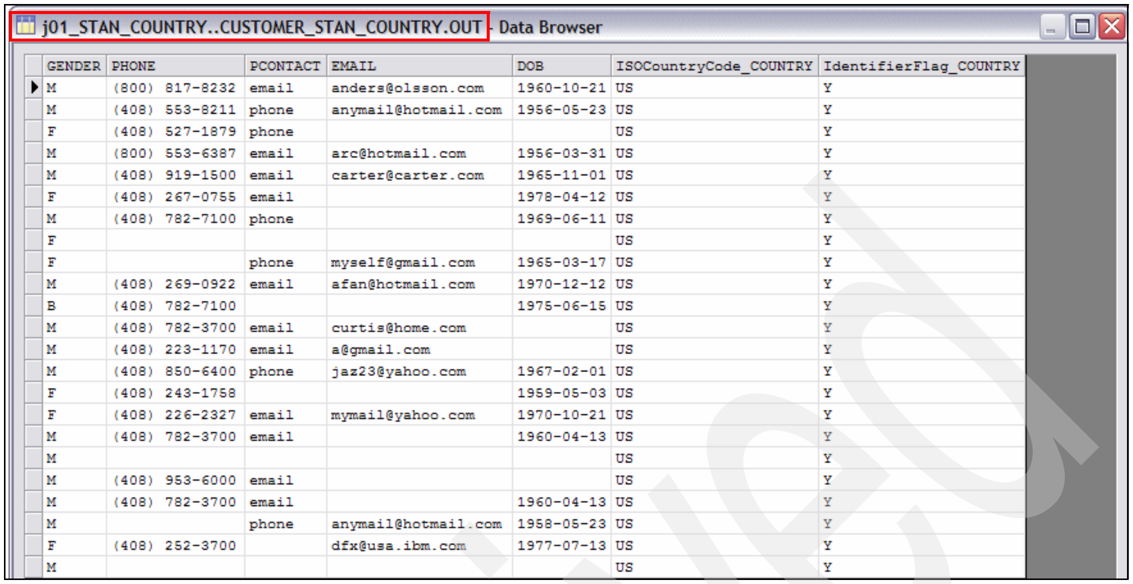


Figure 1-166   Create J01_STAN_COUNTRY job 30/31



j01_STAN_COUNTRY..CUSTOMER_STAN_COUNTRY.OUT - Data Browser

GENDER	PHONE	PCONTACT	EMAIL	DOB	ISOCountryCode_COUNTRY	IdentifierFlag_COUNTRY
M	(800) 817-8232	email	anders@olsson.com	1960-10-21	US	Y
M	(408) 553-8211	phone	anymail@hotmail.com	1956-05-23	US	Y
F	(408) 527-1879	phone			US	Y
M	(800) 553-6387	email	arc@hotmail.com	1956-03-31	US	Y
M	(408) 919-1500	email	carter@carter.com	1965-11-01	US	Y
F	(408) 267-0755	email		1978-04-12	US	Y
M	(408) 782-7100	phone		1969-06-11	US	Y
F					US	Y
F		phone	myself@gmail.com	1965-03-17	US	Y
M	(408) 269-0922	email	afan@hotmail.com	1970-12-12	US	Y
B	(408) 782-7100			1975-06-15	US	Y
M	(408) 782-3700	email	curtis@home.com		US	Y
M	(408) 223-1170	email	a@gmail.com		US	Y
M	(408) 850-6400	phone	jaz23@yahoo.com	1967-02-01	US	Y
F	(408) 243-1758			1959-05-03	US	Y
F	(408) 226-2327	email	mymail@yahoo.com	1970-10-21	US	Y
M	(408) 782-3700	email		1960-04-13	US	Y
M					US	Y
M	(408) 953-6000	email			US	Y
M	(408) 782-3700	email		1960-04-13	US	Y
M		phone	anymail@hotmail.com	1958-05-23	US	Y
F	(408) 252-3700		dfx@usa.ibm.com	1977-07-13	US	Y
M					US	Y

Figure 1-167 Create J01_STAN_COUNTRY job 31/31

J02_INVCC_ISCODE

In this job, we analyze the ISO codes that were generated by the "J01_STAN_COUNTRY" on page 168 job by the Investigate stage using character discrete with the *C* mask to obtain frequency distribution. This step identifies whether the addresses in the credit card customer file belong to more than one country and identifies the codes of the countries in the addresses. In this case, all the addresses are U.S. addresses.

Figure 1-168 on page 188 through Figure 1-181 on page 196 describe the steps using Designer Client to build and execute the DataStage job to perform this task.

The steps are as follows:

1. Figure 1-168 on page 188 shows the various stages that are used in this job, including the data set that was created in "J01_STAN_COUNTRY" on page 168, an Investigate stage, and an output Sequential File stage. We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object that was created earlier.
2. Configure the input STAN_COUNTRY_COUNTRY data set with the table definition CUSTOMER_STAN_CNTRY that was saved in "J01_STAN_COUNTRY" on page 168. We do not repeat this process here because it is similar to the steps shown in Figure 1-138 on page 172 through Figure 1-142 on page 174.

3. Right-click the **INV_CC_ISOCODE** icon and select **Properties** to configure this stage as follows:
   a. Click **Character Concatenate Investigate**. The columns propagated from the input data set are shown in the Available Data Columns list. Select ISOCountryCode_Country column, and click **Add to Selected Columns** as shown in Figure 1-169 on page 188.
   b. In the Mask Column Selection window, click **All C** for each of the three characters in this column as shown in Figure 1-170 on page 189 and Figure 1-171 on page 189.
   c. Repeat this process for the IdentifierFlag_COUNTRY column. We do not show this process here.
   d. Figure 1-172 on page 190 shows the two columns that are selected with the *C* masks for each character.
   e. Click **Stage Properties** in Figure 1-172 on page 190 and when the INV_CC_ISCODE - Investigate windows opens, select **Mapping** in the **Output** tab—the Output name is *IN*. Map all the columns in the Columns list to the IN list as shown in Figure 1-173 on page 191. Click **OK**.
4. Configure the output sequential file ISOCODE_REPORT object to sort the incoming records on descending sequence of the count of distinct values in the two concatenated character columns. The objective is to determine the predominant country addresses and identify the ISO codes that are present in the data.

   Figure 1-174 on page 191 through Figure 1-177 on page 193 show the configuration of the sequential file.
5. After saving, compiling, and running this job, the job statistics are shown in Figure 1-178 on page 194.
6. The contents of the data input to the Investigate stage is shown in Figure 1-179 on page 195 through Figure 1-180 on page 195.
7. The output of the Investigate stage that is written to the sequential file is shown in Figure 1-181 on page 196. It shows a single record with a concatenated value of `US Y` in 100% of the records.

Now, proceed to “J03_STAN_USPREP” on page 196.

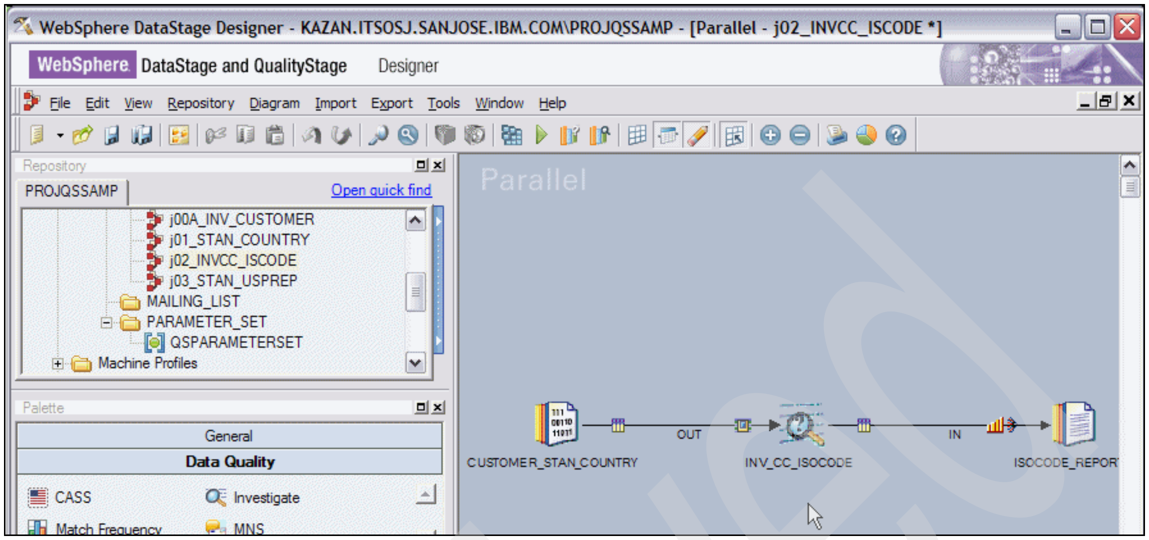
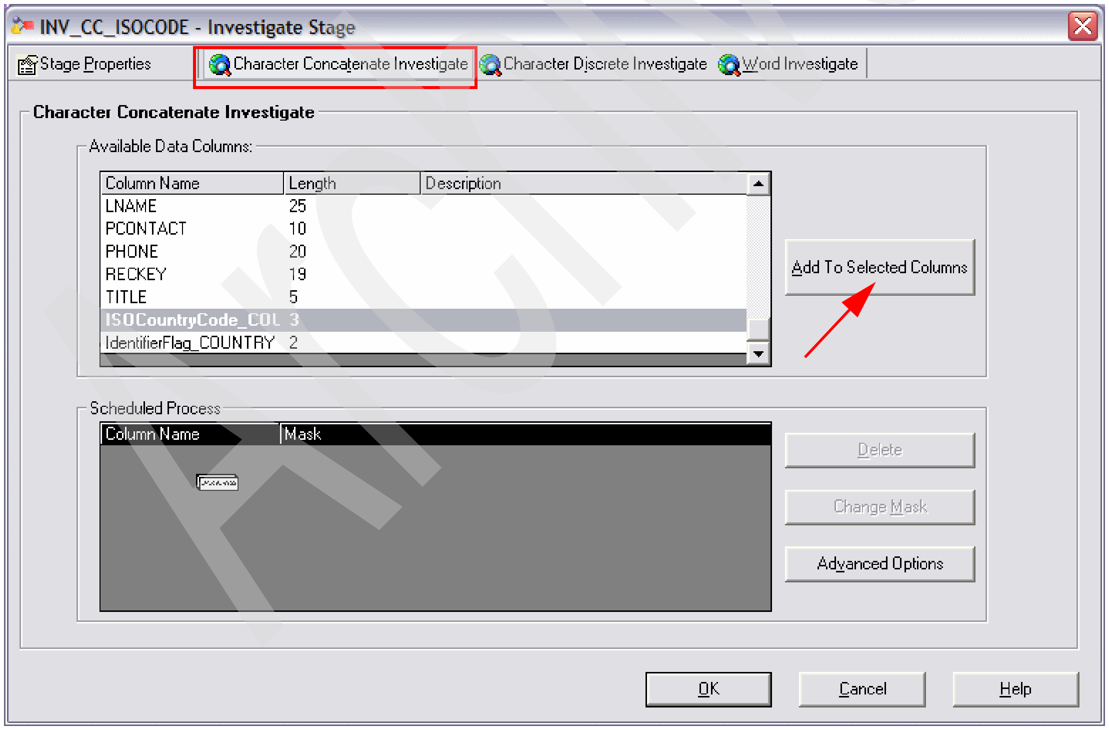


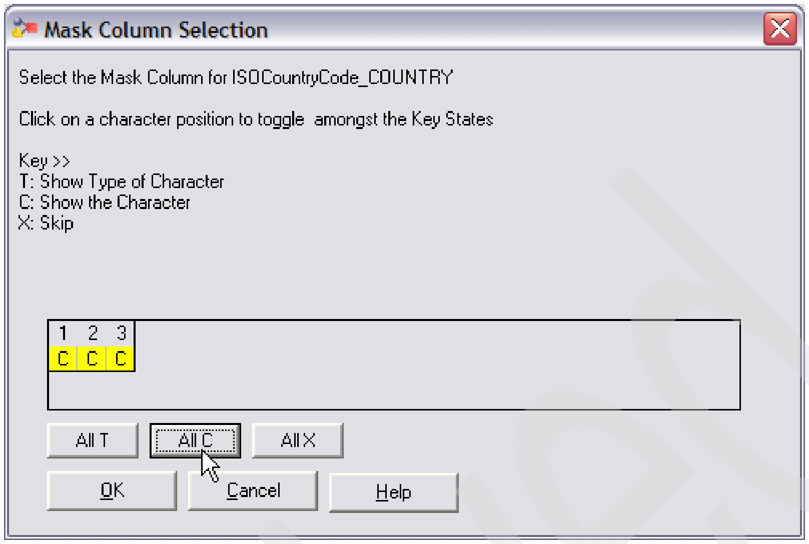
Figure 1-168 Create J02_INVCC_CODE 1/14



Column Name	Length	Description
LNAME	25	
PCONTACT	10	
PHONE	20	
RECKEY	19	
TITLE	5	
ISOCountryCode_COL	3	
IdentifierFlag_COUNTRY	2	

Column Name	Mask

Figure 1-169 Create J02_INVCC_CODE 2/14


Mask Column Selection

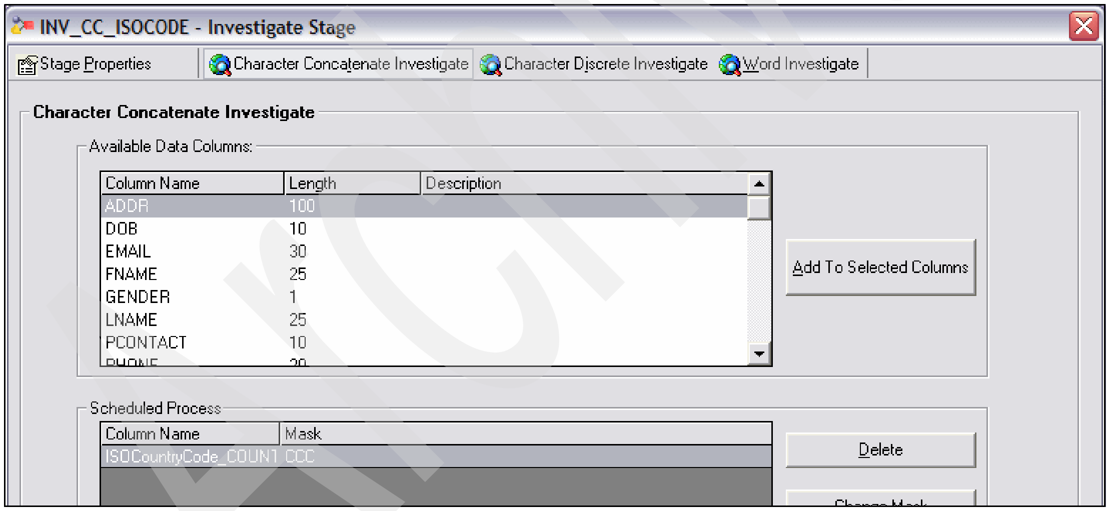
Select the Mask Column for ISOCountryCode_COUNTRY

Click on a character position to toggle amongst the Key States

Key >>
T: Show Type of Character
C: Show the Character
X: Skip

1	2	3
C	C	C

Figure 1-170   Create J02_INVCC_CODE 3/14


INV_CC_ISOCODE - Investigate Stage

Stage Properties
Character Concatenate Investigate
Character Discrete Investigate
Word Investigate

Character Concatenate Investigate

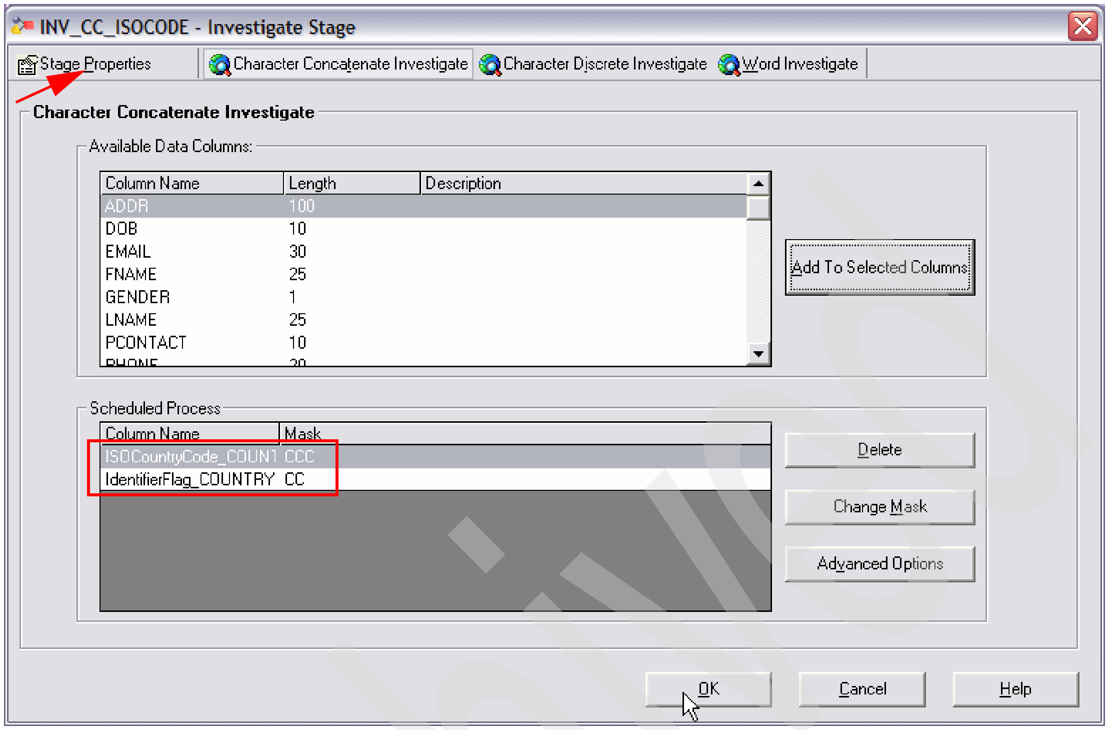
Available Data Columns:

Column Name	Length	Description
ADDR	100	
DOB	10	
EMAIL	30	
FNAME	25	
GENDER	1	
LNAME	25	
PCONTACT	10	
PHONE	20	

Scheduled Process

Column Name	Mask
ISOCountryCode_COUNT	CCC

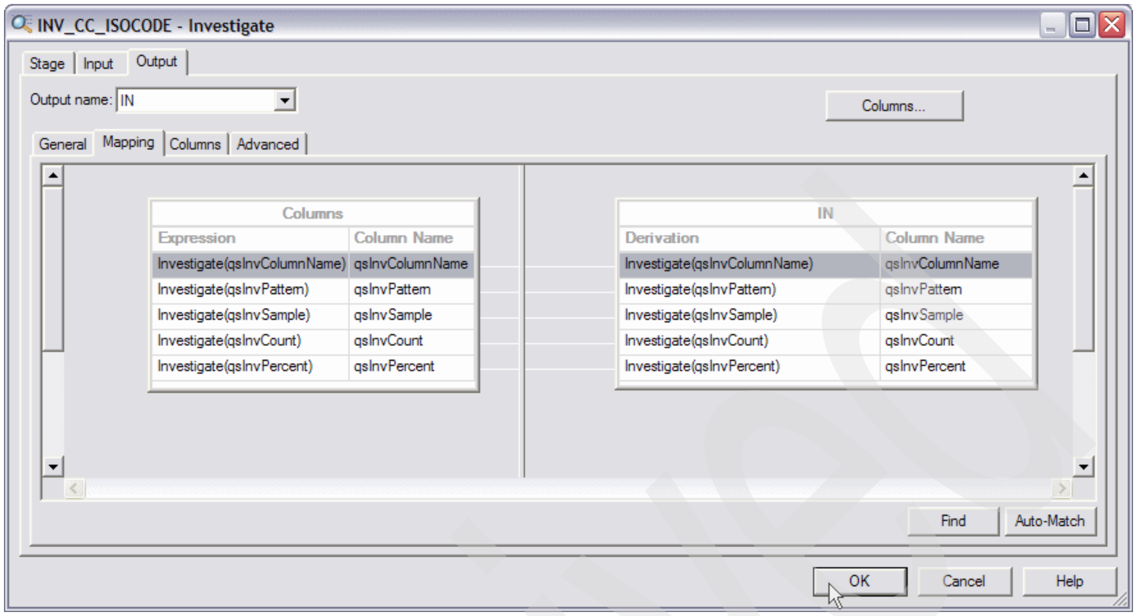
Figure 1-171   Create J02_INVCC_CODE 4/14



Column Name	Length	Description
ADDR	100	
DOB	10	
EMAIL	30	
FNAME	25	
GENDER	1	
LNAME	25	
PCONTACT	10	
PHONE	20	

Column Name	Mask
ISOCountryCode_COUNT	CCC
IdentifierFlag_COUNTRY	CC

Figure 1-172 Create J02_INVCC_CODE 5/14

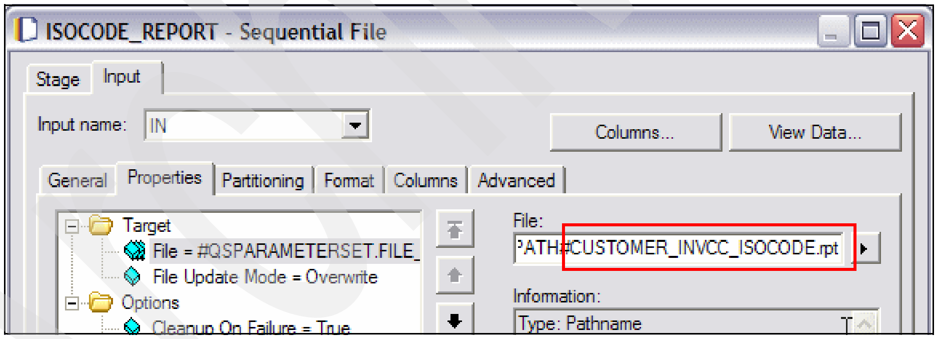

INV_CC_ISOCODE - Investigate

Output name: IN

Columns	
Expression	Column Name
Investigate(qsInvColumnName)	qsInvColumnName
Investigate(qsInvPattern)	qsInvPattern
Investigate(qsInvSample)	qsInvSample
Investigate(qsInvCount)	qsInvCount
Investigate(qsInvPercent)	qsInvPercent

IN	
Derivation	Column Name
Investigate(qsInvColumnName)	qsInvColumnName
Investigate(qsInvPattern)	qsInvPattern
Investigate(qsInvSample)	qsInvSample
Investigate(qsInvCount)	qsInvCount
Investigate(qsInvPercent)	qsInvPercent

Figure 1-173   Create J02_INVCC_CODE 6/14


ISOCODE_REPORT - Sequential File

Input name: IN

Target
File = #QSPARAMETERSET.FILE_
File Update Mode = Overwrite
Options
Cleanup On Failure = True

File: ʾATH#CUSTOMER_INVCC_ISOCODE.rpt

Information:
Type: Pathname

Figure 1-174   Create J02_INVCC_CODE 7/14

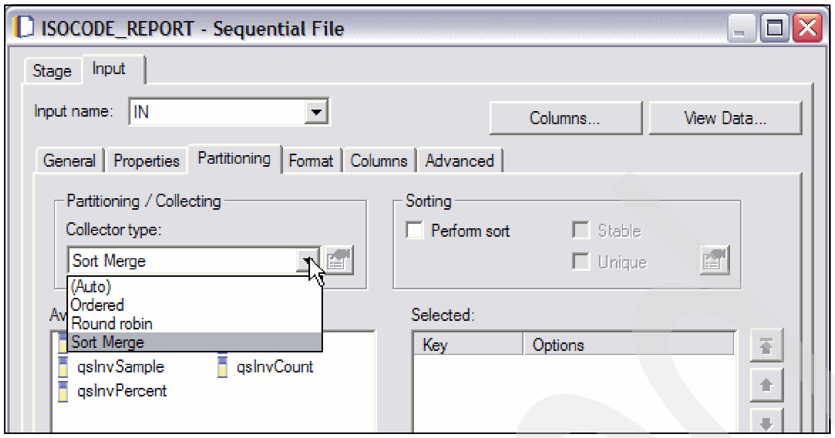


Figure 1-175   Create J02_INVCC_CODE 8/14

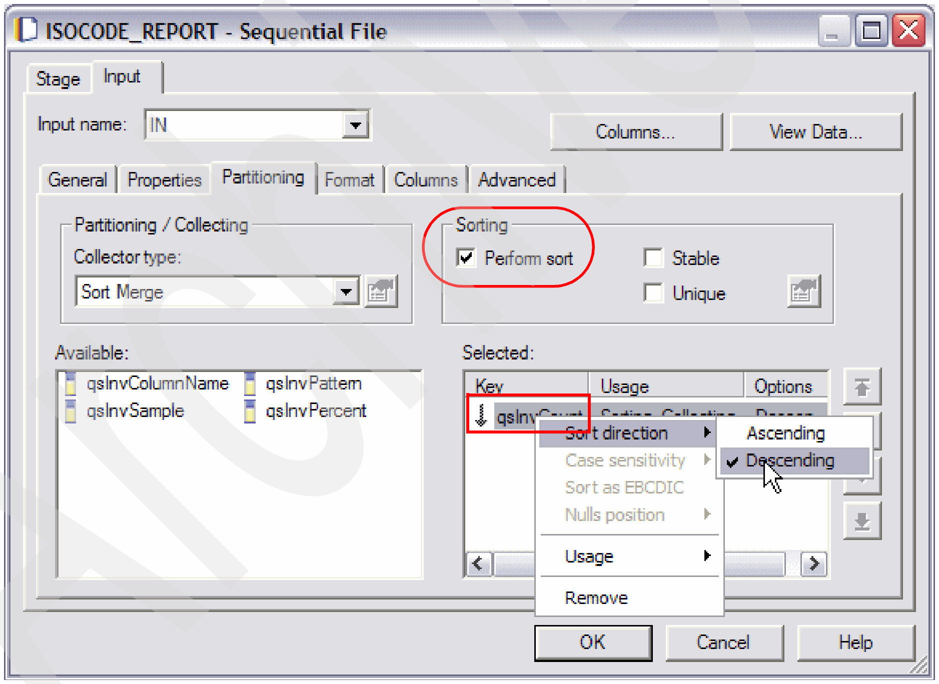
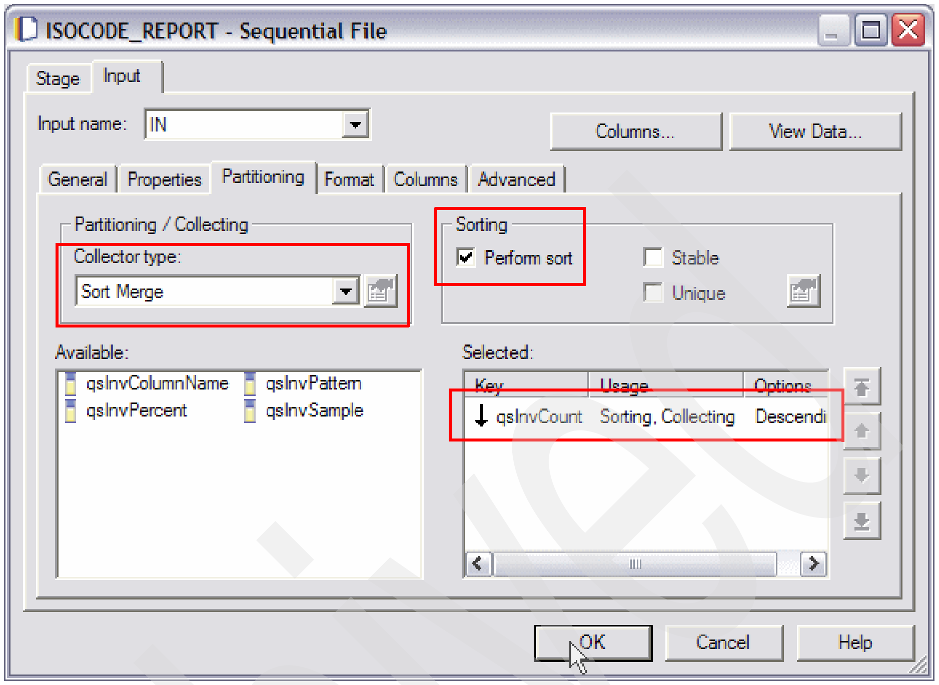


Figure 1-176   Create J02_INVCC_CODE 9/14


ISOCODE_REPORT - Sequential File

Stage | Input

Input name: IN

Columns... | View Data...

General | Properties | Partitioning | Format | Columns | Advanced

Partitioning / Collecting

Collector type: Sort Merge

Sorting

[✓] Perform sort

[ ] Stable

[ ] Unique

Available:

qsInvColumnName qsInvPattern

qsInvPercent qsInvSample

Selected:

Key	Usage	Options
qsInvCount	Sorting, Collecting	Descendi

OK | Cancel | Help

Figure 1-177 Create J02_INVCC_CODE 10/14

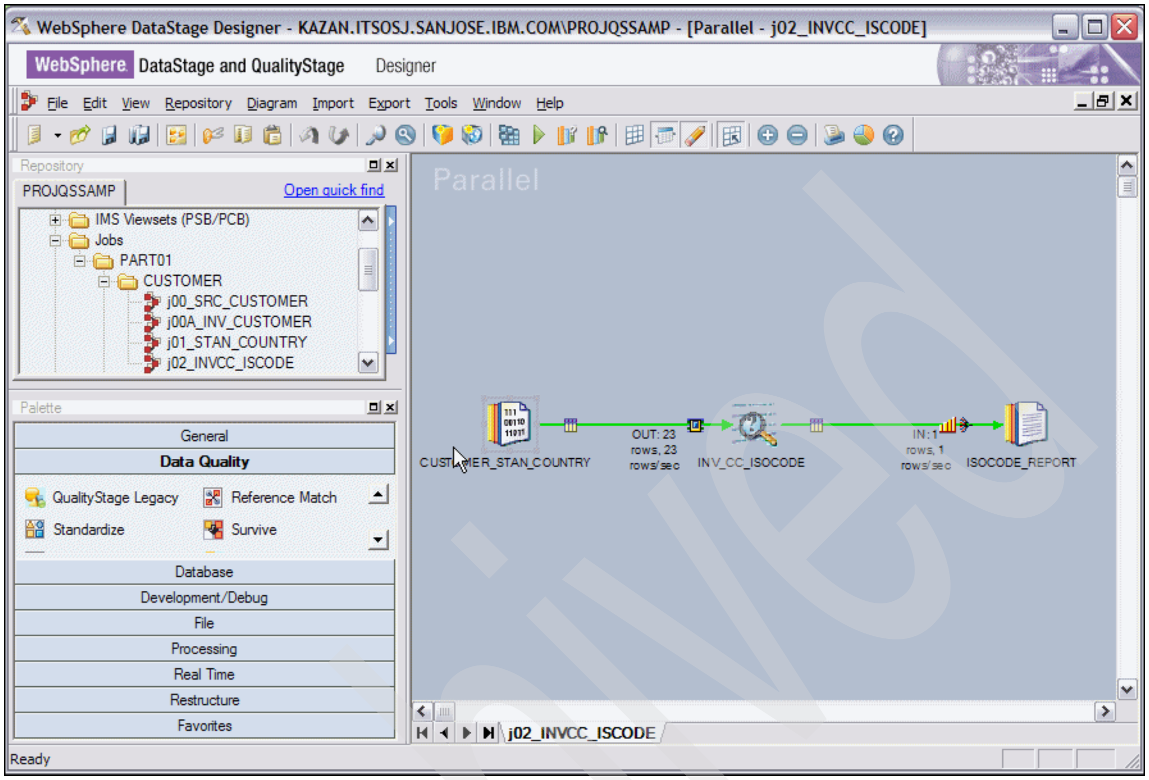
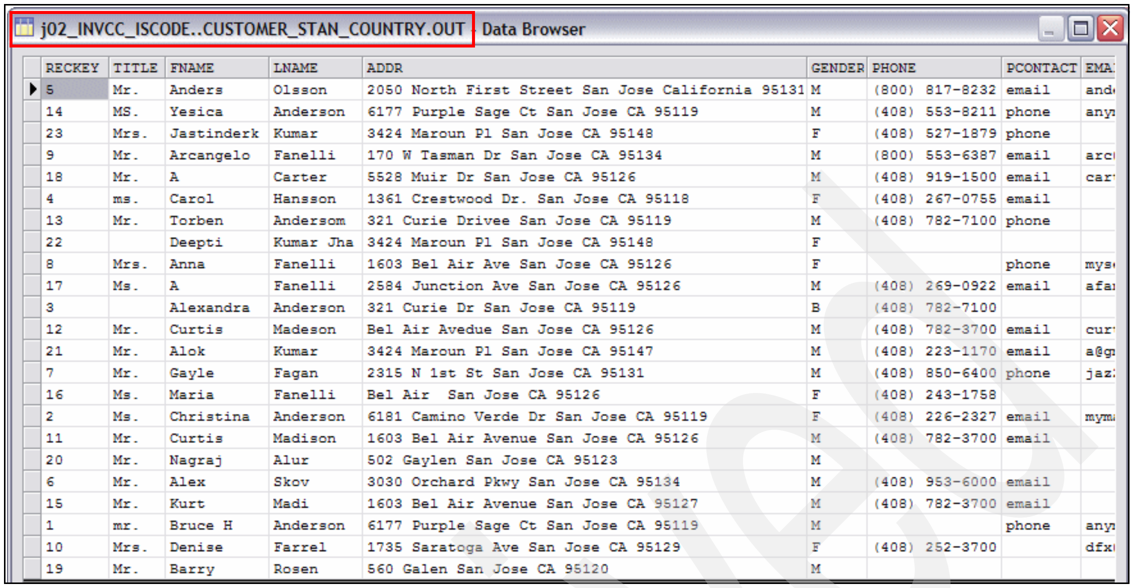


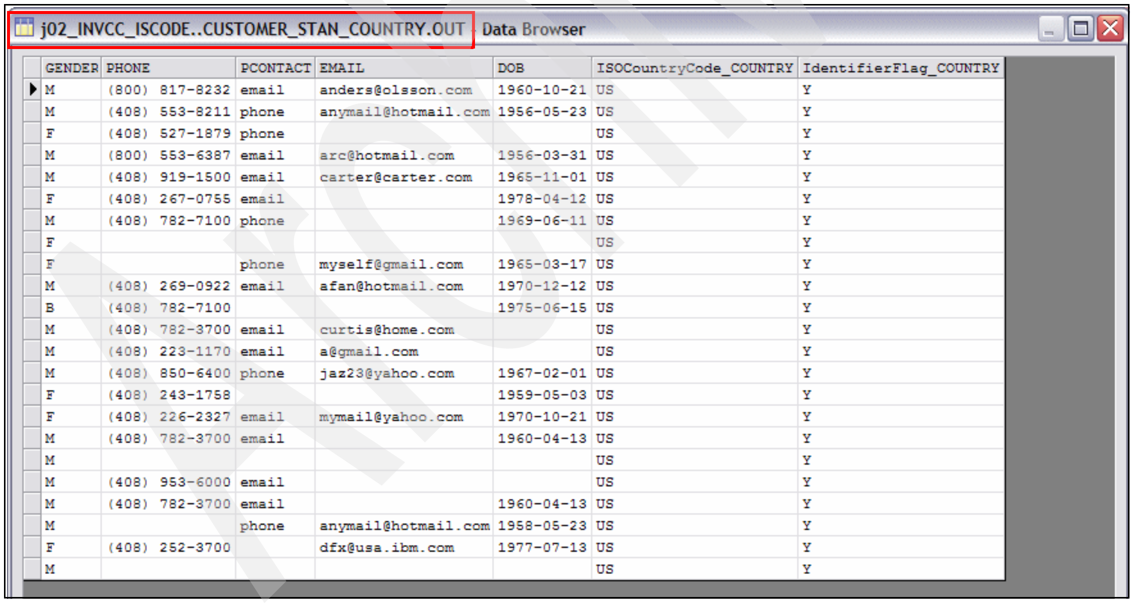
Figure 1-178 Create J02_INVCC_CODE 11/14



j02_INVCC_ISCODE..CUSTOMER_STAN_COUNTRY.OUT - Data Browser

RECKEY	TITLE	FNAME	LNAME	ADDR	GENDER	PHONE	PCONTACT	EMA
5	Mr.	Anders	Olsson	2050 North First Street San Jose California 95131	M	(800) 817-8232	email	and
14	MS.	Yesica	Anderson	6177 Purple Sage Ct San Jose CA 95119	M	(408) 553-8211	phone	any
23	Mrs.	Jastinderk	Kumar	3424 Maroun Pl San Jose CA 95148	F	(408) 527-1879	phone	
9	Mr.	Arcangelo	Fanelli	170 W Tasman Dr San Jose CA 95134	M	(800) 553-6387	email	arc
18	Mr.	A	Carter	5528 Muir Dr San Jose CA 95126	M	(408) 919-1500	email	car
4	ms.	Carol	Hansson	1361 Crestwood Dr. San Jose CA 95118	F	(408) 267-0755	email	
13	Mr.	Torben	Andersom	321 Curie Drivee San Jose CA 95119	M	(408) 782-7100	phone	
22		Deepti	Kumar Jha	3424 Maroun Pl San Jose CA 95148	F			
8	Mrs.	Anna	Fanelli	1603 Bel Air Ave San Jose CA 95126	F		phone	mys
17	Ms.	A	Fanelli	2584 Junction Ave San Jose CA 95126	M	(408) 269-0922	email	afa
3		Alexandra	Anderson	321 Curie Dr San Jose CA 95119	B	(408) 782-7100		
12	Mr.	Curtis	Madeson	Bel Air Avedue San Jose CA 95126	M	(408) 782-3700	email	cur
21	Mr.	Alok	Kumar	3424 Maroun Pl San Jose CA 95147	M	(408) 223-1170	email	a@g
7	Mr.	Gayle	Fagan	2315 N 1st St San Jose CA 95131	M	(408) 850-6400	phone	jaz
16	Ms.	Maria	Fanelli	Bel Air San Jose CA 95126	F	(408) 243-1758		
2	Ms.	Christina	Anderson	6181 Camino Verde Dr San Jose CA 95119	F	(408) 226-2327	email	mym
11	Mr.	Curtis	Madison	1603 Bel Air Avenue San Jose CA 95126	M	(408) 782-3700	email	
20	Mr.	Nagraj	Alur	502 Gaylen San Jose CA 95123	M			
6	Mr.	Alex	Skov	3030 Orchard Pkwy San Jose CA 95134	M	(408) 953-6000	email	
15	Mr.	Kurt	Madi	1603 Bel Air Avenue San Jose CA 95127	M	(408) 782-3700	email	
1	mr.	Bruce H	Anderson	6177 Purple Sage Ct San Jose CA 95119	M		phone	any
10	Mrs.	Denise	Farrel	1735 Saratoga Ave San Jose CA 95129	F	(408) 252-3700		dfx
19	Mr.	Barry	Rosen	560 Galen San Jose CA 95120	M			

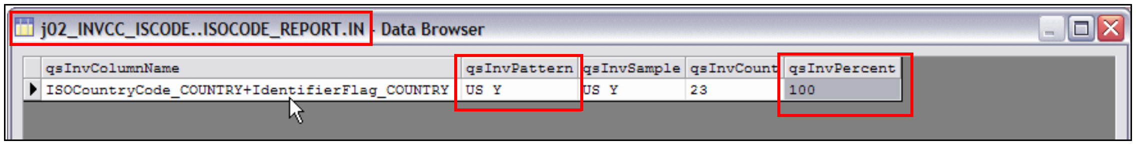
Figure 1-179 Create J02_INVCC_CODE 12/14



j02_INVCC_ISCODE..CUSTOMER_STAN_COUNTRY.OUT - Data Browser

GENDER	PHONE	PCONTACT	EMAIL	DOB	ISOCountryCode_COUNTRY	IdentifierFlag_COUNTRY
M	(800) 817-8232	email	anders@olsson.com	1960-10-21	US	Y
M	(408) 553-8211	phone	anymail@hotmail.com	1956-05-23	US	Y
F	(408) 527-1879	phone			US	Y
M	(800) 553-6387	email	arc@hotmail.com	1956-03-31	US	Y
M	(408) 919-1500	email	carter@carter.com	1965-11-01	US	Y
F	(408) 267-0755	email		1978-04-12	US	Y
M	(408) 782-7100	phone		1969-06-11	US	Y
F					US	Y
F		phone	myself@gmail.com	1965-03-17	US	Y
M	(408) 269-0922	email	afan@hotmail.com	1970-12-12	US	Y
B	(408) 782-7100			1975-06-15	US	Y
M	(408) 782-3700	email	curtis@home.com		US	Y
M	(408) 223-1170	email	a@gmail.com		US	Y
M	(408) 850-6400	phone	jaz23@yahoo.com	1967-02-01	US	Y
F	(408) 243-1758			1959-05-03	US	Y
F	(408) 226-2327	email	mymail@yahoo.com	1970-10-21	US	Y
M	(408) 782-3700	email		1960-04-13	US	Y
M					US	Y
M	(408) 953-6000	email			US	Y
M	(408) 782-3700	email		1960-04-13	US	Y
M		phone	anymail@hotmail.com	1958-05-23	US	Y
F	(408) 252-3700		dfx@usa.ibm.com	1977-07-13	US	Y
M					US	Y

Figure 1-180 Create J02_INVCC_CODE 13/14



qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
ISOCountryCode_COUNTRY+IdentifierFlag_COUNTRY	US Y	US Y	23	100

Figure 1-181 Create J02_INVCC_CODE 14/14

J03_STAN_USPREP

After verifying that all the addresses are U.S. addresses, we use the Standardize stage with the domain-preprocessor rule set USPREP to move name and address data in the credit card customer file into Name, Address, and Area domains.

In this job, the output of the "J01_STAN_COUNTRY" on page 168 job is standardized by the Standardize stage using the domain preprocessor rule set USPREP, because all the addresses are U.S. addresses.

Note: If addresses from other countries existed in the credit card customer file, we would have split the file into as many sub-files as the number of country addresses that were detected and then processed each file separately using the appropriate country's domain preprocessor rule set if available.

Figure 1-182 on page 198 through Figure 1-192 on page 203 describe the steps using Designer Client to build and execute the DataStage job to perform this task.

The steps are as follows:

1. Figure 1-182 on page 198 shows the various stages that are used in this job, including the data set that was created in "J01_STAN_COUNTRY" on page 168, a Standardize stage and an output Data Set stage. We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object that was created earlier.
2. Configure the input STAN_COUNTRY data set with the table definition CUSTOMER_STAN_CNTRY that was saved in "J01_STAN_COUNTRY" on page 168. We do not repeat this process here because it is similar to the steps shown in Figure 1-138 on page 172 through Figure 1-142 on page 174.
3. Next, configure the Standardize stage with the domain preprocessor USPREP rule set as follows:
   a. Right-click the **USPREP_STAN** Standardize stage object and select **Properties** as shown in Figure 1-183 on page 198.
   b. In the Standardize Stage window, click **New Process** to open the Standardize Rule Process window. Then, click the ... button to browse

the available rule sets. We do not repeat this process here because we described it earlier in Figure 1-144 on page 175 through Figure 1-147 on page 176.

c. In the Rule Sets window, expand the Standardization Rules folder, scroll down, and select the USPREP folder and USPREP (rule set) under the USA folder as shown in Figure 1-184 on page 199. Right-click **USPREP** (rule set) and select **Provision All**.

   We do not show the progress and successful completion of the provisioning here, but it is similar to that shown in Figure 1-149 on page 177 and Figure 1-150 on page 177.

d. In the Standardize Rule Process window, the Rule Set shows USPREP.SET. Insert literals ZQNAMEZQ[19] and ZQMIXAZQ[20] interspersed by columns TITLE, FNAME, LNAME, and ADDR as shown in Figure 1-185 on page 199 and Figure 1-186 on page 200. Click **OK**.

e. Click **Stage Properties** in the USPREP_STAN - Standardize Stage window to map the Standardize stage output columns. Figure 1-187 on page 200 shows a partial list of the columns that are mapped.

4. Finally, configure the output STAN_USPREP data set object with the column metadata shown in Figure 1-189 on page 201. You can save this set of columns as a table definition by clicking **Save.**

   This set of columns is saved as CUSTOMER_USPREP. We do not show this process here.

5. After saving, compiling, and running this job j03_STAN_USPREP, you view the results as shown in Figure 1-188 on page 201.

6. The content of the input data set object STAN_COUNTRY is the same as shown in Figure 1-179 on page 195 through Figure 1-180 on page 195.

[19] When the ZQNAMEZQ literal is used, field overrides and field modifications are applied. It then checks for common Name patterns. If not found, it looks for Address and Area patterns. If not found, the field is defaulted to Name.

[20] When the ZQMIXAZQ literal is used, field overrides and field modifications are applied. It then looks the field for name, address, and area data (in that order). Any information that is not assigned a domain is defaulted to Address.

7. The content of the STAN_USPREP is shown in Figure 1-190 on page 202 through Figure 1-192 on page 203.

   The report shows the following:

   – Columns NameDomain_USPREP (contains prefix, first name, last name, suffix tokens), AddressDomain_USPREP (contains apartment, street name and street type tokens), and AreaDomain_USPREP (contains state, ZIP code tokens) that were parsed from the input columns.
   – InputPattern_USPREP and OutboundPattern_USPREP columns that contain the patterns generated after processing the name and address columns in the input file.

Now, proceed to “J04_INVW_USPREP” on page 203.

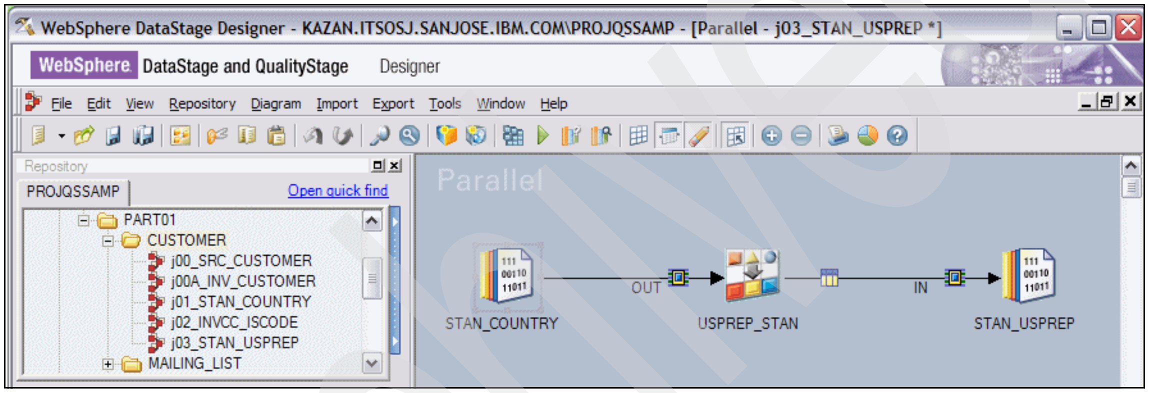

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j03_STAN_USPREP *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Repository
PROJQSSAMP
Open quick find
PART01
CUSTOMER
j00_SRC_CUSTOMER
j00A_INV_CUSTOMER
j01_STAN_COUNTRY
j02_INVCC_ISCODE
j03_STAN_USPREP
MAILING_LIST
Parallel
OUT
IN
STAN_COUNTRY
USPREP_STAN
STAN_USPREP

Figure 1-182 Create J03_STAN_USPREP job 1/11

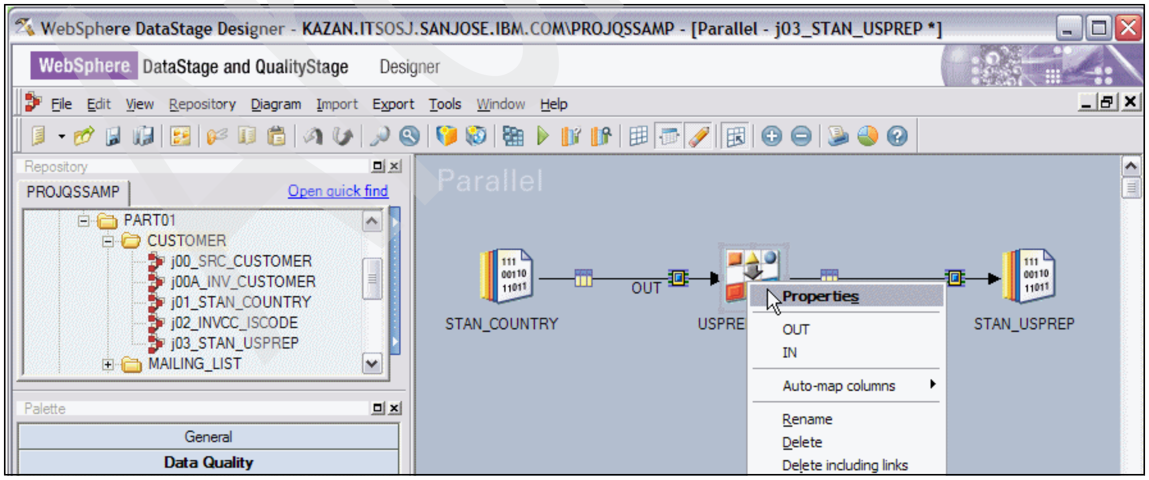


Figure 1-183 Create J03_STAN_USPREP job 2/11

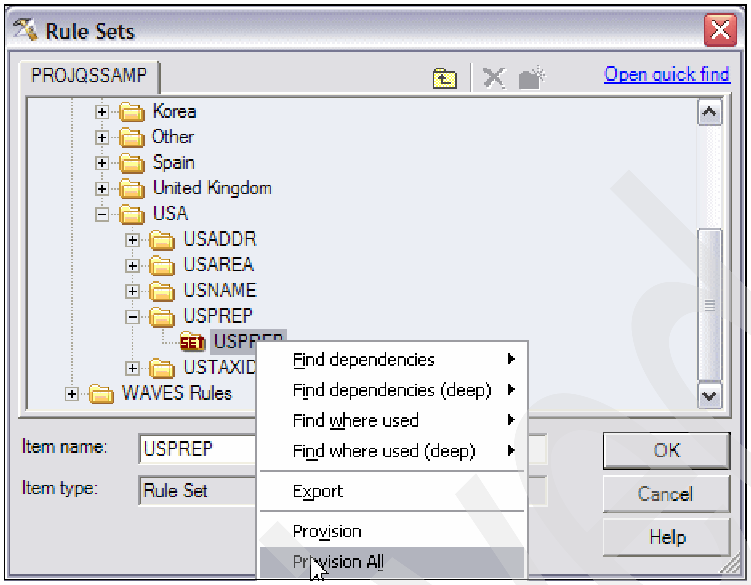


Figure 1-184 Create J03_STAN_USPREP job 3/11

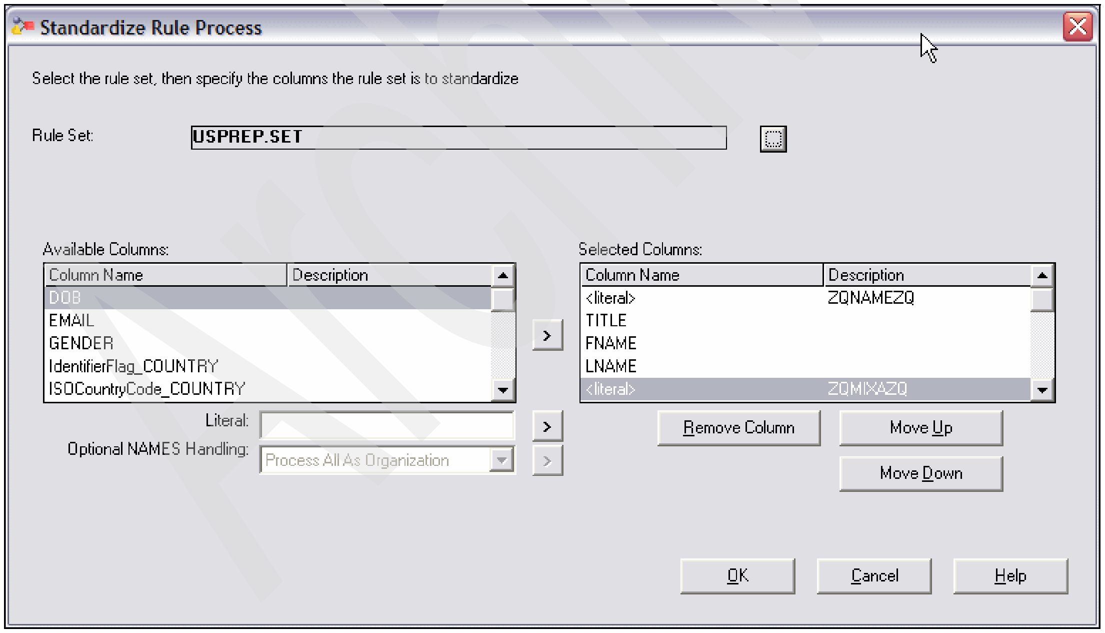
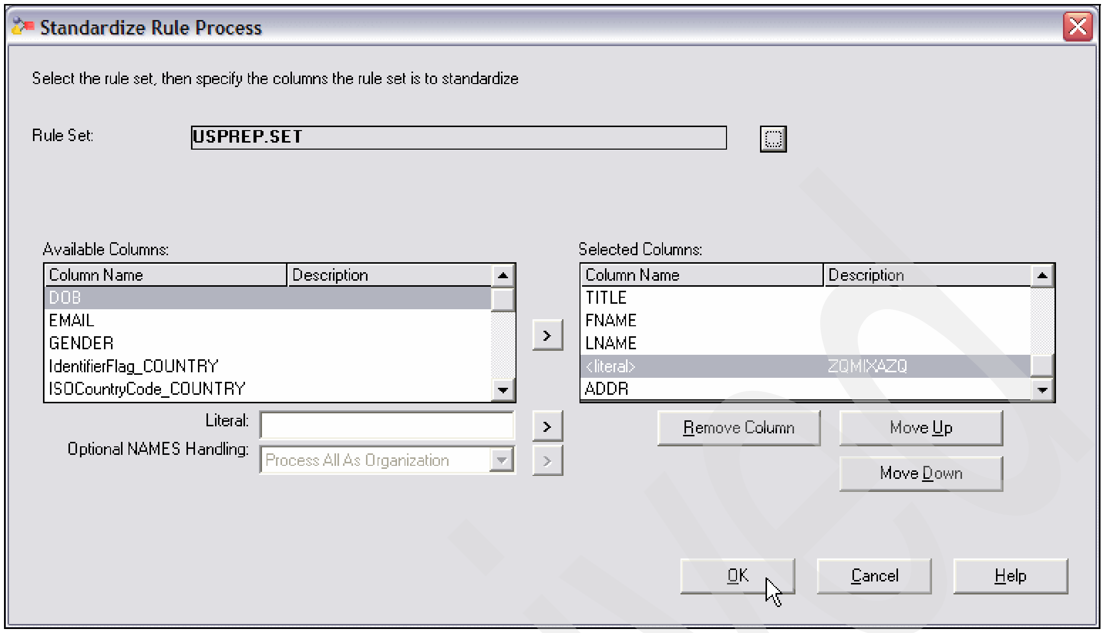


Figure 1-185 Create J03_STAN_USPREP job 4/11


Standardize Rule Process

Select the rule set, then specify the columns the rule set is to standardize

Rule Set: USPREP.SET

Available Columns:

Column Name	Description
DOB	
EMAIL	
GENDER	
IdentifierFlag_COUNTRY	
ISOCountryCode_COUNTRY	

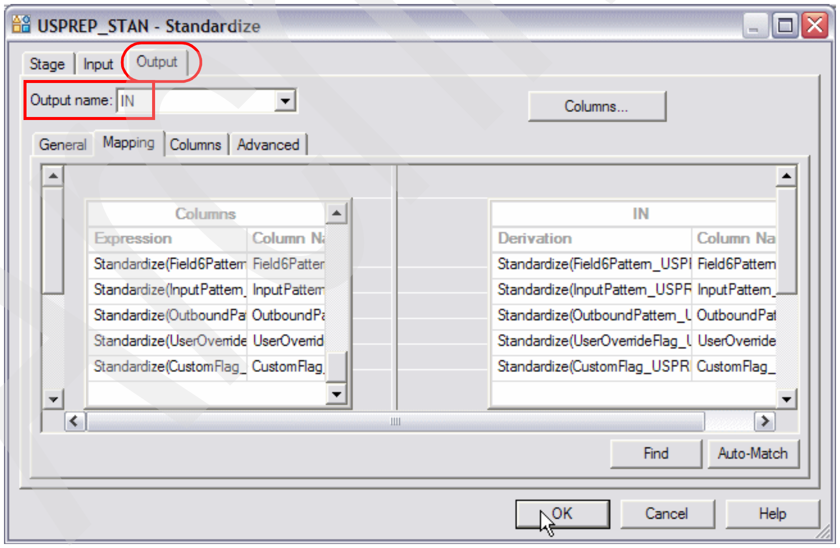
Selected Columns:

Column Name	Description
TITLE	
FNAME	
LNAME	
<literal>	ZQMIXAZQ
ADDR	

Literal:

Optional NAMES Handling: Process All As Organization

Figure 1-186 Create J03_STAN_USPREP job 5/11


USPREP_STAN - Standardize

Output name: IN

Columns

Expression	Column N
Standardize(Field6Pattern	Field6Patter
Standardize(InputPattern_	InputPattern
Standardize(OutboundPa	OutboundP
Standardize(UserOverride	UserOverrid
Standardize(CustomFlag_	CustomFlag

IN

Derivation	Column Na
Standardize(Field6Pattern_USPI	Field6Pattern
Standardize(InputPattern_USPR	InputPattern_
Standardize(OutboundPattern_U	OutboundPat
Standardize(UserOverrideFlag_U	UserOverride
Standardize(CustomFlag_USPR	CustomFlag_

Figure 1-187 Create J03_STAN_USPREP job 6/11

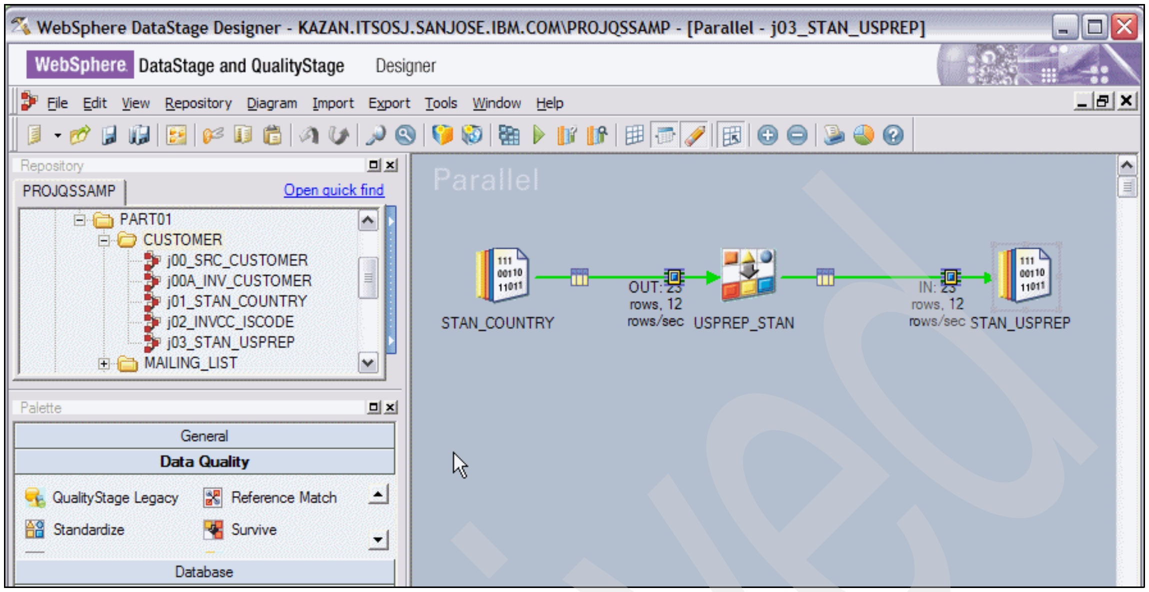
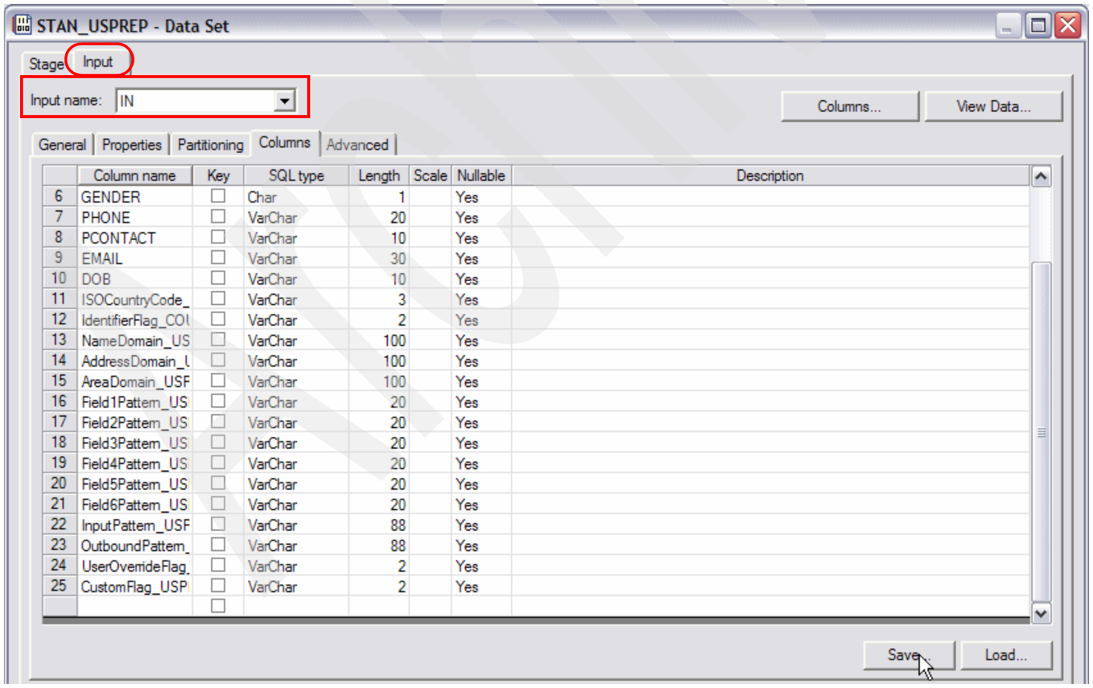


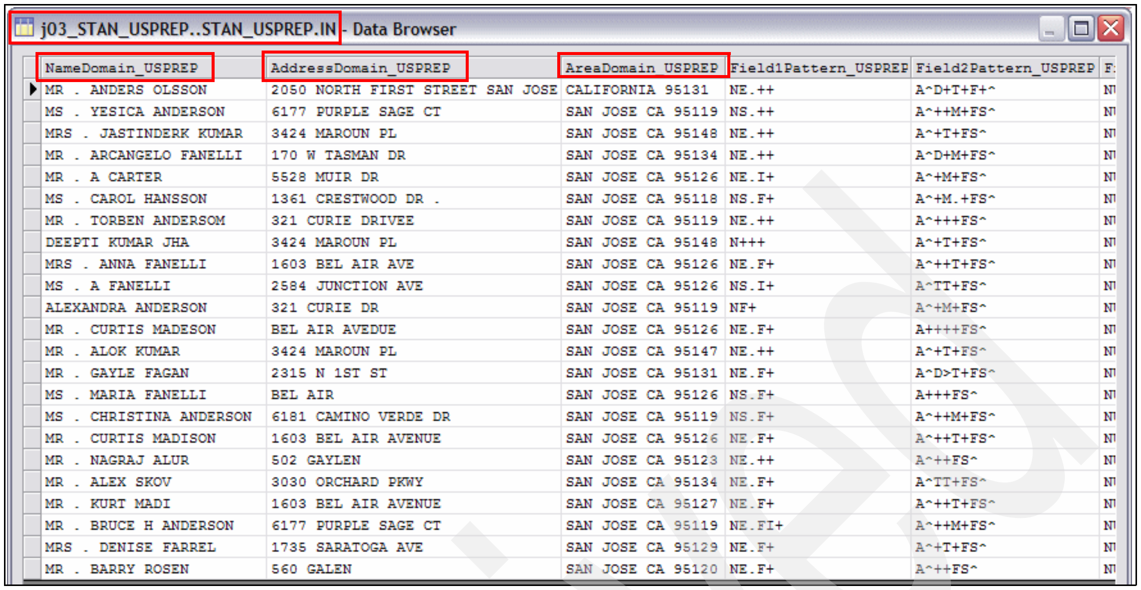
Figure 1-188 Create J03_STAN_USPREP job 7/11



Input name: IN

	Column name	Key	SQL type	Length	Scale	Nullable	Description
6	GENDER	☐	Char	1		Yes	
7	PHONE	☐	VarChar	20		Yes	
8	PCONTACT	☐	VarChar	10		Yes	
9	EMAIL	☐	VarChar	30		Yes	
10	DOB	☐	VarChar	10		Yes	
11	ISOCountryCode_	☐	VarChar	3		Yes	
12	IdentifierFlag_COI	☐	VarChar	2		Yes	
13	NameDomain_US	☐	VarChar	100		Yes	
14	AddressDomain_l	☐	VarChar	100		Yes	
15	AreaDomain_USF	☐	VarChar	100		Yes	
16	Field1Pattern_US	☐	VarChar	20		Yes	
17	Field2Pattern_US	☐	VarChar	20		Yes	
18	Field3Pattern_US	☐	VarChar	20		Yes	
19	Field4Pattern_US	☐	VarChar	20		Yes	
20	Field5Pattern_US	☐	VarChar	20		Yes	
21	Field6Pattern_US	☐	VarChar	20		Yes	
22	InputPattern_USF	☐	VarChar	88		Yes	
23	OutboundPattern_	☐	VarChar	88		Yes	
24	UserOverrideFlag_	☐	VarChar	2		Yes	
25	CustomFlag_USP	☐	VarChar	2		Yes	
		☐					

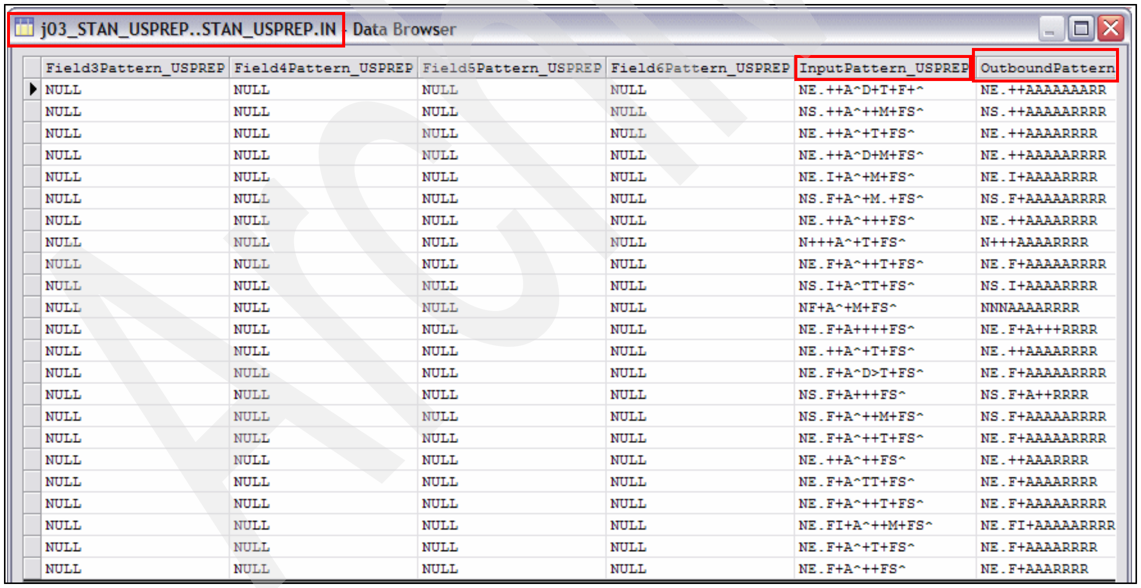
Figure 1-189 Create J03_STAN_USPREP job 8/11



j03_STAN_USPREP..STAN_USPREP.IN - Data Browser

NameDomain_USPREP	AddressDomain_USPREP	AreaDomain_USPREP	Field1Pattern_USPREP	Field2Pattern_USPREP	F
MR . ANDERS OLSSON	2050 NORTH FIRST STREET SAN JOSE	CALIFORNIA 95131	NE.++	A^D+T+F+^	N
MS . YESICA ANDERSON	6177 PURPLE SAGE CT	SAN JOSE CA 95119	NS.++	A^++M+FS^	N
MRS . JASTINDERK KUMAR	3424 MAROUN PL	SAN JOSE CA 95148	NE.++	A^+T+FS^	N
MR . ARCANGELO FANELLI	170 W TASMAN DR	SAN JOSE CA 95134	NE.++	A^D+M+FS^	N
MR . A CARTER	5528 MUIR DR	SAN JOSE CA 95126	NE.I+	A^+M+FS^	N
MS . CAROL HANSSON	1361 CRESTWOOD DR .	SAN JOSE CA 95118	NS.F+	A^+M.+FS^	N
MR . TORBEN ANDERSOM	321 CURIE DRIVEE	SAN JOSE CA 95119	NE.++	A^+++FS^	N
DEEPTI KUMAR JHA	3424 MAROUN PL	SAN JOSE CA 95148	N+++	A^+T+FS^	N
MRS . ANNA FANELLI	1603 BEL AIR AVE	SAN JOSE CA 95126	NE.F+	A^++T+FS^	N
MS . A FANELLI	2584 JUNCTION AVE	SAN JOSE CA 95126	NS.I+	A^TT+FS^	N
ALEXANDRA ANDERSON	321 CURIE DR	SAN JOSE CA 95119	NF+	A^+M+FS^	N
MR . CURTIS MADESON	BEL AIR AVEDUE	SAN JOSE CA 95126	NE.F+	A++++FS^	N
MR . ALOK KUMAR	3424 MAROUN PL	SAN JOSE CA 95147	NE.++	A^+T+FS^	N
MR . GAYLE FAGAN	2315 N 1ST ST	SAN JOSE CA 95131	NE.F+	A^D>T+FS^	N
MS . MARIA FANELLI	BEL AIR	SAN JOSE CA 95126	NS.F+	A+++FS^	N
MS . CHRISTINA ANDERSON	6181 CAMINO VERDE DR	SAN JOSE CA 95119	NS.F+	A^++M+FS^	N
MR . CURTIS MADISON	1603 BEL AIR AVENUE	SAN JOSE CA 95126	NE.F+	A^++T+FS^	N
MR . NAGRAJ ALUR	502 GAYLEN	SAN JOSE CA 95123	NE.++	A^++FS^	N
MR . ALEX SKOV	3030 ORCHARD PKWY	SAN JOSE CA 95134	NE.F+	A^TT+FS^	N
MR . KURT MADI	1603 BEL AIR AVENUE	SAN JOSE CA 95127	NE.F+	A^++T+FS^	N
MR . BRUCE H ANDERSON	6177 PURPLE SAGE CT	SAN JOSE CA 95119	NE.FI+	A^++M+FS^	N
MRS . DENISE FARREL	1735 SARATOGA AVE	SAN JOSE CA 95129	NE.F+	A^+T+FS^	N
MR . BARRY ROSEN	560 GALEN	SAN JOSE CA 95120	NE.F+	A^++FS^	N

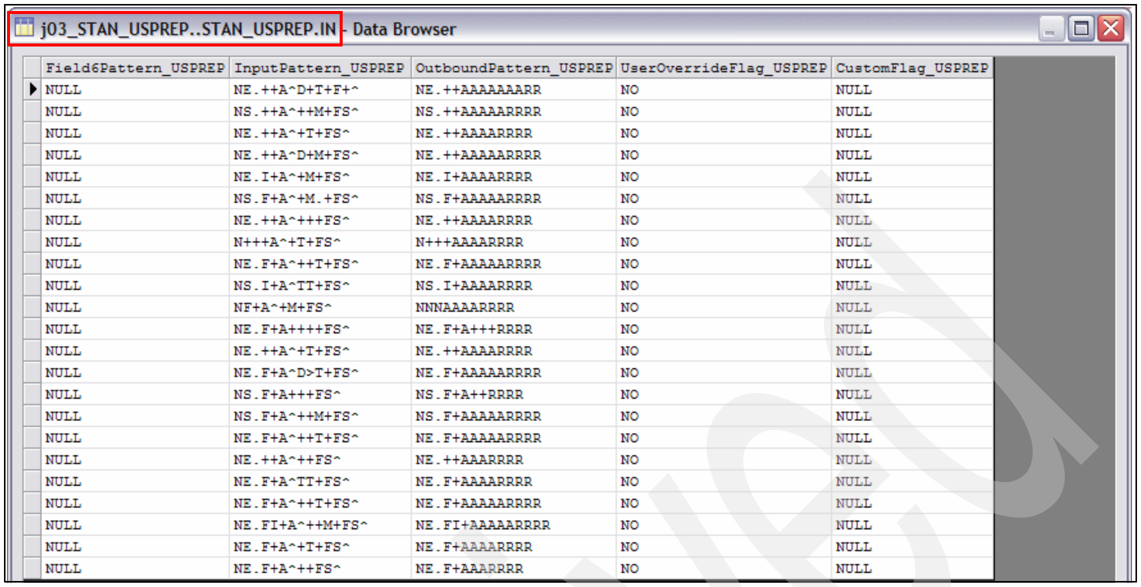
Figure 1-190 Create J03_STAN_USPREP job 9/11



j03_STAN_USPREP..STAN_USPREP.IN - Data Browser

Field3Pattern_USPREP	Field4Pattern_USPREP	Field5Pattern_USPREP	Field6Pattern_USPREP	InputPattern_USPREP	OutboundPattern
NULL	NULL	NULL	NULL	NE.++A^D+T+F+^	NE.++AAAAAAARR
NULL	NULL	NULL	NULL	NS.++A^++M+FS^	NS.++AAAAARRRR
NULL	NULL	NULL	NULL	NE.++A^+T+FS^	NE.++AAAARRRR
NULL	NULL	NULL	NULL	NE.++A^D+M+FS^	NE.++AAAAARRRR
NULL	NULL	NULL	NULL	NE.I+A^+M+FS^	NE.I+AAAARRRR
NULL	NULL	NULL	NULL	NS.F+A^+M.+FS^	NS.F+AAAAARRRR
NULL	NULL	NULL	NULL	NE.++A^+++FS^	NE.++AAAARRRR
NULL	NULL	NULL	NULL	N+++A^+T+FS^	N+++AAAARRRR
NULL	NULL	NULL	NULL	NE.F+A^++T+FS^	NE.F+AAAAARRRR
NULL	NULL	NULL	NULL	NS.I+A^TT+FS^	NS.I+AAAARRRR
NULL	NULL	NULL	NULL	NF+A^+M+FS^	NNNAAAARRRR
NULL	NULL	NULL	NULL	NE.F+A++++FS^	NE.F+A+++RRRR
NULL	NULL	NULL	NULL	NE.++A^+T+FS^	NE.++AAAARRRR
NULL	NULL	NULL	NULL	NE.F+A^D>T+FS^	NE.F+AAAAARRRR
NULL	NULL	NULL	NULL	NS.F+A+++FS^	NS.F+A++RRRR
NULL	NULL	NULL	NULL	NS.F+A^++M+FS^	NS.F+AAAAARRRR
NULL	NULL	NULL	NULL	NE.F+A^++T+FS^	NE.F+AAAAARRRR
NULL	NULL	NULL	NULL	NE.++A^++FS^	NE.++AAARRRR
NULL	NULL	NULL	NULL	NE.F+A^TT+FS^	NE.F+AAAARRRR
NULL	NULL	NULL	NULL	NE.F+A^++T+FS^	NE.F+AAAAARRRR
NULL	NULL	NULL	NULL	NE.FI+A^++M+FS^	NE.FI+AAAAARRRR
NULL	NULL	NULL	NULL	NE.F+A^+T+FS^	NE.F+AAAARRRR
NULL	NULL	NULL	NULL	NE.F+A^++FS^	NE.F+AAARRRR

Figure 1-191 Create J03_STAN_USPREP job 10/11


j03_STAN_USPREP..STAN_USPREP.IN - Data Browser

Field6Pattern_USPREP	InputPattern_USPREP	OutboundPattern_USPREP	UserOverrideFlag_USPREP	CustomFlag_USPREP
NULL	NE.++A^D+T+F+^	NE.++AAAAAAARR	NO	NULL
NULL	NS.++A^++M+FS^	NS.++AAAAARRRR	NO	NULL
NULL	NE.++A^+T+FS^	NE.++AAAARRRR	NO	NULL
NULL	NE.++A^D+M+FS^	NE.++AAAAARRRR	NO	NULL
NULL	NE.I+A^+M+FS^	NE.I+AAAARRRR	NO	NULL
NULL	NS.F+A^+M.+FS^	NS.F+AAAAARRRR	NO	NULL
NULL	NE.++A^+++FS^	NE.++AAAARRRR	NO	NULL
NULL	N+++A^+T+FS^	N+++AAAARRRR	NO	NULL
NULL	NE.F+A^++T+FS^	NE.F+AAAAARRRR	NO	NULL
NULL	NS.I+A^TT+FS^	NS.I+AAAARRRR	NO	NULL
NULL	NF+A^+M+FS^	NNNAAAARRRR	NO	NULL
NULL	NE.F+A++++FS^	NE.F+A+++RRRR	NO	NULL
NULL	NE.++A^+T+FS^	NE.++AAAARRRR	NO	NULL
NULL	NE.F+A^D>T+FS^	NE.F+AAAAARRRR	NO	NULL
NULL	NS.F+A+++FS^	NS.F+A++RRRR	NO	NULL
NULL	NS.F+A^++M+FS^	NS.F+AAAAARRRR	NO	NULL
NULL	NE.F+A^++T+FS^	NE.F+AAAAARRRR	NO	NULL
NULL	NE.++A^++FS^	NE.++AAARRRR	NO	NULL
NULL	NE.F+A^TT+FS^	NE.F+AAAARRRR	NO	NULL
NULL	NE.F+A^++T+FS^	NE.F+AAAAARRRR	NO	NULL
NULL	NE.FI+A^++M+FS^	NE.FI+AAAAARRRR	NO	NULL
NULL	NE.F+A^+T+FS^	NE.F+AAAARRRR	NO	NULL
NULL	NE.F+A^++FS^	NE.F+AAARRRR	NO	NULL

Figure 1-192 Create J03_STAN_USPREP job 11/11

J04_INVW_USPREP

In this job, we use the Investigate stage using word investigate on the USPREP generated Name, Address, and Area domains to determine whether the domain-preprocessor USPREP rule set successfully parsed the tokens in the name and address fields into the correct domains.

We analyze the credit card customer file containing the name, address, and area buckets that are generated by the “J03_STAN_USPREP” on page 196 job by the Investigate stage using word investigate and the domain-specific USNAME, USADDR and USAREA rule sets to determine the degree of success that is achieved by the Standardize stage in moving the tokens to the right domain buckets.

Because a single Investigate stage can only have a single rule set associated with it, the credit card customer file needs to be split (using a Copy stage) and processed by three independent Investigate stages, with each stage using a particular domain-specific rule set. Both the pattern and token reports are generated in each Investigate stage.

Figure 1-193 on page 206 through Figure 1-210 on page 214 describe the steps using Designer Client to build and execute the DataStage job to perform this task.

The steps are as follows:

1. Figure 1-193 on page 206 shows the various stages that are used in this job, including the data set that was created in “J03_STAN_USPREP” on page 196, a Copy stage, three Investigate stages that each use a different domain-specific rule set, and two sequential file stages (one each for the token report and pattern report) for each Investigate stage. The names of the stages are modified as shown, and this job uses the same QSPARAMETERSET object that was created earlier.
2. Configure the input STAN_USPREP data set with the table definition CUSTOMER_USPREP saved in “J03_STAN_USPREP” on page 196. We do not repeat this process here because it is similar to the steps shown in Figure 1-138 on page 172 through Figure 1-142 on page 174.
3. The Copy stage duplicates the source data and sends it to the three Investigate stages. The Copy stage also duplicates the metadata and sends the output metadata to the three Investigate stages.

   Configure the Copy stage as follows:

   a. Right-click the **COPY** stage icon and select **Properties** as shown in Figure 1-194 on page 206.

   b. In the COPY - Copy window, the Input tab and Columns tab shows the metadata that is propagated by the COPY stage. We do not show this process here.

   c. In the same window, click the Output tab and then the Mapping tab. ¨ Mapping. Map the columns that display in the left Columns box to the right box as shown in Figure 1-195 on page 207. This applies to the link NAME in the Output name field.

      Selecting the correct output link assures that the data goes to the correct Investigate stage.

      To copy the data from the Columns pane to the Name pane, simply drag and drop, or complete the following steps:

      i. Place the cursor in the Columns pane, right-click, and choose **Select All**.

      ii. Then, choose **Copy**.

      iii. Place the cursor in the Name pane, right-click, and choose **Paste Column**. The column metadata copies into the Name pane and lines show the columns linking from Columns to Name.

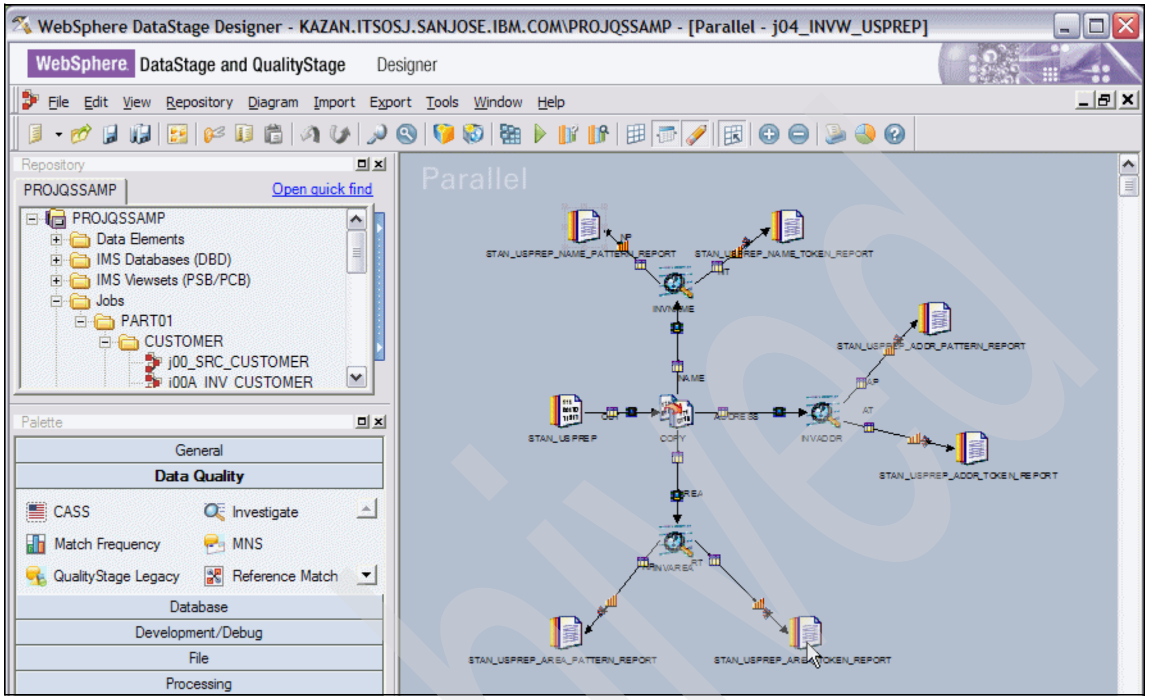
      Repeat the process for Output name fields ADDRESS and AREA.

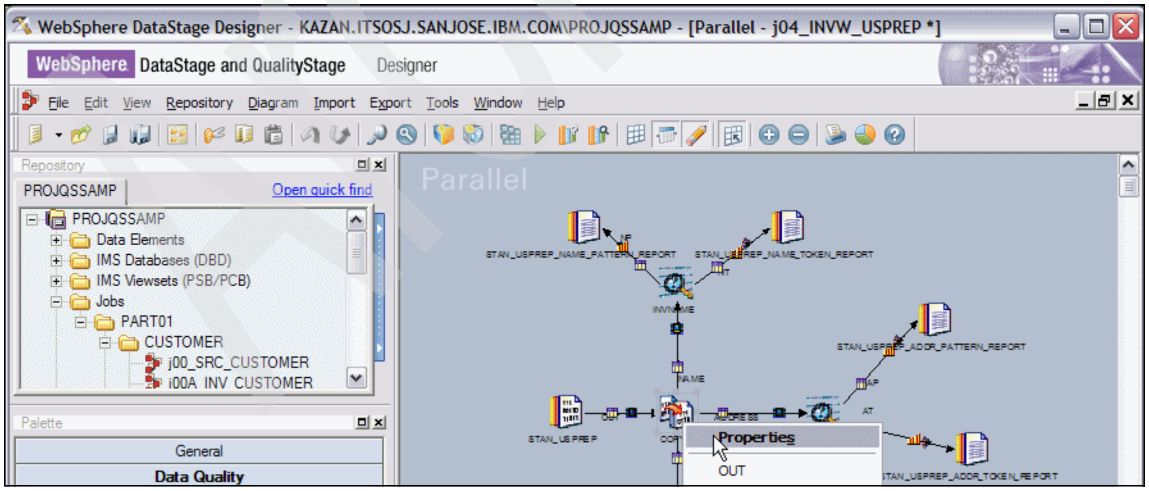
4. Configure the INVNAME Investigate stage as follows:
   a. Right-click that icon and select **Properties** as shown in Figure 1-196 on page 207.
   b. Select the domain-specific USNAME rule set and provision it. We do not repeat this process here because it is similar to the process described in Figure 1-144 on page 175 through Figure 1-151 on page 178.
   c. In the INVNAME - Investigate Stage, click **Word Investigate**. The Rule Set field shows USNAME.SET. The columns that are propagated from the input data set are shown in the Available Columns list. Select NameDomain_USPREP and move it to the Standard Columns pane using the > button as shown in Figure 1-197 on page 208. The Pattern Report and Token Report boxes are selected in the Output Dataset field.
   d. Click **Stage Properties** in Figure 1-197 on page 208 and when the INVNAME - Investigate windows opens, select Mapping in the Output tab—the Output name is *NP*. Map all the columns in the Columns list to the NP list as shown in Figure 1-198 on page 208. This corresponds to the pattern report. Click **OK**.

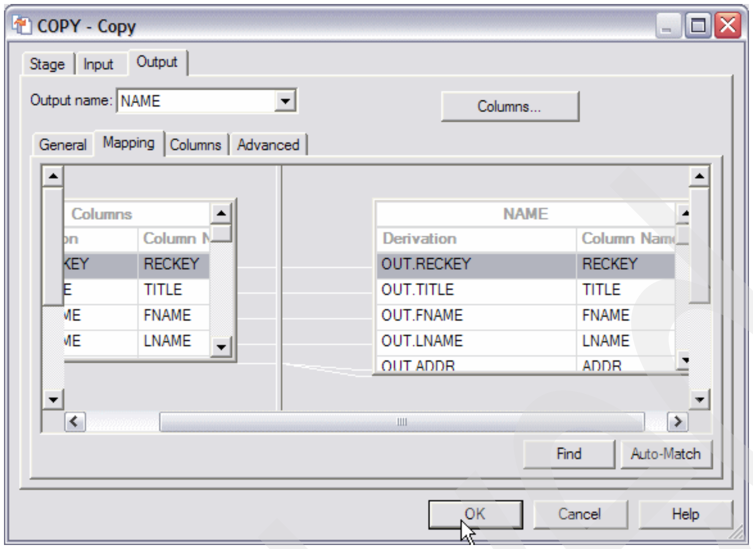
      Attention: Ensure that the link ordering is set correctly so that the data is directed to the correct output path.

   e. Repeat the process for the Output name NT as shown in Figure 1-199 on page 209. This corresponds to the token report.
   f. Configure the sequential files for the pattern (Figure 1-200 on page 209) and token (Figure 1-201 on page 210) reports.
5. Repeat the process for configuring the INVADDR Investigate stage. The AddressDomain_USPREP column is analyzed with the domain-specific USADDR rule set as shown in Figure 1-202 on page 210.
6. Repeat the process for configuring the INVAREA Investigate stage. The AreaDomain_USPREP column is analyzed with the domain-specific USAREA rule set as shown in Figure 1-203 on page 211.
7. After saving, compiling, and running this job, the contents of the output of each investigate stage are listed here:
   a. The contents of the data input to the Investigate stage are shown in Figure 1-179 on page 195 through Figure 1-180 on page 195.
   b. The outputs of the Investigate stage that are written to the sequential file are shown in Figure 1-204 on page 211 through Figure 1-210 on page 214.

After analyzing the results of this job, proceed to “J03_Z_Override_And_After” on page 215.


WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j04_INVW_USPREP]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Repository
PROJQSSAMP
Open quick find
PROJQSSAMP
Data Elements
IMS Databases (DBD)
IMS Viewsets (PSB/PCB)
Jobs
PART01
CUSTOMER
j00_SRC_CUSTOMER
i00A INV CUSTOMER
Palette
General
Data Quality
CASS
Investigate
Match Frequency
MNS
QualityStage Legacy
Reference Match
Database
Development/Debug
File
Processing
Parallel
STAN_USPREP_NAME_PATTERN_REPORT
STAN_USPREP_NAME_TOKEN_REPORT
INVNAME
NAME
STAN_USPREP_ADDR_PATTERN_REPORT
STAN_USPREP
COPY
ADDRESS
INVADDR
STAN_USPREP_ADDR_TOKEN_REPORT
AREA
INVAREA
STAN_USPREP_AREA_PATTERN_REPORT
STAN_USPREP_AREA_TOKEN_REPORT
Figure 1-193 Create J04_INVW_USPREP job 1/18


WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j04_INVW_USPREP *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Repository
PROJQSSAMP
Open quick find
PROJQSSAMP
Data Elements
IMS Databases (DBD)
IMS Viewsets (PSB/PCB)
Jobs
PART01
CUSTOMER
j00_SRC_CUSTOMER
i00A INV CUSTOMER
Palette
General
Data Quality
Parallel
STAN_USPREP_NAME_PATTERN_REPORT
STAN_USPREP_NAME_TOKEN_REPORT
INVNAME
NAME
STAN_USPREP_ADDR_PATTERN_REPORT
STAN_USPREP
COPY
Properties
OUT
STAN_USPREP_ADDR_TOKEN_REPORT
Figure 1-194 Create J04_INVW_USPREP job 2/18


COPY - Copy

Output name: NAME

Derivation	Column Name
OUT.RECKEY	RECKEY
OUT.TITLE	TITLE
OUT.FNAME	FNAME
OUT.LNAME	LNAME
OUT.ADDR	ADDR

Figure 1-195 Create J04_INVW_USPREP job 3/18

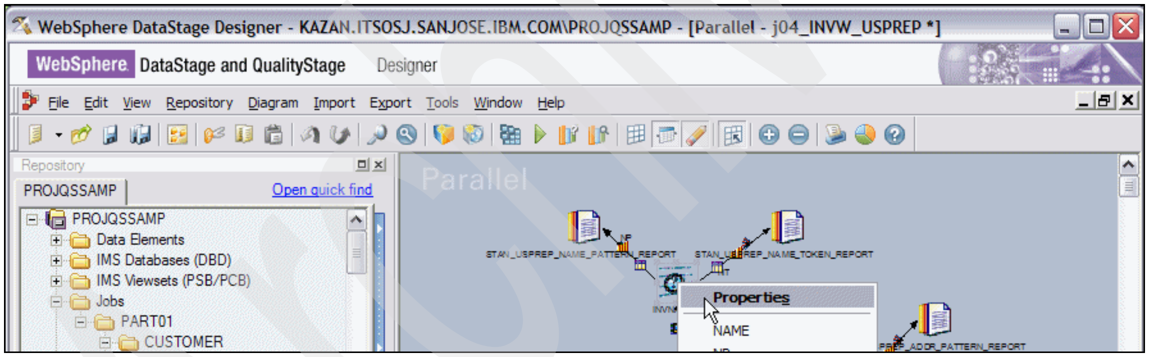
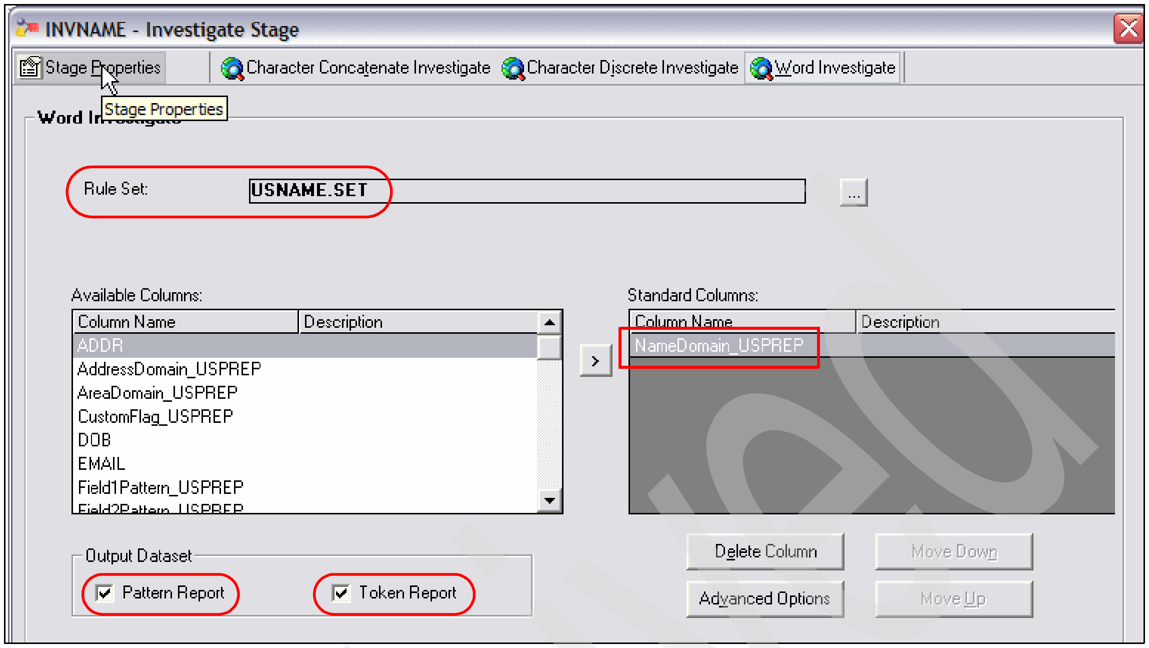

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j04_INVW_USPREP *]

Figure 1-196 Create J04_INVW_USPREP job 4/18


INVNAME - Investigate Stage

Stage Properties | Character Concatenate Investigate | Character Discrete Investigate | Word Investigate

Word Investigate

Rule Set: USNAME.SET

Available Columns:

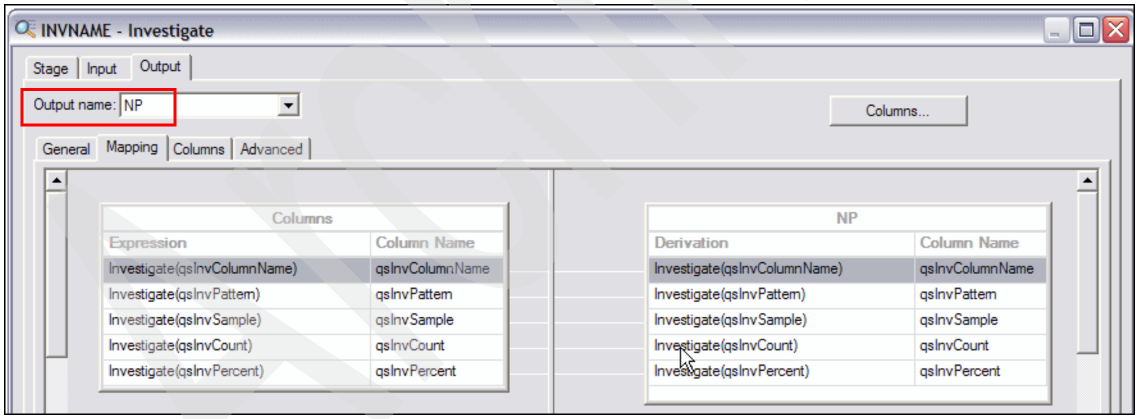
Column Name	Description
ADDR	
AddressDomain_USPREP	
AreaDomain_USPREP	
CustomFlag_USPREP	
DOB	
EMAIL	
Field1Pattern_USPREP	
Field2Pattern_USPREP	

Standard Columns:

Column Name	Description
NameDomain_USPREP	

Output Dataset: [✓ Pattern Report] [✓ Token Report]

Figure 1-197   Create J04_INVW_USPREP job 5/18


INVNAME - Investigate

Stage | Input | Output

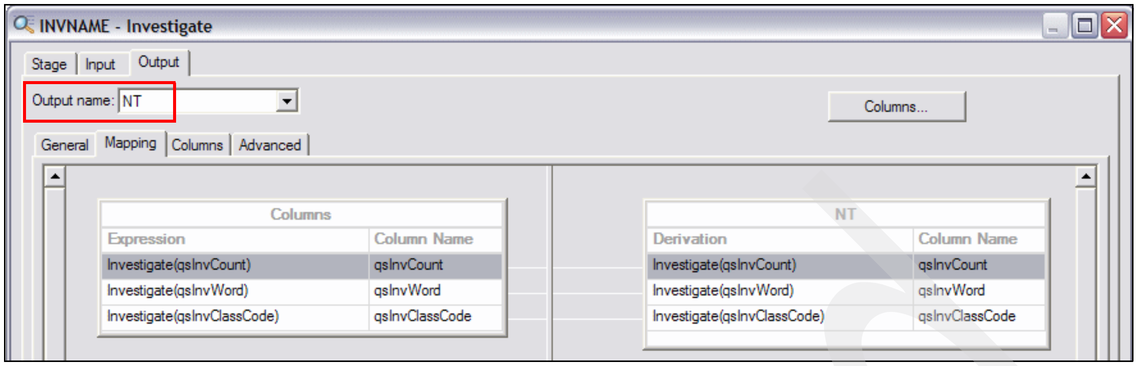
Output name: NP

General | Mapping | Columns | Advanced

Expression	Column Name
Investigate(qsInvColumnName)	qsInvColumnName
Investigate(qsInvPattern)	qsInvPattern
Investigate(qsInvSample)	qsInvSample
Investigate(qsInvCount)	qsInvCount
Investigate(qsInvPercent)	qsInvPercent

Derivation	Column Name
Investigate(qsInvColumnName)	qsInvColumnName
Investigate(qsInvPattern)	qsInvPattern
Investigate(qsInvSample)	qsInvSample
Investigate(qsInvCount)	qsInvCount
Investigate(qsInvPercent)	qsInvPercent

Figure 1-198   Create J04_INVW_USPREP job 6/18


INVNAME - Investigate

Stage | Input | Output

Output name: NT

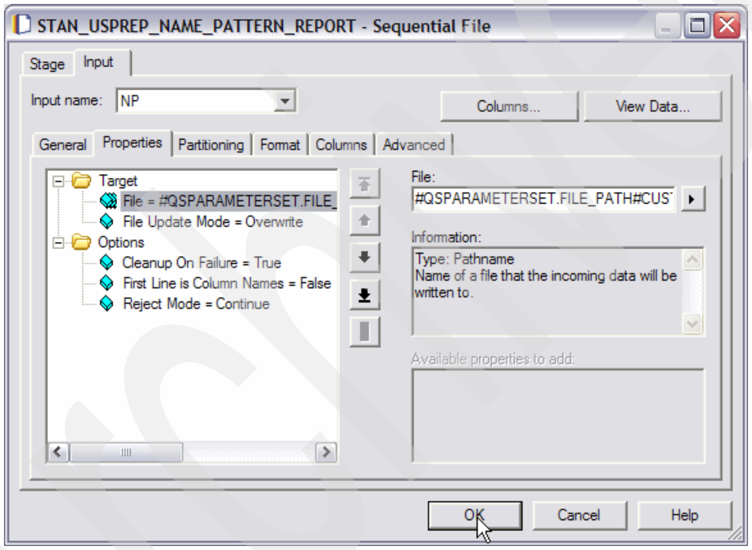
Columns...

General | Mapping | Columns | Advanced

Columns	
Expression	Column Name
Investigate(qsInvCount)	qsInvCount
Investigate(qsInvWord)	qsInvWord
Investigate(qsInvClassCode)	qsInvClassCode

NT	
Derivation	Column Name
Investigate(qsInvCount)	qsInvCount
Investigate(qsInvWord)	qsInvWord
Investigate(qsInvClassCode)	qsInvClassCode

Figure 1-199 Create J04_INVW_USPREP job 7/18


STAN_USPREP_NAME_PATTERN_REPORT - Sequential File

Stage | Input

Input name: NP

Columns... | View Data...

General | Properties | Partitioning | Format | Columns | Advanced

- Target
  - File = #QSPARAMETERSET.FILE_
  - File Update Mode = Overwrite
- Options
  - Cleanup On Failure = True
  - First Line is Column Names = False
  - Reject Mode = Continue

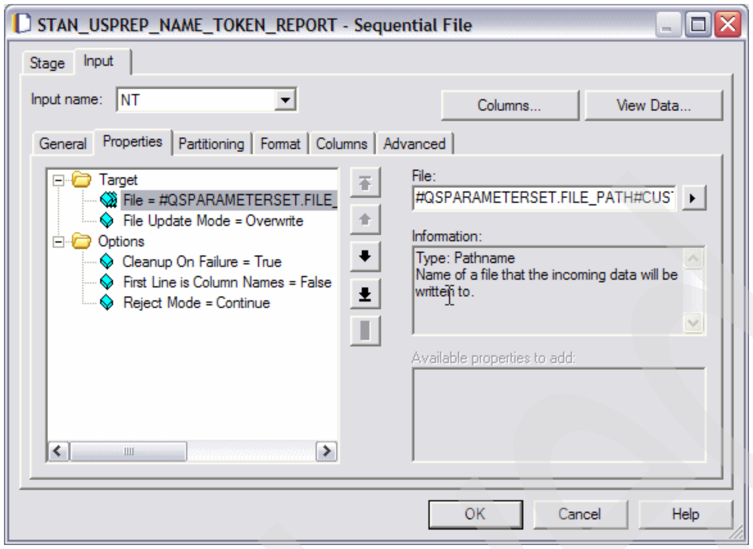
File:
#QSPARAMETERSET.FILE_PATH#CUS

Information:
Type: Pathname
Name of a file that the incoming data will be written to.

Available properties to add:

OK | Cancel | Help

Figure 1-200 Create J04_INVW_USPREP job 8/18


STAN_USPREP_NAME_TOKEN_REPORT - Sequential File

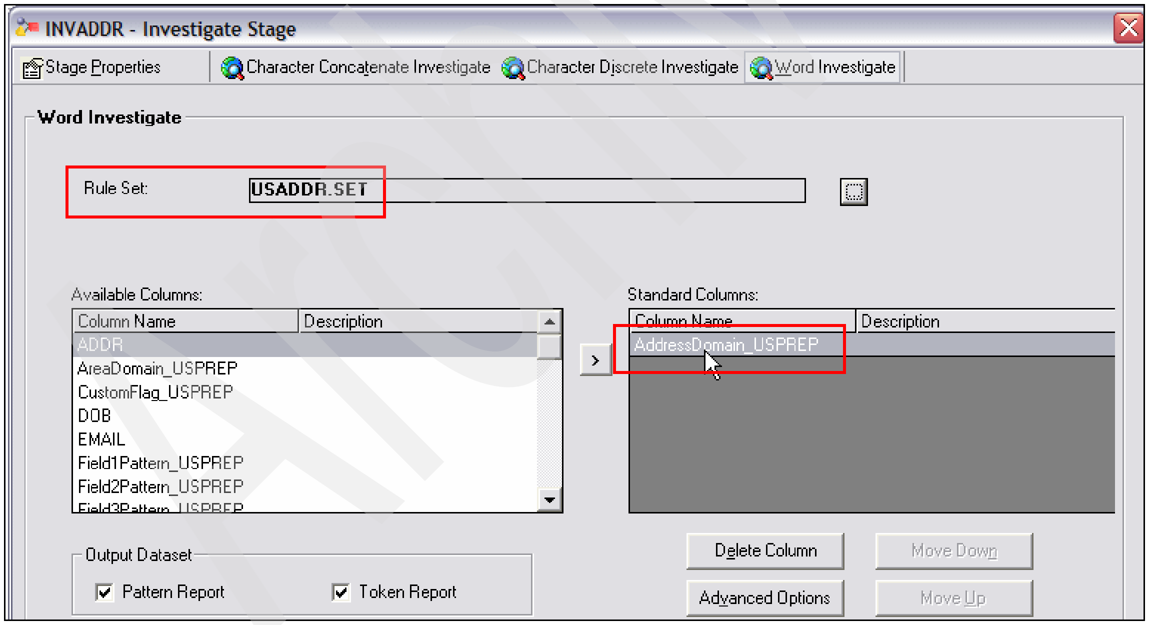
Input name: NT

- Target
  - File = #QSPARAMETERSET.FILE_
  - File Update Mode = Overwrite
- Options
  - Cleanup On Failure = True
  - First Line is Column Names = False
  - Reject Mode = Continue

File: #QSPARAMETERSET.FILE_PATH#CUS

Information:
Type: Pathname
Name of a file that the incoming data will be written to.

Figure 1-201 Create J04_INVW_USPREP job 9/18


INVADDR - Investigate Stage

Word Investigate

Rule Set: USADDR.SET

Available Columns:

Column Name	Description
ADDR	
AreaDomain_USPREP	
CustomFlag_USPREP	
DOB	
EMAIL	
Field1Pattern_USPREP	
Field2Pattern_USPREP	
Field3Pattern_USPREP	

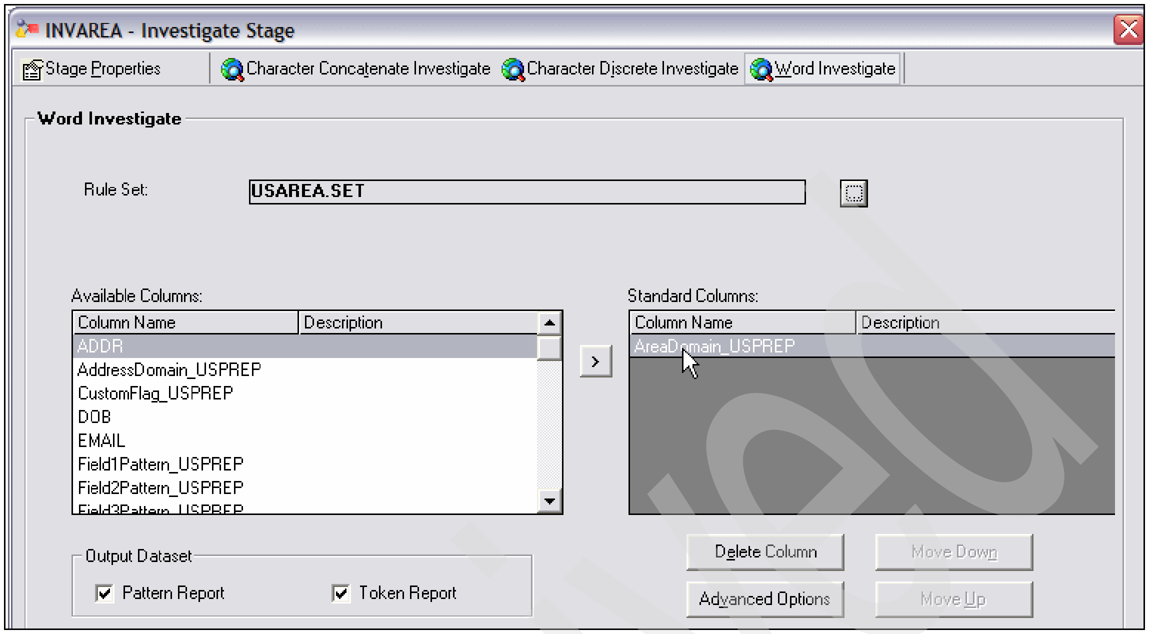
Standard Columns:

Column Name	Description
AddressDomain_USPREP	

Output Dataset

☑ Pattern Report ☑ Token Report

Figure 1-202 Create J04_INVW_USPREP job 10/18


INVAREA - Investigate Stage
Stage Properties
Character Concatenate Investigate
Character Discrete Investigate
Word Investigate
Word Investigate

Rule Set: USAREA.SET

Available Columns:

Column Name	Description
ADDR	
AddressDomain_USPREP	
CustomFlag_USPREP	
DOB	
EMAIL	
Field1Pattern_USPREP	
Field2Pattern_USPREP	
Field3Pattern_USPREP	

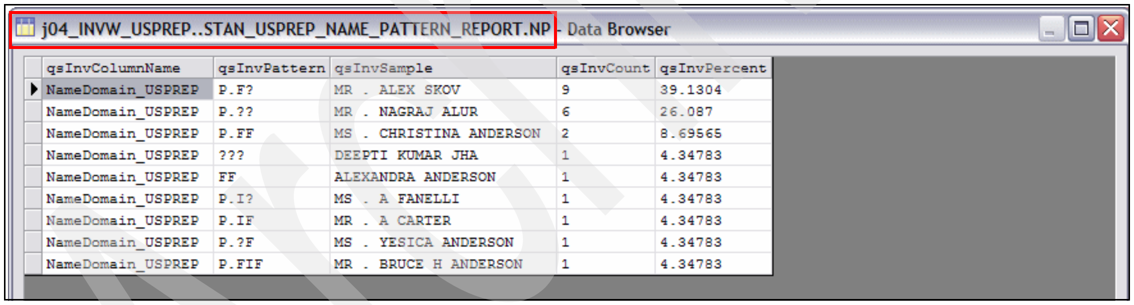
Standard Columns:

Column Name	Description
AreaDomain_USPREP	

Output Dataset

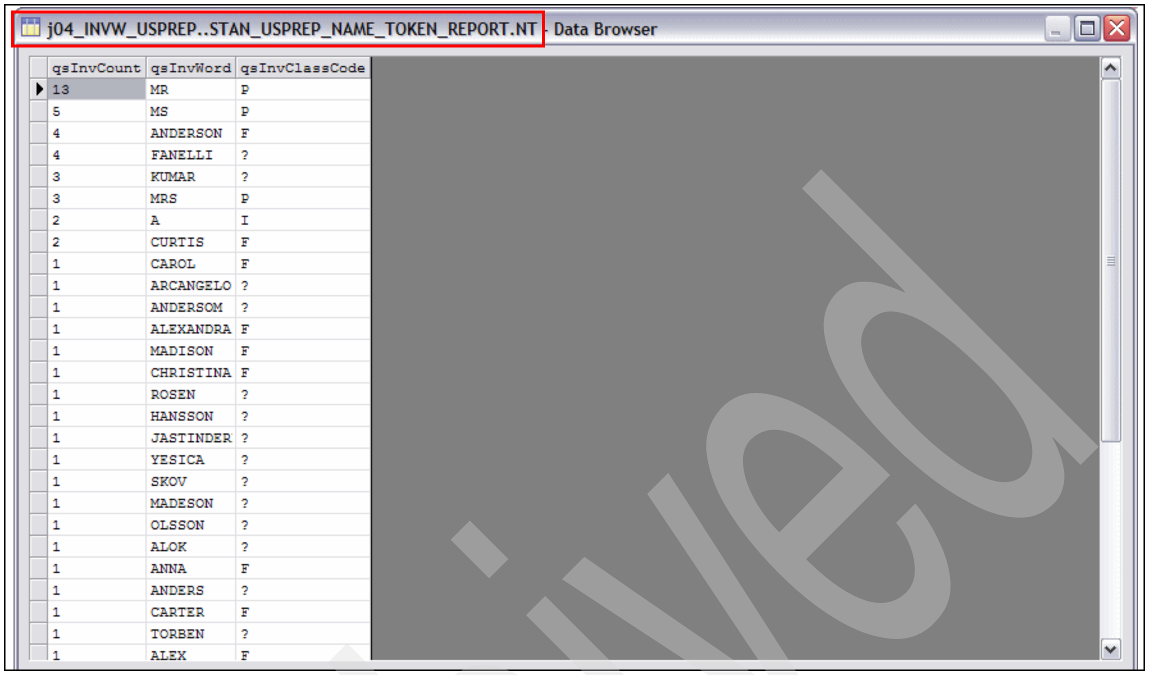
[✓] Pattern Report [✓] Token Report

Figure 1-203 Create J04_INVW_USPREP job 11/18


j04_INVW_USPREP..STAN_USPREP_NAME_PATTERN_REPORT.NP - Data Browser

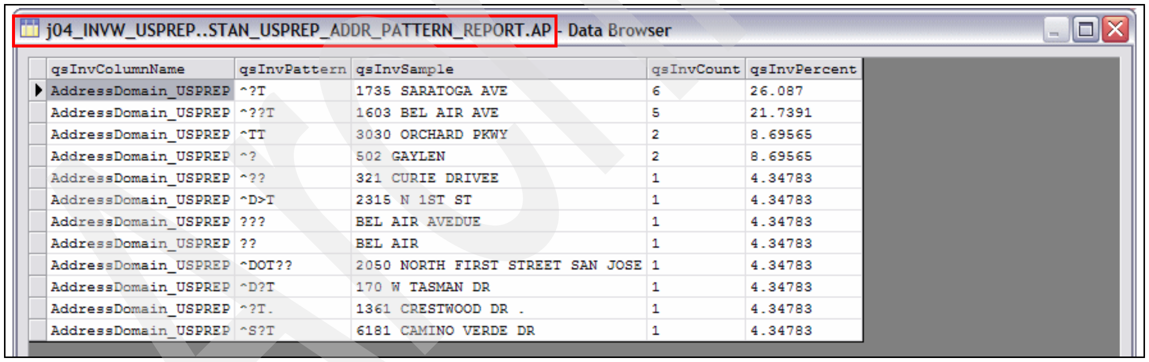
qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
NameDomain_USPREP	P.F?	MR . ALEX SKOV	9	39.1304
NameDomain_USPREP	P.??	MR . NAGRAJ ALUR	6	26.087
NameDomain_USPREP	P.FF	MS . CHRISTINA ANDERSON	2	8.69565
NameDomain_USPREP	???	DEEPTI KUMAR JHA	1	4.34783
NameDomain_USPREP	FF	ALEXANDRA ANDERSON	1	4.34783
NameDomain_USPREP	P.I?	MS . A FANELLI	1	4.34783
NameDomain_USPREP	P.IF	MR . A CARTER	1	4.34783
NameDomain_USPREP	P.?F	MS . YESICA ANDERSON	1	4.34783
NameDomain_USPREP	P.FIF	MR . BRUCE H ANDERSON	1	4.34783

Figure 1-204 Create J04_INVW_USPREP job 12/18


j04_INVW_USPREP..STAN_USPREP_NAME_TOKEN_REPORT.NT - Data Browser

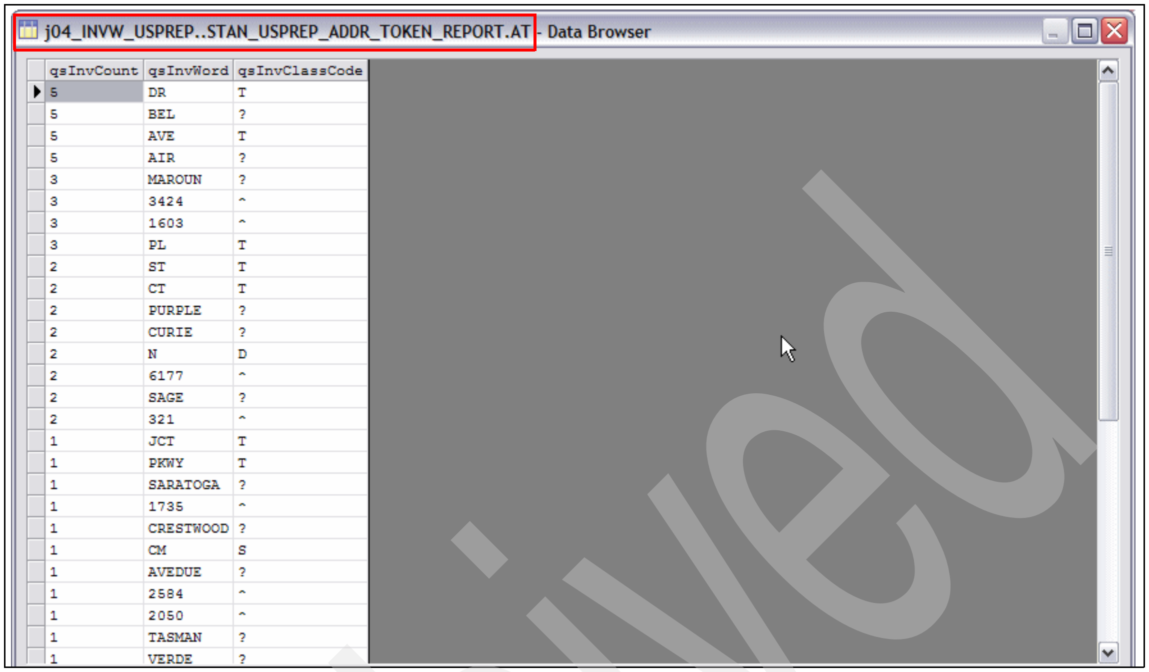
qsInvCount	qsInvWord	qsInvClassCode
13	MR	P
5	MS	P
4	ANDERSON	F
4	FANELLI	?
3	KUMAR	?
3	MRS	P
2	A	I
2	CURTIS	F
1	CAROL	F
1	ARCANGELO	?
1	ANDERSOM	?
1	ALEXANDRA	F
1	MADISON	F
1	CHRISTINA	F
1	ROSEN	?
1	HANSSON	?
1	JASTINDER	?
1	YESICA	?
1	SKOV	?
1	MADESON	?
1	OLSSON	?
1	ALOK	?
1	ANNA	F
1	ANDERS	?
1	CARTER	F
1	TORBEN	?
1	ALEX	F

Figure 1-205 Create J04_INVW_USPREP job 13/18


j04_INVW_USPREP..STAN_USPREP_ADDR_PATTERN_REPORT.AP - Data Browser

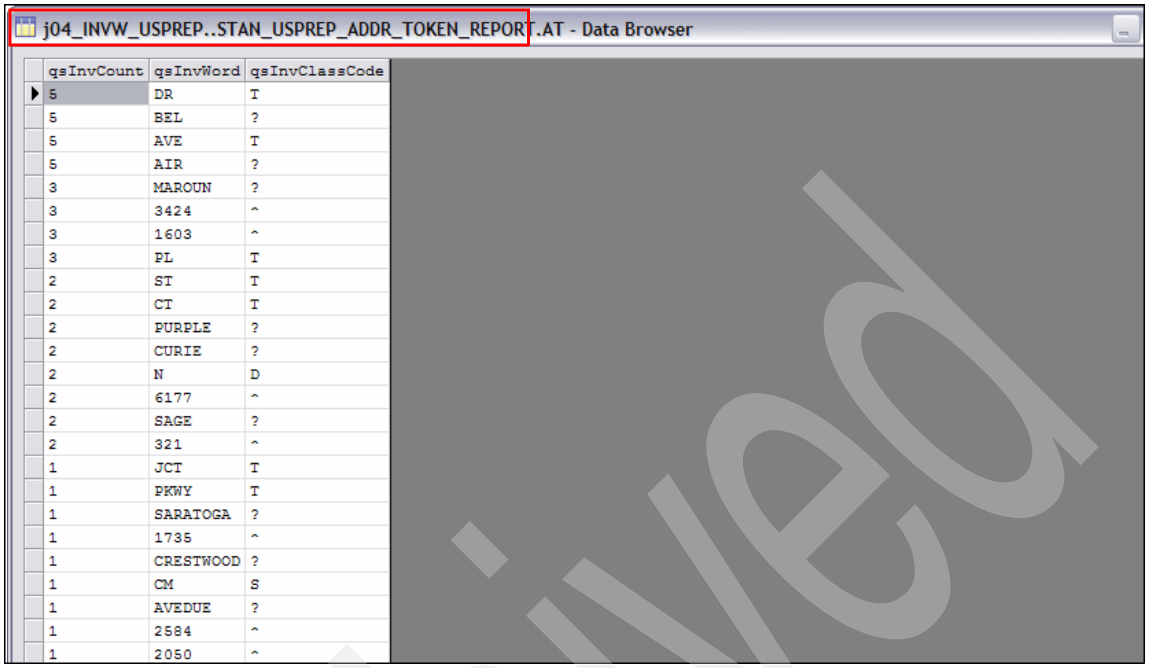
qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
AddressDomain_USPREP	^?T	1735 SARATOGA AVE	6	26.087
AddressDomain_USPREP	^??T	1603 BEL AIR AVE	5	21.7391
AddressDomain_USPREP	^TT	3030 ORCHARD PKWY	2	8.69565
AddressDomain_USPREP	^?	502 GAYLEN	2	8.69565
AddressDomain_USPREP	^??	321 CURIE DRIVEE	1	4.34783
AddressDomain_USPREP	^D>T	2315 N 1ST ST	1	4.34783
AddressDomain_USPREP	???	BEL AIR AVEDUE	1	4.34783
AddressDomain_USPREP	??	BEL AIR	1	4.34783
AddressDomain_USPREP	^DOT??	2050 NORTH FIRST STREET SAN JOSE	1	4.34783
AddressDomain_USPREP	^D?T	170 W TASMAN DR	1	4.34783
AddressDomain_USPREP	^?T.	1361 CRESTWOOD DR .	1	4.34783
AddressDomain_USPREP	^S?T	6181 CAMINO VERDE DR	1	4.34783

Figure 1-206 Create J04_INVW_USPREP job 14/18


j04_INVW_USPREP..STAN_USPREP_ADDR_TOKEN_REPORT.AT - Data Browser

qsInvCount	qsInvWord	qsInvClassCode
5	DR	T
5	BEL	?
5	AVE	T
5	AIR	?
3	MAROUN	?
3	3424	^
3	1603	^
3	PL	T
2	ST	T
2	CT	T
2	PURPLE	?
2	CURIE	?
2	N	D
2	6177	^
2	SAGE	?
2	321	^
1	JCT	T
1	PKWY	T
1	SARATOGA	?
1	1735	^
1	CRESTWOOD	?
1	CM	S
1	AVEDUE	?
1	2584	^
1	2050	^
1	TASMAN	?
1	VERDE	?

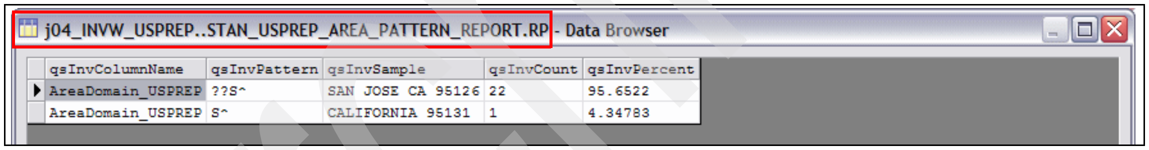
Figure 1-207 Create J04_INVW_USPREP job 15/18



j04_INVW_USPREP..STAN_USPREP_ADDR_TOKEN_REPORT.AT - Data Browser

qsInvCount	qsInvWord	qsInvClassCode
5	DR	T
5	BEL	?
5	AVE	T
5	AIR	?
3	MAROUN	?
3	3424	^
3	1603	^
3	PL	T
2	ST	T
2	CT	T
2	PURPLE	?
2	CURIE	?
2	N	D
2	6177	^
2	SAGE	?
2	321	^
1	JCT	T
1	PKWY	T
1	SARATOGA	?
1	1735	^
1	CRESTWOOD	?
1	CM	S
1	AVEDUE	?
1	2584	^
1	2050	^

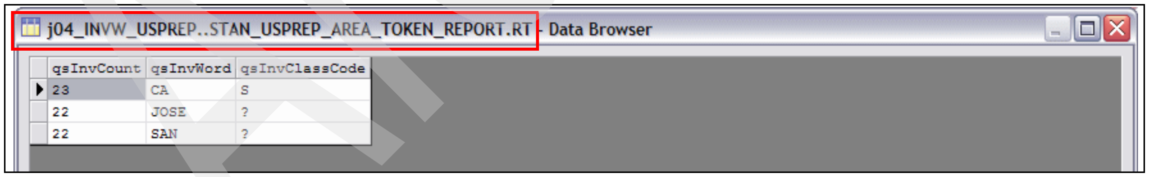
Figure 1-208 Create J04_INVW_USPREP job 16/18



j04_INVW_USPREP..STAN_USPREP_AREA_PATTERN_REPORT.RP - Data Browser

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
AreaDomain_USPREP	??S^	SAN JOSE CA 95126	22	95.6522
AreaDomain_USPREP	S^	CALIFORNIA 95131	1	4.34783

Figure 1-209 Create J04_INVW_USPREP job 17/18



j04_INVW_USPREP..STAN_USPREP_AREA_TOKEN_REPORT.RT - Data Browser

qsInvCount	qsInvWord	qsInvClassCode
23	CA	S
22	JOSE	?
22	SAN	?

Figure 1-210 Create J04_INVW_USPREP job 18/18

J03_Z_Override_And_After

In this job, we perform a visual analysis of the token and pattern reports of the "J04_INVW_USPREP" on page 203 step to determine if the parsing is successful.

We find an error in the Area pattern report shown in Figure 1-209 on page 214. A single occurrence of the string "CALIFORNIA" is identified, and the city name is missing in the AreaDomain_USPREP column—this corresponds to the address of Anders Olson of 2050 NORTH FIRST STREET SAN JOSE CALIFORNIA 95131 as shown in Figure 1-190 on page 202. We determine that the string `CALIFORNIA` refers to a state that should be abbreviated to `CA`.

We choose to perform a classification override of the USPREP rule set for the token `CALIFORNIA` as shown in Figure 1-211 on page 216 through Figure 1-217 on page 219.

The steps are as follows:

1. From the Designer client File menu, click **Open** as shown in Figure 1-211 on page 216.
2. In the Open window, expand Standardization Rules folder and click **USPREP** as shown in Figure 1-212 on page 216.
3. In the Rules Management - USPREP.SET window, click **Overrides** to add an override for that rule set as shown in Figure 1-213 on page 217.
4. In the Classification - USPREP window in Figure 1-214 on page 217 and Figure 1-215 on page 218, click the Classification tab.
   a. In the Input Token field, type the word for which you want to override the classification (`CALIFORNIA` in this case).
   b. In the Standard Form field, type the standardized spelling of the token (`CA` in this case).
   c. From the Classification menu, select the one-character tag that indicates the class of the token word (`S` for state in this case).
   d. In the Comparison Threshold field, type a value that defines the degree of uncertainty to tolerate in the spelling of the token word (`850` in this case).
   e. Click **Add** to add the override to the pane at the bottom of the window as shown in Figure 1-216 on page 218. Click **OK** to close the window.

   After you create the override, the next time you run the rule set, the word tokens are classified with the designations you specified and appear with the appropriate standard form.
5. You need to provision this rule set after the changes as shown in Figure 1-217 on page 219, and then compile and run the job (not shown here). The

override resulted in fixing the problem as seen in the Figure 1-218 on page 219 where the AreaDomain_USPREP domain bucket has the city and the state name correctly included.

Now, proceed to “J05_CASS_USPREP” on page 219.

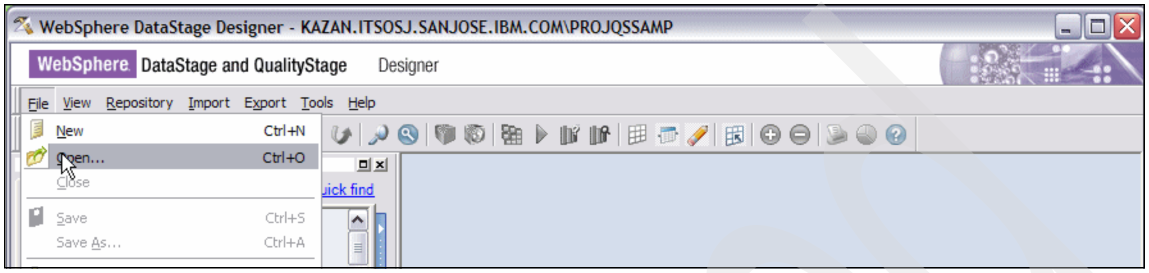

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP
WebSphere DataStage and QualityStage Designer
File View Repository Import Export Tools Help
New Ctrl+N
Open... Ctrl+O
Close
Save Ctrl+S
Save As... Ctrl+A
uick find

Figure 1-211 Create J03_Z_Override_And_After job 1/7

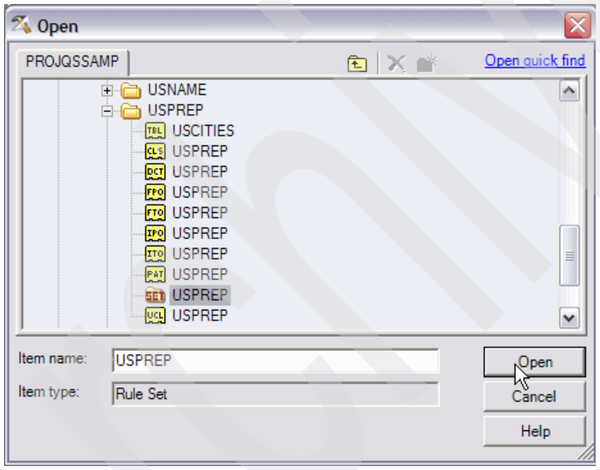
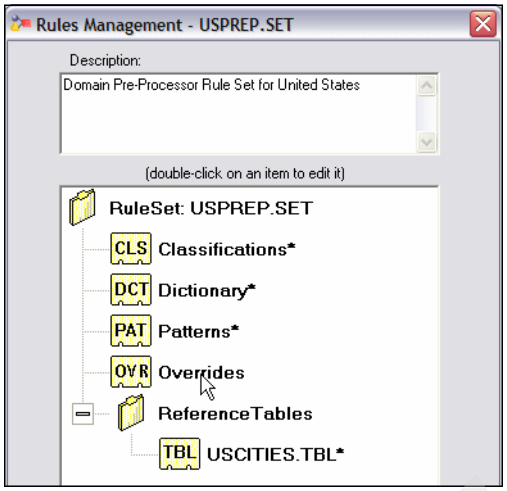


Figure 1-212 Create J03_Z_Override_And_After job 2/7


Rules Management - USPREP.SET

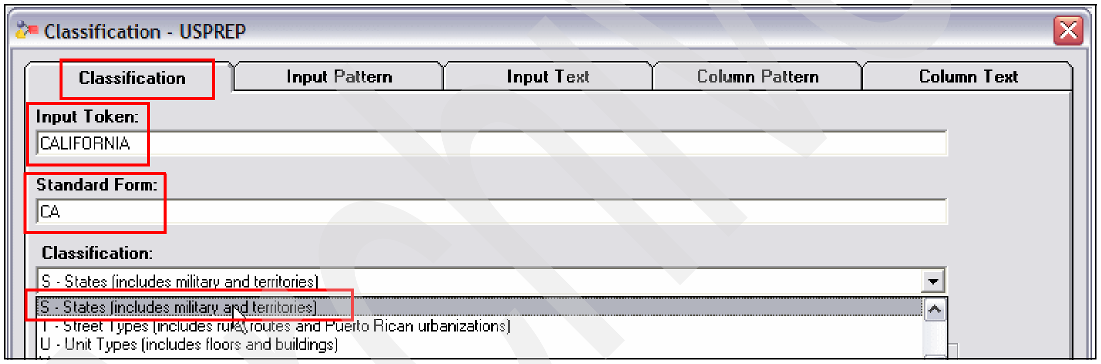
Description:

Domain Pre-Processor Rule Set for United States

(double-click on an item to edit it)

RuleSet: USPREP.SET
- CLS Classifications*
- DCT Dictionary*
- PAT Patterns*
- OVR Overrides
- ReferenceTables
  - TBL USCITIES.TBL*

Figure 1-213   Create J03_Z_Override_And_After job 3/7


Classification - USPREP

Classification | Input Pattern | Input Text | Column Pattern | Column Text

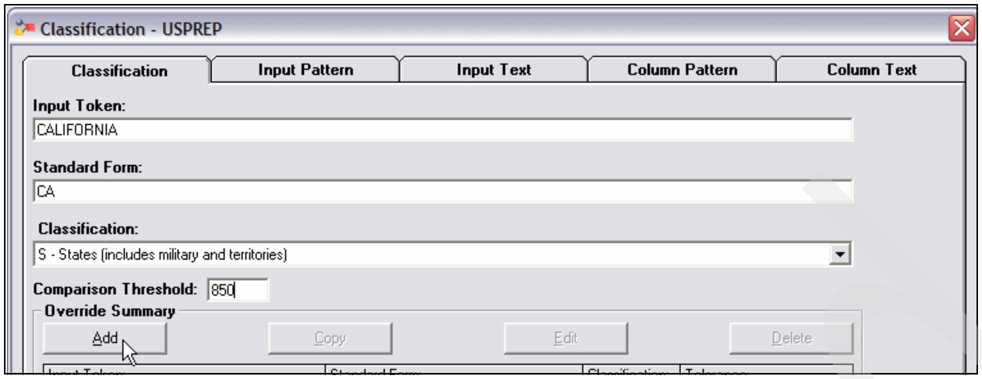
Input Token: CALIFORNIA

Standard Form: CA

Classification: S - States (includes military and territories)

S - States (includes military and territories)
T - Street Types (includes rural routes and Puerto Rican urbanizations)
U - Unit Types (includes floors and buildings)

Figure 1-214   Create J03_Z_Override_And_After job 4/7



Input Token: CALIFORNIA

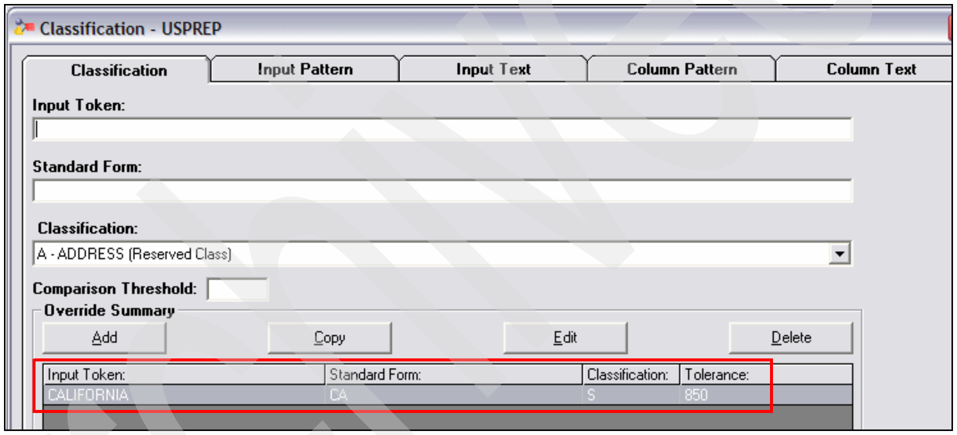
Standard Form: CA

Classification: S - States (includes military and territories)

Comparison Threshold: 850

Override Summary

Figure 1-215 Create J03_Z_Override_And_After job 5/8



Input Token:

Standard Form:

Classification: A - ADDRESS (Reserved Class)

Comparison Threshold:

Override Summary

Input Token:	Standard Form:	Classification:	Tolerance:
CALIFORNIA	CA	S	850

Figure 1-216 Create J03_Z_Override_And_After job 6/8

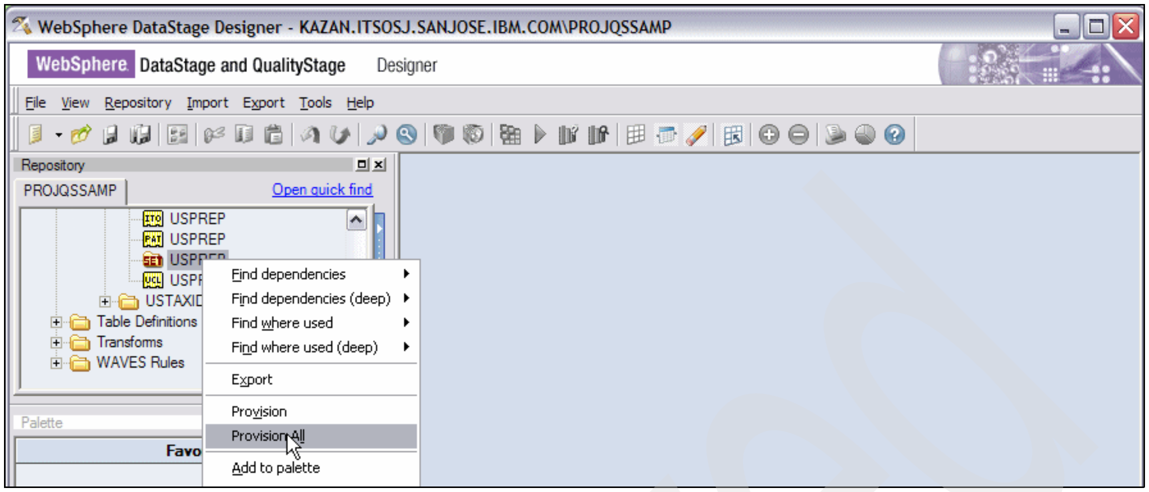
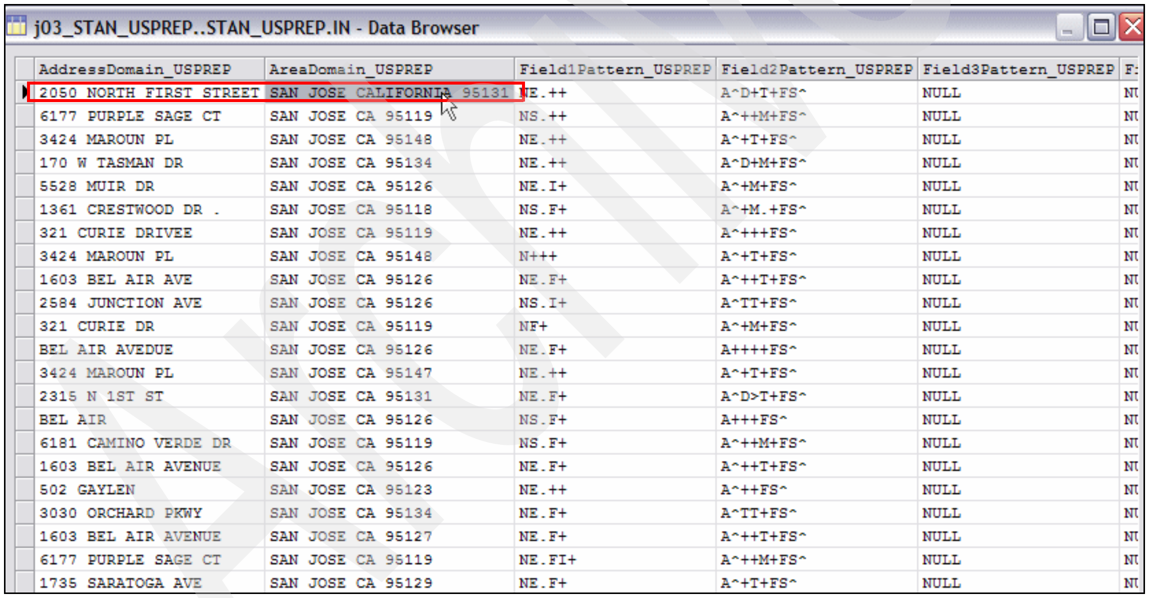


Figure 1-217 Create J03_Z_Override_And_After job 7/8


j03_STAN_USPREP..STAN_USPREP.IN - Data Browser

AddressDomain_USPREP	AreaDomain_USPREP	Field1Pattern_USPREP	Field2Pattern_USPREP	Field3Pattern_USPREP	F:
2050 NORTH FIRST STREET	SAN JOSE CALIFORNIA 95131	NE.++	A^D+T+FS^	NULL	NU
6177 PURPLE SAGE CT	SAN JOSE CA 95119	NS.++	A^++M+FS^	NULL	NU
3424 MAROUN PL	SAN JOSE CA 95148	NE.++	A^+T+FS^	NULL	NU
170 W TASMAN DR	SAN JOSE CA 95134	NE.++	A^D+M+FS^	NULL	NU
5528 MUIR DR	SAN JOSE CA 95126	NE.I+	A^+M+FS^	NULL	NU
1361 CRESTWOOD DR .	SAN JOSE CA 95118	NS.F+	A^+M.+FS^	NULL	NU
321 CURIE DRIVEE	SAN JOSE CA 95119	NE.++	A^+++FS^	NULL	NU
3424 MAROUN PL	SAN JOSE CA 95148	N+++	A^+T+FS^	NULL	NU
1603 BEL AIR AVE	SAN JOSE CA 95126	NE.F+	A^++T+FS^	NULL	NU
2584 JUNCTION AVE	SAN JOSE CA 95126	NS.I+	A^TT+FS^	NULL	NU
321 CURIE DR	SAN JOSE CA 95119	NF+	A^+M+FS^	NULL	NU
BEL AIR AVEDUE	SAN JOSE CA 95126	NE.F+	A++++FS^	NULL	NU
3424 MAROUN PL	SAN JOSE CA 95147	NE.++	A^+T+FS^	NULL	NU
2315 N 1ST ST	SAN JOSE CA 95131	NE.F+	A^D>T+FS^	NULL	NU
BEL AIR	SAN JOSE CA 95126	NS.F+	A+++FS^	NULL	NU
6181 CAMINO VERDE DR	SAN JOSE CA 95119	NS.F+	A^++M+FS^	NULL	NU
1603 BEL AIR AVENUE	SAN JOSE CA 95126	NE.F+	A^++T+FS^	NULL	NU
502 GAYLEN	SAN JOSE CA 95123	NE.++	A^++FS^	NULL	NU
3030 ORCHARD PKWY	SAN JOSE CA 95134	NE.F+	A^TT+FS^	NULL	NU
1603 BEL AIR AVENUE	SAN JOSE CA 95127	NE.F+	A^++T+FS^	NULL	NU
6177 PURPLE SAGE CT	SAN JOSE CA 95119	NE.FI+	A^++M+FS^	NULL	NU
1735 SARATOGA AVE	SAN JOSE CA 95129	NE.F+	A^+T+FS^	NULL	NU

Figure 1-218 Create J03_Z_Override_And_After job 8/8

J05_CASS_USPREP

After all the name and address data is moved to the correct domain buckets, we use the CASS[21] stage to validate, correct, and standardize the U.S. addresses in the Address domain and to write a Postal Service form 3553 to a file. We also included a Transformer stage to add a second address line column to customer file because CASS requires two address lines as input for its processing.

Note: As mentioned earlier, CASS is a separately priced module that requires installation of the CASS module.

Figure 1-219 on page 223 through Figure 1-228 on page 228 describe the steps using Designer Client to build and execute the DataStage job to perform this task.

The steps are as follows:

1. Figure 1-219 on page 223 shows the various stages that are used in this job, including the data set that was created in “J03_STAN_USPREP” on page 196, a Transformer stage, a CASS stage, and a Data Set stage. The names of the stages are modified as shown, and this job uses the same QSPARAMETERSET object created earlier.
2. Configure the input STAN_USPREP data set with the table definition CUSTOMER_USPREP that was saved in “J03_STAN_USPREP” on page 196. We do not repeat this process here because it is similar to the steps shown in Figure 1-138 on page 172 through Figure 1-142 on page 174.
3. Configure the Transformer stage ADD_SPACE_FIELD to add an address line to the input file, because CASS expects two address lines in the input metadata and the credit card customer file only has one address line. Follow these steps:
   a. Right-click the **ADD_SAPCE_FIELD** stage icon and click **Properties** as shown in Figure 1-220 on page 223.
   b. In the ADD_SAPCE_FIELD - Transformer Stage window, select and copy all the columns from the OUT list to the TOUT list as shown in Figure 1-221 on page 224.
   c. To add a column to the output (TOUT), select the top row of the TOUT pane and right-click the first row and select **Insert Row.** Edit the SQL type, length, scale, and nullability to values as highlighted in Figure 1-221 on

[21] The standardization process within the QualityStage CASS stage corrects many defects in the mailing list input file. However, the closer your address data conforms to USPS guidelines, the more address matches you generate. According to the U.S. Postal Service, a complete address is one that has all the address elements necessary to allow an exact match with the current Postal Service format of ZIP code + 4, and city, state files. A standardized address has all the components completely spelled out or abbreviated using the Postal Service standard abbreviations or uses the abbreviations as shown in the current Postal Service ZIP code + 4 file. For more information about USPS address standards, see Postal Addressing Standards, Publication 28, published by the National Customer Support Center. The pub28.pdf document is available at:
http://pe.usps.gov/cpim/ftp/pubs/Pub28/

page 224. Click **OK** to complete the addition of the DUMMY_ADD_FIELD of character length 1 to serve as the second address line required by the CASS stage.

4. Configure the CASS stage by double-clicking the **CASS** stage icon to open the stage. On the Properties tab:
   a. Select **Address Line 1** under Input Address Columns. On the right side of the tab, a list of all columns in the input stage displays under Address Line 1, which generally corresponds to the first line of a street address.
   b. From the list, select the column AddressDomain_USPREP that corresponds to the first address line in your data.
   c. Select DUMMY_ADD_FIELD as the column that corresponds to the Address Line 2.
   d. Select City/State/ZIP combined and select **Yes**. Select **AreaDomain_Area** as the input column that contains the combined information.
   e. Add information for the USPSForm 3553 such as Company Name (#QSPARAMETERSET.CASS_COMPANY_NAME#), List Identifier (#QSPARAMETERSET.CASS_List_ID#), and Output File[22] (#QSPARAMETERSET_CASS_REPORT_PATH#CUSTOMER_CASS.rpt #). Form 3553 is written to the output file when you run the job.
   f. Under Reference Database, specify the Path (#QSPARAMETERSET.CASS_PATH#) of the location of the CASS database.

   Note: Figure 1-222 on page 225 shows the CASS stage properties at the end of configuration.

   g. Next, map the output columns. In the CASS window, on the Output tab for the Output name (IN), click the Mapping tab. Copy all the columns in the left Columns pane as shown in Figure 1-223 on page 225. Click **OK** to save changes and close the CASS stage.
5. Configure the CASS_CUSTOMER Data Set stage, and after saving, compiling, and running this job (Figure 1-224 on page 226), review the contents of the output of the CASS stage.
6. The contents of the data input to the CASS stage are shown in Figure 1-190 on page 202 through Figure 1-192 on page 203.

[22] The CASS results are written to an output data set during CASS processing. The results data set contains all the input record information.

7. The output of the CASS stage is shown in Figure 1-225 on page 226 through Figure 1-228 on page 228.

   The CASS results are written to an output data set during CASS processing. The results data set contains all the input record information. Those records where the address matched successfully to the CASS database also contain the footnotes and flags as a result of CASS processing. You can search the results file to identify unqualified addresses and correct where necessary.

   The CASS stage standardizes all address records while processing records through DPV and LACSLink, if necessary. The standardize process that the CASS stage performs results in address records being matched to the USPS standard addresses. The following tasks are performed during the (CASS) standardize process:

   – Parses all input records and places components in specific match key columns
   – Expands city abbreviations, such as *SF* to *San Francisco*
   – Standardizes specific abbreviations, such as *CONN* to *CT*
   – Matches the standardized addresses against the USPS standard addresses that is contained in the IBM WebSphere QualityStage CASS data. Matching results are defined as follows:
     • If the address information matches, the CASS stage assigns CASS delivery data including ZIP Code, ZIP code + 4, delivery point, and carrier route. Delivery Point Verification (DPV) processes the ZIP code + 4 information to ensure that it is a valid delivery point.
     • If the address information does not match, the CASS stage assigns CASS delivery data of ZIP code and carrier route only. DPV could not confirm the address as a valid delivery point.
     • If the record is also identified as a LACSLink record, then the CASS stage performs LACSLink processing.
     • If LACSLink processing returns a converted address, the converted address goes through DPV processing.
     • If a new LACSLink address is not produced, the address remains unchanged.

   The records that do not have an exact match in the USPS file can still be processed. The following guidelines are applied to those records:

   – For most unmatched columns, the columns for mailing name, street address, and city/state/ZIP are copied from the input record to the output record
   – For most unmatched records, if an input record does not match the USPS file, the output column for the four-digit ZIP code add-on, the carrier route

number, and the delivery point code number are left blank in the output record

– For an unmatched street address but with valid city and state or ZIP Code, if the street address does not match the USPS file, the address qualifies as a *small town default*.

Now, proceed to “J06_INVCC_CASS” on page 228.

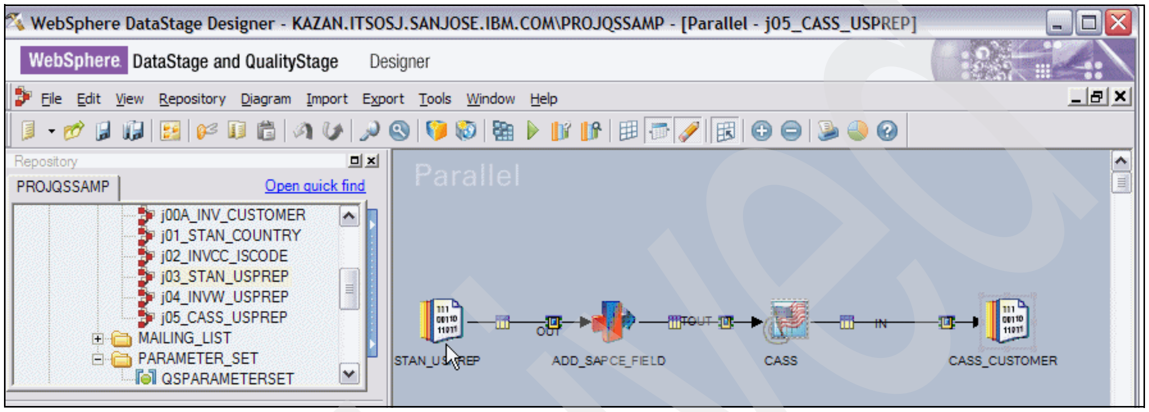


Figure 1-219 Create J05_CASS_USPREP job 1/10

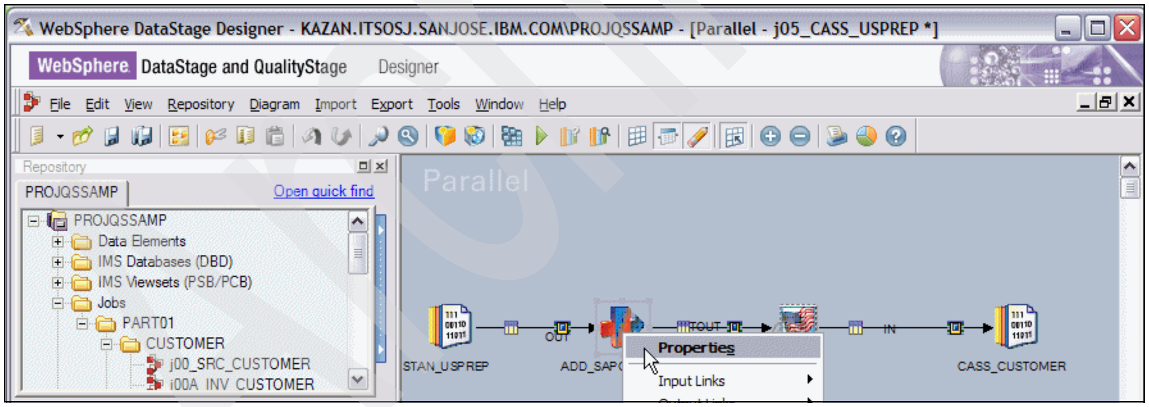
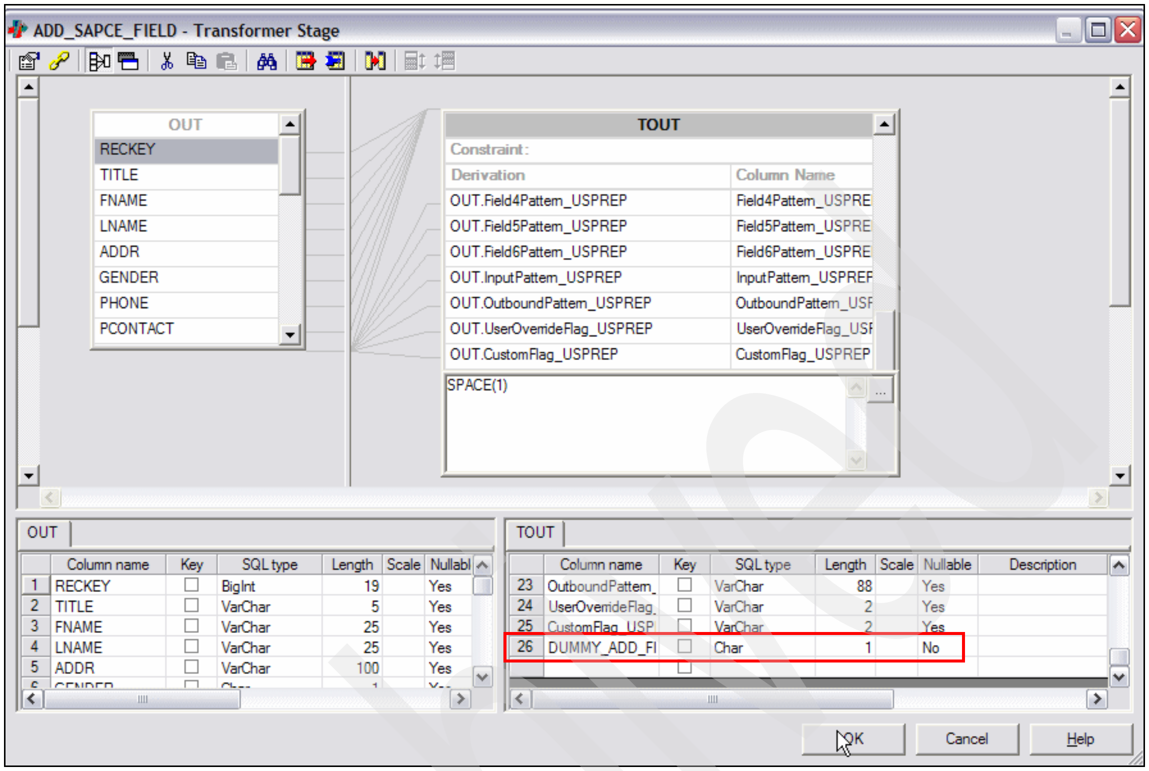


Figure 1-220 Create J05_CASS_USPREP job 2/10


ADD_SAPCE_FIELD - Transformer Stage

OUT
RECKEY
TITLE
FNAME
LNAME
ADDR
GENDER
PHONE
PCONTACT

TOUT

Constraint:

Derivation	Column Name
OUT.Field4Pattern_USPREP	Field4Pattern_USPRE
OUT.Field5Pattern_USPREP	Field5Pattern_USPRE
OUT.Field6Pattern_USPREP	Field6Pattern_USPRE
OUT.InputPattern_USPREP	InputPattern_USPREF
OUT.OutboundPattern_USPREP	OutboundPattern_USF
OUT.UserOverrideFlag_USPREP	UserOverrideFlag_USF
OUT.CustomFlag_USPREP	CustomFlag_USPREP

SPACE(1)

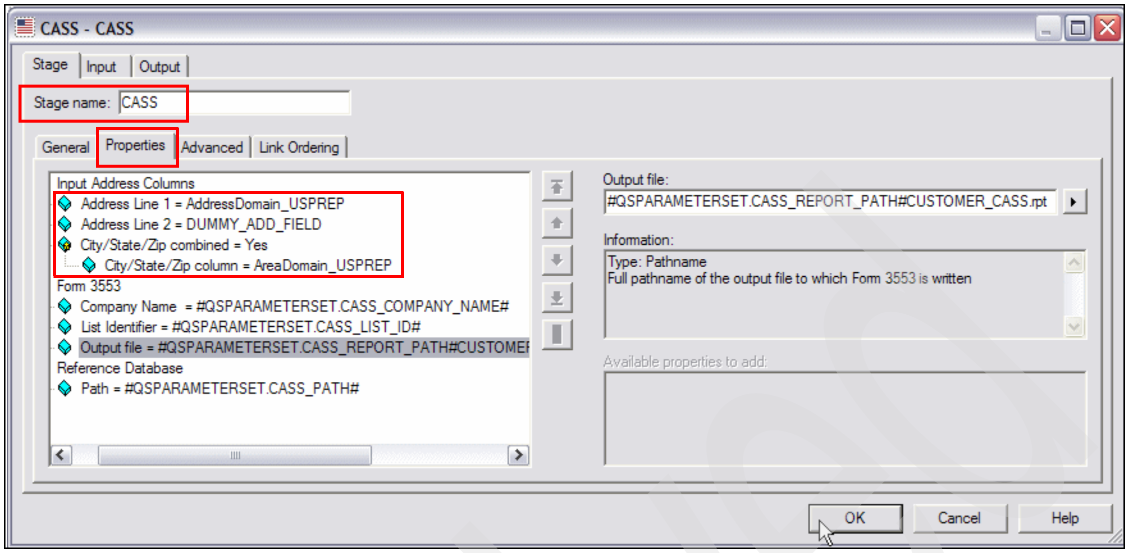
OUT

	Column name	Key	SQL type	Length	Scale	Nullabl
1	RECKEY	☐	BigInt	19		Yes
2	TITLE	☐	VarChar	5		Yes
3	FNAME	☐	VarChar	25		Yes
4	LNAME	☐	VarChar	25		Yes
5	ADDR	☐	VarChar	100		Yes

TOUT

	Column name	Key	SQL type	Length	Scale	Nullable	Description
23	OutboundPattern_	☐	VarChar	88		Yes	
24	UserOverrideFlag_	☐	VarChar	2		Yes	
25	CustomFlag_USP	☐	VarChar	2		Yes	
26	DUMMY_ADD_FI	☐	Char	1		No	

Figure 1-221 Create J05_CASS_USPREP job 3/10


CASS - CASS

Stage | Input | Output

Stage name: CASS

General | Properties | Advanced | Link Ordering

```
Input Address Columns
  Address Line 1 = AddressDomain_USPREP
  Address Line 2 = DUMMY_ADD_FIELD
  City/State/Zip combined = Yes
    City/State/Zip column = AreaDomain_USPREP
Form 3553
  Company Name  = #QSPARAMETERSET.CASS_COMPANY_NAME#
  List Identifier = #QSPARAMETERSET.CASS_LIST_ID#
  Output file = #QSPARAMETERSET.CASS_REPORT_PATH#CUSTOMEF
Reference Database
  Path = #QSPARAMETERSET.CASS_PATH#
```

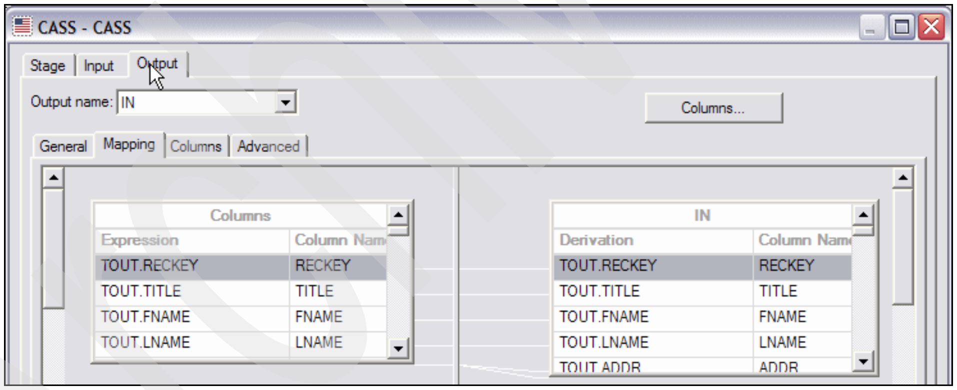
Output file: #QSPARAMETERSET.CASS_REPORT_PATH#CUSTOMER_CASS.rpt

Information:
Type: Pathname
Full pathname of the output file to which Form 3553 is written

Available properties to add:

OK | Cancel | Help

Figure 1-222 Create J05_CASS_USPREP job 4/10


CASS - CASS

Stage | Input | Output

Output name: IN

Columns...

General | Mapping | Columns | Advanced

Columns	
Expression	Column Nam
TOUT.RECKEY	RECKEY
TOUT.TITLE	TITLE
TOUT.FNAME	FNAME
TOUT.LNAME	LNAME

IN	
Derivation	Column Nam
TOUT.RECKEY	RECKEY
TOUT.TITLE	TITLE
TOUT.FNAME	FNAME
TOUT.LNAME	LNAME
TOUT.ADDR	ADDR

Figure 1-223 Create J05_CASS_USPREP job 5/10

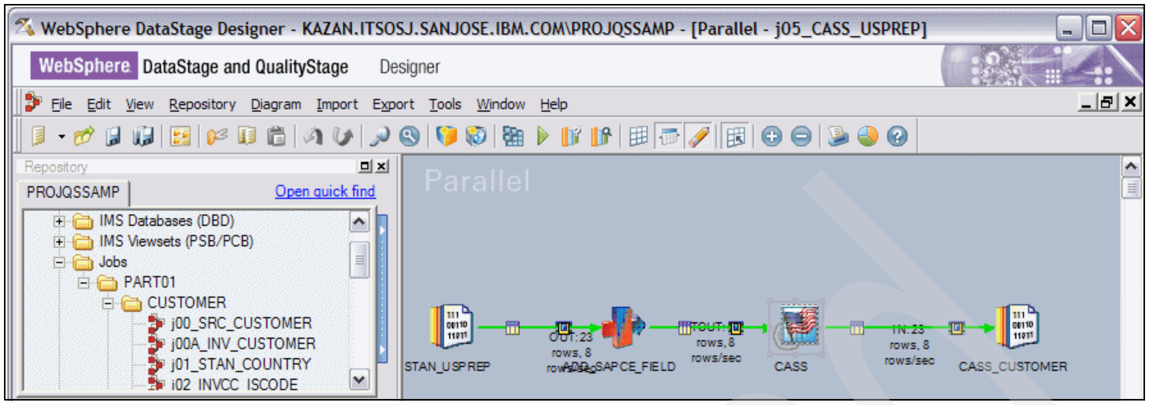
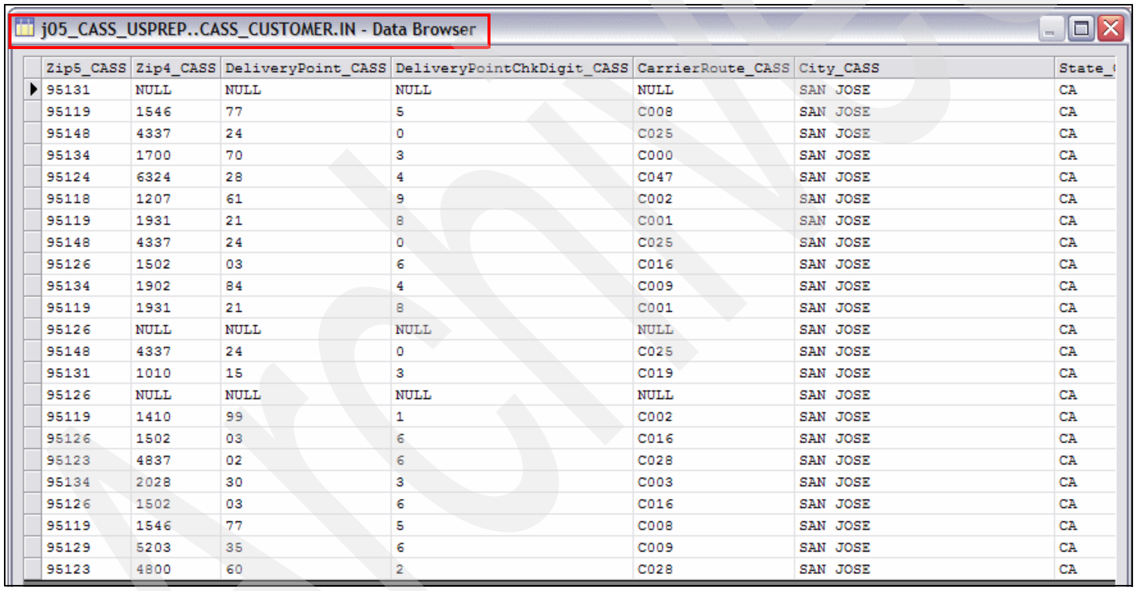
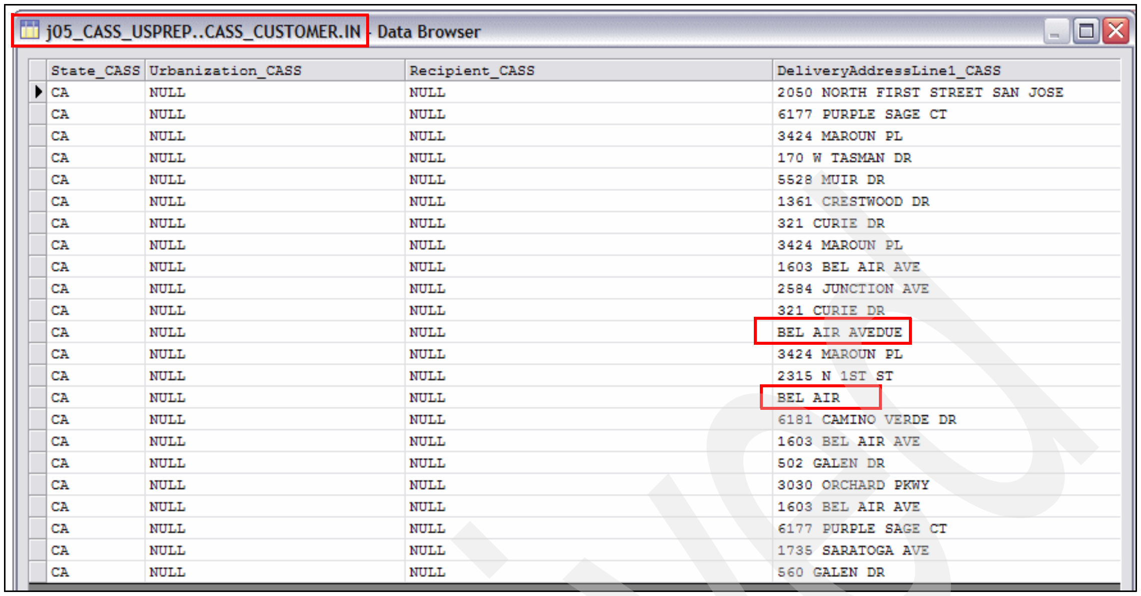


Figure 1-224 Create J05_CASS_USPREP job 6/10


j05_CASS_USPREP..CASS_CUSTOMER.IN - Data Browser

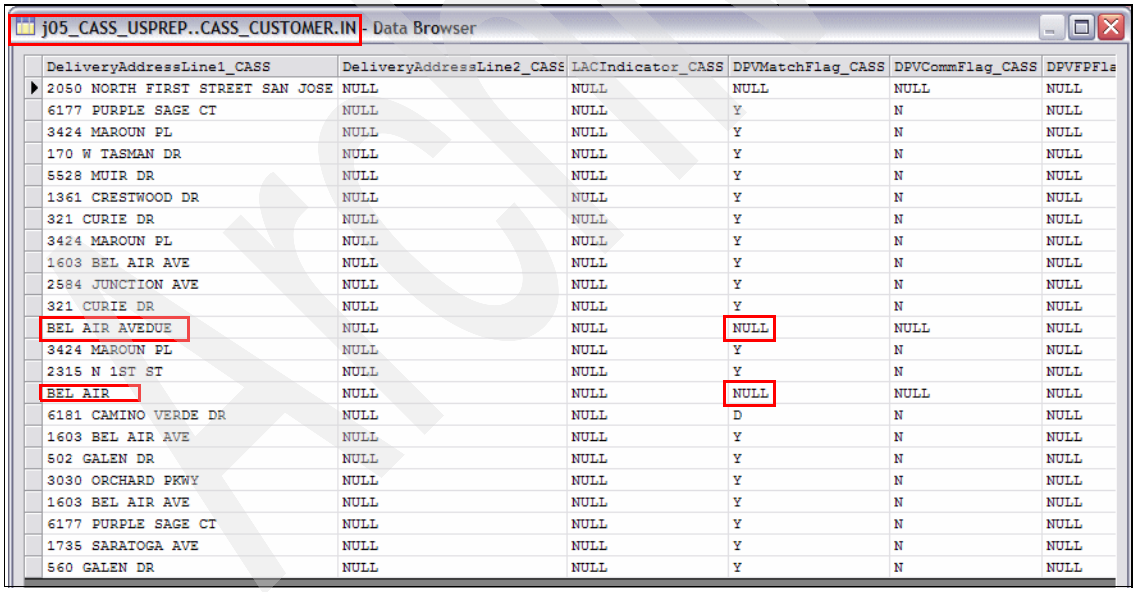
Zip5_CASS	Zip4_CASS	DeliveryPoint_CASS	DeliveryPointChkDigit_CASS	CarrierRoute_CASS	City_CASS	State_
95131	NULL	NULL	NULL	NULL	SAN JOSE	CA
95119	1546	77	5	C008	SAN JOSE	CA
95148	4337	24	0	C025	SAN JOSE	CA
95134	1700	70	3	C000	SAN JOSE	CA
95124	6324	28	4	C047	SAN JOSE	CA
95118	1207	61	9	C002	SAN JOSE	CA
95119	1931	21	8	C001	SAN JOSE	CA
95148	4337	24	0	C025	SAN JOSE	CA
95126	1502	03	6	C016	SAN JOSE	CA
95134	1902	84	4	C009	SAN JOSE	CA
95119	1931	21	8	C001	SAN JOSE	CA
95126	NULL	NULL	NULL	NULL	SAN JOSE	CA
95148	4337	24	0	C025	SAN JOSE	CA
95131	1010	15	3	C019	SAN JOSE	CA
95126	NULL	NULL	NULL	NULL	SAN JOSE	CA
95119	1410	99	1	C002	SAN JOSE	CA
95126	1502	03	6	C016	SAN JOSE	CA
95123	4837	02	6	C028	SAN JOSE	CA
95134	2028	30	3	C003	SAN JOSE	CA
95126	1502	03	6	C016	SAN JOSE	CA
95119	1546	77	5	C008	SAN JOSE	CA
95129	5203	35	6	C009	SAN JOSE	CA
95123	4800	60	2	C028	SAN JOSE	CA

Figure 1-225 Create J05_CASS_USPREP job 7/10


j05_CASS_USPREP..CASS_CUSTOMER.IN - Data Browser

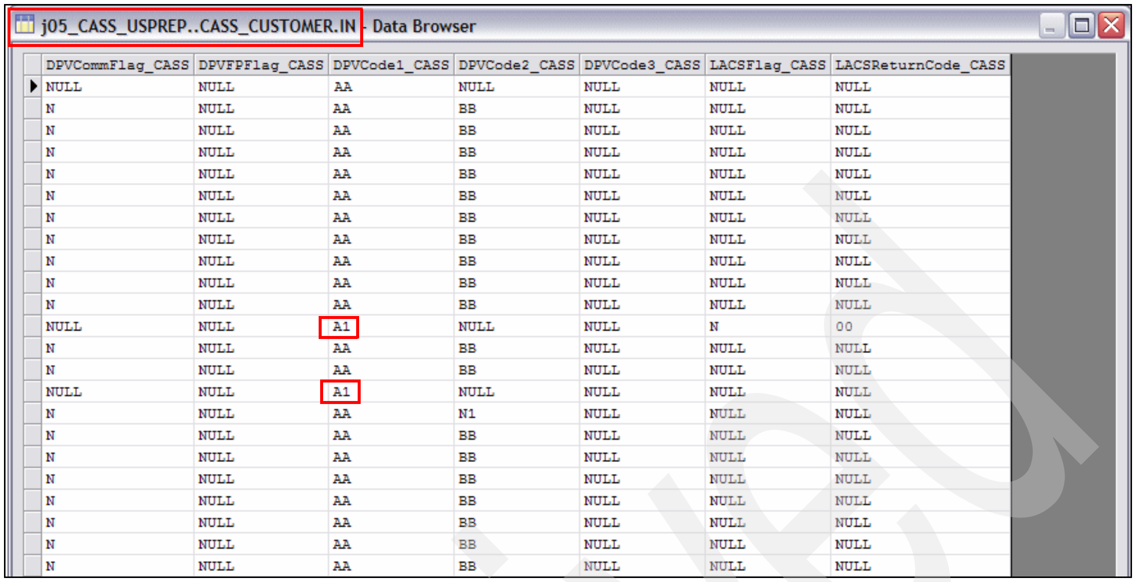
State_CASS	Urbanization_CASS	Recipient_CASS	DeliveryAddressLine1_CASS
CA	NULL	NULL	2050 NORTH FIRST STREET SAN JOSE
CA	NULL	NULL	6177 PURPLE SAGE CT
CA	NULL	NULL	3424 MAROUN PL
CA	NULL	NULL	170 W TASMAN DR
CA	NULL	NULL	5528 MUIR DR
CA	NULL	NULL	1361 CRESTWOOD DR
CA	NULL	NULL	321 CURIE DR
CA	NULL	NULL	3424 MAROUN PL
CA	NULL	NULL	1603 BEL AIR AVE
CA	NULL	NULL	2584 JUNCTION AVE
CA	NULL	NULL	321 CURIE DR
CA	NULL	NULL	BEL AIR AVEDUE
CA	NULL	NULL	3424 MAROUN PL
CA	NULL	NULL	2315 N 1ST ST
CA	NULL	NULL	BEL AIR
CA	NULL	NULL	6181 CAMINO VERDE DR
CA	NULL	NULL	1603 BEL AIR AVE
CA	NULL	NULL	502 GALEN DR
CA	NULL	NULL	3030 ORCHARD PKWY
CA	NULL	NULL	1603 BEL AIR AVE
CA	NULL	NULL	6177 PURPLE SAGE CT
CA	NULL	NULL	1735 SARATOGA AVE
CA	NULL	NULL	560 GALEN DR

Figure 1-226 Create J05_CASS_USPREP job 8/10


j05_CASS_USPREP..CASS_CUSTOMER.IN - Data Browser

DeliveryAddressLine1_CASS	DeliveryAddressLine2_CASS	LACIndicator_CASS	DPVMatchFlag_CASS	DPVCommFlag_CASS	DPVFPFla
2050 NORTH FIRST STREET SAN JOSE	NULL	NULL	NULL	NULL	NULL
6177 PURPLE SAGE CT	NULL	NULL	Y	N	NULL
3424 MAROUN PL	NULL	NULL	Y	N	NULL
170 W TASMAN DR	NULL	NULL	Y	N	NULL
5528 MUIR DR	NULL	NULL	Y	N	NULL
1361 CRESTWOOD DR	NULL	NULL	Y	N	NULL
321 CURIE DR	NULL	NULL	Y	N	NULL
3424 MAROUN PL	NULL	NULL	Y	N	NULL
1603 BEL AIR AVE	NULL	NULL	Y	N	NULL
2584 JUNCTION AVE	NULL	NULL	Y	N	NULL
321 CURIE DR	NULL	NULL	Y	N	NULL
BEL AIR AVEDUE	NULL	NULL	NULL	NULL	NULL
3424 MAROUN PL	NULL	NULL	Y	N	NULL
2315 N 1ST ST	NULL	NULL	Y	N	NULL
BEL AIR	NULL	NULL	NULL	NULL	NULL
6181 CAMINO VERDE DR	NULL	NULL	D	N	NULL
1603 BEL AIR AVE	NULL	NULL	Y	N	NULL
502 GALEN DR	NULL	NULL	Y	N	NULL
3030 ORCHARD PKWY	NULL	NULL	Y	N	NULL
1603 BEL AIR AVE	NULL	NULL	Y	N	NULL
6177 PURPLE SAGE CT	NULL	NULL	Y	N	NULL
1735 SARATOGA AVE	NULL	NULL	Y	N	NULL
560 GALEN DR	NULL	NULL	Y	N	NULL

Figure 1-227 Create J05_CASS_USPREP job 9/10


j05_CASS_USPREP..CASS_CUSTOMER.IN - Data Browser

	DPVCommFlag_CASS	DPVFPFlag_CASS	DPVCode1_CASS	DPVCode2_CASS	DPVCode3_CASS	LACSFlag_CASS	LACSReturnCode_CASS
▶	NULL	NULL	AA	NULL	NULL	NULL	NULL
	N	NULL	AA	BB	NULL	NULL	NULL
	N	NULL	AA	BB	NULL	NULL	NULL
	N	NULL	AA	BB	NULL	NULL	NULL
	N	NULL	AA	BB	NULL	NULL	NULL
	N	NULL	AA	BB	NULL	NULL	NULL
	N	NULL	AA	BB	NULL	NULL	NULL
	N	NULL	AA	BB	NULL	NULL	NULL
	N	NULL	AA	BB	NULL	NULL	NULL
	N	NULL	AA	BB	NULL	NULL	NULL
	N	NULL	AA	BB	NULL	NULL	NULL
	NULL	NULL	A1	NULL	NULL	N	00
	N	NULL	AA	BB	NULL	NULL	NULL
	N	NULL	AA	BB	NULL	NULL	NULL
	NULL	NULL	A1	NULL	NULL	NULL	NULL
	N	NULL	AA	N1	NULL	NULL	NULL
	N	NULL	AA	BB	NULL	NULL	NULL
	N	NULL	AA	BB	NULL	NULL	NULL
	N	NULL	AA	BB	NULL	NULL	NULL
	N	NULL	AA	BB	NULL	NULL	NULL
	N	NULL	AA	BB	NULL	NULL	NULL
	N	NULL	AA	BB	NULL	NULL	NULL
	N	NULL	AA	BB	NULL	NULL	NULL

Figure 1-228 Create J05_CASS_USPREP job 10/10

J06_INVCC_CASS

In this job, we run the Investigate stage with character concatenate on the results of the job “J05_CASS_USPREP” on page 219 to determine addresses that are not recognized by CASS (delivery point verification or DPV). We investigate the output of CASS using character concatenate (on CASS generated columns DPVMATCHFLAG_CASS and DPVCODE1_CASS) using a *C* mask. A value of `A1` in the DPVCODE1_CASS field indicated a potential problem.

Note: Due to a bug with handling nulls, we introduced a Transformer stage to convert nulls to a space using column derivation.

Figure 1-229 on page 230 through Figure 1-240 on page 234 describe the steps using Designer Client to build and execute the DataStage job to perform this task.

The steps are as follows:

1. Figure 1-229 on page 230 shows the various stages that are used in this job, including the data set that was created in “J05_CASS_USPREP” on page 219, a Transformer stage for handling nulls, an Investigate stage, and an output Sequential File stage. We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object that was created earlier.

2. Configure the input CASS_CUSTOMER data set with the table definition CUSTOMER_CASS that was saved in “J05_CASS_USPREP” on page 219. We do not repeat this here because it is similar to the steps shown in Figure 1-138 on page 172 through Figure 1-142 on page 174.
3. Configure the HANDLE_NULL Transformer stage to convert nulls to spaces as follows:
   a. Double-click the **HANDLE_NULL** stage icon, and when the HANDLE_NULL - Transformer Stage window opens, copy all the columns from the OUT left hand pane to the IN right hand pane.
   b. Then select all rows in the right-hand pane, right-click it, and select **Derivation Substitution** as shown in Figure 1-230 on page 231.
   c. In the Expression Substitution window, select **Whole expression** in the Substitute field, and enter the following expression as shown in Figure 1-231 on page 231.

      ```
      If ISNULL($1) then “ else $1
      ```

      Then, click **OK**. This derivative directs that if a value is null, then replace it with an empty string.
4. A Confirm Action window opens. Click **Yes To All** to overwrite the expressions of all columns as shown in Figure 1-232 on page 231. The total number of expressions updated is shown in Figure 1-233 on page 232. Click **OK**.
5. Configure the INV_DPV_INFO Investigate stage by right-clicking the INV_DPV_INFO icon and selecting **Properties** in Figure 1-234 on page 232. Then, follow these steps:
   a. Click **Character Concatenate Investigate**. The columns propagated from the input data set are shown in the Available Data Columns list. Select the following columns:
      - DPVMatchFlag_CASS with C mask
      - DPVCode1_CASS with C mask
      - Delivery AddressLine1_CASS with X mask
      - City_CASS with X mask
      - State_CASS with X mask
      - Zip5_CASS with X mask

      After adding these columns with these masks, click **Stage Properties** as shown in Figure 1-235 on page 232.
   b. In the INV_DPV_INFO - Investigate windows opens, select Mapping in the Output tab—the Output name is *IN*. Map all the columns in the Columns list to the IN list as shown in Figure 1-236 on page 233. Click **OK**.
6. Configure the output sequential file CUSTOMER_CASS_REPORT object to sort the incoming records on descending sequence of the count of distinct

values in the concatenated character columns. The objective is to determine the predominant country addresses and identify the ISO codes present in the data.

Figure 1-237 on page 233 shows the properties of the configured file.

7. After saving, compiling, and running this job, the job statistics are shown in Figure 1-238 on page 234.
8. The contents of the data input to the Investigate stage are shown in Figure 1-225 on page 226 through Figure 1-228 on page 228.
9. The output of the Investigate stage that are written to the sequential file is shown in Figure 1-239 on page 234 and Figure 1-240 on page 234. It shows a count of two records with values `A1` in the DPVMatchFlag_CASS and DPVCode1_CASS character concatenated columns.

   As discussed in "J05_CASS_USPREP" on page 219, a value of `A1` in the DPVCode1_CASS indicates an address that did not match.

   Based on the data in the qsInvSample column of the report (in Figure 1-239 on page 234 and Figure 1-240 on page 234) and an analysis of the input file as reported in Figure 1-227 on page 227 and Figure 1-228 on page 228, two input addresses have a value of `A1` in the DPVCode1_CASS indicator, which indicates that the address does not match a USPS standard address. These input addresses are as follows:

   – BEL AIR AVEDUE
   – BEL AIR

   Note: *AVEDUE* is spelled incorrectly here. We will address this misspelling later with an override. The street number is either incorrect or missing and will need some other mechanism for resolution.

Now, proceed to "J07_STAN_CUSTOMER_Domain_Preprocessor" on page 234.

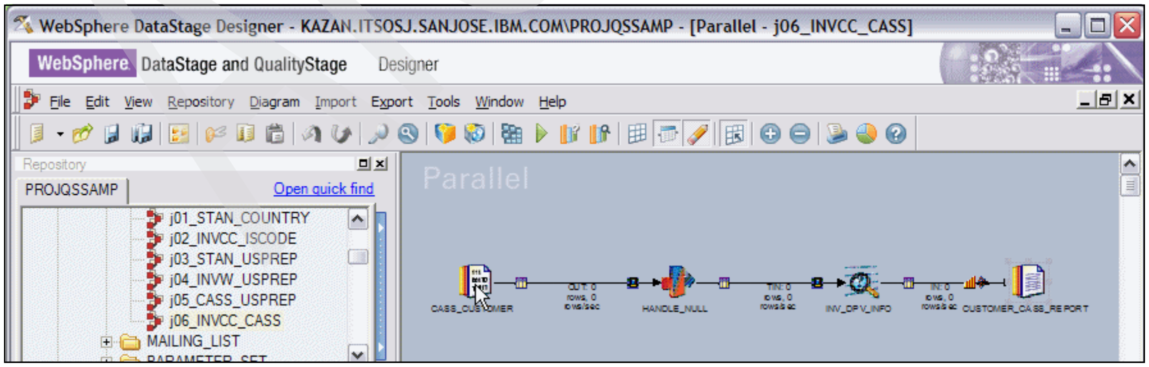


Figure 1-229 Create J06_INVCC_CASS job 1/12

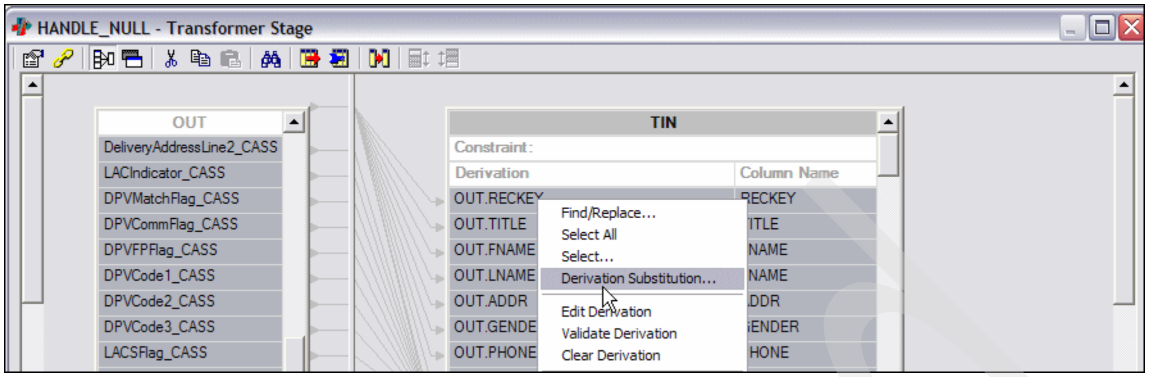
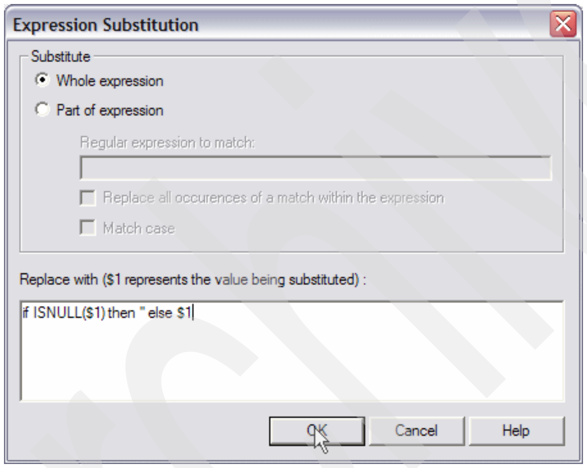
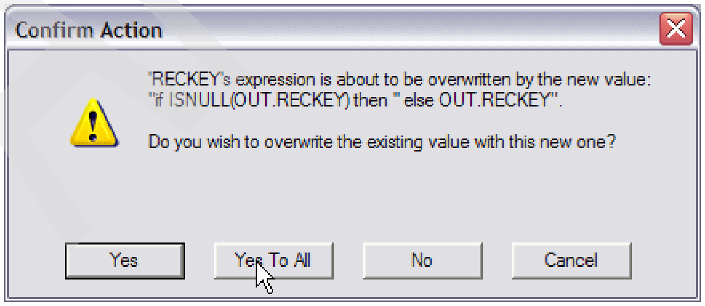


Figure 1-230 Create J06_INVCC_CASS job 2/12


Expression Substitution
Substitute
Whole expression
Part of expression
Regular expression to match:
Replace all occurences of a match within the expression
Match case
Replace with ($1 represents the value being substituted) :
OK
Cancel
Help

```
if ISNULL($1) then '' else $1
```

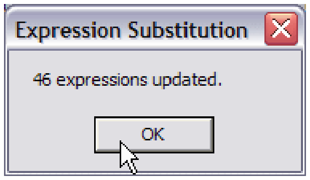
Figure 1-231 Create J06_INVCC_CASS job 3/12



'RECKEY''s expression is about to be overwritten by the new value:
"if ISNULL(OUT.RECKEY) then '' else OUT.RECKEY".

Do you wish to overwrite the existing value with this new one?

Figure 1-232 Create J06_INVCC_CASS job 4/12


Expression Substitution

46 expressions updated.

Figure 1-233   Create J06_INVCC_CASS job 5/12

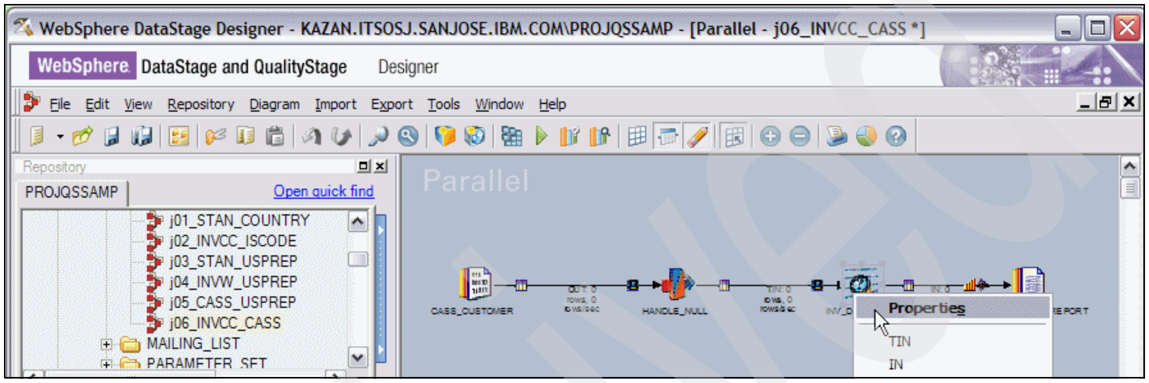
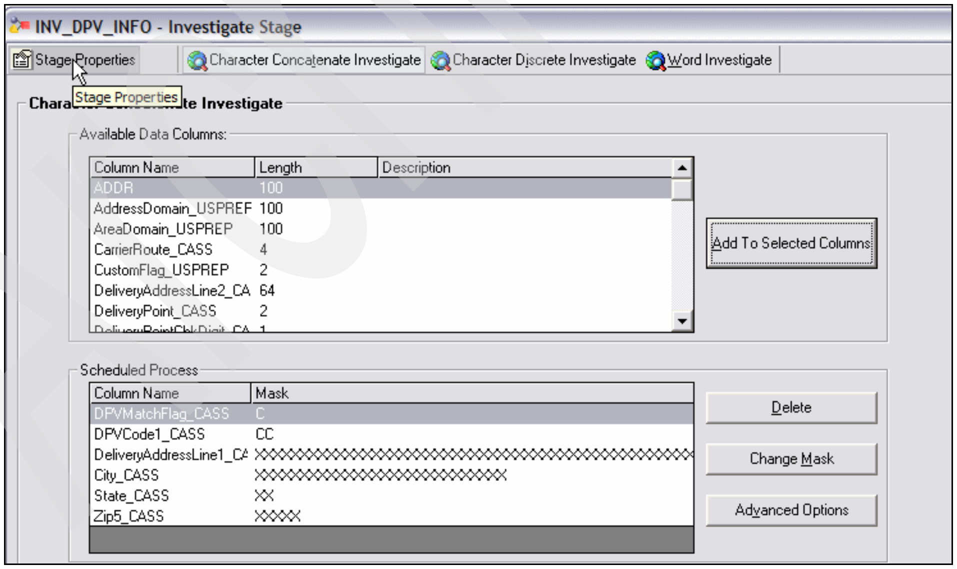

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j06_INVCC_CASS *]

Figure 1-234   Create J06_INVCC_CASS job 6/12


INV_DPV_INFO - Investigate Stage

Stage Properties
Character Concatenate Investigate
Character Discrete Investigate
Word Investigate

Chara... te Investigate

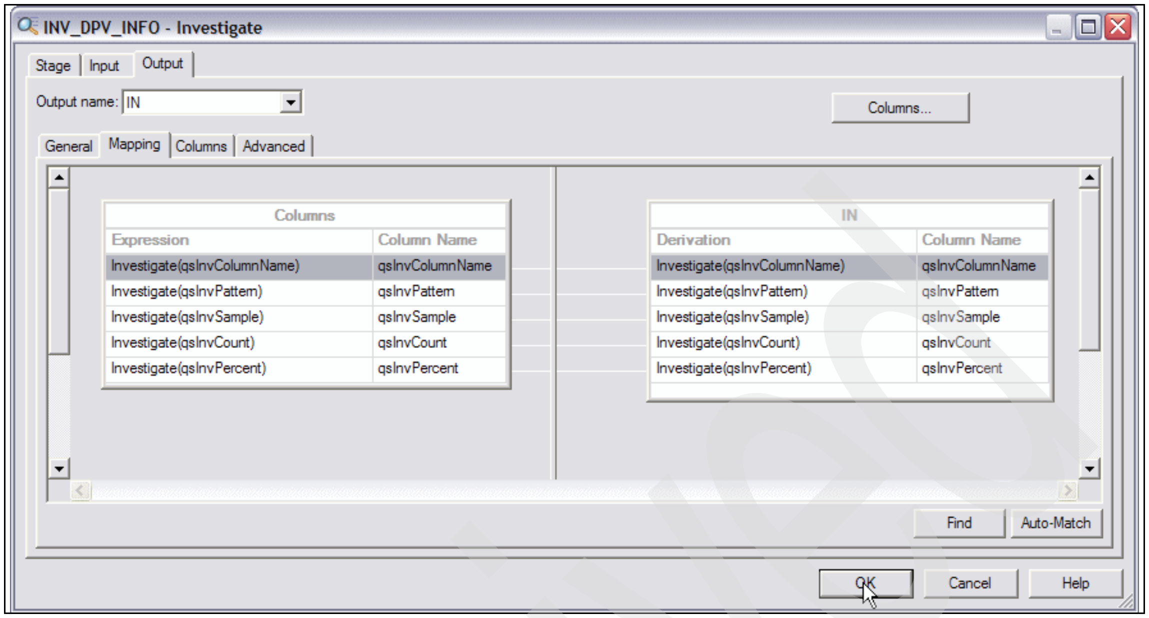
Available Data Columns:

Column Name	Length	Description
ADDR	100	
AddressDomain_USPREP	100	
AreaDomain_USPREP	100	
CarrierRoute_CASS	4	
CustomFlag_USPREP	2	
DeliveryAddressLine2_CA	64	
DeliveryPoint_CASS	2	

Scheduled Process

Column Name	Mask
DPVMatchFlag_CASS	C
DPVCode1_CASS	CC
DeliveryAddressLine1_CA	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
City_CASS	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
State_CASS	XX
Zip5_CASS	XXXXX

Figure 1-235   Create J06_INVCC_CASS job 7/12

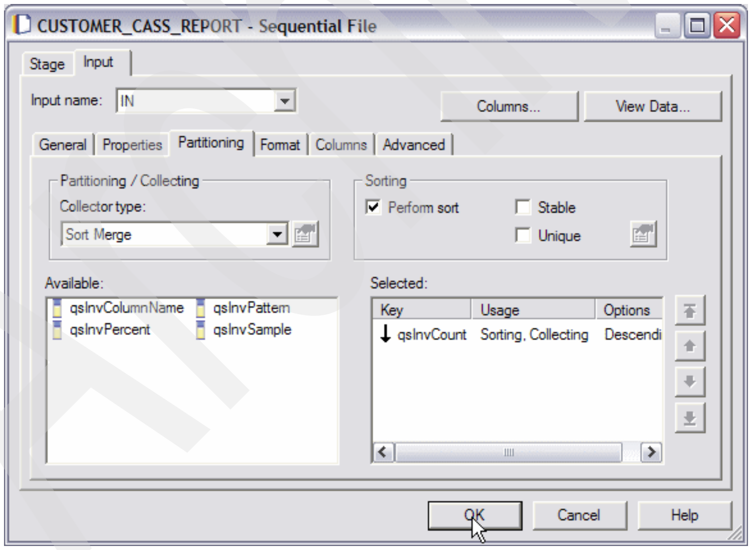

INV_DPV_INFO - Investigate

Output name: IN

Expression	Column Name
Investigate(qsInvColumnName)	qsInvColumnName
Investigate(qsInvPattern)	qsInvPattern
Investigate(qsInvSample)	qsInvSample
Investigate(qsInvCount)	qsInvCount
Investigate(qsInvPercent)	qsInvPercent

Derivation	Column Name
Investigate(qsInvColumnName)	qsInvColumnName
Investigate(qsInvPattern)	qsInvPattern
Investigate(qsInvSample)	qsInvSample
Investigate(qsInvCount)	qsInvCount
Investigate(qsInvPercent)	qsInvPercent

Figure 1-236 Create J06_INVCC_CASS job 8/12


CUSTOMER_CASS_REPORT - Sequential File

Input name: IN

Collector type: Sort Merge

Sorting: [✓ Perform sort] Stable Unique

Available: qsInvColumnName, qsInvPattern, qsInvPercent, qsInvSample

Selected:

Key	Usage	Options
qsInvCount	Sorting, Collecting	Descendi

Figure 1-237 Create J06_INVCC_CASS job 9/12

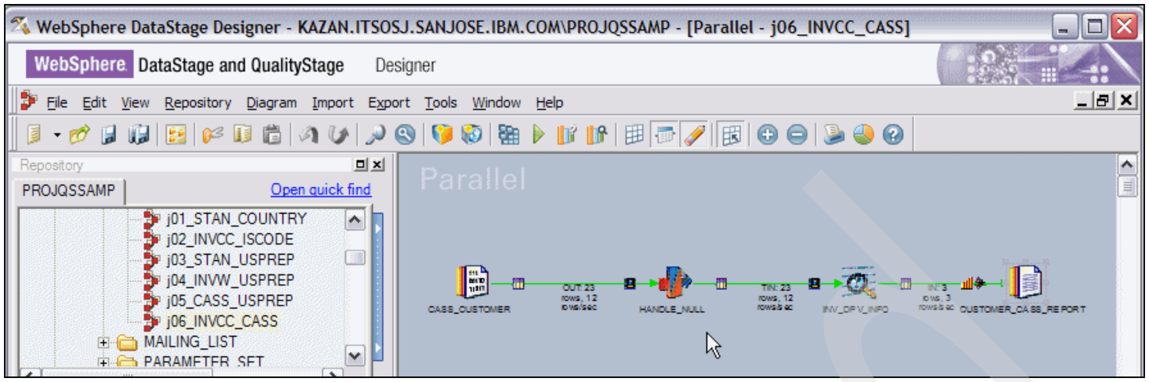
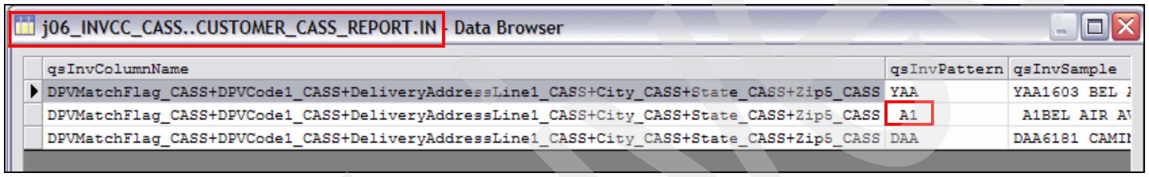
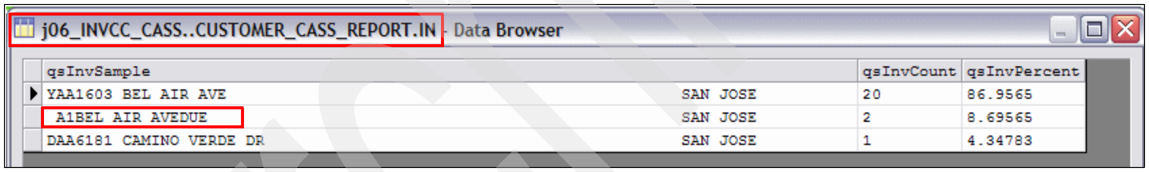

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j06_INVCC_CASS]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Repository
PROJQSSAMP
Open quick find
j01_STAN_COUNTRY
j02_INVCC_ISCODE
j03_STAN_USPREP
j04_INVW_USPREP
j05_CASS_USPREP
j06_INVCC_CASS
MAILING_LIST
PARAMETER_SET
Parallel
CASS_CUSTOMER
OUT: 23 rows, 12 rows/sec
HANDLE_NULL
TIN: 23 rows, 12 rows/sec
INV_DPV_INFO
IN: 3 rows, 3 rows/sec
CUSTOMER_CASS_REPORT

Figure 1-238 Create J06_INVCC_CASS job 10/12


j06_INVCC_CASS..CUSTOMER_CASS_REPORT.IN - Data Browser

qsInvColumnName	qsInvPattern	qsInvSample
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_CASS+City_CASS+State_CASS+Zip5_CASS	YAA	YAA1603 BEL A
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_CASS+City_CASS+State_CASS+Zip5_CASS	A1	A1BEL AIR AV
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_CASS+City_CASS+State_CASS+Zip5_CASS	DAA	DAA6181 CAMIN

Figure 1-239 Create J06_INVCC_CASS job 11/12


j06_INVCC_CASS..CUSTOMER_CASS_REPORT.IN - Data Browser

qsInvSample		qsInvCount	qsInvPercent
YAA1603 BEL AIR AVE	SAN JOSE	20	86.9565
A1BEL AIR AVEDUE	SAN JOSE	2	8.69565
DAA6181 CAMINO VERDE DR	SAN JOSE	1	4.34783

Figure 1-240 Create J06_INVCC_CASS job 12/12

J07_STAN_CUSTOMER_Domain_Preprocessor

In this job, we standardize the name and address contents of the output of job “J05_CASS_USPREP” on page 219 using the domain-preprocessor USPREP rule set. We also add a column to the output that only had the first three characters of the ZIP code using a Transformer stage. This new column (ZIP3) is used as a blocking variable in the following matching stage.

After CASS validates and corrects the address fields but not the name fields, the Standardize stage is run again with the domain-preprocessor rule set USPREP on the name fields and CASS corrected address fields into Name, Address, and Area domains.

Figure 1-241 on page 236 through Figure 1-249 on page 241 describe the steps using Designer Client to build and execute the DataStage job to perform this task.

The steps are as follows:

1. Figure 1-241 on page 236 shows the various stages that are used in this job, including the data set that was created in “J05_CASS_USPREP” on page 219, a Standardize stage, a Transformer stage to add a column, and an output Data Set stage. We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object created earlier.
2. Configure the input CASS_CUSTOMER data set with the table definition that was saved in “J05_CASS_USPREP” on page 219. We do not repeat this process here because it is similar to the steps shown in Figure 1-138 on page 172 through Figure 1-142 on page 174.
3. Next, configure the Standardize stage with the domain preprocessor USPREP rule set as described here.

   Note: This process is somewhat similar to the configuration of the Standardize stage in “J03_STAN_USPREP” on page 196, except for differences in the columns input for processing. Therefore, we cover the differences here.

   In the Standardize Rule Process window, the Rule Set shows USPREP.SET. Insert literals ZQNAMEZQ, ZQPUTAZQ[23], and ZQPUTRZQ[24] between columns TITLE, FNAME, LNAME, DeliveryAddressLine1_CASS, City_CASS, State_CASS, and Zip5_CASS in ADDR as shown in Figure 1-242 on page 237 through Figure 1-245 on page 238.

   The literal placement directs the following actions:

   – Fields TITLE, FNAME and LNAME are to default to the Name domain

   – Field DeliveryAddressLine1_CASS defaults to the Address domain

   – Fields City_CASS, State_CASS, and Zip5_CASS defaults to the Area domain
4. Configure the ADD_ZIP3 Transformer stage to add a column that only contains the first three digits of the 5-digit ZIP5_CASS field. Figure 1-246 on page 239 shows the ADD_ZIP3 - Transformer Stage window with the addition of the new column ZIP3 to the output.
5. Finally, configure the output CUSTOMER_CASS_USPREP data set object. We do not repeat this process here.

[23] ZQPUTAZQ defaults the entire field to the Address Domain automatically.
[24] ZQPUTRZQ defaults the entire field to the Area Domain automatically.

6. After saving this job j07_STAN_CUSTOMER_Domain_Preprocessor, compiling and running it, you view the content of the CUSTOMER_CASS_USPREP data set object as shown in Figure 1-247 on page 240 through Figure 1-249 on page 241.

   As discussed before, this report shows the following:

   – Columns NameDomain_USPREP (contains prefix, first name, last name, and suffix tokens), AddressDomain_USPREP (contains apartment, street name, and street type tokens), and AreaDomain_USPREP (contains state and ZIP code tokens) that were parsed from the input columns.

   – InputPattern_USPREP and OutboundPattern_USPREP columns that contain the patterns that are generated after processing the name and address columns in the input file.

A visual analysis of the report shows no actionable items. In the real world, the volume of data would be too large to attempt a visual analysis of the report. Therefore, proceed to “J08_STAN_CUSTOMER_Domain_Specific” on page 241.

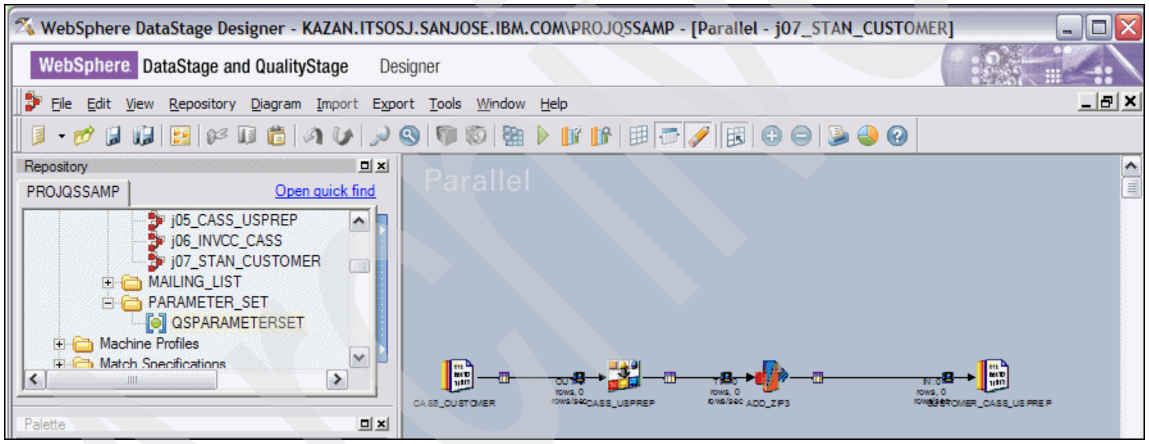
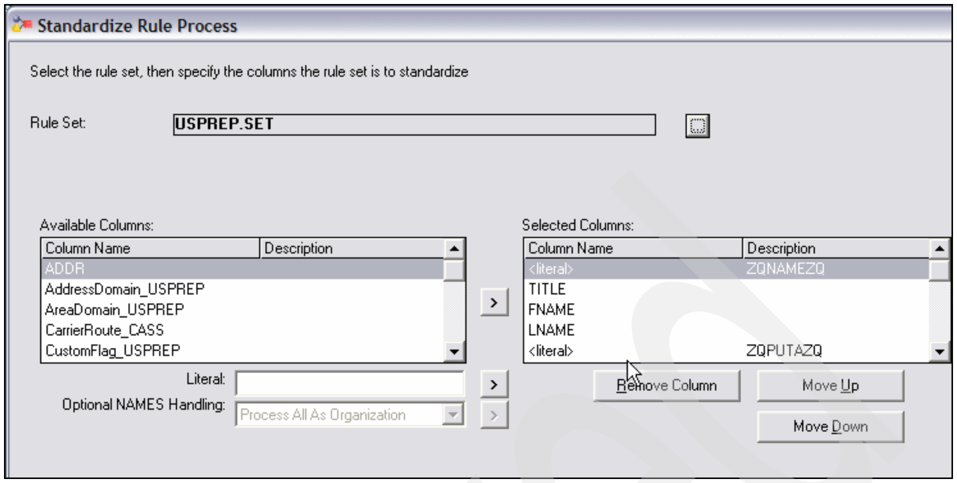


Figure 1-241 Create J07_STAN_CUSTOMER_Domain_Preprocessor job 1/9


Standardize Rule Process

Select the rule set, then specify the columns the rule set is to standardize

Rule Set: USPREP.SET

Available Columns:

Column Name	Description
ADDR	
AddressDomain_USPREP	
AreaDomain_USPREP	
CarrierRoute_CASS	
CustomFlag_USPREP	

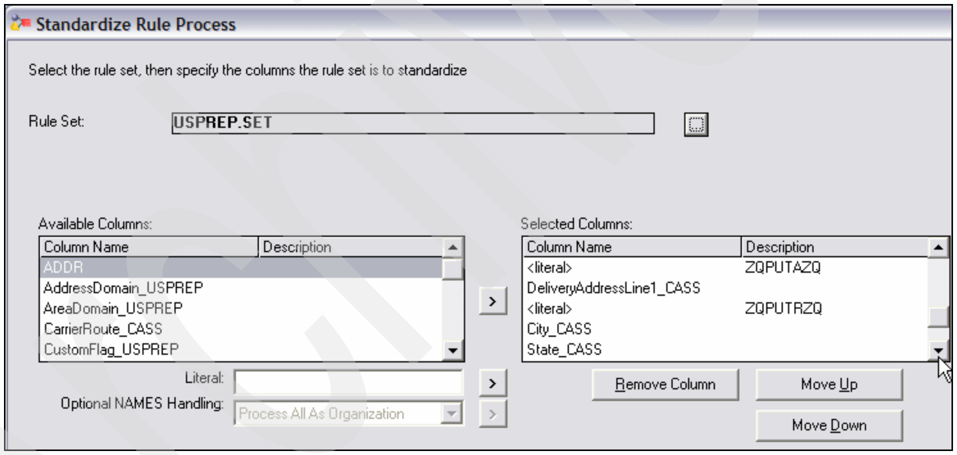
Selected Columns:

Column Name	Description
<literal>	ZQNAMEZQ
TITLE	
FNAME	
LNAME	
<literal>	ZQPUTAZQ

Literal:

Optional NAMES Handling: Process All As Organization

Figure 1-242 Create J07_STAN_CUSTOMER_Domain_Preprocessor job 2/9


Standardize Rule Process

Select the rule set, then specify the columns the rule set is to standardize

Rule Set: USPREP.SET

Available Columns:

Column Name	Description
ADDR	
AddressDomain_USPREP	
AreaDomain_USPREP	
CarrierRoute_CASS	
CustomFlag_USPREP	

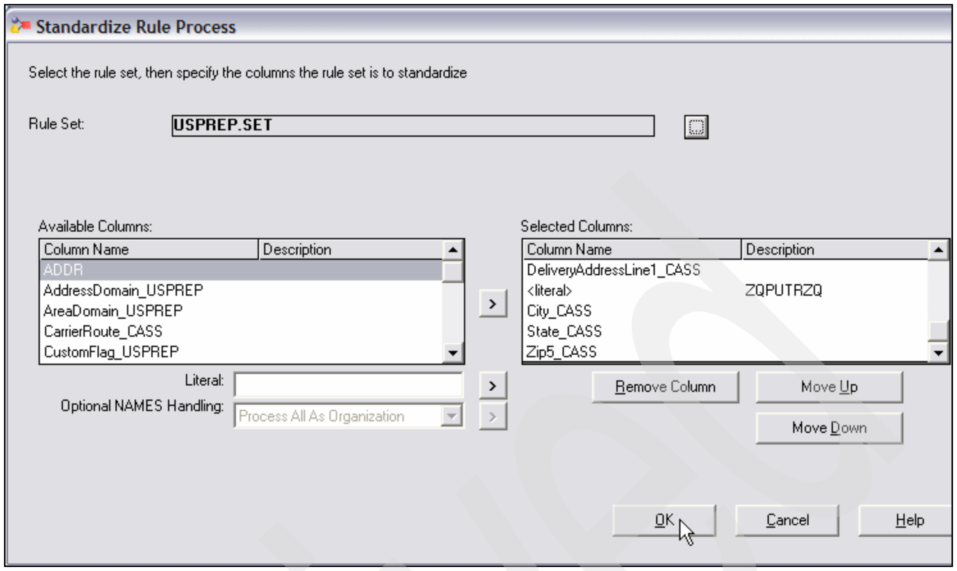
Selected Columns:

Column Name	Description
<literal>	ZQPUTAZQ
DeliveryAddressLine1_CASS	
<literal>	ZQPUTRZQ
City_CASS	
State_CASS	

Literal:

Optional NAMES Handling: Process All As Organization

Figure 1-243 Create J07_STAN_CUSTOMER_Domain_Preprocessor job 3/9


Standardize Rule Process

Select the rule set, then specify the columns the rule set is to standardize

Rule Set: USPREP.SET

Available Columns:

Column Name	Description
ADDR	
AddressDomain_USPREP	
AreaDomain_USPREP	
CarrierRoute_CASS	
CustomFlag_USPREP	

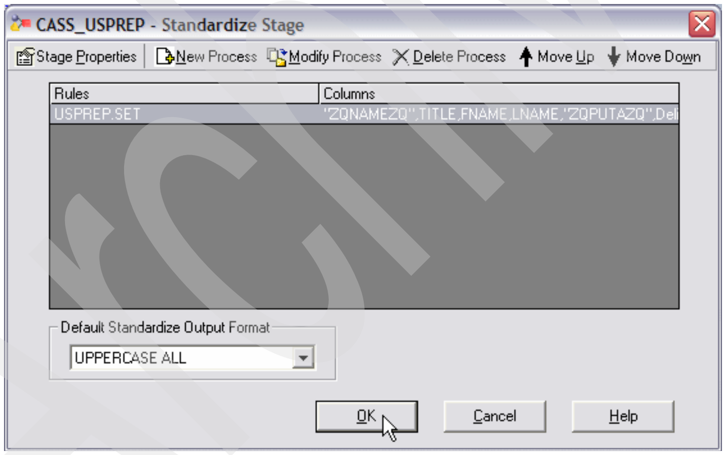
Selected Columns:

Column Name	Description
DeliveryAddressLine1_CASS	
<literal>	ZQPUTRZQ
City_CASS	
State_CASS	
Zip5_CASS	

Literal:

Optional NAMES Handling: Process All As Organization

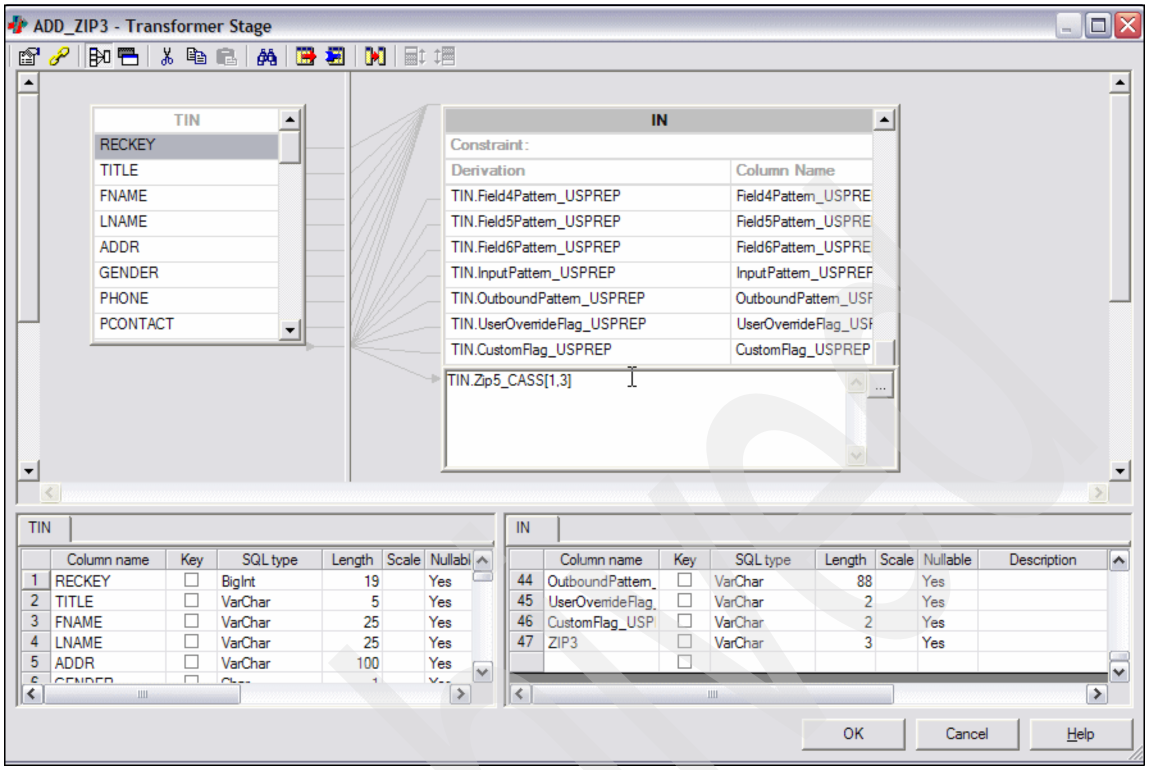
Figure 1-244   Create J07_STAN_CUSTOMER_Domain_Preprocessor job 4/9


CASS_USPREP - Standardize Stage

Rules	Columns
USPREP.SET	"ZQNAMEZQ",TITLE,FNAME,LNAME,"ZQPUTAZQ",Del

Default Standardize Output Format: UPPERCASE ALL

Figure 1-245   Create J07_STAN_CUSTOMER_Domain_Preprocessor job 5/9



ADD_ZIP3 - Transformer Stage

TIN: RECKEY, TITLE, FNAME, LNAME, ADDR, GENDER, PHONE, PCONTACT

IN

Constraint:

Derivation	Column Name
TIN.Field4Pattern_USPREP	Field4Pattern_USPRE
TIN.Field5Pattern_USPREP	Field5Pattern_USPRE
TIN.Field6Pattern_USPREP	Field6Pattern_USPRE
TIN.InputPattern_USPREP	InputPattern_USPREF
TIN.OutboundPattern_USPREP	OutboundPattern_USF
TIN.UserOverrideFlag_USPREP	UserOverrideFlag_USF
TIN.CustomFlag_USPREP	CustomFlag_USPREP

TIN.Zip5_CASS[1,3]

TIN

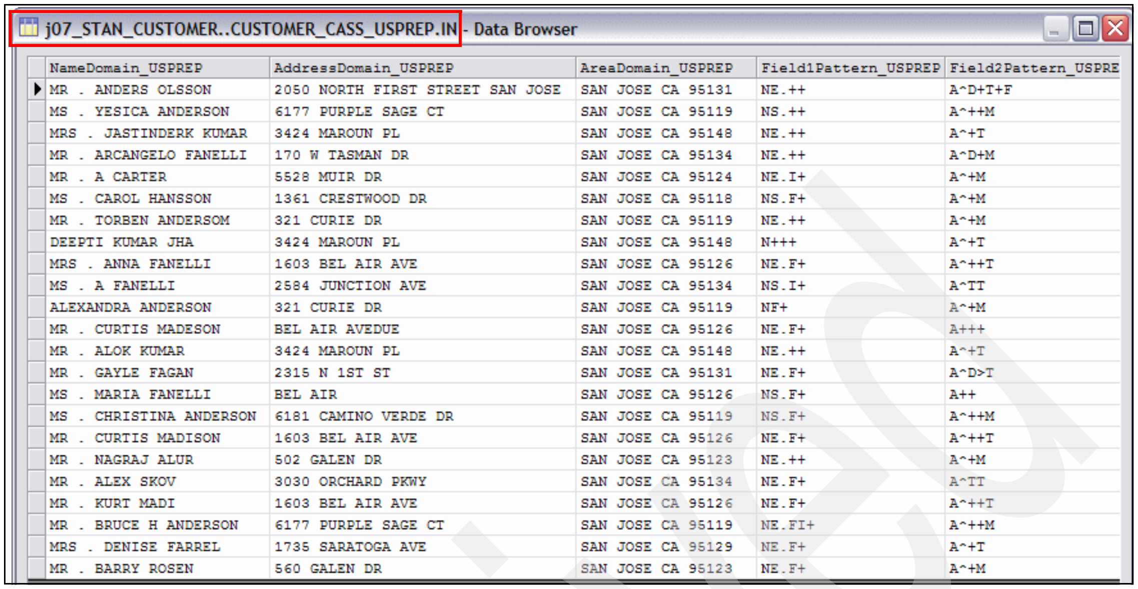
	Column name	Key	SQL type	Length	Scale	Nullabl
1	RECKEY	☐	BigInt	19		Yes
2	TITLE	☐	VarChar	5		Yes
3	FNAME	☐	VarChar	25		Yes
4	LNAME	☐	VarChar	25		Yes
5	ADDR	☐	VarChar	100		Yes

IN

	Column name	Key	SQL type	Length	Scale	Nullable	Description
44	OutboundPattern_	☐	VarChar	88		Yes	
45	UserOverrideFlag_	☐	VarChar	2		Yes	
46	CustomFlag_USP	☐	VarChar	2		Yes	
47	ZIP3	☐	VarChar	3		Yes	

OK Cancel Help

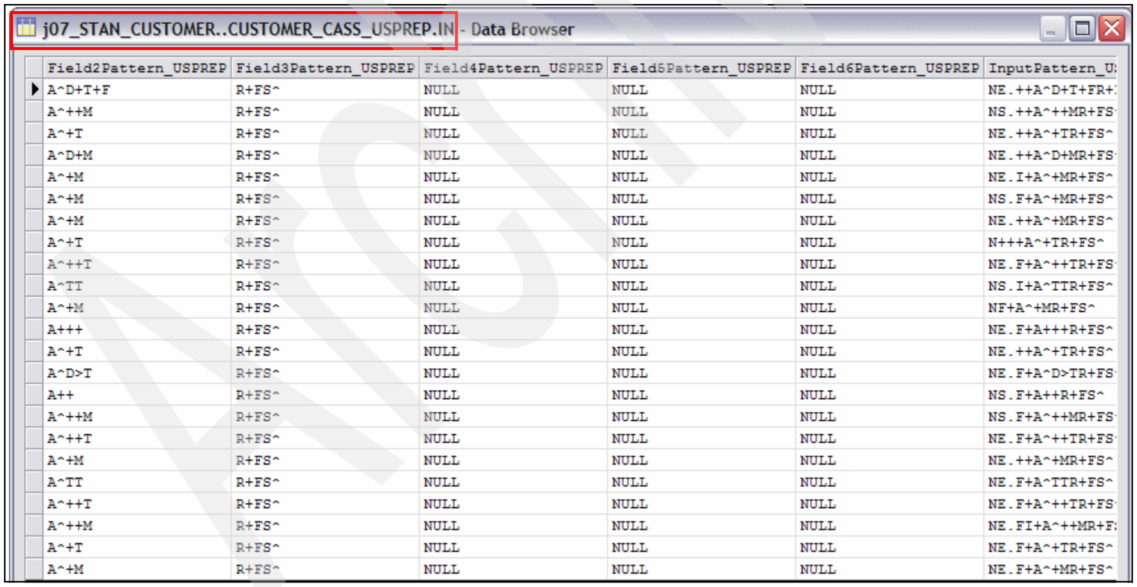
Figure 1-246 Create J07_STAN_CUSTOMER_Domain_Preprocessor job 6/9



j07_STAN_CUSTOMER..CUSTOMER_CASS_USPREP.IN - Data Browser

NameDomain_USPREP	AddressDomain_USPREP	AreaDomain_USPREP	Field1Pattern_USPREP	Field2Pattern_USPRE
MR . ANDERS OLSSON	2050 NORTH FIRST STREET SAN JOSE	SAN JOSE CA 95131	NE.++	A^D+T+F
MS . YESICA ANDERSON	6177 PURPLE SAGE CT	SAN JOSE CA 95119	NS.++	A^++M
MRS . JASTINDERK KUMAR	3424 MAROUN PL	SAN JOSE CA 95148	NE.++	A^+T
MR . ARCANGELO FANELLI	170 W TASMAN DR	SAN JOSE CA 95134	NE.++	A^D+M
MR . A CARTER	5528 MUIR DR	SAN JOSE CA 95124	NE.I+	A^+M
MS . CAROL HANSSON	1361 CRESTWOOD DR	SAN JOSE CA 95118	NS.F+	A^+M
MR . TORBEN ANDERSOM	321 CURIE DR	SAN JOSE CA 95119	NE.++	A^+M
DEEPTI KUMAR JHA	3424 MAROUN PL	SAN JOSE CA 95148	N+++	A^+T
MRS . ANNA FANELLI	1603 BEL AIR AVE	SAN JOSE CA 95126	NE.F+	A^++T
MS . A FANELLI	2584 JUNCTION AVE	SAN JOSE CA 95134	NS.I+	A^TT
ALEXANDRA ANDERSON	321 CURIE DR	SAN JOSE CA 95119	NF+	A^+M
MR . CURTIS MADESON	BEL AIR AVEDUE	SAN JOSE CA 95126	NE.F+	A+++
MR . ALOK KUMAR	3424 MAROUN PL	SAN JOSE CA 95148	NE.++	A^+T
MR . GAYLE FAGAN	2315 N 1ST ST	SAN JOSE CA 95131	NE.F+	A^D>T
MS . MARIA FANELLI	BEL AIR	SAN JOSE CA 95126	NS.F+	A++
MS . CHRISTINA ANDERSON	6181 CAMINO VERDE DR	SAN JOSE CA 95119	NS.F+	A^++M
MR . CURTIS MADISON	1603 BEL AIR AVE	SAN JOSE CA 95126	NE.F+	A^++T
MR . NAGRAJ ALUR	502 GALEN DR	SAN JOSE CA 95123	NE.++	A^+M
MR . ALEX SKOV	3030 ORCHARD PKWY	SAN JOSE CA 95134	NE.F+	A^TT
MR . KURT MADI	1603 BEL AIR AVE	SAN JOSE CA 95126	NE.F+	A^++T
MR . BRUCE H ANDERSON	6177 PURPLE SAGE CT	SAN JOSE CA 95119	NE.FI+	A^++M
MRS . DENISE FARREL	1735 SARATOGA AVE	SAN JOSE CA 95129	NE.F+	A^+T
MR . BARRY ROSEN	560 GALEN DR	SAN JOSE CA 95123	NE.F+	A^+M

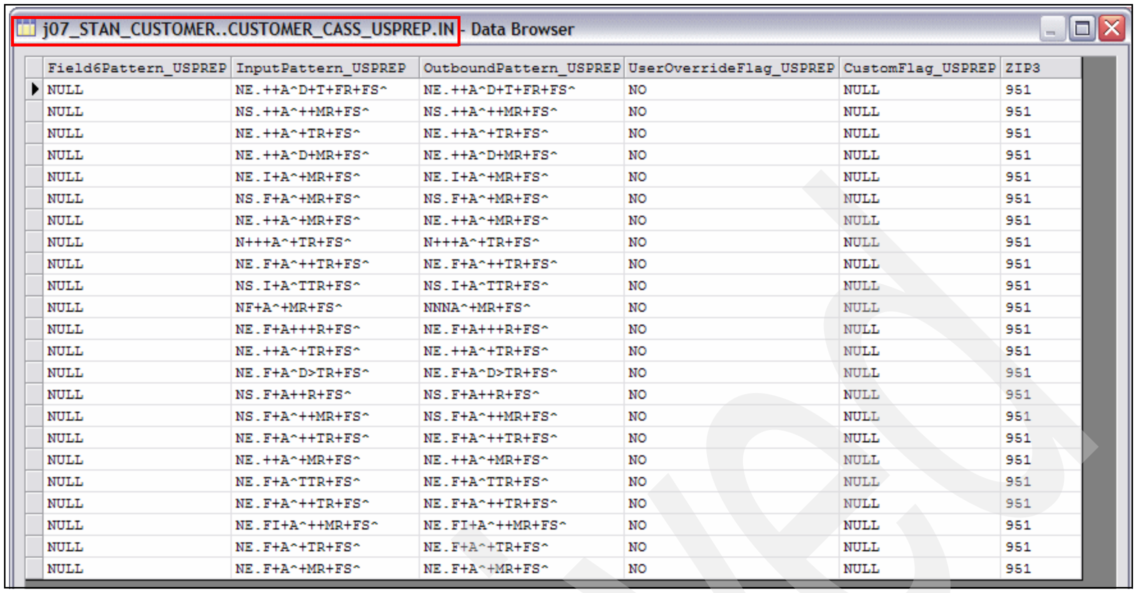
Figure 1-247 Create J07_STAN_CUSTOMER_Domain_Preprocessor job 7/9



j07_STAN_CUSTOMER..CUSTOMER_CASS_USPREP.IN - Data Browser

Field2Pattern_USPREP	Field3Pattern_USPREP	Field4Pattern_USPREP	Field5Pattern_USPREP	Field6Pattern_USPREP	InputPattern_U
A^D+T+F	R+FS^	NULL	NULL	NULL	NE.++A^D+T+FR+
A^++M	R+FS^	NULL	NULL	NULL	NS.++A^++MR+FS
A^+T	R+FS^	NULL	NULL	NULL	NE.++A^+TR+FS^
A^D+M	R+FS^	NULL	NULL	NULL	NE.++A^D+MR+FS
A^+M	R+FS^	NULL	NULL	NULL	NE.I+A^+MR+FS^
A^+M	R+FS^	NULL	NULL	NULL	NS.F+A^+MR+FS^
A^+M	R+FS^	NULL	NULL	NULL	NE.++A^+MR+FS^
A^+T	R+FS^	NULL	NULL	NULL	N+++A^+TR+FS^
A^++T	R+FS^	NULL	NULL	NULL	NE.F+A^++TR+FS
A^TT	R+FS^	NULL	NULL	NULL	NS.I+A^TTR+FS^
A^+M	R+FS^	NULL	NULL	NULL	NF+A^+MR+FS^
A+++	R+FS^	NULL	NULL	NULL	NE.F+A+++R+FS^
A^+T	R+FS^	NULL	NULL	NULL	NE.++A^+TR+FS^
A^D>T	R+FS^	NULL	NULL	NULL	NE.F+A^D>TR+FS
A++	R+FS^	NULL	NULL	NULL	NS.F+A++R+FS^
A^++M	R+FS^	NULL	NULL	NULL	NS.F+A^++MR+FS
A^++T	R+FS^	NULL	NULL	NULL	NE.F+A^++TR+FS
A^+M	R+FS^	NULL	NULL	NULL	NE.++A^+MR+FS^
A^TT	R+FS^	NULL	NULL	NULL	NE.F+A^TTR+FS^
A^++T	R+FS^	NULL	NULL	NULL	NE.F+A^++TR+FS
A^++M	R+FS^	NULL	NULL	NULL	NE.FI+A^++MR+F
A^+T	R+FS^	NULL	NULL	NULL	NE.F+A^+TR+FS^
A^+M	R+FS^	NULL	NULL	NULL	NE.F+A^+MR+FS^

Figure 1-248 Create J07_STAN_CUSTOMER_Domain_Preprocessor job 8/9


j07_STAN_CUSTOMER..CUSTOMER_CASS_USPREP.IN - Data Browser

Field6Pattern_USPREP	InputPattern_USPREP	OutboundPattern_USPREP	UserOverrideFlag_USPREP	CustomFlag_USPREP	ZIP3
NULL	NE.++A^D+T+FR+FS^	NE.++A^D+T+FR+FS^	NO	NULL	951
NULL	NS.++A^++MR+FS^	NS.++A^++MR+FS^	NO	NULL	951
NULL	NE.++A^+TR+FS^	NE.++A^+TR+FS^	NO	NULL	951
NULL	NE.++A^D+MR+FS^	NE.++A^D+MR+FS^	NO	NULL	951
NULL	NE.I+A^+MR+FS^	NE.I+A^+MR+FS^	NO	NULL	951
NULL	NS.F+A^+MR+FS^	NS.F+A^+MR+FS^	NO	NULL	951
NULL	NE.++A^+MR+FS^	NE.++A^+MR+FS^	NO	NULL	951
NULL	N+++A^+TR+FS^	N+++A^+TR+FS^	NO	NULL	951
NULL	NE.F+A^++TR+FS^	NE.F+A^++TR+FS^	NO	NULL	951
NULL	NS.I+A^TTR+FS^	NS.I+A^TTR+FS^	NO	NULL	951
NULL	NF+A^+MR+FS^	NNNA^+MR+FS^	NO	NULL	951
NULL	NE.F+A+++R+FS^	NE.F+A+++R+FS^	NO	NULL	951
NULL	NE.++A^+TR+FS^	NE.++A^+TR+FS^	NO	NULL	951
NULL	NE.F+A^D>TR+FS^	NE.F+A^D>TR+FS^	NO	NULL	951
NULL	NS.F+A++R+FS^	NS.F+A++R+FS^	NO	NULL	951
NULL	NS.F+A^++MR+FS^	NS.F+A^++MR+FS^	NO	NULL	951
NULL	NE.F+A^++TR+FS^	NE.F+A^++TR+FS^	NO	NULL	951
NULL	NE.++A^+MR+FS^	NE.++A^+MR+FS^	NO	NULL	951
NULL	NE.F+A^TTR+FS^	NE.F+A^TTR+FS^	NO	NULL	951
NULL	NE.F+A^++TR+FS^	NE.F+A^++TR+FS^	NO	NULL	951
NULL	NE.FI+A^++MR+FS^	NE.FI+A^++MR+FS^	NO	NULL	951
NULL	NE.F+A^+TR+FS^	NE.F+A^+TR+FS^	NO	NULL	951
NULL	NE.F+A^+MR+FS^	NE.F+A^+MR+FS^	NO	NULL	951

Figure 1-249 Create J07_STAN_CUSTOMER_Domain_Preprocessor job 9/9

J08_STAN_CUSTOMER_Domain_Specific

In this step, we standardize the name and address contents of the output of job "J07_STAN_CUSTOMER_Domain_Preprocessor" on page 234 using the domain-specific USNAME (with column NameDomain_USPREP), USADDR (with column AddressDomain_USPREP) and USAREA (with column AreaDomain_USPREP) rule sets. Three separate processes are defined—one for each rule set.

Note: Here again, we had to introduce a null handling Transformer stage to circumvent a bug.

Figure 1-250 on page 243 through Figure 1-271 on page 252 describe the steps using Designer Client to build and execute the DataStage job to perform this task.

The steps are as follows:

1. Figure 1-250 on page 243 shows the various stages that are used in this job, including the data set that was created in "J07_STAN_CUSTOMER_Domain_Preprocessor" on page 234, a Standardize stage, a Transformer stage to handle nulls, an output Data Set stage. We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object that was created earlier.

2. Configure the input CUSTOMER_CASS_USPREP data set that was created in “J07_STAN_CUSTOMER_Domain_Preprocessor” on page 234. We do not repeat this process here.
3. Next, configure the Standardize stage with the three domain specific rule sets: USNAME, USADDR, and USAREA.

   Note: This process is somewhat similar to the configuration of the Standardize stage that we describe in “J03_STAN_USPREP” on page 196, except for differences in the columns input for processing. Therefore, we describe the differences here.

   Because three rule sets are involved, you define three separate processes—one for each rule set.

   – Figure 1-251 on page 243 shows the USNAME rule set for the NameDomain_USPREP column
   – Figure 1-252 on page 243 shows the USADDR rule set for the AddressDomain_USPREP column
   – Figure 1-253 on page 244 shows the USAREA rule set for the AreaDomain_USPREP column

   Figure 1-254 on page 244 shows the three rules that are used in this Standardize job.
4. Configure the HANDLE_NULL Transformer stage to convert nulls to spaces similar that are described in “J06_INVCC_CASS” on page 228. Figure 1-255 on page 244 shows the Derivation Substitution for each of the columns.
5. Finally, configure the output STAN_CUSTOMER data set object. We do not repeat this process here.
6. After saving, compiling, and running this job j08_STAN_CUSTOMER_Domain_Specific, view the content of the STAN_CUSTOMER data set object as shown in Figure 1-269 on page 251 through Figure 1-271 on page 252.

   This report shows Standardize Stage added columns such as FirstName_USNAME and PrimaryName_USNAME, InputPattern_USNAME, UnhandledPattern_USNAME, UnhandledData_USNAME, HouseNumber_USADDR, StreetName_USADDR, UnitType_USADDR, CityName_USAREA, ZipCode_USAREA, and CountryCode_USAREA.

Now, proceed to that identifies any unhandled patterns and classifications by performing an investigate, as described in “J09_INVCC_STAN_CUSTOMER” on page 253.

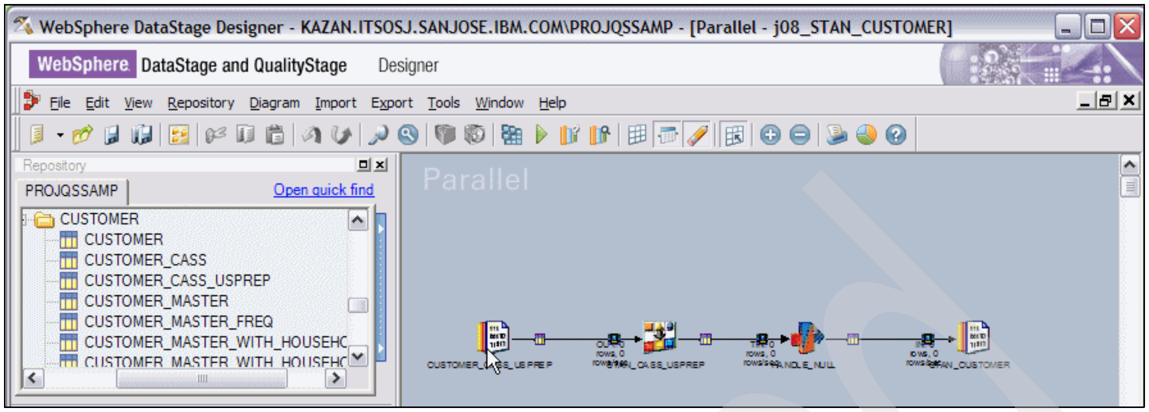
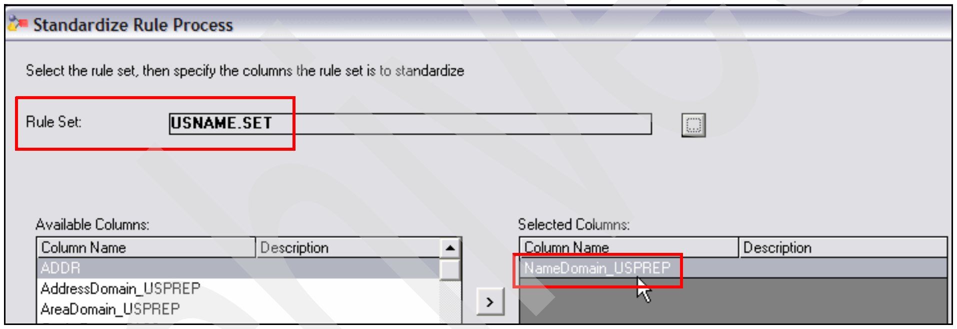

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j08_STAN_CUSTOMER]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Repository
PROJQSSAMP
Open quick find
CUSTOMER
CUSTOMER
CUSTOMER_CASS
CUSTOMER_CASS_USPREP
CUSTOMER_MASTER
CUSTOMER_MASTER_FREQ
CUSTOMER_MASTER_WITH_HOUSEHO
CUSTOMER_MASTER_WITH_HOUSEHO
Parallel
CUSTOMER_CASS_USPREP
STAN_CASS_USPREP
HANDLE_NULL
STAN_CUSTOMER

Figure 1-250 Create J08_STAN_CUSTOMER_Domain_Specific 1/22



Select the rule set, then specify the columns the rule set is to standardize

Rule Set: USNAME.SET

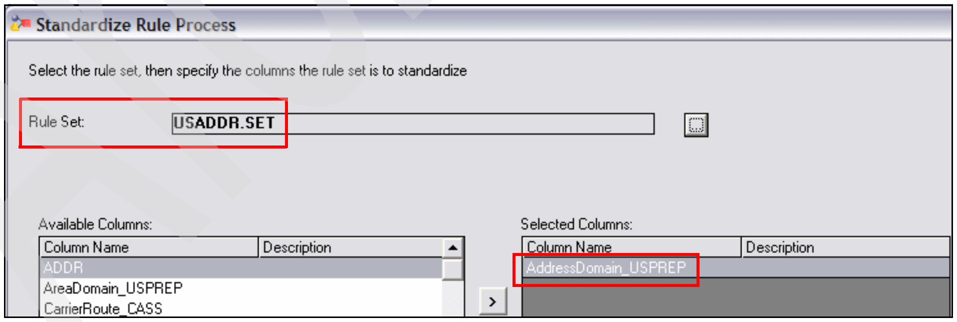
Available Columns:

Column Name	Description
ADDR	
AddressDomain_USPREP	
AreaDomain_USPREP	

Selected Columns:

Column Name	Description
NameDomain_USPREP	

Figure 1-251 Create J08_STAN_CUSTOMER_Domain_Specific 2/22



Select the rule set, then specify the columns the rule set is to standardize

Rule Set: USADDR.SET

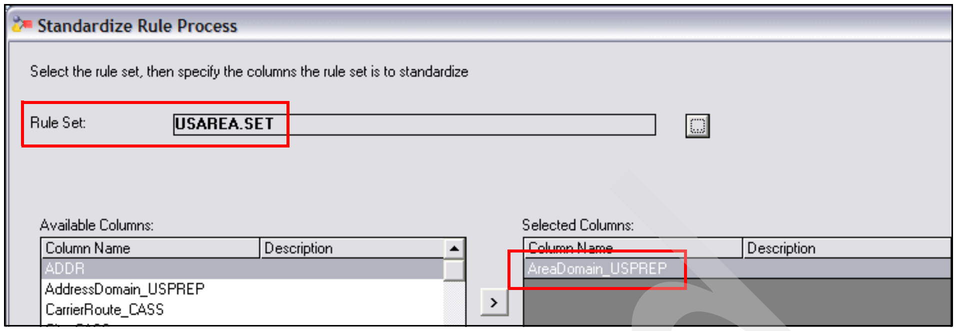
Available Columns:

Column Name	Description
ADDR	
AreaDomain_USPREP	
CarrierRoute_CASS	

Selected Columns:

Column Name	Description
AddressDomain_USPREP	

Figure 1-252 Create J08_STAN_CUSTOMER_Domain_Specific 3/22


Standardize Rule Process

Select the rule set, then specify the columns the rule set is to standardize

Rule Set: USAREA.SET

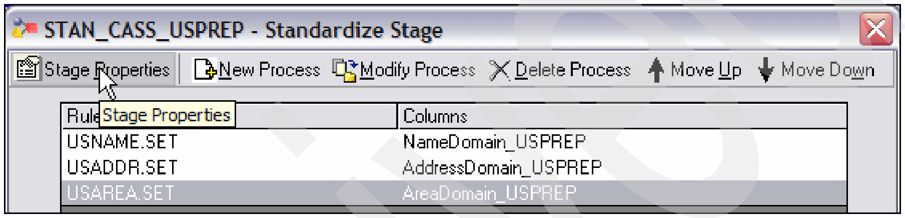
Available Columns:

Column Name	Description
ADDR	
AddressDomain_USPREP	
CarrierRoute_CASS	

Selected Columns:

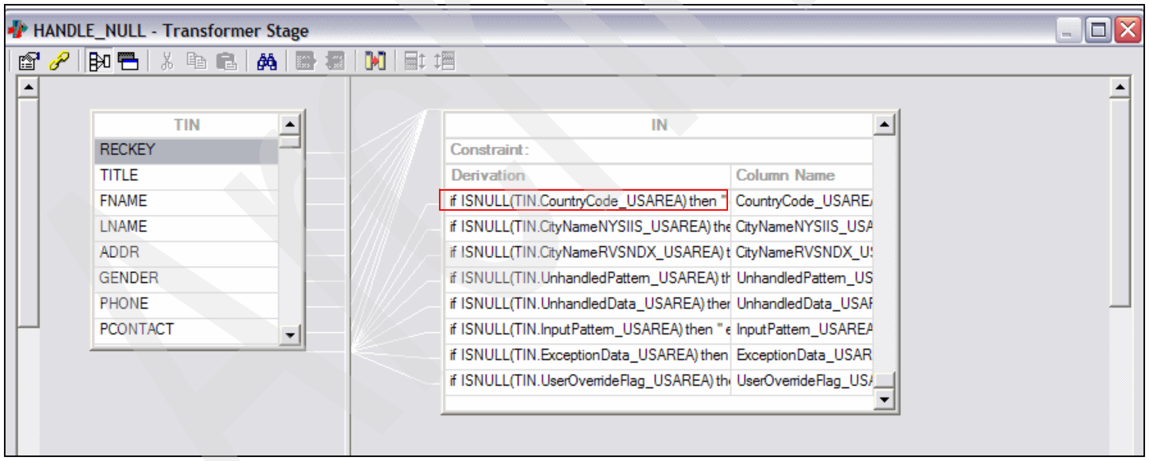
Column Name	Description
AreaDomain_USPREP	

Figure 1-253 Create J08_STAN_CUSTOMER_Domain_Specific 4/22


STAN_CASS_USPREP - Standardize Stage

Rule Set	Columns
USNAME.SET	NameDomain_USPREP
USADDR.SET	AddressDomain_USPREP
USAREA.SET	AreaDomain_USPREP

Figure 1-254 Create J08_STAN_CUSTOMER_Domain_Specific 5/22


HANDLE_NULL - Transformer Stage

TIN

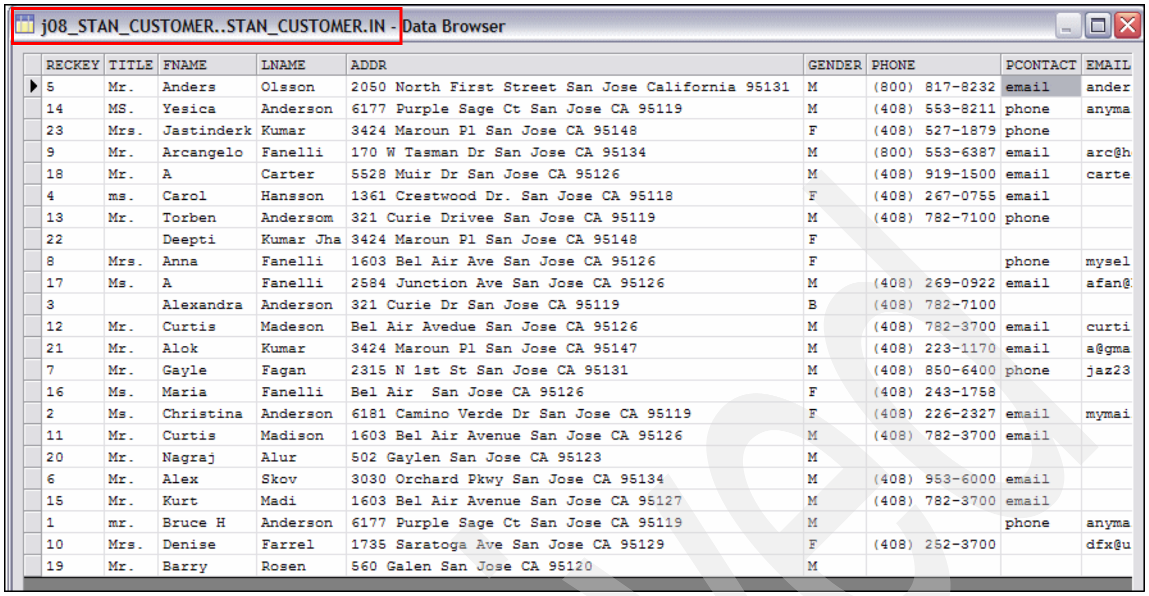
RECKEY
TITLE
FNAME
LNAME
ADDR
GENDER
PHONE
PCONTACT

IN

Constraint:

Derivation	Column Name
if ISNULL(TIN.CountryCode_USAREA) then "	CountryCode_USARE
if ISNULL(TIN.CityNameNYSIIS_USAREA) the	CityNameNYSIIS_USA
if ISNULL(TIN.CityNameRVSNDX_USAREA) t	CityNameRVSNDX_U
if ISNULL(TIN.UnhandledPattern_USAREA) th	UnhandledPattern_US
if ISNULL(TIN.UnhandledData_USAREA) ther	UnhandledData_USAF
if ISNULL(TIN.InputPattern_USAREA) then "e	InputPattern_USAREA
if ISNULL(TIN.ExceptionData_USAREA) then	ExceptionData_USAR
if ISNULL(TIN.UserOverrideFlag_USAREA) th	UserOverrideFlag_US

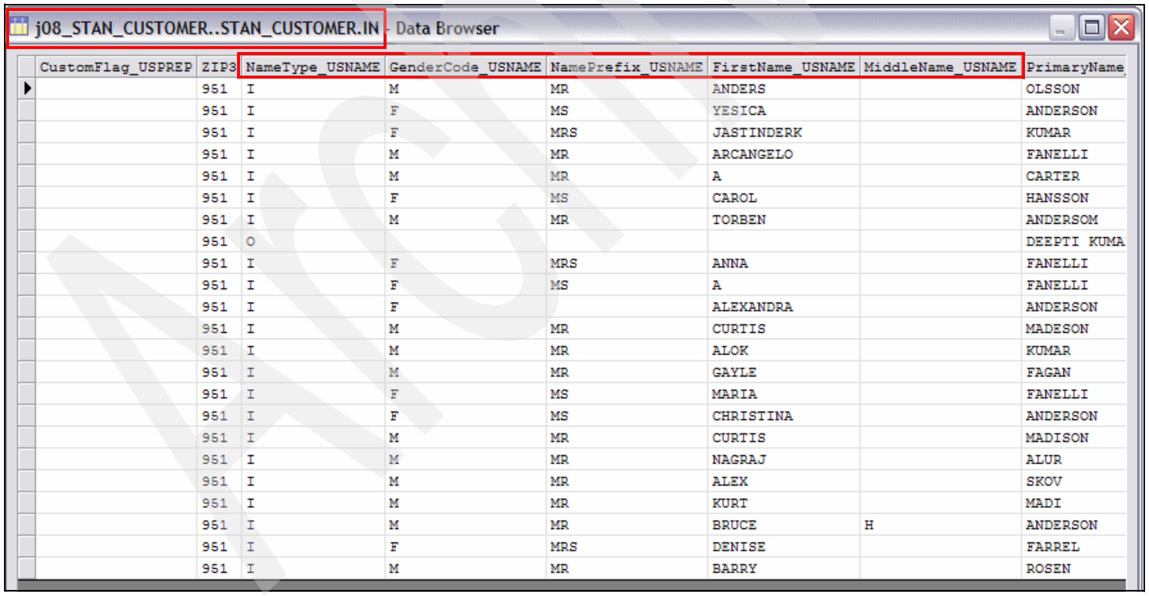
Figure 1-255 Create J08_STAN_CUSTOMER_Domain_Specific 6/22



j08_STAN_CUSTOMER..STAN_CUSTOMER.IN - Data Browser

RECKEY	TITLE	FNAME	LNAME	ADDR	GENDER	PHONE	PCONTACT	EMAIL
5	Mr.	Anders	Olsson	2050 North First Street San Jose California 95131	M	(800) 817-8232	email	ander
14	MS.	Yesica	Anderson	6177 Purple Sage Ct San Jose CA 95119	M	(408) 553-8211	phone	anyma
23	Mrs.	Jastinderk	Kumar	3424 Maroun Pl San Jose CA 95148	F	(408) 527-1879	phone	
9	Mr.	Arcangelo	Fanelli	170 W Tasman Dr San Jose CA 95134	M	(800) 553-6387	email	arc@h
18	Mr.	A	Carter	5528 Muir Dr San Jose CA 95126	M	(408) 919-1500	email	carte
4	ms.	Carol	Hansson	1361 Crestwood Dr. San Jose CA 95118	F	(408) 267-0755	email	
13	Mr.	Torben	Andersom	321 Curie Drivee San Jose CA 95119	M	(408) 782-7100	phone	
22		Deepti	Kumar Jha	3424 Maroun Pl San Jose CA 95148	F			
8	Mrs.	Anna	Fanelli	1603 Bel Air Ave San Jose CA 95126	F		phone	mysel
17	Ms.	A	Fanelli	2584 Junction Ave San Jose CA 95126	M	(408) 269-0922	email	afan@
3		Alexandra	Anderson	321 Curie Dr San Jose CA 95119	B	(408) 782-7100		
12	Mr.	Curtis	Madeson	Bel Air Avedue San Jose CA 95126	M	(408) 782-3700	email	curti
21	Mr.	Alok	Kumar	3424 Maroun Pl San Jose CA 95147	M	(408) 223-1170	email	a@gma
7	Mr.	Gayle	Fagan	2315 N 1st St San Jose CA 95131	M	(408) 850-6400	phone	jaz23
16	Ms.	Maria	Fanelli	Bel Air San Jose CA 95126	F	(408) 243-1758		
2	Ms.	Christina	Anderson	6181 Camino Verde Dr San Jose CA 95119	F	(408) 226-2327	email	mymai
11	Mr.	Curtis	Madison	1603 Bel Air Avenue San Jose CA 95126	M	(408) 782-3700	email	
20	Mr.	Nagraj	Alur	502 Gaylen San Jose CA 95123	M			
6	Mr.	Alex	Skov	3030 Orchard Pkwy San Jose CA 95134	M	(408) 953-6000	email	
15	Mr.	Kurt	Madi	1603 Bel Air Avenue San Jose CA 95127	M	(408) 782-3700	email	
1	mr.	Bruce H	Anderson	6177 Purple Sage Ct San Jose CA 95119	M		phone	anyma
10	Mrs.	Denise	Farrel	1735 Saratoga Ave San Jose CA 95129	F	(408) 252-3700		dfx@u
19	Mr.	Barry	Rosen	560 Galen San Jose CA 95120	M			

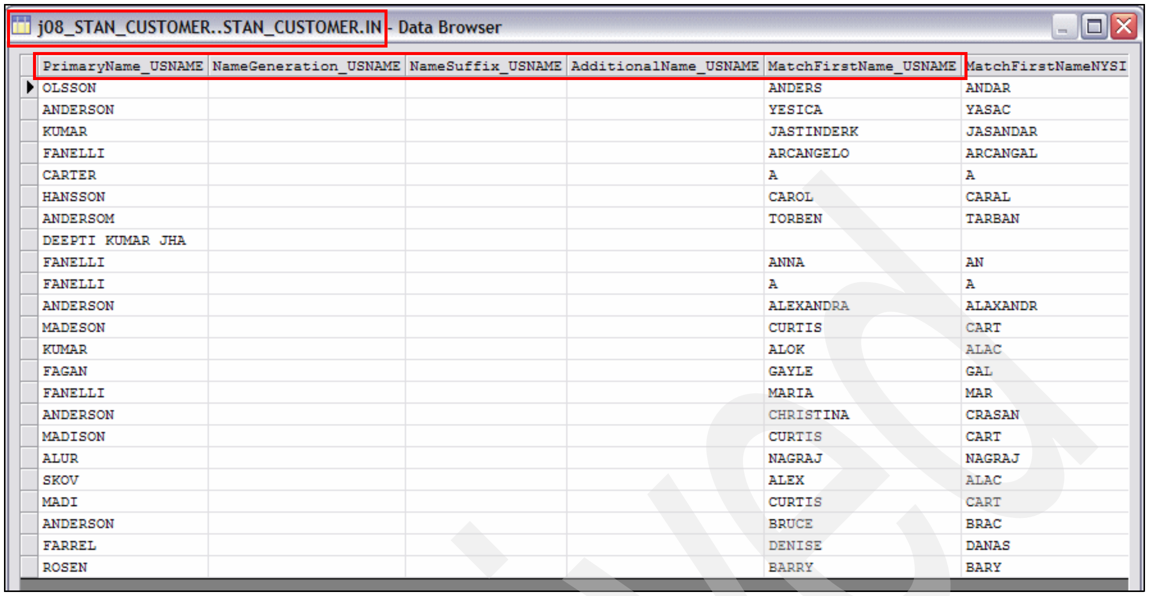
Figure 1-256   Create J08_STAN_CUSTOMER_Domain_Specific 7/22



j08_STAN_CUSTOMER..STAN_CUSTOMER.IN - Data Browser

CustomFlag_USPREP	ZIP3	NameType_USNAME	GenderCode_USNAME	NamePrefix_USNAME	FirstName_USNAME	MiddleName_USNAME	PrimaryName
	951	I	M	MR	ANDERS		OLSSON
	951	I	F	MS	YESICA		ANDERSON
	951	I	F	MRS	JASTINDERK		KUMAR
	951	I	M	MR	ARCANGELO		FANELLI
	951	I	M	MR	A		CARTER
	951	I	F	MS	CAROL		HANSSON
	951	I	M	MR	TORBEN		ANDERSOM
	951	O					DEEPTI KUMA
	951	I	F	MRS	ANNA		FANELLI
	951	I	F	MS	A		FANELLI
	951	I	F		ALEXANDRA		ANDERSON
	951	I	M	MR	CURTIS		MADESON
	951	I	M	MR	ALOK		KUMAR
	951	I	M	MR	GAYLE		FAGAN
	951	I	F	MS	MARIA		FANELLI
	951	I	F	MS	CHRISTINA		ANDERSON
	951	I	M	MR	CURTIS		MADISON
	951	I	M	MR	NAGRAJ		ALUR
	951	I	M	MR	ALEX		SKOV
	951	I	M	MR	KURT		MADI
	951	I	M	MR	BRUCE	H	ANDERSON
	951	I	F	MRS	DENISE		FARREL
	951	I	M	MR	BARRY		ROSEN

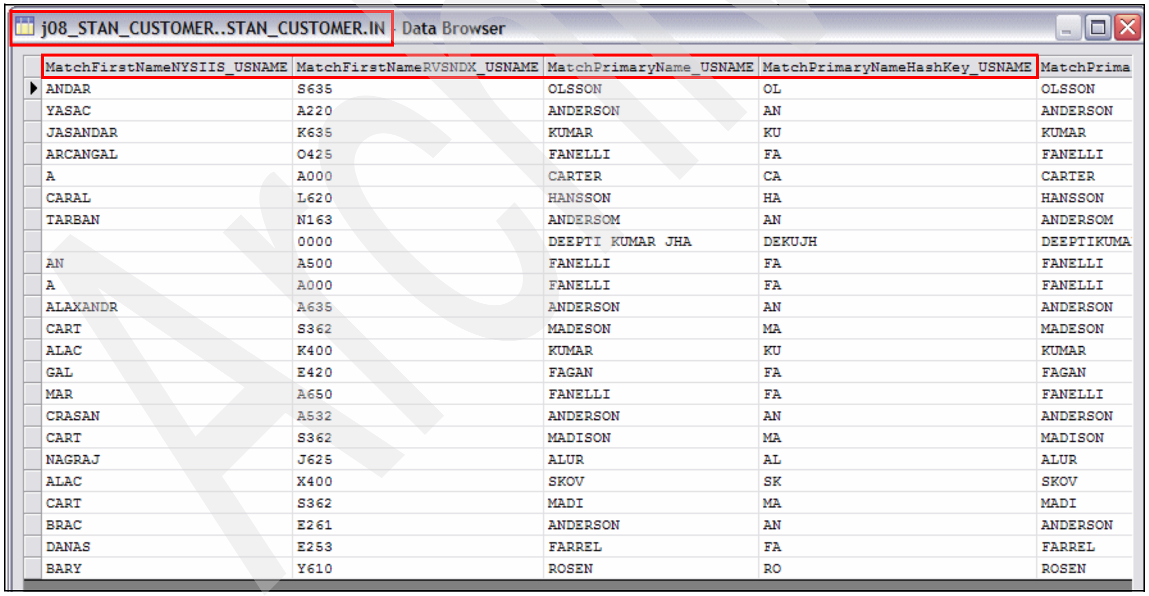
Figure 1-257   Create J08_STAN_CUSTOMER_Domain_Specific 8/22



j08_STAN_CUSTOMER..STAN_CUSTOMER.IN - Data Browser

PrimaryName_USNAME	NameGeneration_USNAME	NameSuffix_USNAME	AdditionalName_USNAME	MatchFirstName_USNAME	MatchFirstNameNYSI
OLSSON				ANDERS	ANDAR
ANDERSON				YESICA	YASAC
KUMAR				JASTINDERK	JASANDAR
FANELLI				ARCANGELO	ARCANGAL
CARTER				A	A
HANSSON				CAROL	CARAL
ANDERSOM				TORBEN	TARBAN
DEEPTI KUMAR JHA					
FANELLI				ANNA	AN
FANELLI				A	A
ANDERSON				ALEXANDRA	ALAXANDR
MADESON				CURTIS	CART
KUMAR				ALOK	ALAC
FAGAN				GAYLE	GAL
FANELLI				MARIA	MAR
ANDERSON				CHRISTINA	CRASAN
MADISON				CURTIS	CART
ALUR				NAGRAJ	NAGRAJ
SKOV				ALEX	ALAC
MADI				CURTIS	CART
ANDERSON				BRUCE	BRAC
FARREL				DENISE	DANAS
ROSEN				BARRY	BARY

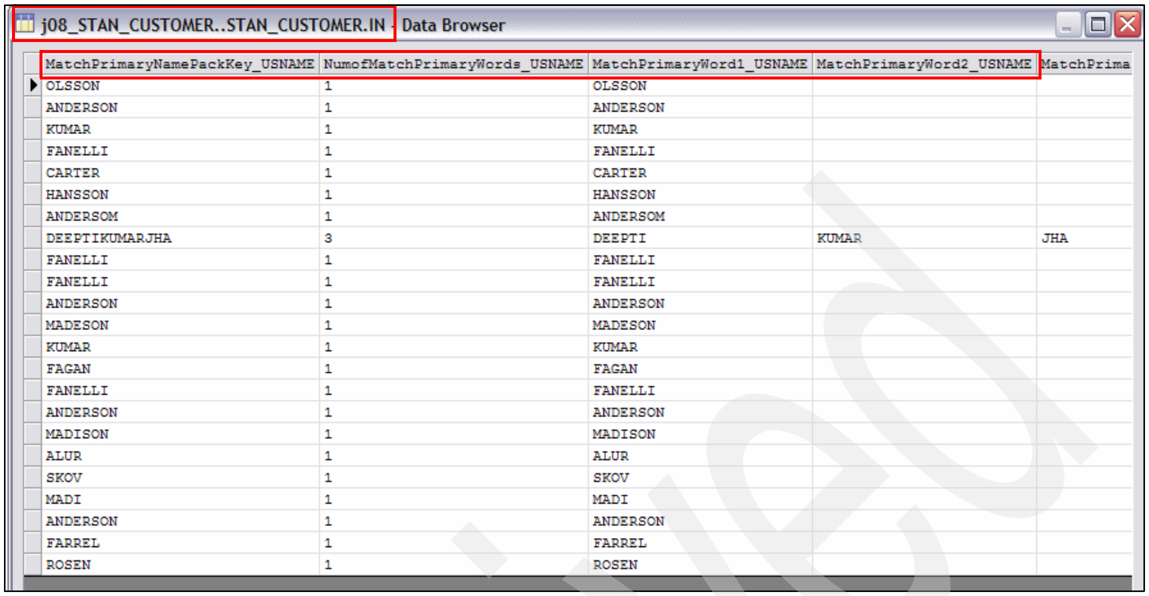
Figure 1-258   Create J08_STAN_CUSTOMER_Domain_Specific 9/22



j08_STAN_CUSTOMER..STAN_CUSTOMER.IN - Data Browser

MatchFirstNameNYSIIS_USNAME	MatchFirstNameRVSNDX_USNAME	MatchPrimaryName_USNAME	MatchPrimaryNameHashKey_USNAME	MatchPrima
ANDAR	S635	OLSSON	OL	OLSSON
YASAC	A220	ANDERSON	AN	ANDERSON
JASANDAR	K635	KUMAR	KU	KUMAR
ARCANGAL	O425	FANELLI	FA	FANELLI
A	A000	CARTER	CA	CARTER
CARAL	L620	HANSSON	HA	HANSSON
TARBAN	N163	ANDERSOM	AN	ANDERSOM
	0000	DEEPTI KUMAR JHA	DEKUJH	DEEPTIKUMA
AN	A500	FANELLI	FA	FANELLI
A	A000	FANELLI	FA	FANELLI
ALAXANDR	A635	ANDERSON	AN	ANDERSON
CART	S362	MADESON	MA	MADESON
ALAC	K400	KUMAR	KU	KUMAR
GAL	E420	FAGAN	FA	FAGAN
MAR	A650	FANELLI	FA	FANELLI
CRASAN	A532	ANDERSON	AN	ANDERSON
CART	S362	MADISON	MA	MADISON
NAGRAJ	J625	ALUR	AL	ALUR
ALAC	X400	SKOV	SK	SKOV
CART	S362	MADI	MA	MADI
BRAC	E261	ANDERSON	AN	ANDERSON
DANAS	E253	FARREL	FA	FARREL
BARY	Y610	ROSEN	RO	ROSEN

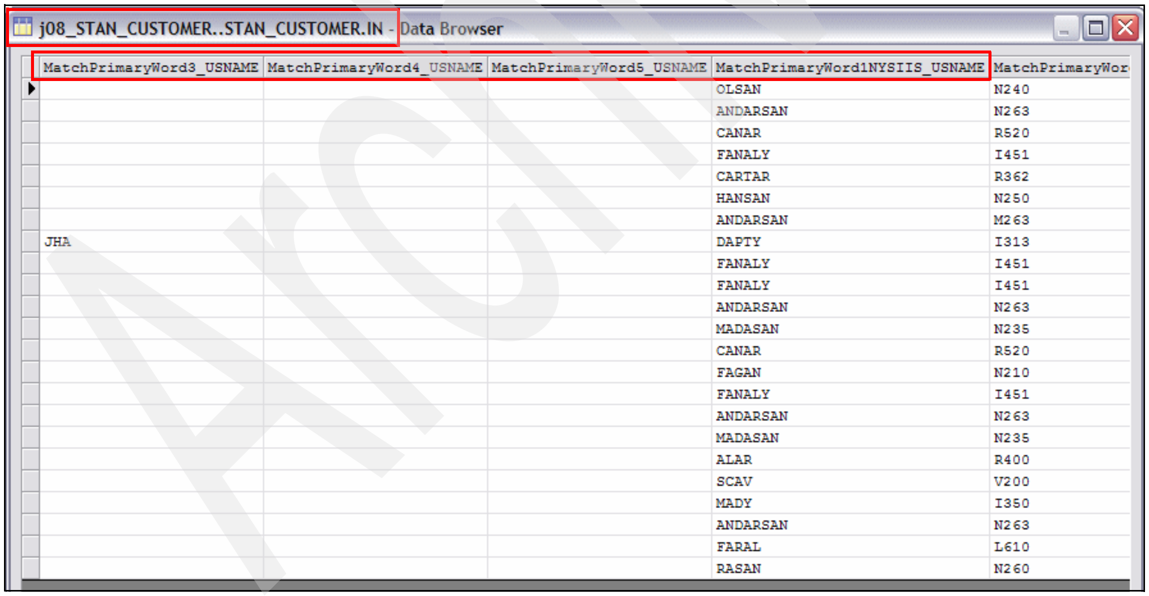
Figure 1-259   Create J08_STAN_CUSTOMER_Domain_Specific 10/22



j08_STAN_CUSTOMER..STAN_CUSTOMER.IN - Data Browser

MatchPrimaryNamePackKey_USNAME	NumofMatchPrimaryWords_USNAME	MatchPrimaryWord1_USNAME	MatchPrimaryWord2_USNAME	MatchPrima
OLSSON	1	OLSSON		
ANDERSON	1	ANDERSON		
KUMAR	1	KUMAR		
FANELLI	1	FANELLI		
CARTER	1	CARTER		
HANSSON	1	HANSSON		
ANDERSOM	1	ANDERSOM		
DEEPTIKUMARJHA	3	DEEPTI	KUMAR	JHA
FANELLI	1	FANELLI		
FANELLI	1	FANELLI		
ANDERSON	1	ANDERSON		
MADESON	1	MADESON		
KUMAR	1	KUMAR		
FAGAN	1	FAGAN		
FANELLI	1	FANELLI		
ANDERSON	1	ANDERSON		
MADISON	1	MADISON		
ALUR	1	ALUR		
SKOV	1	SKOV		
MADI	1	MADI		
ANDERSON	1	ANDERSON		
FARREL	1	FARREL		
ROSEN	1	ROSEN		

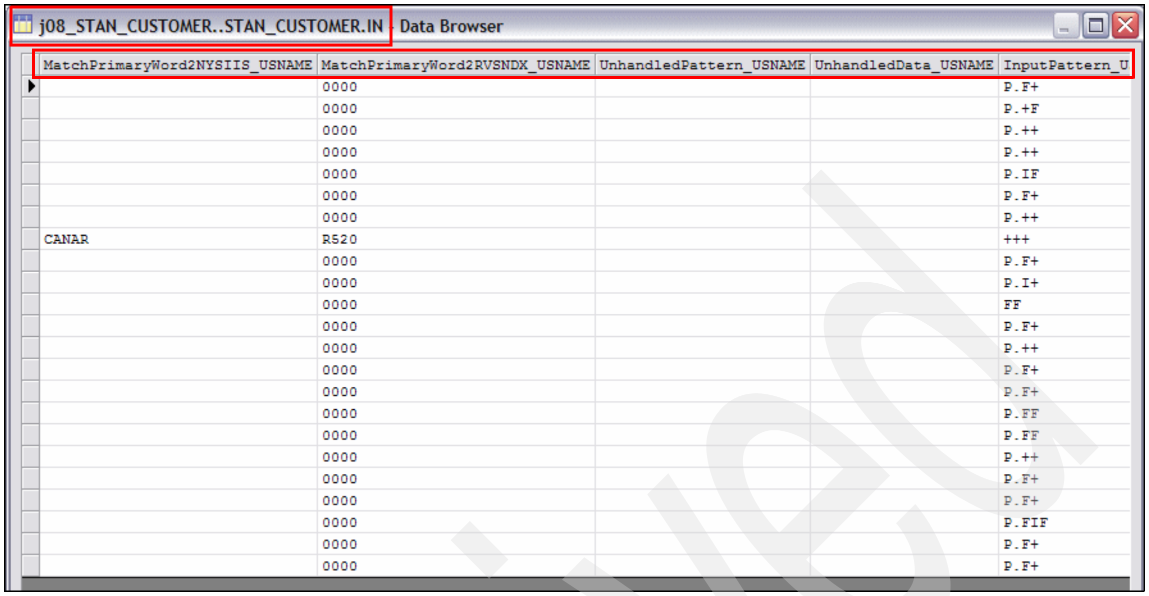
Figure 1-260 Create J08_STAN_CUSTOMER_Domain_Specific 11/22



j08_STAN_CUSTOMER..STAN_CUSTOMER.IN - Data Browser

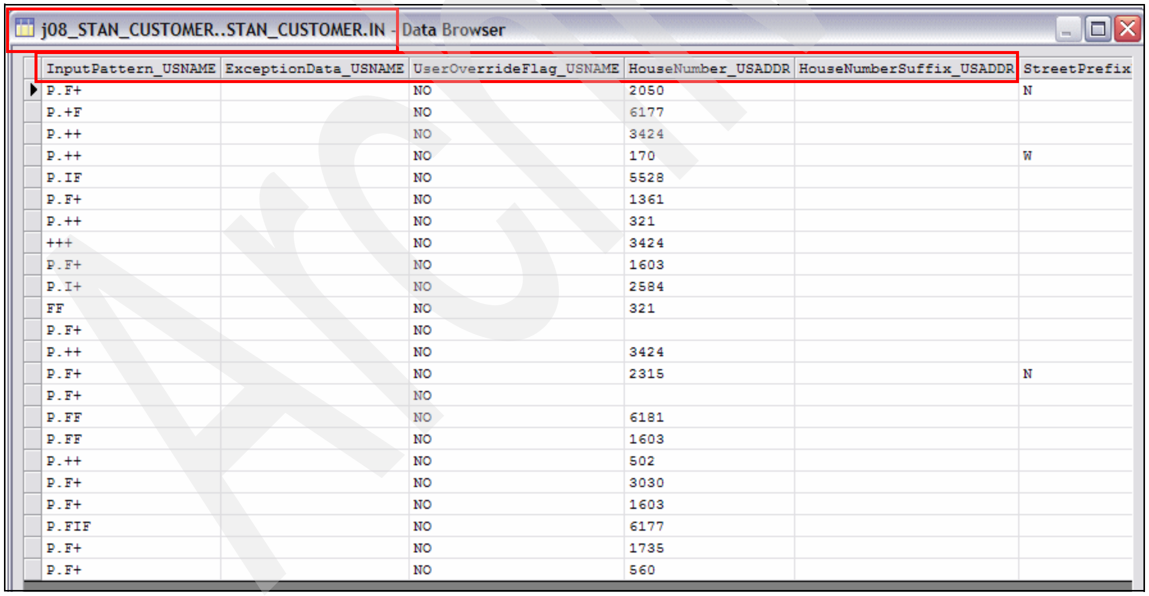
MatchPrimaryWord3_USNAME	MatchPrimaryWord4_USNAME	MatchPrimaryWord5_USNAME	MatchPrimaryWord1NYSIIS_USNAME	MatchPrimaryWor
			OLSAN	N240
			ANDARSAN	N263
			CANAR	R520
			FANALY	I451
			CARTAR	R362
			HANSAN	N250
			ANDARSAN	M263
JHA			DAPTY	I313
			FANALY	I451
			FANALY	I451
			ANDARSAN	N263
			MADASAN	N235
			CANAR	R520
			FAGAN	N210
			FANALY	I451
			ANDARSAN	N263
			MADASAN	N235
			ALAR	R400
			SCAV	V200
			MADY	I350
			ANDARSAN	N263
			FARAL	L610
			RASAN	N260

Figure 1-261 Create J08_STAN_CUSTOMER_Domain_Specific 12/22


j08_STAN_CUSTOMER..STAN_CUSTOMER.IN - Data Browser

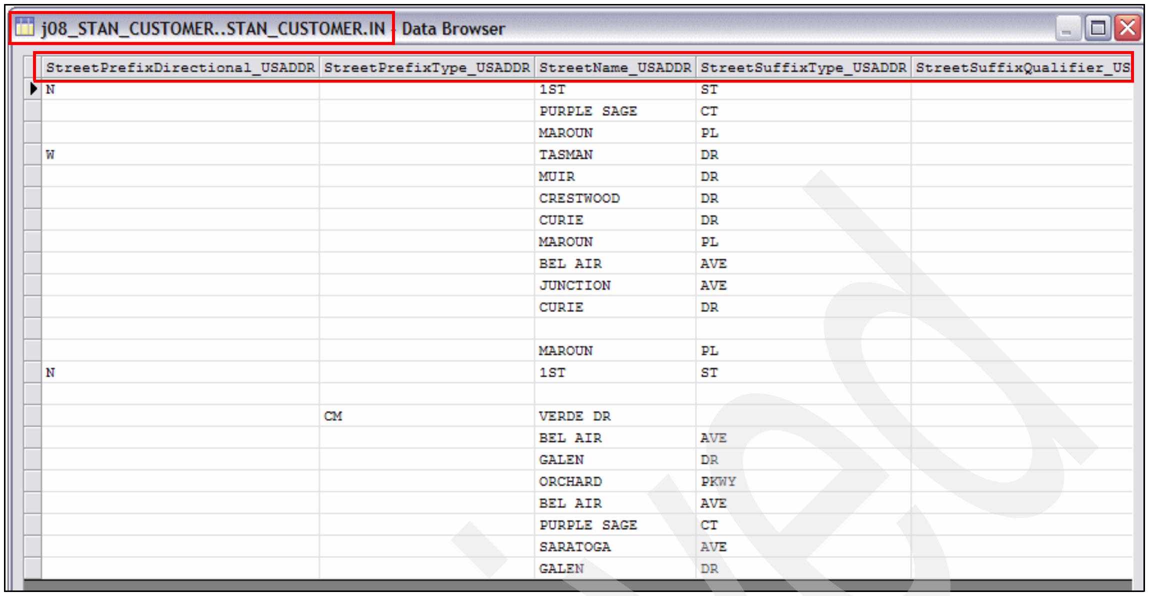
MatchPrimaryWord2NYSIIS_USNAME	MatchPrimaryWord2RVSNDX_USNAME	UnhandledPattern_USNAME	UnhandledData_USNAME	InputPattern_U
	0000			P.F+
	0000			P.+F
	0000			P.++
	0000			P.++
	0000			P.IF
	0000			P.F+
	0000			P.++
CANAR	R520			+++
	0000			P.F+
	0000			P.I+
	0000			FF
	0000			P.F+
	0000			P.++
	0000			P.F+
	0000			P.F+
	0000			P.FF
	0000			P.FF
	0000			P.++
	0000			P.F+
	0000			P.F+
	0000			P.FIF
	0000			P.F+
	0000			P.F+

Figure 1-262 Create J08_STAN_CUSTOMER_Domain_Specific 13/22


j08_STAN_CUSTOMER..STAN_CUSTOMER.IN - Data Browser

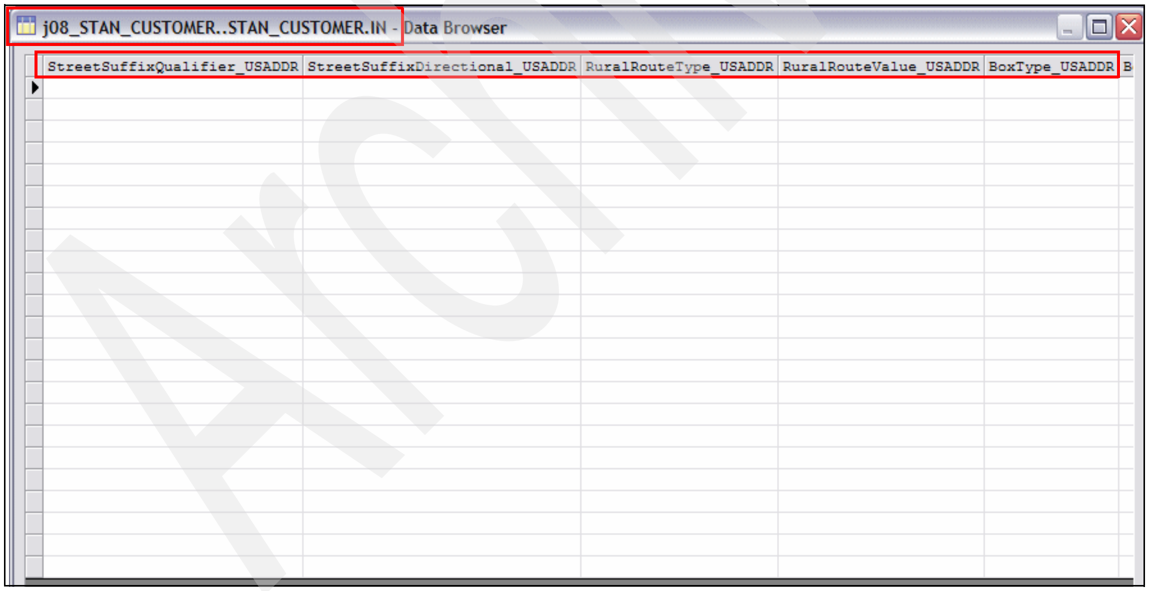
InputPattern_USNAME	ExceptionData_USNAME	UserOverrideFlag_USNAME	HouseNumber_USADDR	HouseNumberSuffix_USADDR	StreetPrefix
P.F+		NO	2050		N
P.+F		NO	6177		
P.++		NO	3424		
P.++		NO	170		W
P.IF		NO	5528		
P.F+		NO	1361		
P.++		NO	321		
+++		NO	3424		
P.F+		NO	1603		
P.I+		NO	2584		
FF		NO	321		
P.F+		NO			
P.++		NO	3424		
P.F+		NO	2315		N
P.F+		NO			
P.FF		NO	6181		
P.FF		NO	1603		
P.++		NO	502		
P.F+		NO	3030		
P.F+		NO	1603		
P.FIF		NO	6177		
P.F+		NO	1735		
P.F+		NO	560		

Figure 1-263 Create J08_STAN_CUSTOMER_Domain_Specific 14/22


j08_STAN_CUSTOMER..STAN_CUSTOMER.IN - Data Browser

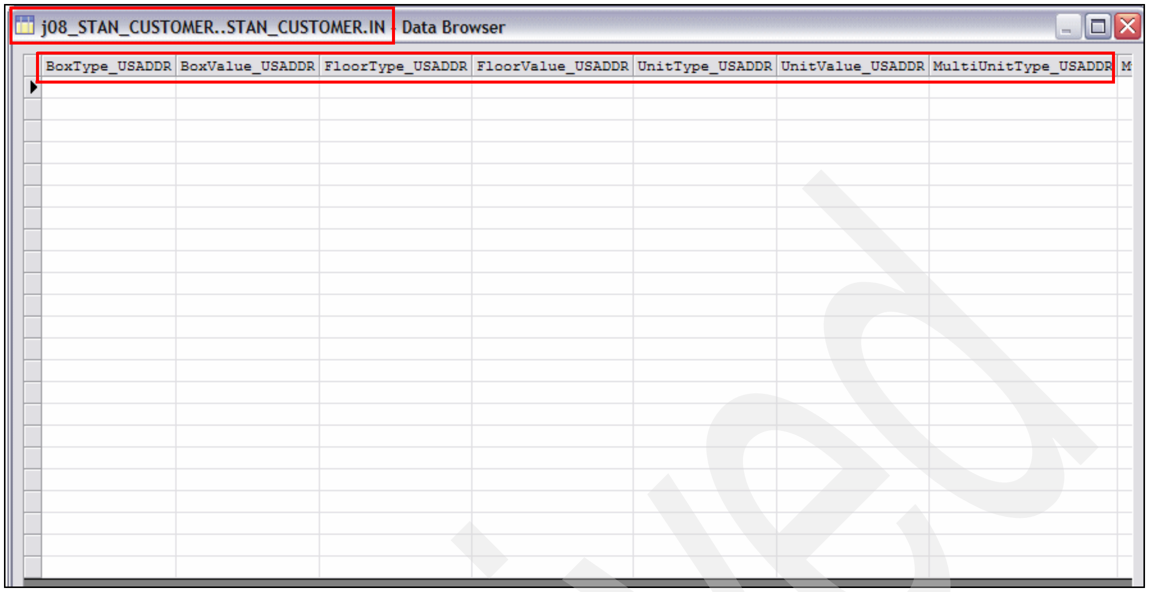
StreetPrefixDirectional_USADDR	StreetPrefixType_USADDR	StreetName_USADDR	StreetSuffixType_USADDR	StreetSuffixQualifier_US
N		1ST	ST	
		PURPLE SAGE	CT	
		MAROUN	PL	
W		TASMAN	DR	
		MUIR	DR	
		CRESTWOOD	DR	
		CURIE	DR	
		MAROUN	PL	
		BEL AIR	AVE	
		JUNCTION	AVE	
		CURIE	DR	
		MAROUN	PL	
N		1ST	ST	
	CM	VERDE DR		
		BEL AIR	AVE	
		GALEN	DR	
		ORCHARD	PKWY	
		BEL AIR	AVE	
		PURPLE SAGE	CT	
		SARATOGA	AVE	
		GALEN	DR	

Figure 1-264 Create J08_STAN_CUSTOMER_Domain_Specific 15/22


j08_STAN_CUSTOMER..STAN_CUSTOMER.IN - Data Browser

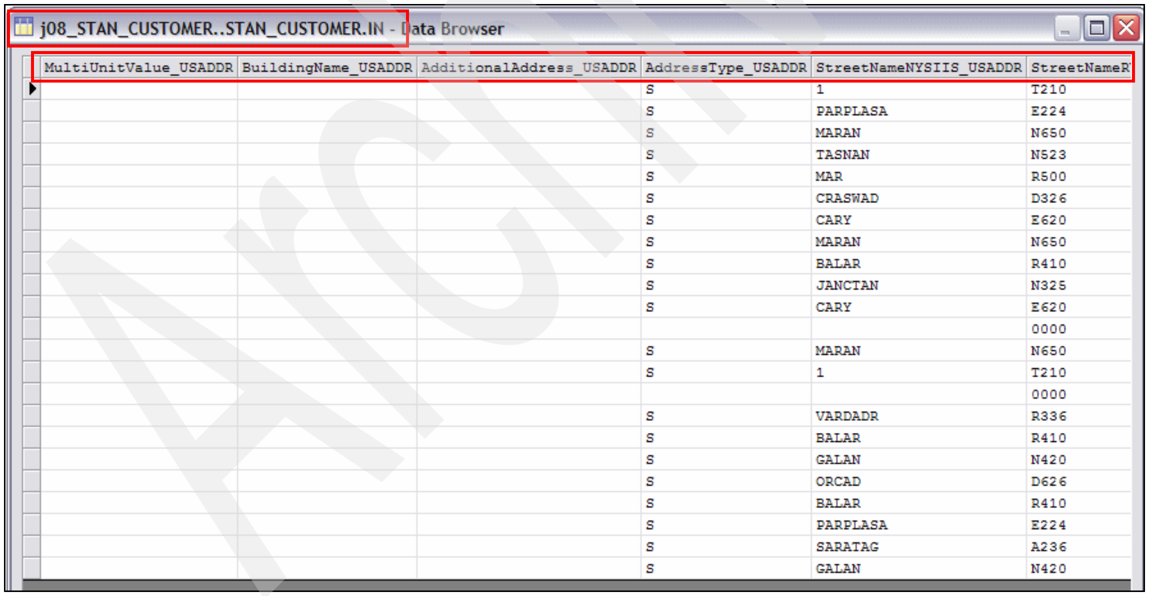
StreetSuffixQualifier_USADDR	StreetSuffixDirectional_USADDR	RuralRouteType_USADDR	RuralRouteValue_USADDR	BoxType_USADDR	B

Figure 1-265 Create J08_STAN_CUSTOMER_Domain_Specific 16/22


j08_STAN_CUSTOMER..STAN_CUSTOMER.IN - Data Browser

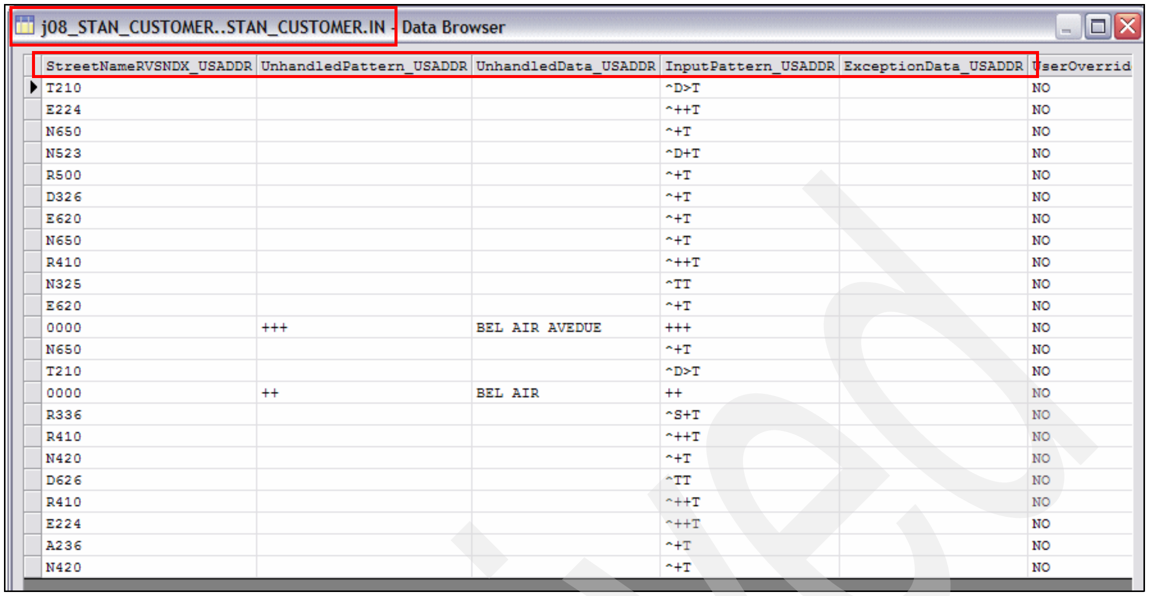
BoxType_USADDR	BoxValue_USADDR	FloorType_USADDR	FloorValue_USADDR	UnitType_USADDR	UnitValue_USADDR	MultiUnitType_USADDR	M

Figure 1-266   Create J08_STAN_CUSTOMER_Domain_Specific 17/22


j08_STAN_CUSTOMER..STAN_CUSTOMER.IN - Data Browser

MultiUnitValue_USADDR	BuildingName_USADDR	AdditionalAddress_USADDR	AddressType_USADDR	StreetNameNYSIIS_USADDR	StreetNameR
			S	1	T210
			S	PARPLASA	E224
			S	MARAN	N650
			S	TASNAN	N523
			S	MAR	R500
			S	CRASWAD	D326
			S	CARY	E620
			S	MARAN	N650
			S	BALAR	R410
			S	JANCTAN	N325
			S	CARY	E620
					0000
			S	MARAN	N650
			S	1	T210
					0000
			S	VARDADR	R336
			S	BALAR	R410
			S	GALAN	N420
			S	ORCAD	D626
			S	BALAR	R410
			S	PARPLASA	E224
			S	SARATAG	A236
			S	GALAN	N420

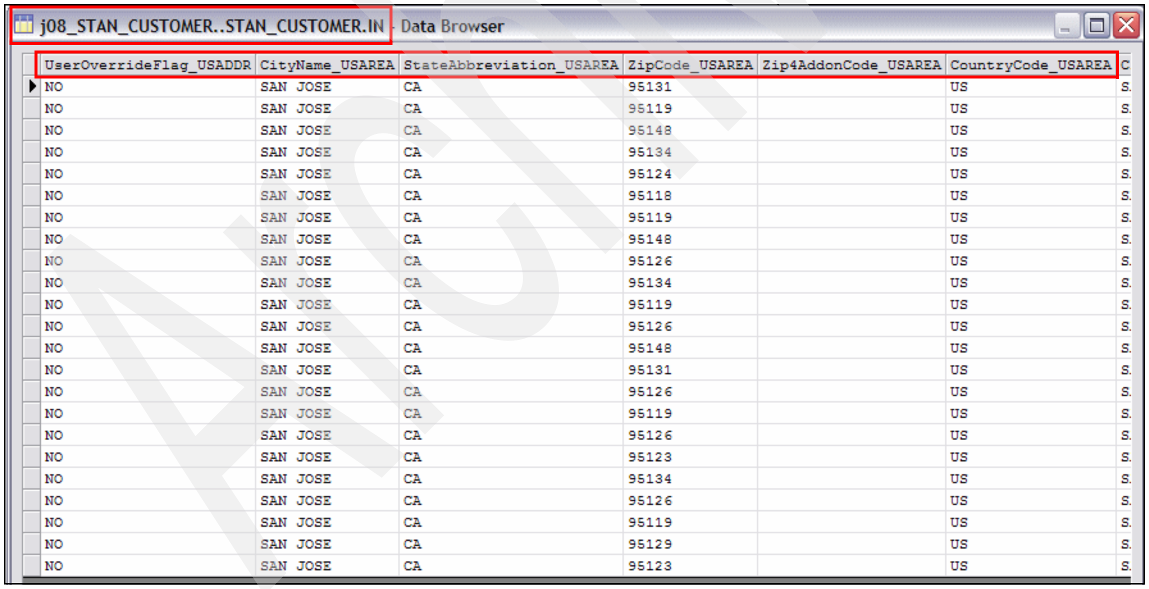
Figure 1-267   Create J08_STAN_CUSTOMER_Domain_Specific 18/22



j08_STAN_CUSTOMER..STAN_CUSTOMER.IN - Data Browser

StreetNameRVSNDX_USADDR	UnhandledPattern_USADDR	UnhandledData_USADDR	InputPattern_USADDR	ExceptionData_USADDR	UserOverrid
T210			^D>T		NO
E224			^++T		NO
N650			^+T		NO
N523			^D+T		NO
R500			^+T		NO
D326			^+T		NO
E620			^+T		NO
N650			^+T		NO
R410			^++T		NO
N325			^TT		NO
E620			^+T		NO
0000	+++	BEL AIR AVEDUE	+++		NO
N650			^+T		NO
T210			^D>T		NO
0000	++	BEL AIR	++		NO
R336			^S+T		NO
R410			^++T		NO
N420			^+T		NO
D626			^TT		NO
R410			^++T		NO
E224			^++T		NO
A236			^+T		NO
N420			^+T		NO

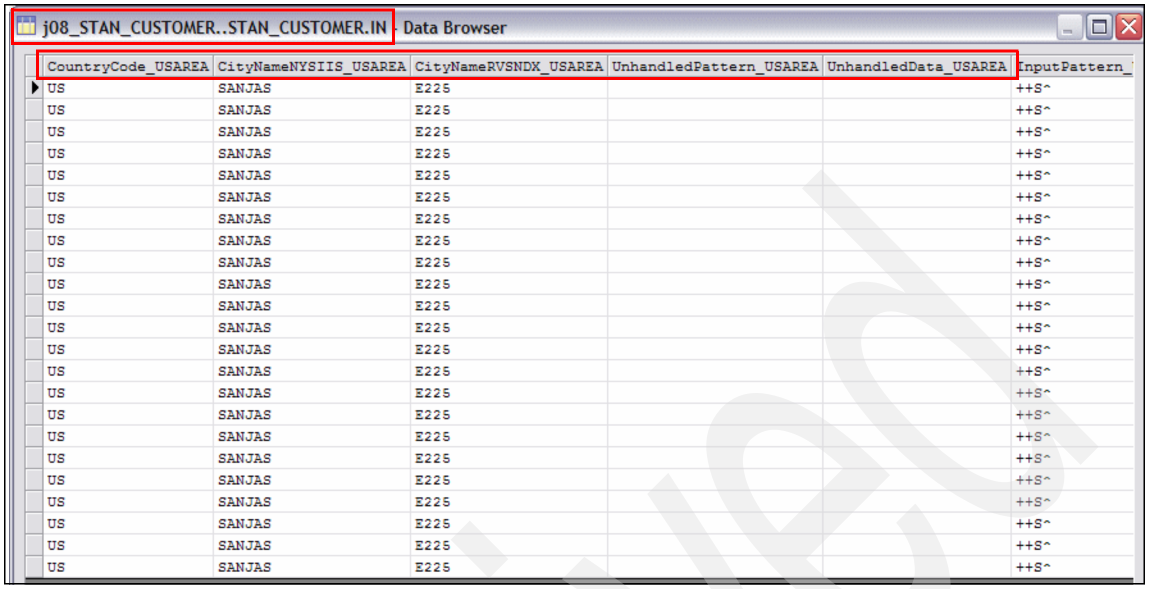
Figure 1-268 Create J08_STAN_CUSTOMER_Domain_Specific 19/22



j08_STAN_CUSTOMER..STAN_CUSTOMER.IN - Data Browser

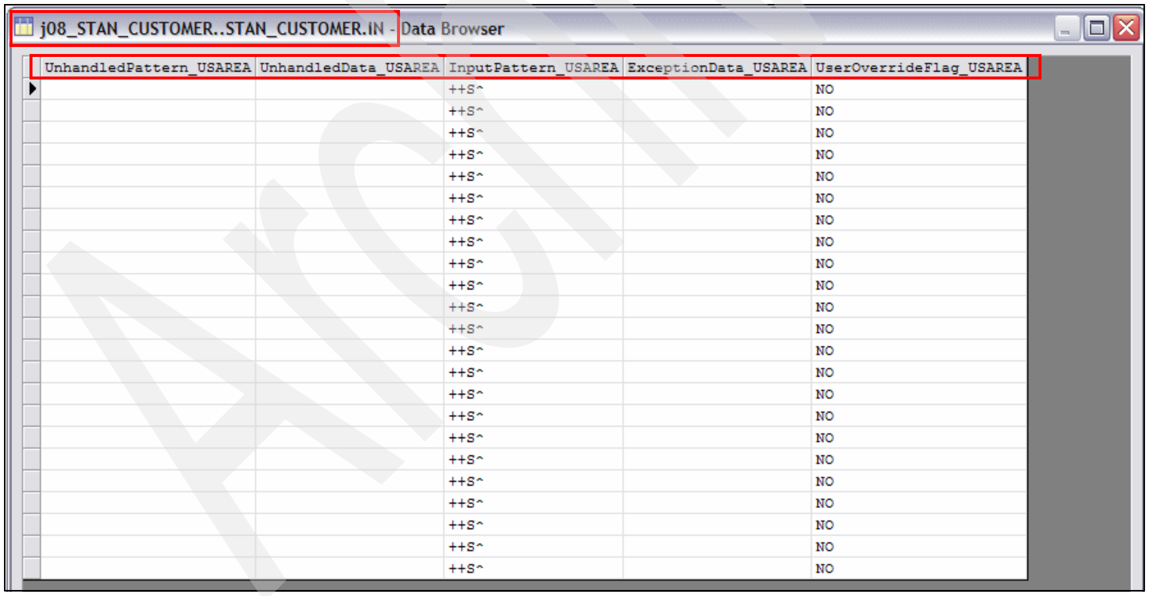
UserOverrideFlag_USADDR	CityName_USAREA	StateAbbreviation_USAREA	ZipCode_USAREA	Zip4AddonCode_USAREA	CountryCode_USAREA	C
NO	SAN JOSE	CA	95131		US	S.
NO	SAN JOSE	CA	95119		US	S.
NO	SAN JOSE	CA	95148		US	S.
NO	SAN JOSE	CA	95134		US	S.
NO	SAN JOSE	CA	95124		US	S.
NO	SAN JOSE	CA	95118		US	S.
NO	SAN JOSE	CA	95119		US	S.
NO	SAN JOSE	CA	95148		US	S.
NO	SAN JOSE	CA	95126		US	S.
NO	SAN JOSE	CA	95134		US	S.
NO	SAN JOSE	CA	95119		US	S.
NO	SAN JOSE	CA	95126		US	S.
NO	SAN JOSE	CA	95148		US	S.
NO	SAN JOSE	CA	95131		US	S.
NO	SAN JOSE	CA	95126		US	S.
NO	SAN JOSE	CA	95119		US	S.
NO	SAN JOSE	CA	95126		US	S.
NO	SAN JOSE	CA	95123		US	S.
NO	SAN JOSE	CA	95134		US	S.
NO	SAN JOSE	CA	95126		US	S.
NO	SAN JOSE	CA	95119		US	S.
NO	SAN JOSE	CA	95129		US	S.
NO	SAN JOSE	CA	95123		US	S.

Figure 1-269 Create J08_STAN_CUSTOMER_Domain_Specific 20/22


j08_STAN_CUSTOMER..STAN_CUSTOMER.IN - Data Browser

CountryCode_USAREA	CityNameNYSIIS_USAREA	CityNameRVSNDX_USAREA	UnhandledPattern_USAREA	UnhandledData_USAREA	InputPattern_
US	SANJAS	E225			++S^
US	SANJAS	E225			++S^
US	SANJAS	E225			++S^
US	SANJAS	E225			++S^
US	SANJAS	E225			++S^
US	SANJAS	E225			++S^
US	SANJAS	E225			++S^
US	SANJAS	E225			++S^
US	SANJAS	E225			++S^
US	SANJAS	E225			++S^
US	SANJAS	E225			++S^
US	SANJAS	E225			++S^
US	SANJAS	E225			++S^
US	SANJAS	E225			++S^
US	SANJAS	E225			++S^
US	SANJAS	E225			++S^
US	SANJAS	E225			++S^
US	SANJAS	E225			++S^
US	SANJAS	E225			++S^
US	SANJAS	E225			++S^
US	SANJAS	E225			++S^
US	SANJAS	E225			++S^
US	SANJAS	E225			++S^

Figure 1-270 Create J08_STAN_CUSTOMER_Domain_Specific 21/22


j08_STAN_CUSTOMER..STAN_CUSTOMER.IN - Data Browser

UnhandledPattern_USAREA	UnhandledData_USAREA	InputPattern_USAREA	ExceptionData_USAREA	UserOverrideFlag_USAREA
		++S^		NO
		++S^		NO
		++S^		NO
		++S^		NO
		++S^		NO
		++S^		NO
		++S^		NO
		++S^		NO
		++S^		NO
		++S^		NO
		++S^		NO
		++S^		NO
		++S^		NO
		++S^		NO
		++S^		NO
		++S^		NO
		++S^		NO
		++S^		NO
		++S^		NO
		++S^		NO
		++S^		NO
		++S^		NO
		++S^		NO

Figure 1-271 Create J08_STAN_CUSTOMER_Domain_Specific 22/22

J09_INVCC_STAN_CUSTOMER

In this step, we investigate unhandled patterns in "J08_STAN_CUSTOMER_Domain_Specific" on page 241. We run the Investigate stage with character concatenate using the *C* mask on the unhandled pattern columns of J08_STAN_CUSTOMER_Domain_Specific—the columns that we investigated corresponded to the name, address, and area domains.

Because a single Investigate stage can only have a single rule set associated with it, we split the output data set of the "J08_STAN_CUSTOMER_Domain_Specific" on page 241 job (using a Copy stage) and processed the data set by three independent Investigate stages, with each stage using a particular domain-specific rule set. The column frequency report was generated in each Investigate stage.

Figure 1-272 on page 255 through Figure 1-287 on page 262 describe the steps using Designer Client to build and execute the DataStage job to perform this task.

The steps are as follows:

1. Figure 1-272 on page 255 shows the various stages that are used in this job, including the data set that was created in "J08_STAN_CUSTOMER_Domain_Specific" on page 241, a Copy stage, three Investigate stages that each use a different domain-specific rule set, and two data set stages (one each for the token report and pattern report) for each Investigate stage. We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object created earlier.
2. Configure the input STAN_CUSTOMER data set. We do not repeat this process here because it is similar to the steps shown in Figure 1-138 on page 172 through Figure 1-142 on page 174.
3. The Copy stage duplicates the source data and sends it to the three Investigate stages. The Copy stage also duplicates the metadata and sends the output metadata to the three Investigate stages.
   a. Figure 1-273 on page 256 shows the mapping of columns for the link IN1 to the INV01 (USNAME rule set) stage
   b. Figure 1-274 on page 257 shows the mapping of columns for the link IN2 to the INV02 (USADDR rule set) stage
   c. Figure 1-275 on page 258 shows the mapping of columns for the link IN3 to the INV01 (USAREA rule set) stage

4. Configure the INV01 Investigate stage with the USNAME rule set. Figure 1-276 on page 258 shows the columns selected for Character Concatenate Investigate. It includes the following columns:

   – UnhandledPattern_USNAME with *C* mask
   – InputPattern_USNAME with *C* mask
   – UnhandledData_USNAME with *X* mask
   – NameDomain_USPREP with *X* mask

   Configure the sequential file for the column frequency report. We do not repeat this process here because it is similar to Figure 1-174 on page 191 through Figure 1-177 on page 193.

5. Repeat the process for the INV02 Investigate stage with the USADDR rule set (Figure 1-277 on page 259) and INV03 Investigate stage with the USAREA rule set (Figure 1-278 on page 259) with the appropriate columns as shown.

6. After saving, compiling, and running this job, the contents of the output of each investigate stage are as follows:

   a. The contents of the data input to the Investigate stage are shown in Figure 1-269 on page 251 and Figure 1-271 on page 252.

   b. The outputs of the Investigate stages that are written to the sequential file are shown in Figure 1-279 on page 260 through Figure 1-287 on page 262.

      i. The output of the INV01 Investigate stage in Figure 1-279 on page 260 through Figure 1-281 on page 260 show that there are no unhandled patterns in the name domain.

      ii. The output of the INV02 Investigate stage in Figure 1-282 on page 261 through Figure 1-284 on page 261 show a total of two types of unhandled patterns for the address domain—one occurrences of the ++ pattern (corresponding to the ++ input pattern), and one occurrence of the +++ pattern (corresponding to the +++ input pattern).

      iii. The output of the INV03 Investigate stage in Figure 1-285 on page 261 through Figure 1-287 on page 262 show that there are no unhandled patterns in the area domain.

Now, proceed to manage the unhandled patterns in the address domain, which are described in the next step, “J09_Z_Override_And_After” on page 262.

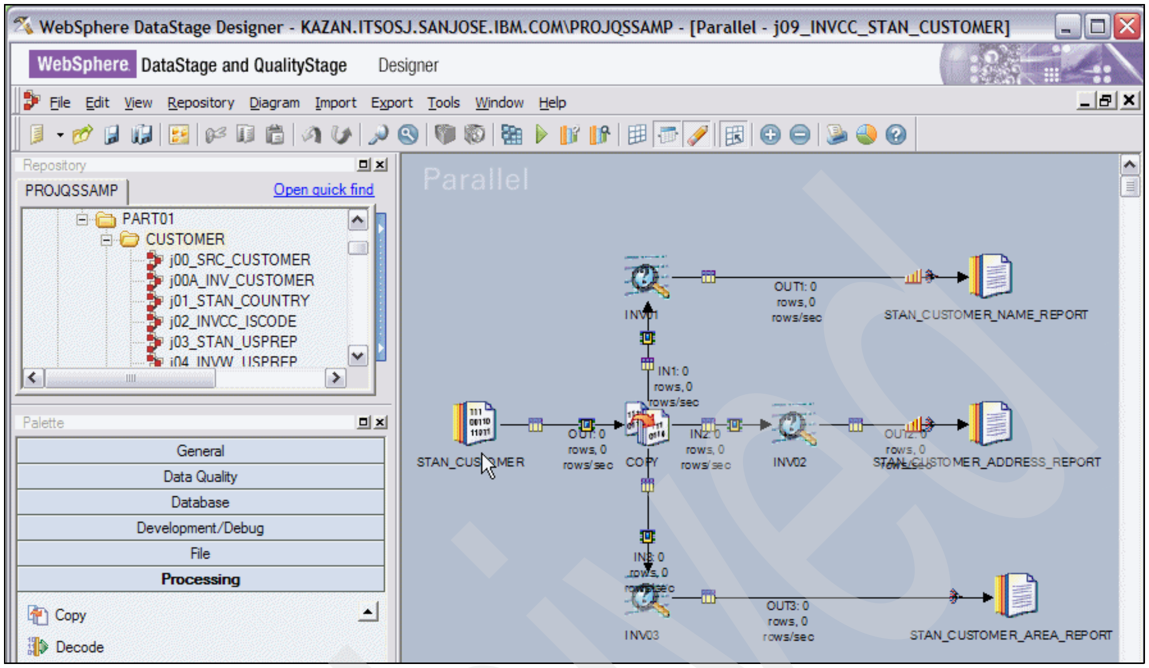
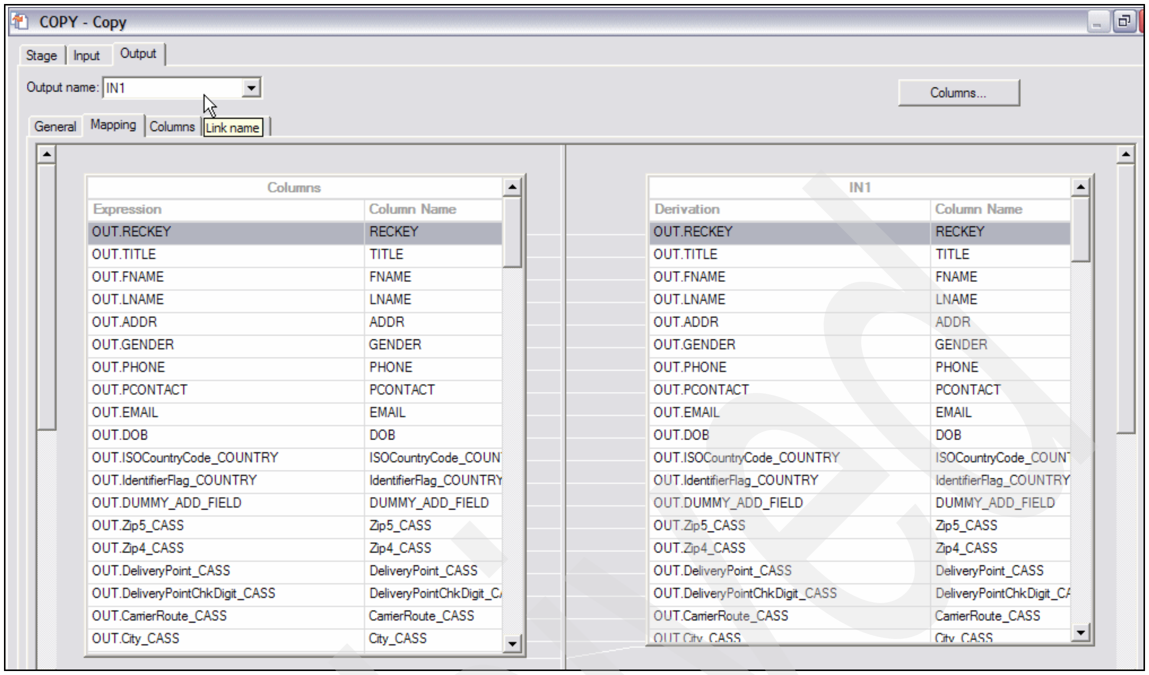


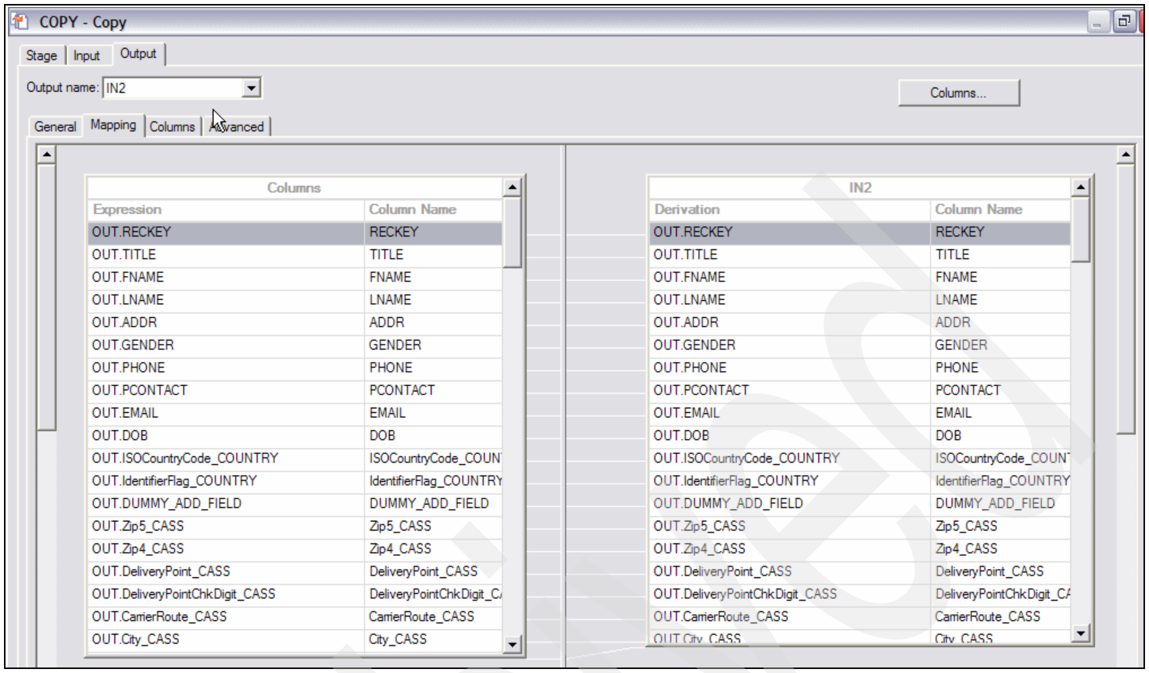
Figure 1-272 Create J09_INVCC_STAN_CUSTOMER 1/16



Columns	
Expression	Column Name
OUT.RECKEY	RECKEY
OUT.TITLE	TITLE
OUT.FNAME	FNAME
OUT.LNAME	LNAME
OUT.ADDR	ADDR
OUT.GENDER	GENDER
OUT.PHONE	PHONE
OUT.PCONTACT	PCONTACT
OUT.EMAIL	EMAIL
OUT.DOB	DOB
OUT.ISOCountryCode_COUNTRY	ISOCountryCode_COUN
OUT.IdentifierFlag_COUNTRY	IdentifierFlag_COUNTRY
OUT.DUMMY_ADD_FIELD	DUMMY_ADD_FIELD
OUT.Zip5_CASS	Zip5_CASS
OUT.Zip4_CASS	Zip4_CASS
OUT.DeliveryPoint_CASS	DeliveryPoint_CASS
OUT.DeliveryPointChkDigit_CASS	DeliveryPointChkDigit_C
OUT.CarrierRoute_CASS	CarrierRoute_CASS
OUT.City_CASS	City_CASS

IN1	
Derivation	Column Name
OUT.RECKEY	RECKEY
OUT.TITLE	TITLE
OUT.FNAME	FNAME
OUT.LNAME	LNAME
OUT.ADDR	ADDR
OUT.GENDER	GENDER
OUT.PHONE	PHONE
OUT.PCONTACT	PCONTACT
OUT.EMAIL	EMAIL
OUT.DOB	DOB
OUT.ISOCountryCode_COUNTRY	ISOCountryCode_COUN
OUT.IdentifierFlag_COUNTRY	IdentifierFlag_COUNTRY
OUT.DUMMY_ADD_FIELD	DUMMY_ADD_FIELD
OUT.Zip5_CASS	Zip5_CASS
OUT.Zip4_CASS	Zip4_CASS
OUT.DeliveryPoint_CASS	DeliveryPoint_CASS
OUT.DeliveryPointChkDigit_CASS	DeliveryPointChkDigit_C
OUT.CarrierRoute_CASS	CarrierRoute_CASS
OUT.City_CASS	City_CASS

Figure 1-273 Create J09_INVCC_STAN_CUSTOMER 2/16


COPY - Copy

Output name: IN2

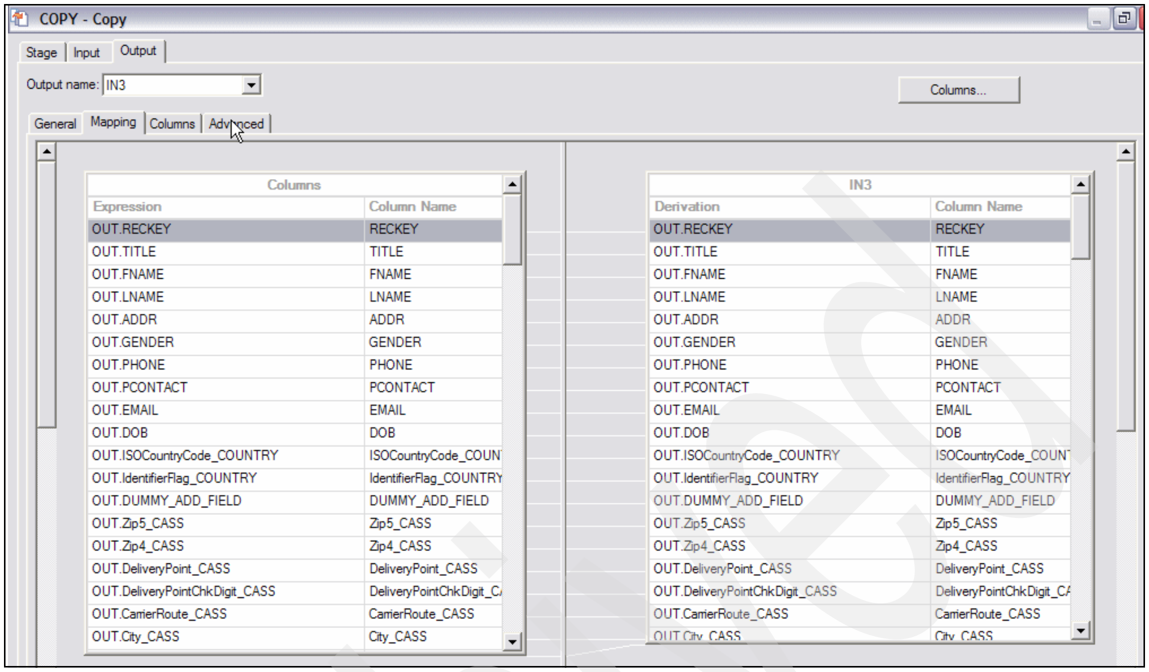
Columns

Expression	Column Name
OUT.RECKEY	RECKEY
OUT.TITLE	TITLE
OUT.FNAME	FNAME
OUT.LNAME	LNAME
OUT.ADDR	ADDR
OUT.GENDER	GENDER
OUT.PHONE	PHONE
OUT.PCONTACT	PCONTACT
OUT.EMAIL	EMAIL
OUT.DOB	DOB
OUT.ISOCountryCode_COUNTRY	ISOCountryCode_COUN
OUT.IdentifierFlag_COUNTRY	IdentifierFlag_COUNTRY
OUT.DUMMY_ADD_FIELD	DUMMY_ADD_FIELD
OUT.Zip5_CASS	Zip5_CASS
OUT.Zip4_CASS	Zip4_CASS
OUT.DeliveryPoint_CASS	DeliveryPoint_CASS
OUT.DeliveryPointChkDigit_CASS	DeliveryPointChkDigit_C
OUT.CarrierRoute_CASS	CarrierRoute_CASS
OUT.City_CASS	City_CASS

IN2

Derivation	Column Name
OUT.RECKEY	RECKEY
OUT.TITLE	TITLE
OUT.FNAME	FNAME
OUT.LNAME	LNAME
OUT.ADDR	ADDR
OUT.GENDER	GENDER
OUT.PHONE	PHONE
OUT.PCONTACT	PCONTACT
OUT.EMAIL	EMAIL
OUT.DOB	DOB
OUT.ISOCountryCode_COUNTRY	ISOCountryCode_COUN
OUT.IdentifierFlag_COUNTRY	IdentifierFlag_COUNTRY
OUT.DUMMY_ADD_FIELD	DUMMY_ADD_FIELD
OUT.Zip5_CASS	Zip5_CASS
OUT.Zip4_CASS	Zip4_CASS
OUT.DeliveryPoint_CASS	DeliveryPoint_CASS
OUT.DeliveryPointChkDigit_CASS	DeliveryPointChkDigit_CA
OUT.CarrierRoute_CASS	CarrierRoute_CASS
OUT.City_CASS	City_CASS

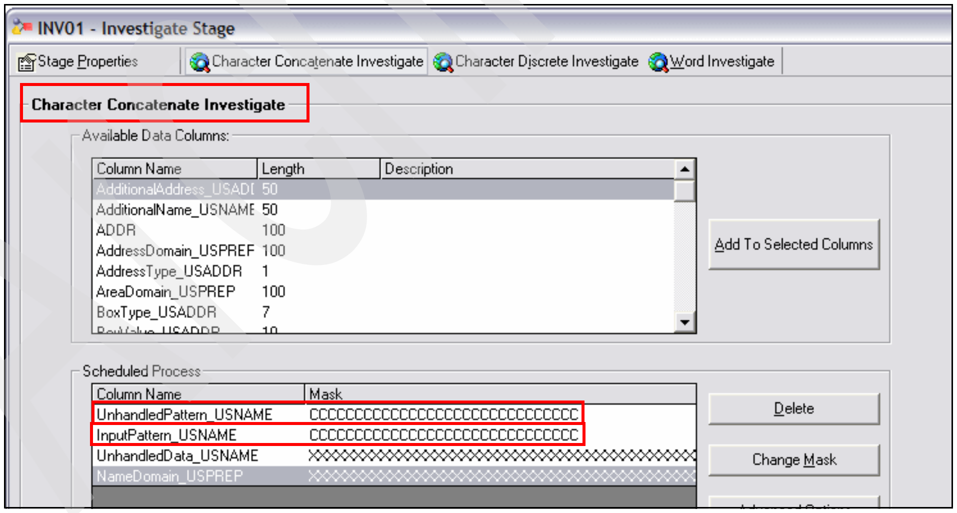
Figure 1-274 Create J09_INVCC_STAN_CUSTOMER 3/16



Columns	
Expression	Column Name
OUT.RECKEY	RECKEY
OUT.TITLE	TITLE
OUT.FNAME	FNAME
OUT.LNAME	LNAME
OUT.ADDR	ADDR
OUT.GENDER	GENDER
OUT.PHONE	PHONE
OUT.PCONTACT	PCONTACT
OUT.EMAIL	EMAIL
OUT.DOB	DOB
OUT.ISOCountryCode_COUNTRY	ISOCountryCode_COUN
OUT.IdentifierFlag_COUNTRY	IdentifierFlag_COUNTRY
OUT.DUMMY_ADD_FIELD	DUMMY_ADD_FIELD
OUT.Zip5_CASS	Zip5_CASS
OUT.Zip4_CASS	Zip4_CASS
OUT.DeliveryPoint_CASS	DeliveryPoint_CASS
OUT.DeliveryPointChkDigit_CASS	DeliveryPointChkDigit_C
OUT.CarrierRoute_CASS	CarrierRoute_CASS
OUT.City_CASS	City_CASS

IN3	
Derivation	Column Name
OUT.RECKEY	RECKEY
OUT.TITLE	TITLE
OUT.FNAME	FNAME
OUT.LNAME	LNAME
OUT.ADDR	ADDR
OUT.GENDER	GENDER
OUT.PHONE	PHONE
OUT.PCONTACT	PCONTACT
OUT.EMAIL	EMAIL
OUT.DOB	DOB
OUT.ISOCountryCode_COUNTRY	ISOCountryCode_COUN
OUT.IdentifierFlag_COUNTRY	IdentifierFlag_COUNTRY
OUT.DUMMY_ADD_FIELD	DUMMY_ADD_FIELD
OUT.Zip5_CASS	Zip5_CASS
OUT.Zip4_CASS	Zip4_CASS
OUT.DeliveryPoint_CASS	DeliveryPoint_CASS
OUT.DeliveryPointChkDigit_CASS	DeliveryPointChkDigit_C
OUT.CarrierRoute_CASS	CarrierRoute_CASS
OUT.City_CASS	City_CASS

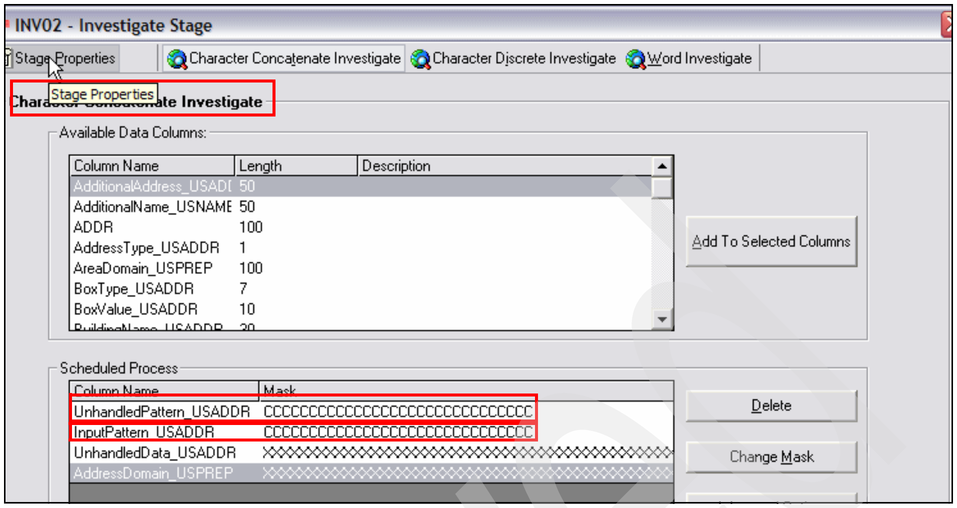
Figure 1-275 Create J09_INVCC_STAN_CUSTOMER 4/16


INV01 - Investigate Stage
Stage Properties
Character Concatenate Investigate
Character Discrete Investigate
Word Investigate
Character Concatenate Investigate
Available Data Columns:
Add To Selected Columns
Scheduled Process
Delete
Change Mask

Column Name	Length	Description
AdditionalAddress_USADI	50	
AdditionalName_USNAME	50	
ADDR	100	
AddressDomain_USPREF	100	
AddressType_USADDR	1	
AreaDomain_USPREP	100	
BoxType_USADDR	7	
BoxValue_USADDR	10	

Column Name	Mask
UnhandledPattern_USNAME	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
InputPattern_USNAME	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
UnhandledData_USNAME	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
NameDomain_USPREP	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Figure 1-276 Create J09_INVCC_STAN_CUSTOMER 5/16


INV02 - Investigate Stage

Stage Properties | Character Concatenate Investigate | Character Discrete Investigate | Word Investigate

Character Concatenate Investigate

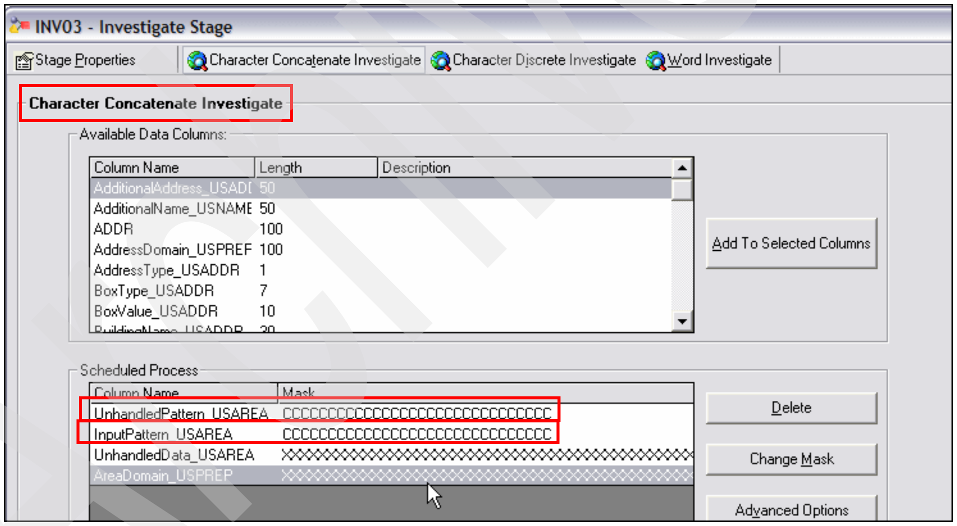
Available Data Columns:

Column Name	Length	Description
AdditionalAddress_USADI	50	
AdditionalName_USNAME	50	
ADDR	100	
AddressType_USADDR	1	
AreaDomain_USPREP	100	
BoxType_USADDR	7	
BoxValue_USADDR	10	

Scheduled Process

Column Name	Mask
UnhandledPattern_USADDR	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
InputPattern_USADDR	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
UnhandledData_USADDR	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
AddressDomain_USPREP	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Figure 1-277 Create J09_INVCC_STAN_CUSTOMER 6/16


INV03 - Investigate Stage

Stage Properties | Character Concatenate Investigate | Character Discrete Investigate | Word Investigate

Character Concatenate Investigate

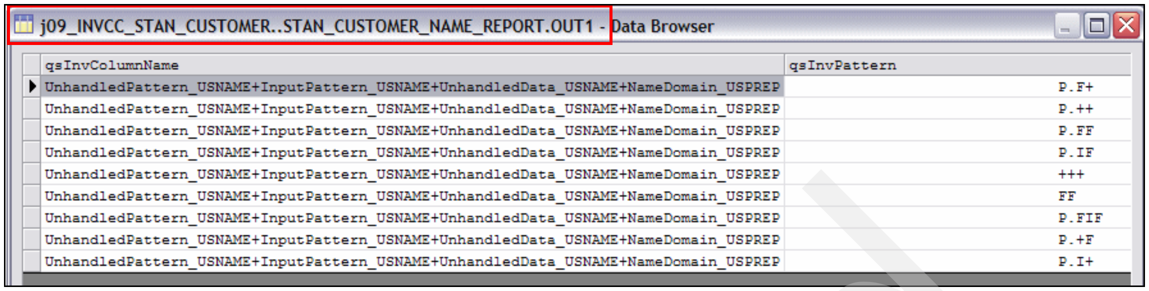
Available Data Columns:

Column Name	Length	Description
AdditionalAddress_USADI	50	
AdditionalName_USNAME	50	
ADDR	100	
AddressDomain_USPREF	100	
AddressType_USADDR	1	
BoxType_USADDR	7	
BoxValue_USADDR	10	

Scheduled Process

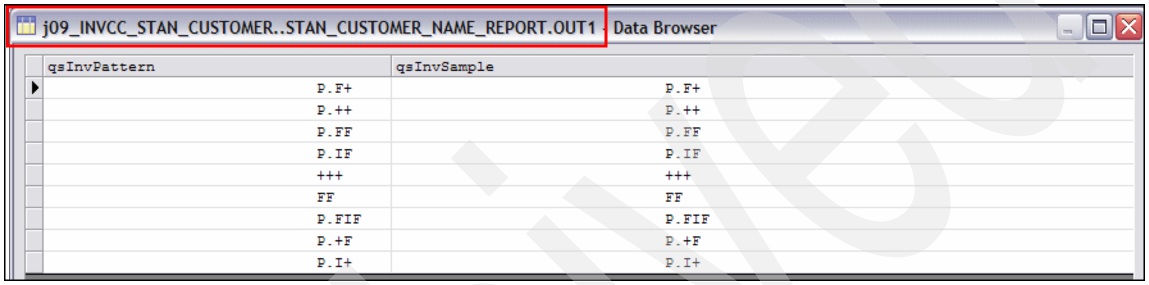
Column Name	Mask
UnhandledPattern_USAREA	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
InputPattern_USAREA	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
UnhandledData_USAREA	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
AreaDomain_USPREP	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Figure 1-278 Create J09_INVCC_STAN_CUSTOMER 7/16


j09_INVCC_STAN_CUSTOMER..STAN_CUSTOMER_NAME_REPORT.OUT1 - Data Browser

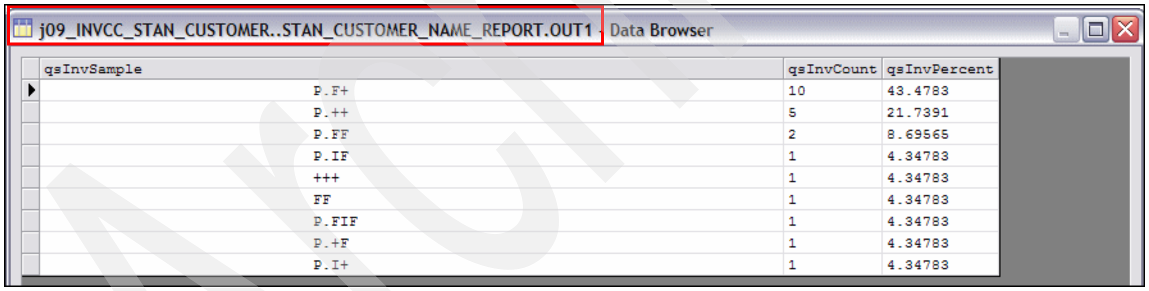
qsInvColumnName	qsInvPattern
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP	P.F+
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP	P.++
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP	P.FF
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP	P.IF
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP	+++
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP	FF
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP	P.FIF
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP	P.+F
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP	P.I+

Figure 1-279 Create J09_INVCC_STAN_CUSTOMER 8/16


j09_INVCC_STAN_CUSTOMER..STAN_CUSTOMER_NAME_REPORT.OUT1 - Data Browser

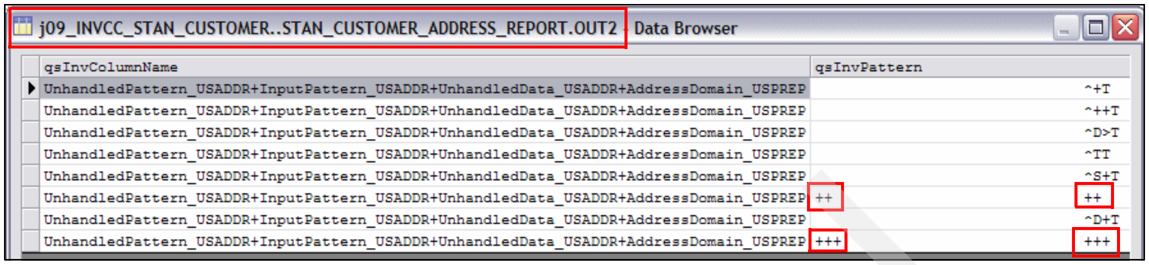
qsInvPattern	qsInvSample
P.F+	P.F+
P.++	P.++
P.FF	P.FF
P.IF	P.IF
+++	+++
FF	FF
P.FIF	P.FIF
P.+F	P.+F
P.I+	P.I+

Figure 1-280 Create J09_INVCC_STAN_CUSTOMER 9/16


j09_INVCC_STAN_CUSTOMER..STAN_CUSTOMER_NAME_REPORT.OUT1 - Data Browser

qsInvSample	qsInvCount	qsInvPercent
P.F+	10	43.4783
P.++	5	21.7391
P.FF	2	8.69565
P.IF	1	4.34783
+++	1	4.34783
FF	1	4.34783
P.FIF	1	4.34783
P.+F	1	4.34783
P.I+	1	4.34783

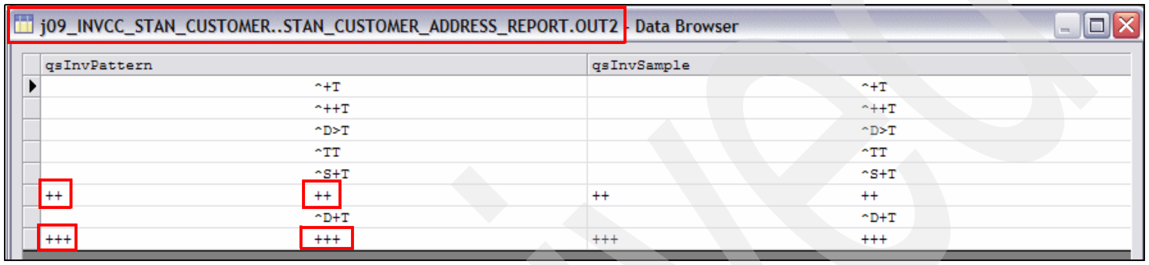
Figure 1-281 Create J09_INVCC_STAN_CUSTOMER 10/16



j09_INVCC_STAN_CUSTOMER..STAN_CUSTOMER_ADDRESS_REPORT.OUT2 - Data Browser

qsInvColumnName	qsInvPattern
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	^+T
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	^++T
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	^D>T
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	^TT
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	^S+T
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	++ ++
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	^D+T
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	+++ +++

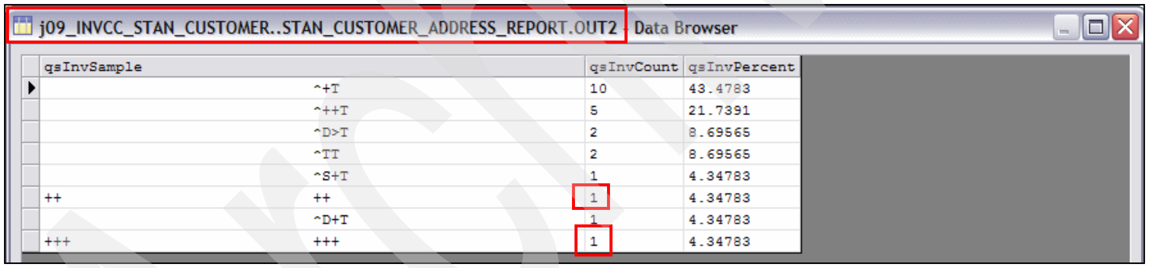
Figure 1-282 Create J09_INVCC_STAN_CUSTOMER 11/16



j09_INVCC_STAN_CUSTOMER..STAN_CUSTOMER_ADDRESS_REPORT.OUT2 - Data Browser

qsInvPattern	qsInvSample
^+T	^+T
^++T	^++T
^D>T	^D>T
^TT	^TT
^S+T	^S+T
++ ++	++ ++
^D+T	^D+T
+++ +++	+++ +++

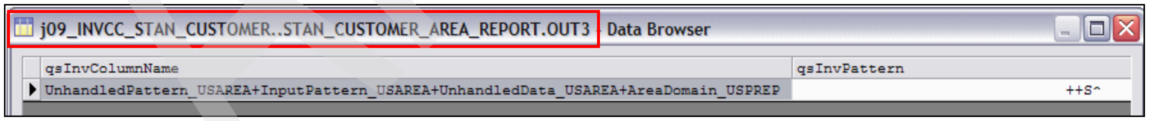
Figure 1-283 Create J09_INVCC_STAN_CUSTOMER 12/16



j09_INVCC_STAN_CUSTOMER..STAN_CUSTOMER_ADDRESS_REPORT.OUT2 - Data Browser

qsInvSample	qsInvCount	qsInvPercent
^+T	10	43.4783
^++T	5	21.7391
^D>T	2	8.69565
^TT	2	8.69565
^S+T	1	4.34783
++ ++	1	4.34783
^D+T	1	4.34783
+++ +++	1	4.34783

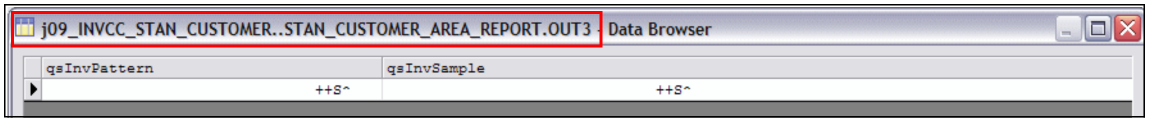
Figure 1-284 Create J09_INVCC_STAN_CUSTOMER 13/16



j09_INVCC_STAN_CUSTOMER..STAN_CUSTOMER_AREA_REPORT.OUT3 - Data Browser

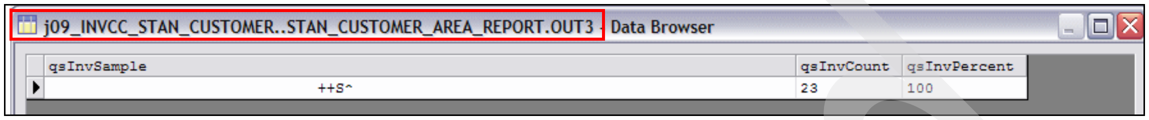
qsInvColumnName	qsInvPattern
UnhandledPattern_USAREA+InputPattern_USAREA+UnhandledData_USAREA+AreaDomain_USPREP	++S^

Figure 1-285 Create J09_INVCC_STAN_CUSTOMER 14/16


j09_INVCC_STAN_CUSTOMER..STAN_CUSTOMER_AREA_REPORT.OUT3 - Data Browser

qsInvPattern	qsInvSample
++S^	++S^

Figure 1-286 Create J09_INVCC_STAN_CUSTOMER 15/16


j09_INVCC_STAN_CUSTOMER..STAN_CUSTOMER_AREA_REPORT.OUT3 - Data Browser

qsInvSample	qsInvCount	qsInvPercent
++S^	23	100

Figure 1-287 Create J09_INVCC_STAN_CUSTOMER 16/16

J09_Z_Override_And_After

Because we found unhandled patterns (such as ++ and +++) in the address domain, we create pattern overrides in the domain-specific USADDR rule set, and re-run the job "J08_STAN_CUSTOMER_Domain_Specific" on page 241. We also use a classification override for AVEDUE, which was discovered in "J06_INVCC_CASS" on page 228.

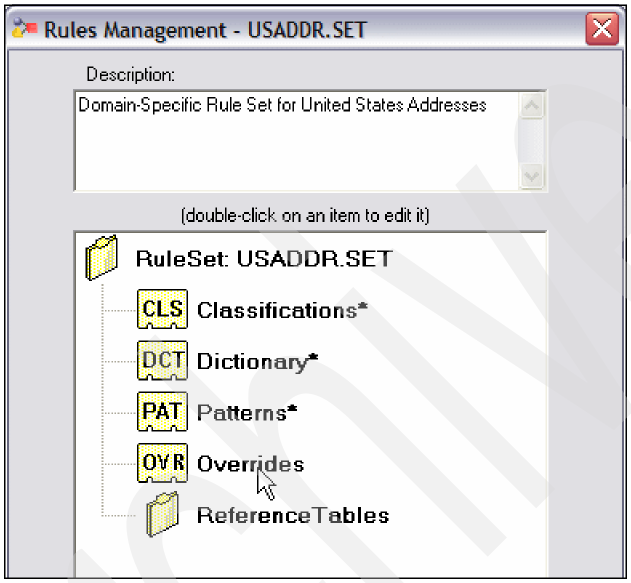
Figure 1-288 on page 263 through Figure 1-293 on page 265 describe the main steps in performing the overrides

The steps are as follows:

1. Expand the Standardization Rules folder in the Designer client repository tree and click the USADDR rule set to open the Rules Management - USADDR.SET window as shown in Figure 1-288 on page 263. Click **Overrides** to add an override for that rule set.
2. In the Input Pattern - USADDR window shown in Figure 1-289 on page 264, the various fields for overriding the unhandled pattern ++ is shown. The first token + is marked with the StreetName override code, while the second token + is marked with the AdditionalAddress override code as seen in the Current Pattern List. Additionally, the override directs the Original Value and Move All Remaining and Leading space to be moved to the target area.
3. Click **Add** in Figure 1-289 on page 264 to add this override to the Override Summary as seen in Figure 1-290 on page 264.
4. In the Classification - USADDR window shown in Figure 1-291 on page 265, the string `AVEDUE` is overridden (by entering it in the Input Token field) with the value `AVE` (by entering `AVE` in the Standard Form field), and assigned a classification of `T` (street type in the Classification field). The Comparison Threshold value chosen is 850.

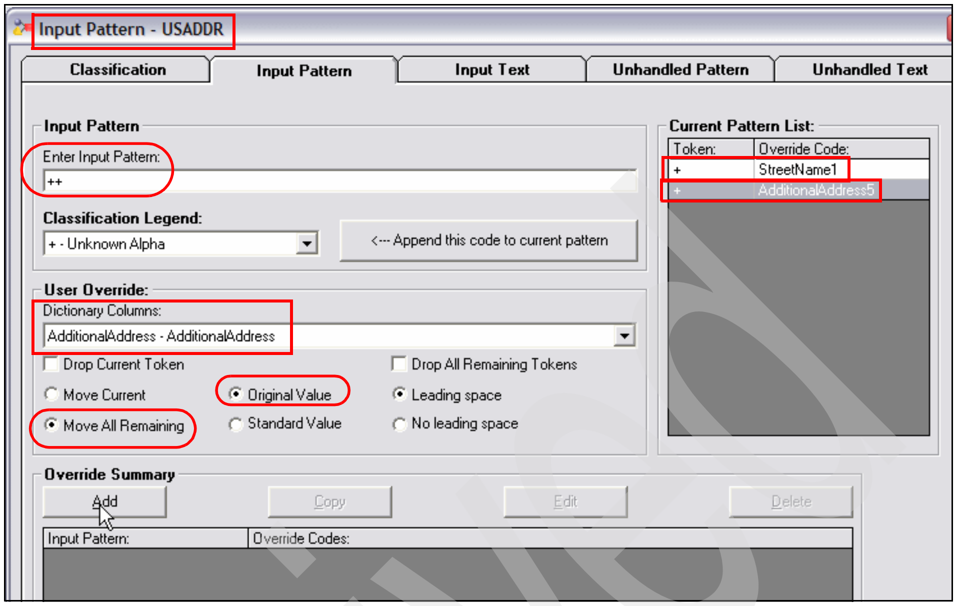
5. Click **Add** in Figure 1-291 on page 265 to add this override to the Override Summary as seen in Figure 1-292 on page 265.
6. After these overrides, jobs J08_STAN_CUSTOMER_Domain_Specific and J09_INVCC_STAN_CUSTOMER was rerun (after provisioning and compiling the jobs) to verify that the overrides resolved the problem. Figure 1-293 on page 265 shows the partial output report of the Investigate stage. It shows no unhandled patterns indicating that the overrides were successful.

Now, proceed to “J10_MATCHFREQ_STAN_CUSTOMER” on page 266.



Domain-Specific Rule Set for United States Addresses

Figure 1-288 J09_Z_Override_And_After job 1/6



Input Pattern
Enter Input Pattern: ++
Classification Legend: + - Unknown Alpha
<--- Append this code to current pattern

User Override:
Dictionary Columns: AdditionalAddress - AdditionalAddress
Drop Current Token [ ]
Drop All Remaining Tokens [ ]
Move Current ( ) | Original Value (●) | Leading space (●)
Move All Remaining (●) | Standard Value ( ) | No leading space ( )

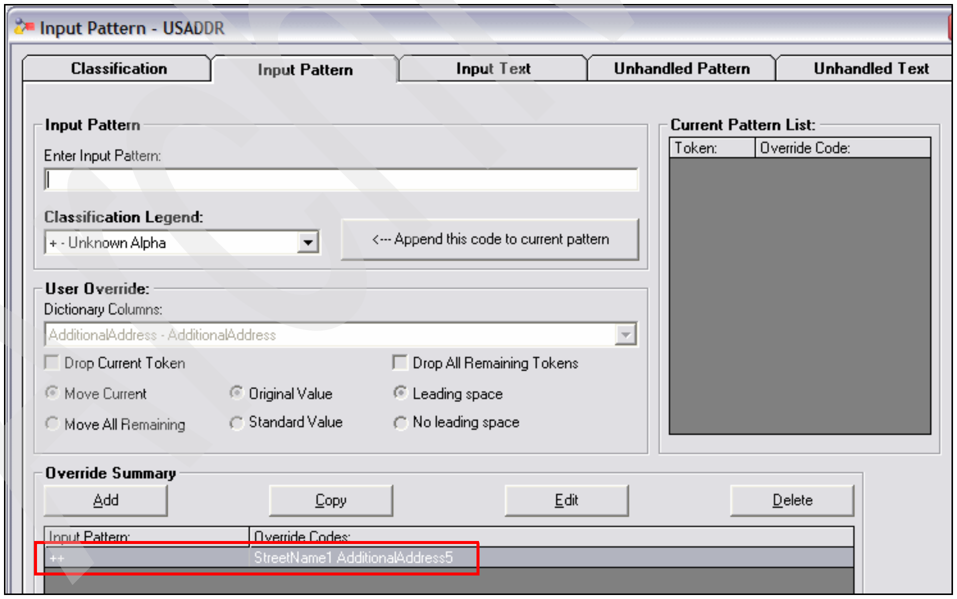
Current Pattern List:

Token:	Override Code:
+	StreetName1
+	AdditionalAddress5

Override Summary

Input Pattern:	Override Codes:

Figure 1-289 J09_Z_Override_And_After job 2/6



Input Pattern
Enter Input Pattern:
Classification Legend: + - Unknown Alpha
<--- Append this code to current pattern

User Override:
Dictionary Columns: AdditionalAddress - AdditionalAddress
Drop Current Token [ ]
Drop All Remaining Tokens [ ]
Move Current (●) | Original Value ( ) | Leading space (●)
Move All Remaining ( ) | Standard Value ( ) | No leading space ( )

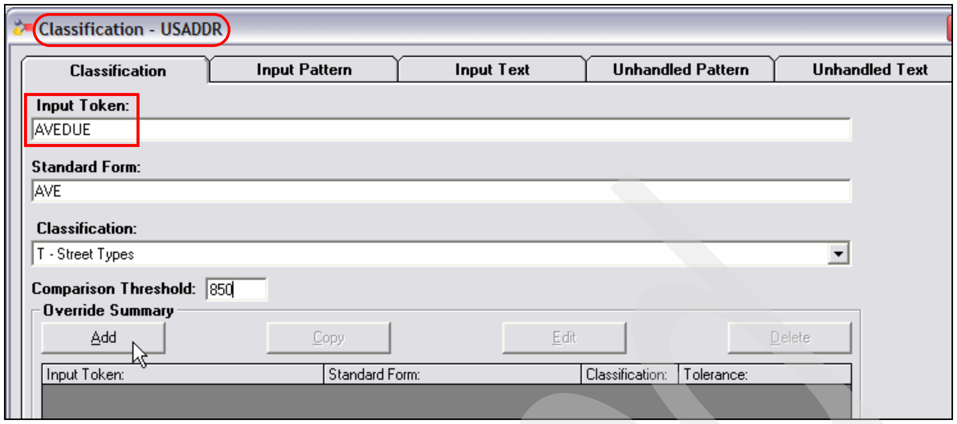
Current Pattern List:

Token:	Override Code:

Override Summary

Input Pattern:	Override Codes:
++	StreetName1 AdditionalAddress5

Figure 1-290 J09_Z_Override_And_After job 3/6


Classification - USADDR

Classification | Input Pattern | Input Text | Unhandled Pattern | Unhandled Text

Input Token: AVEDUE

Standard Form: AVE

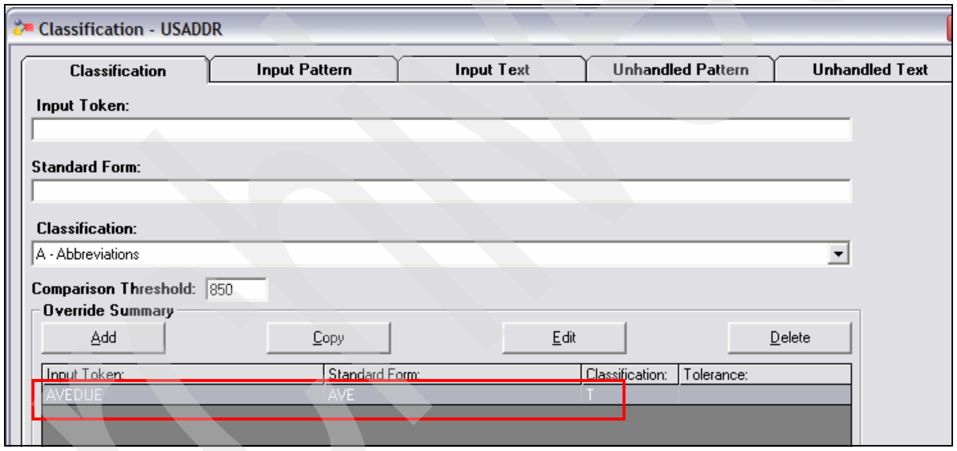
Classification: T - Street Types

Comparison Threshold: 850

Override Summary

Input Token:	Standard Form:	Classification:	Tolerance:

Figure 1-291 J09_Z_Override_And_After job 4/6


Classification - USADDR

Input Token:

Standard Form:

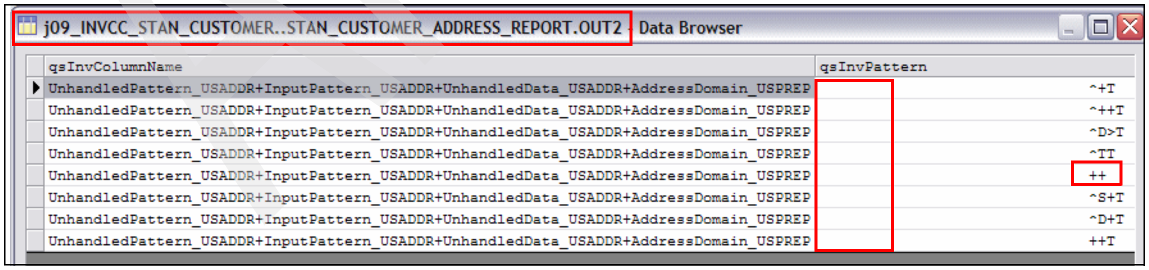
Classification: A - Abbreviations

Comparison Threshold: 850

Override Summary

Input Token:	Standard Form:	Classification:	Tolerance:
AVEDUE	AVE	T	

Figure 1-292 J09_Z_Override_And_After job 5/6


j09_INVCC_STAN_CUSTOMER..STAN_CUSTOMER_ADDRESS_REPORT.OUT2 - Data Browser

qsInvColumnName	qsInvPattern
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	^+T
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	^++T
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	^D>T
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	^TT
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	++
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	^S+T
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	^D+T
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	++T

Figure 1-293 J09_Z_Override_And_After job 6/6

J10_MATCHFREQ_STAN_CUSTOMER

After the unhandled patterns are handled, we proceed to generate frequency distribution on *all* the columns in the credit card customer file using the Match Frequency stage. The idea is to generate match frequency for all the columns so that it can be used with any match specification.

Figure 1-294 on page 267 through Figure 1-299 on page 269 describe the steps using Designer Client to build and execute the DataStage job to perform this task.

The steps are as follows:

1. Figure 1-294 on page 267 shows the various stages that are used in this job, including the data set that was created in "J08_STAN_CUSTOMER_Domain_Specific" on page 241, a Match Frequency stage, and a Sequential File stage. We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object created earlier.
2. Configure the input STAN_CUSTOMER data set. We do not repeat this process here because it is similar to the steps shown in Figure 1-138 on page 172 through Figure 1-142 on page 174.
3. The Match Frequency stage generates frequency information using any data that provides the columns needed by a match. The Match Frequency stage processes frequency data independently from executing a match. The output link of the stage carries four columns: qsFreqVal, qsFreqCounts, qsFreqColumnID, and qsFreqHeaderFlag.

   Configure the Match Frequency stage as follows:

   a. Right-click the **MATCH_FREQUENCY** stage icon and select **Properties** as shown in Figure 1-295 on page 267.
   b. In the MATCH_FREQUENCY - Match Frequency Stage window that opens, select **Do not use a Match Specification**, and click **Stage Properties** as shown in Figure 1-296 on page 267.
   c. From the Output tab, for the link IN (Output name) click the Mapping tab, and copy the columns in the left pane to the right pane as shown in Figure 1-297 on page 268.
4. Configure the sequential file for the output of this stage. We do not repeat this process here because it is similar to that shown in Figure 1-174 on page 191 through Figure 1-177 on page 193.
5. After saving, compiling, and running this job (Figure 1-298 on page 268), the contents of the output of this stage are listed in Figure 1-299 on page 269. The interpretation of the format and content of this file is not documented.

Now, proceed to “J10_Undup_MatchSpec_STAN_CUSTOMER” on page 269.

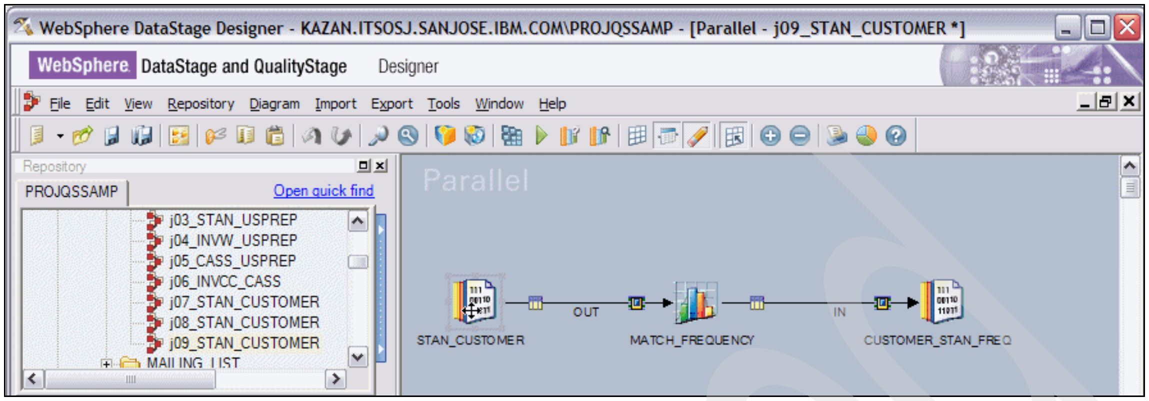


Figure 1-294 J10_MATCHFREQ_STAN_CUSTOMER job 1/6

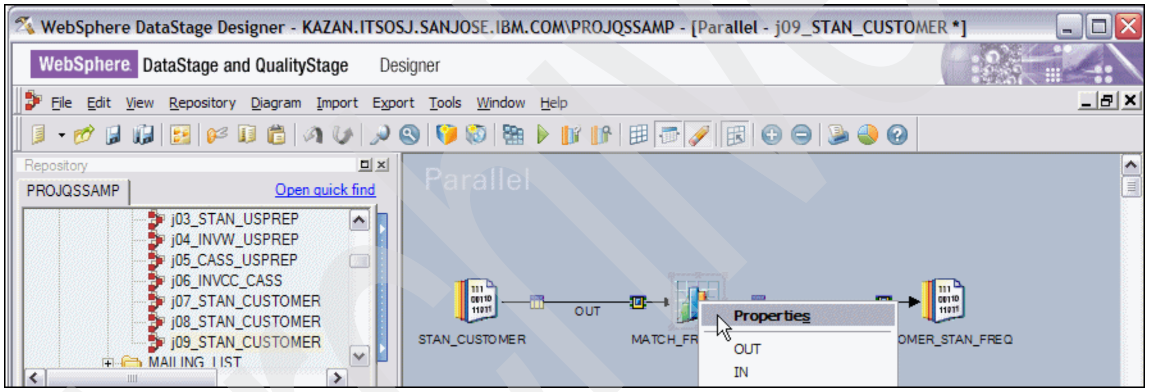
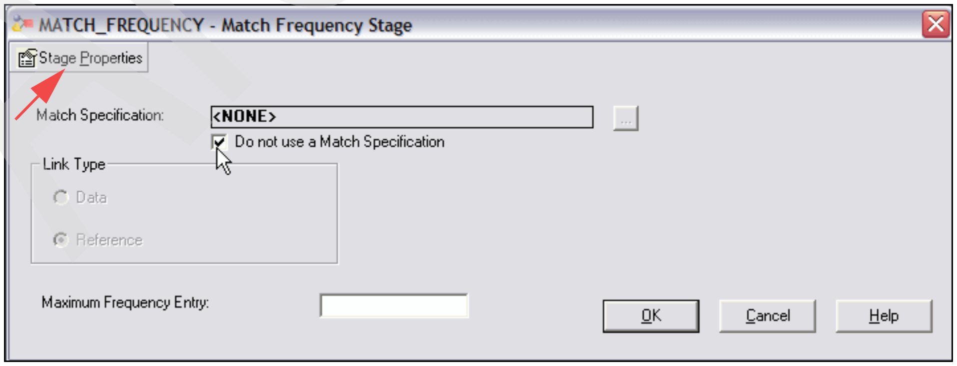


Figure 1-295 J10_MATCHFREQ_STAN_CUSTOMER job 2/6


MATCH_FREQUENCY - Match Frequency Stage

Stage Properties

Match Specification: <NONE>

[✓] Do not use a Match Specification

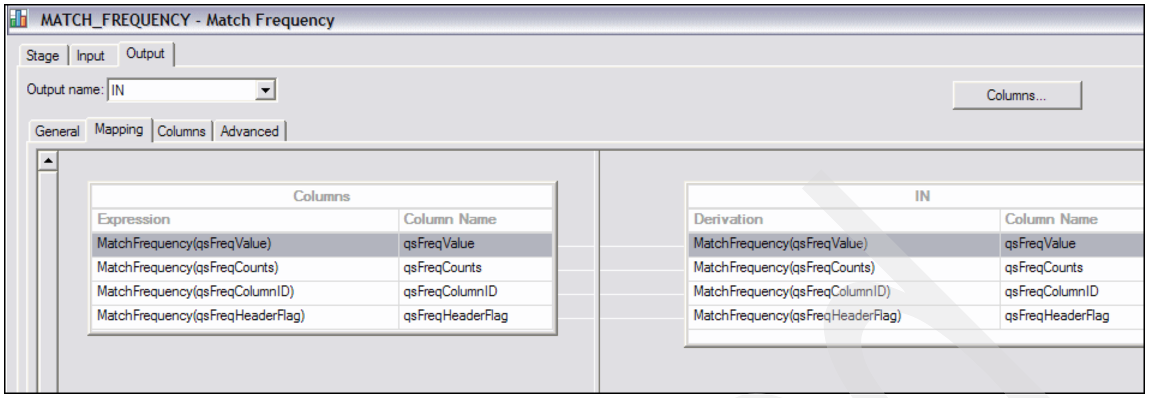
Link Type

( ) Data

(•) Reference

Maximum Frequency Entry:

Figure 1-296 J10_MATCHFREQ_STAN_CUSTOMER job 3/6


MATCH_FREQUENCY - Match Frequency

Stage | Input | Output

Output name: IN

Columns...

General | Mapping | Columns | Advanced

Columns	
Expression	Column Name
MatchFrequency(qsFreqValue)	qsFreqValue
MatchFrequency(qsFreqCounts)	qsFreqCounts
MatchFrequency(qsFreqColumnID)	qsFreqColumnID
MatchFrequency(qsFreqHeaderFlag)	qsFreqHeaderFlag

IN	
Derivation	Column Name
MatchFrequency(qsFreqValue)	qsFreqValue
MatchFrequency(qsFreqCounts)	qsFreqCounts
MatchFrequency(qsFreqColumnID)	qsFreqColumnID
MatchFrequency(qsFreqHeaderFlag)	qsFreqHeaderFlag

Figure 1-297 J10_MATCHFREQ_STAN_CUSTOMER job 4/6

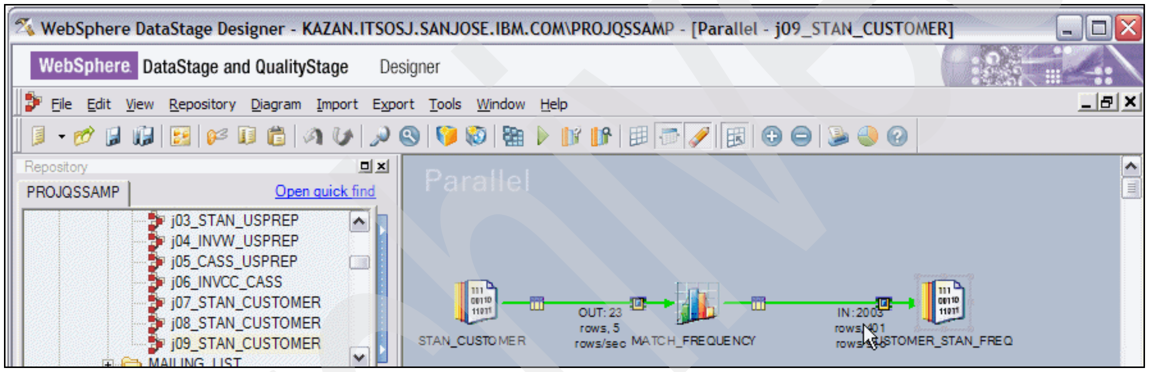
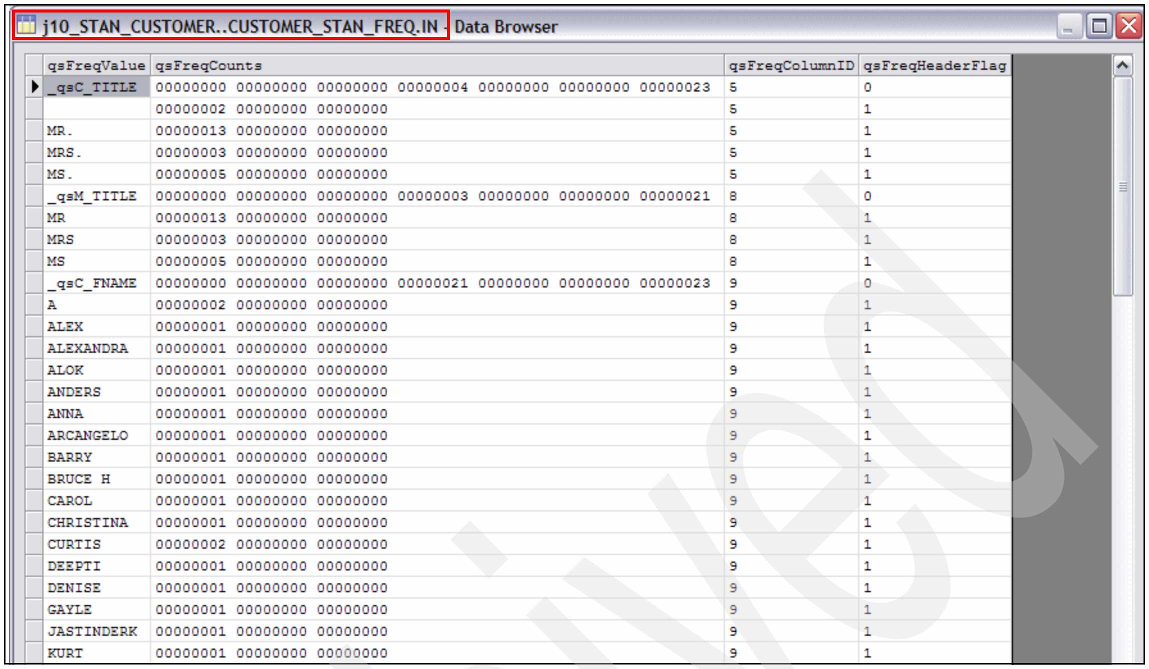

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j09_STAN_CUSTOMER]

Figure 1-298 J10_MATCHFREQ_STAN_CUSTOMER job 5/6


j10_STAN_CUSTOMER..CUSTOMER_STAN_FREQ.IN - Data Browser

qsFreqValue	qsFreqCounts	qsFreqColumnID	qsFreqHeaderFlag
_qsC_TITLE	00000000 00000000 00000000 00000004 00000000 00000000 00000023	5	0
	00000002 00000000 00000000	5	1
MR.	00000013 00000000 00000000	5	1
MRS.	00000003 00000000 00000000	5	1
MS.	00000005 00000000 00000000	5	1
_qsM_TITLE	00000000 00000000 00000000 00000003 00000000 00000000 00000021	8	0
MR	00000013 00000000 00000000	8	1
MRS	00000003 00000000 00000000	8	1
MS	00000005 00000000 00000000	8	1
_qsC_FNAME	00000000 00000000 00000000 00000021 00000000 00000000 00000023	9	0
A	00000002 00000000 00000000	9	1
ALEX	00000001 00000000 00000000	9	1
ALEXANDRA	00000001 00000000 00000000	9	1
ALOK	00000001 00000000 00000000	9	1
ANDERS	00000001 00000000 00000000	9	1
ANNA	00000001 00000000 00000000	9	1
ARCANGELO	00000001 00000000 00000000	9	1
BARRY	00000001 00000000 00000000	9	1
BRUCE H	00000001 00000000 00000000	9	1
CAROL	00000001 00000000 00000000	9	1
CHRISTINA	00000001 00000000 00000000	9	1
CURTIS	00000002 00000000 00000000	9	1
DEEPTI	00000001 00000000 00000000	9	1
DENISE	00000001 00000000 00000000	9	1
GAYLE	00000001 00000000 00000000	9	1
JASTINDERK	00000001 00000000 00000000	9	1
KURT	00000001 00000000 00000000	9	1

Figure 1-299 J10_MATCHFREQ_STAN_CUSTOMER job 6/6

J10_Undup_MatchSpec_STAN_CUSTOMER

In this step, we generate a match specification for an Unduplicate match stage using as input the match frequency data created in the "J10_Undup_MatchSpec_STAN_CUSTOMER" on page 269 job. The specification included two passes:

- ► The first pass blocks on name, address, and area.
- ► The second pass blocks on the first three digits of the ZIP code.

Figure 1-300 on page 271 through Figure 1-316 on page 282 describe the main steps using Designer Client to build and execute the Unduplicate Match Specification job.

The main steps are as follows:

1. From the File menu of the Designer Client, click **New**. In the New window that opens, select the Match Specification icon, and click **OK** as shown in Figure 1-300 on page 271.
2. In the Match Designer - Specification window, select **Unduplicate** for the Match Type. Click the **Define input for this Specification** icon as shown in Figure 1-301 on page 272.

3. In the Input Columns window that opens, click **Load** to open the Table Definitions window as shown in Figure 1-302 on page 272.
4. In the Table Definitions window, navigate to the CUSTOMER_STAN table definition in the Table Definition folder structure, and click **OK** as shown in Figure 1-303 on page 273.
5. The window closes and the columns display in the Input Columns window as shown in Figure 1-304 on page 273.
6. Rename the default MyPass to CUSTOMER_NAME_ADDR as shown in Figure 1-305 on page 274 and Figure 1-306 on page 274.
7. Click **Add** in Blocking Columns section to add the blocking column MatchPrimaryWord1NYIIS_USNAME using character comparison as shown in Figure 1-306 on page 274 through Figure 1-309 on page 277. Additional blocking columns (not shown here) that we added were AddressType_USADDR, StreetNameNYIIS_USADDR, and ZipCode_USAREA.
8. To add Match Commands for the GENDER column, click **Add** in the Match Commands section as shown in Figure 1-310 on page 277. Additional match commands (not shown here) added were MiddleName_USNAME, PrimaryName_USNAME, HouseNumber_USADDR, StreetPrefixDirectional_USADDR, and StreetName_USADDR.
9. Specify the Cutoff Values of 2 for Clerical and 5 for Match.
10. On the toolbar, click **Configure Specification** and select **Test Environment** to configure the Match specification to specify sample data, frequency information, and results database connection[25] information as shown in Figure 1-311 on page 278. Click **Update** to save any changed settings.
11. Click **Test All Passes** in the toolbar as shown in Figure 1-312 on page 279. When the test runs are completed for all active passes, you can click any pass to view results as shown in Figure 1-313 on page 279.
12. Click the CUSTOMER_NAME_ADDR pass to view the results as shown in Figure 1-314 on page 280.

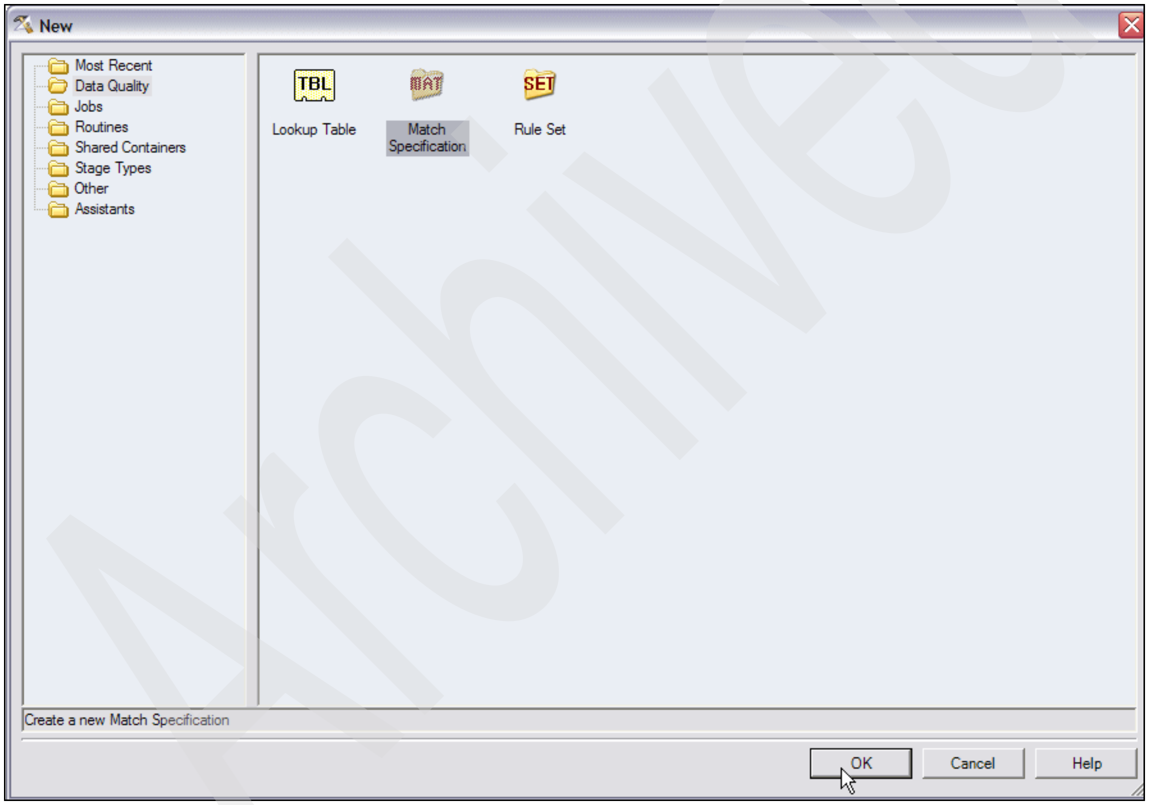
    In the Test Results pane, there are three matched blocks (as indicated by the unique SetID) with duplicates (DA), master records (XA), and clerical review (CP) records. The weights indicate whether the record is XA, DA, or CP within a block based on the Clerical and Match Cutoff Values.
13. Click the CUSTOMER_ZIP3 pass to view the results as shown in Figure 1-315 on page 281. The Test Results pane shows two matched blocks (as indicated by the unique SetID) with duplicates (DA), master records (XA), and clerical review (CP) records.

[25] It is an ODBC connection defined on the client computer.

14. To view the statistics of each of these passes, click the Pass Statistics tab in Figure 1-315 on page 281 to view the results in Figure 1-316 on page 282. This tab provides total statistics for all the passes as well as those related to each pass in the specification. You can also view the results graphically.

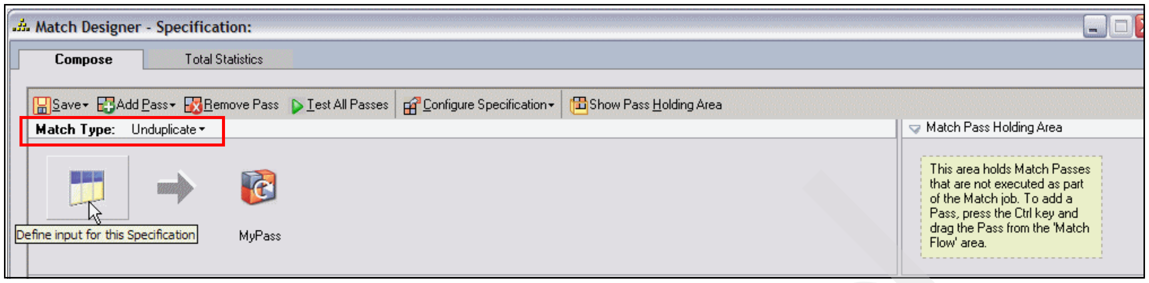
    Of particular interest is the statistic OVERFLOW blocks—a non-zero value indicates the need to increase the block size or define more restrictive blocking columns.

The test of the match specification with the full volume of data appeared to deliver results that were accurate. We use this specification in an Unduplicate match stage, as described in "J11_UNDUP_DEP_MATCH_CUSTOMER" on page 282.



Create a new Match Specification

Figure 1-300 Create J10_Undup_MatchSpec_STAN_CUSTOMER job 1/17



This area holds Match Passes that are not executed as part of the Match job. To add a Pass, press the Ctrl key and drag the Pass from the 'Match Flow' area.

Figure 1-301 Create J10_Undup_MatchSpec_STAN_CUSTOMER job 2/17

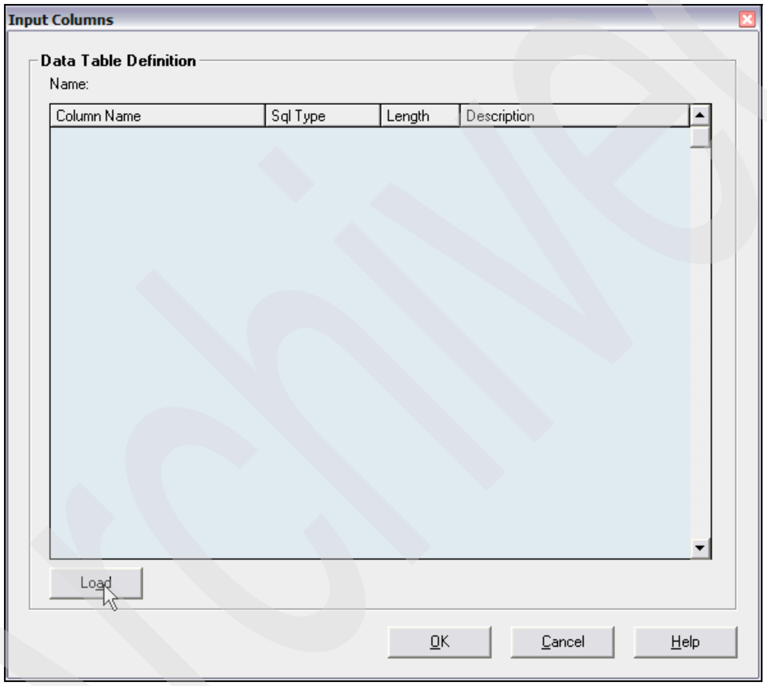


Figure 1-302 Create J10_Undup_MatchSpec_STAN_CUSTOMER job 3/17

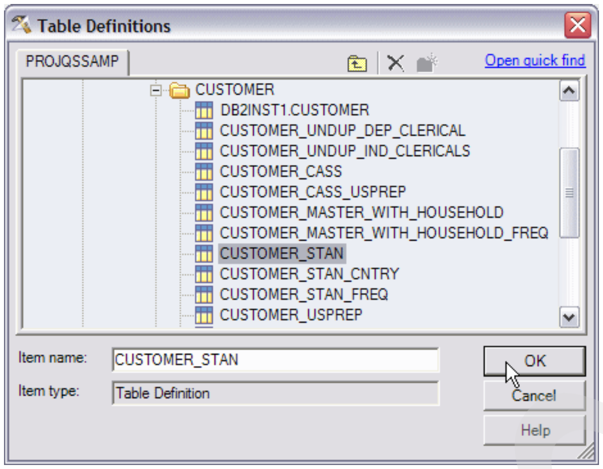


Table Definitions

PROJQSSAMP

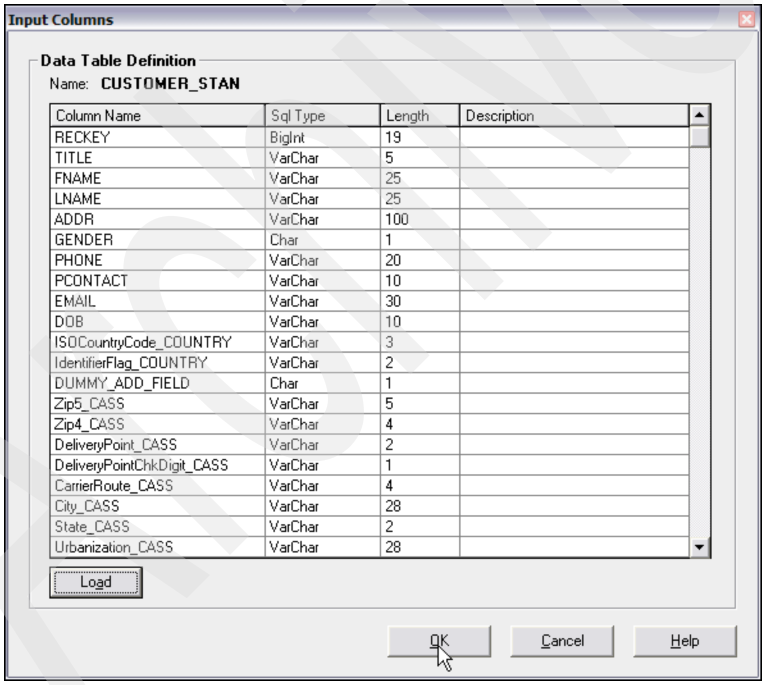
Open quick find

- CUSTOMER
  - DB2INST1.CUSTOMER
  - CUSTOMER_UNDUP_DEP_CLERICAL
  - CUSTOMER_UNDUP_IND_CLERICALS
  - CUSTOMER_CASS
  - CUSTOMER_CASS_USPREP
  - CUSTOMER_MASTER_WITH_HOUSEHOLD
  - CUSTOMER_MASTER_WITH_HOUSEHOLD_FREQ
  - CUSTOMER_STAN
  - CUSTOMER_STAN_CNTRY
  - CUSTOMER_STAN_FREQ
  - CUSTOMER_USPREP

Item name: CUSTOMER_STAN

Item type: Table Definition

Figure 1-303   Create J10_Undup_MatchSpec_STAN_CUSTOMER job 4/17



Input Columns

Data Table Definition

Name: CUSTOMER_STAN

Column Name	Sql Type	Length	Description
RECKEY	BigInt	19	
TITLE	VarChar	5	
FNAME	VarChar	25	
LNAME	VarChar	25	
ADDR	VarChar	100	
GENDER	Char	1	
PHONE	VarChar	20	
PCONTACT	VarChar	10	
EMAIL	VarChar	30	
DOB	VarChar	10	
ISOCountryCode_COUNTRY	VarChar	3	
IdentifierFlag_COUNTRY	VarChar	2	
DUMMY_ADD_FIELD	Char	1	
Zip5_CASS	VarChar	5	
Zip4_CASS	VarChar	4	
DeliveryPoint_CASS	VarChar	2	
DeliveryPointChkDigit_CASS	VarChar	1	
CarrierRoute_CASS	VarChar	4	
City_CASS	VarChar	28	
State_CASS	VarChar	2	
Urbanization_CASS	VarChar	28	

Figure 1-304   Create J10_Undup_MatchSpec_STAN_CUSTOMER job 5/17

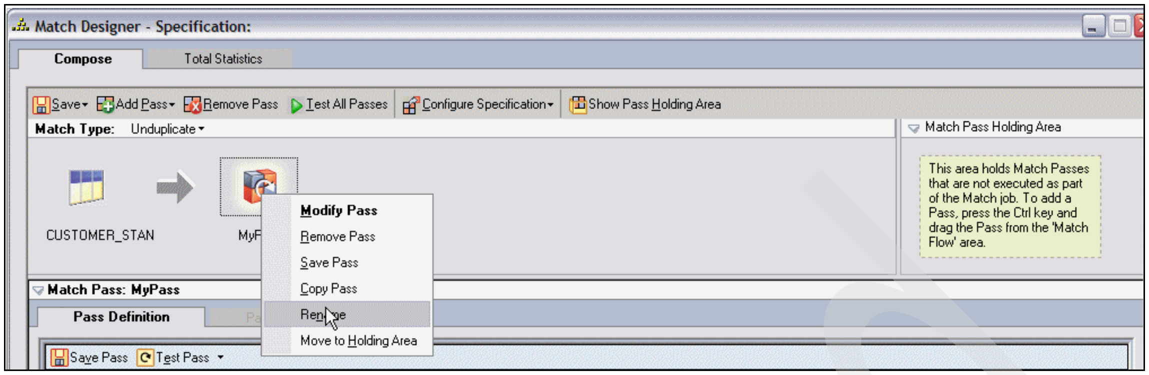


Figure 1-305 Create J10_Undup_MatchSpec_STAN_CUSTOMER job 6/17

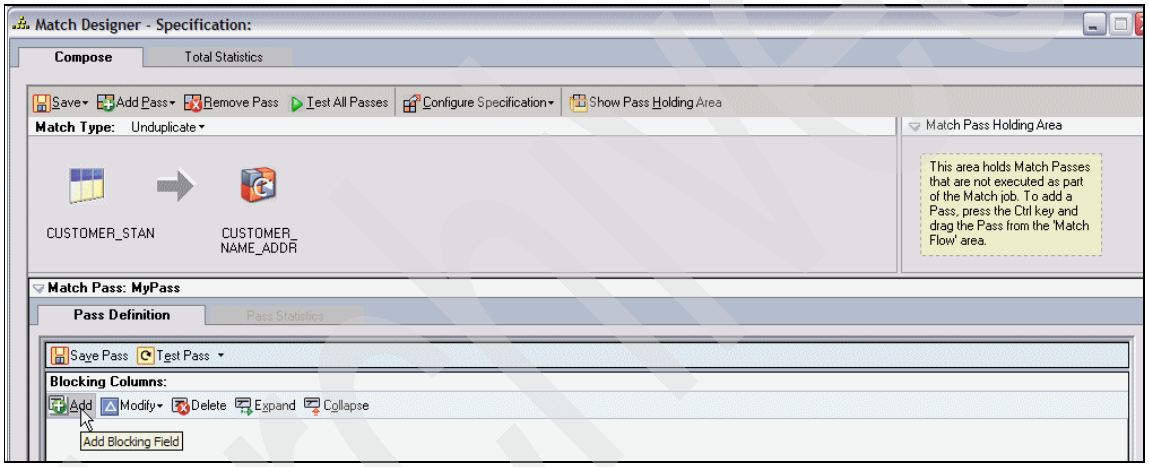
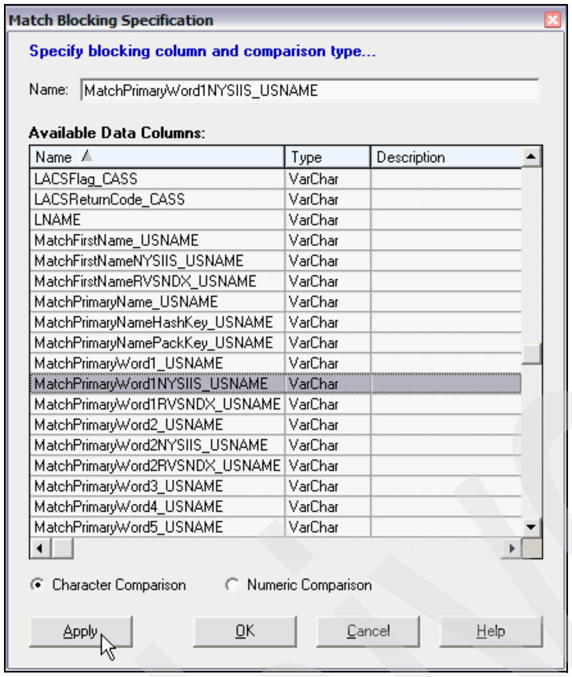


Figure 1-306 Create J10_Undup_MatchSpec_STAN_CUSTOMER job 7/17



Match Blocking Specification

Specify blocking column and comparison type...

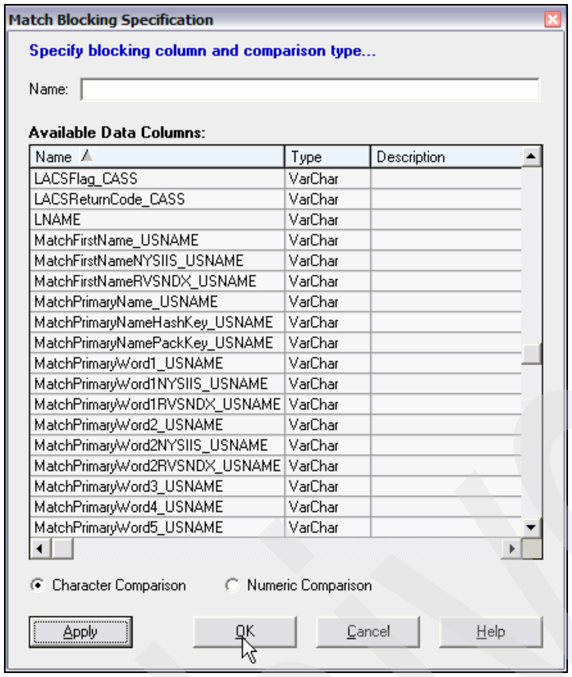
Name: MatchPrimaryWord1NYSIIS_USNAME

Available Data Columns:

Name	Type	Description
LACSFlag_CASS	VarChar	
LACSReturnCode_CASS	VarChar	
LNAME	VarChar	
MatchFirstName_USNAME	VarChar	
MatchFirstNameNYSIIS_USNAME	VarChar	
MatchFirstNameRVSNDX_USNAME	VarChar	
MatchPrimaryName_USNAME	VarChar	
MatchPrimaryNameHashKey_USNAME	VarChar	
MatchPrimaryNamePackKey_USNAME	VarChar	
MatchPrimaryWord1_USNAME	VarChar	
MatchPrimaryWord1NYSIIS_USNAME	VarChar	
MatchPrimaryWord1RVSNDX_USNAME	VarChar	
MatchPrimaryWord2_USNAME	VarChar	
MatchPrimaryWord2NYSIIS_USNAME	VarChar	
MatchPrimaryWord2RVSNDX_USNAME	VarChar	
MatchPrimaryWord3_USNAME	VarChar	
MatchPrimaryWord4_USNAME	VarChar	
MatchPrimaryWord5_USNAME	VarChar	

[✓ Character Comparison] Numeric Comparison

Figure 1-307 Create J10_Undup_MatchSpec_STAN_CUSTOMER job 8/17


Match Blocking Specification

Specify blocking column and comparison type...

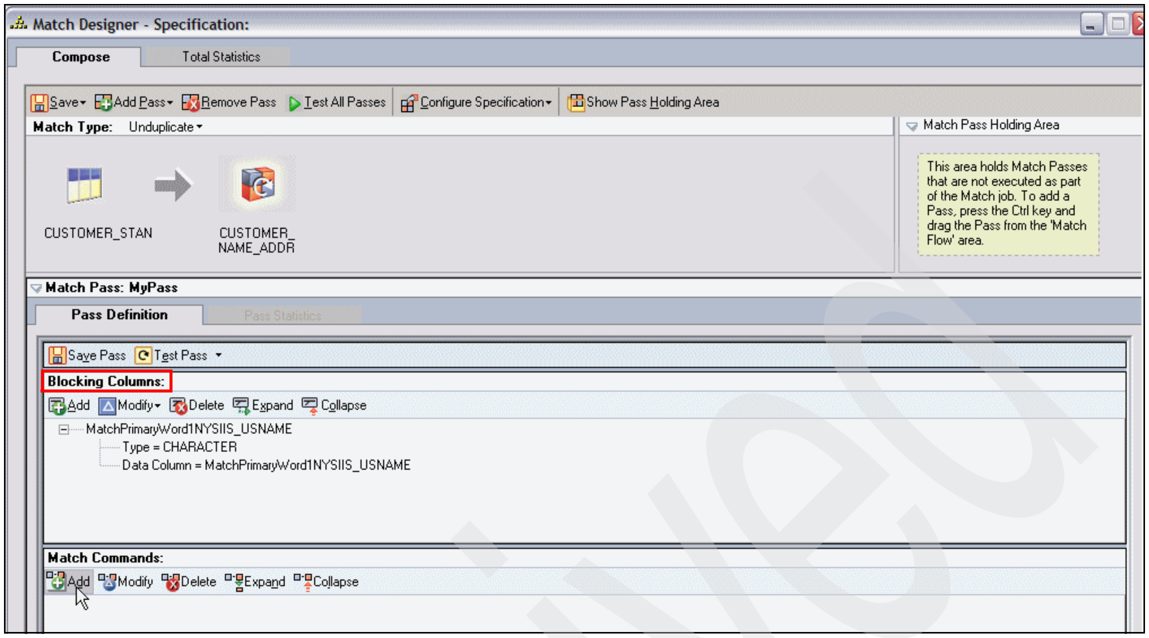
Name:

Available Data Columns:

Name	Type	Description
LACSFlag_CASS	VarChar	
LACSReturnCode_CASS	VarChar	
LNAME	VarChar	
MatchFirstName_USNAME	VarChar	
MatchFirstNameNYSIIS_USNAME	VarChar	
MatchFirstNameRVSNDX_USNAME	VarChar	
MatchPrimaryName_USNAME	VarChar	
MatchPrimaryNameHashKey_USNAME	VarChar	
MatchPrimaryNamePackKey_USNAME	VarChar	
MatchPrimaryWord1_USNAME	VarChar	
MatchPrimaryWord1NYSIIS_USNAME	VarChar	
MatchPrimaryWord1RVSNDX_USNAME	VarChar	
MatchPrimaryWord2_USNAME	VarChar	
MatchPrimaryWord2NYSIIS_USNAME	VarChar	
MatchPrimaryWord2RVSNDX_USNAME	VarChar	
MatchPrimaryWord3_USNAME	VarChar	
MatchPrimaryWord4_USNAME	VarChar	
MatchPrimaryWord5_USNAME	VarChar	

[✓ Character Comparison] Numeric Comparison

Figure 1-308 Create J10_Undup_MatchSpec_STAN_CUSTOMER job 9/17


Match Designer - Specification:

Compose | Total Statistics

Save | Add Pass | Remove Pass | Test All Passes | Configure Specification | Show Pass Holding Area

Match Type: Unduplicate

CUSTOMER_STAN → CUSTOMER_NAME_ADDR

Match Pass Holding Area

This area holds Match Passes that are not executed as part of the Match job. To add a Pass, press the Ctrl key and drag the Pass from the 'Match Flow' area.

Match Pass: MyPass

Pass Definition | Pass Statistics

Save Pass | Test Pass

Blocking Columns:

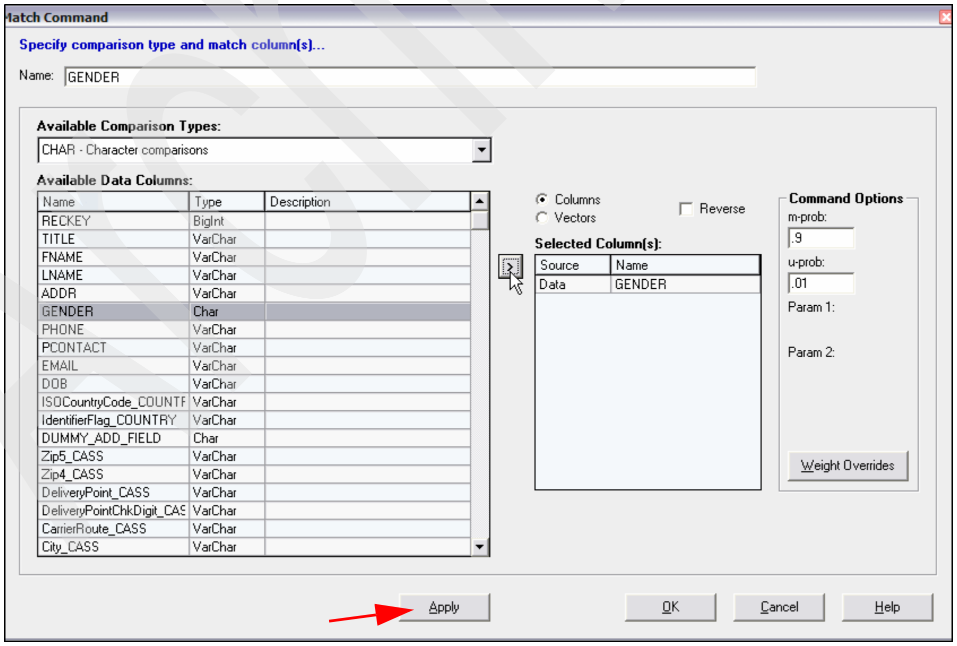
Add | Modify | Delete | Expand | Collapse

```
MatchPrimaryWord1NYSIIS_USNAME
    Type = CHARACTER
    Data Column = MatchPrimaryWord1NYSIIS_USNAME
```

Match Commands:

Add | Modify | Delete | Expand | Collapse

Figure 1-309 Create J10_Undup_MatchSpec_STAN_CUSTOMER job 10/17


Match Command

Specify comparison type and match column(s)...

Name: GENDER

Available Comparison Types: CHAR - Character comparisons

Available Data Columns:

Name	Type	Description
RECKEY	BigInt	
TITLE	VarChar	
FNAME	VarChar	
LNAME	VarChar	
ADDR	VarChar	
GENDER	Char	
PHONE	VarChar	
PCONTACT	VarChar	
EMAIL	VarChar	
DOB	VarChar	
ISOCountryCode_COUNTF	VarChar	
IdentifierFlag_COUNTRY	VarChar	
DUMMY_ADD_FIELD	Char	
Zip5_CASS	VarChar	
Zip4_CASS	VarChar	
DeliveryPoint_CASS	VarChar	
DeliveryPointChkDigit_CAS	VarChar	
CarrierRoute_CASS	VarChar	
City_CASS	VarChar	

[✓ Columns] Vectors

Reverse

Selected Column(s):

Source	Name
Data	GENDER

Command Options

m-prob: .9

u-prob: .01

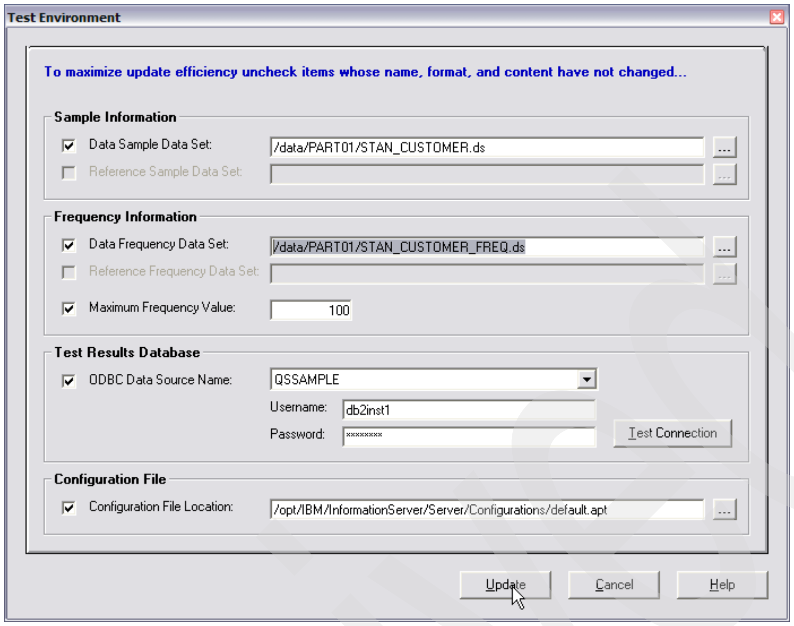
Param 1:

Param 2:

Weight Overrides

Apply | OK | Cancel | Help

Figure 1-310 Create J10_Undup_MatchSpec_STAN_CUSTOMER job 11/17



Test Environment

To maximize update efficiency uncheck items whose name, format, and content have not changed...

Sample Information

- [✓] Data Sample Data Set: /data/PART01/STAN_CUSTOMER.ds
- [ ] Reference Sample Data Set:

Frequency Information

- [✓] Data Frequency Data Set: /data/PART01/STAN_CUSTOMER_FREQ.ds
- [ ] Reference Frequency Data Set:
- [✓] Maximum Frequency Value: 100

Test Results Database

- [✓] ODBC Data Source Name: QSSAMPLE

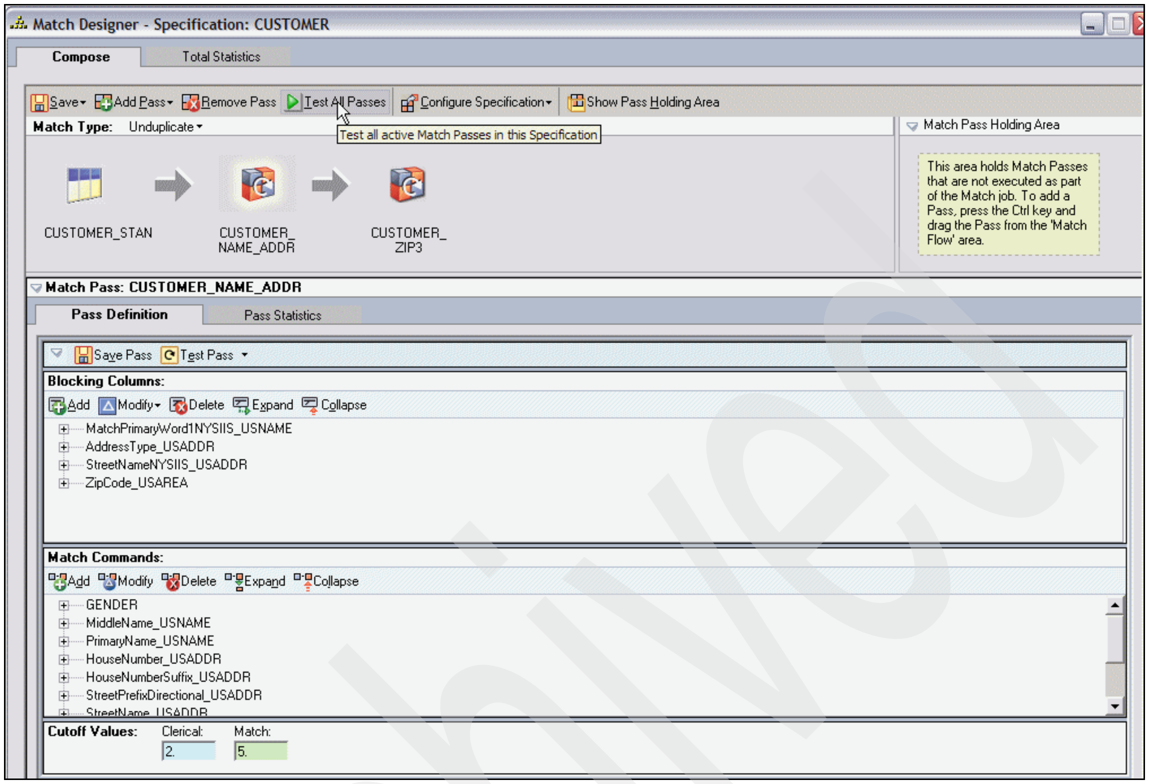
Username: db2inst1

Password: ********

Configuration File

- [✓] Configuration File Location: /opt/IBM/InformationServer/Server/Configurations/default.apt

Figure 1-311 Create J10_Undup_MatchSpec_STAN_CUSTOMER job 12/17



Match Type: Unduplicate

This area holds Match Passes that are not executed as part of the Match job. To add a Pass, press the Ctrl key and drag the Pass from the 'Match Flow' area.

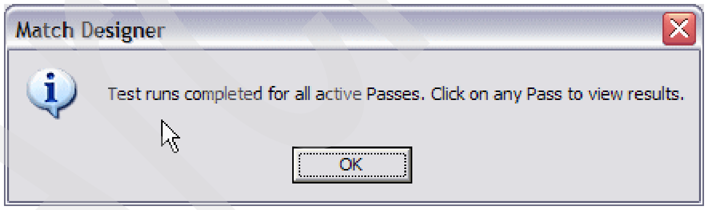
Match Pass: CUSTOMER_NAME_ADDR

Blocking Columns:
- MatchPrimaryWord1NYSIIS_USNAME
- AddressType_USADDR
- StreetNameNYSIIS_USADDR
- ZipCode_USAREA

Match Commands:
- GENDER
- MiddleName_USNAME
- PrimaryName_USNAME
- HouseNumber_USADDR
- HouseNumberSuffix_USADDR
- StreetPrefixDirectional_USADDR
- StreetName_USADDR

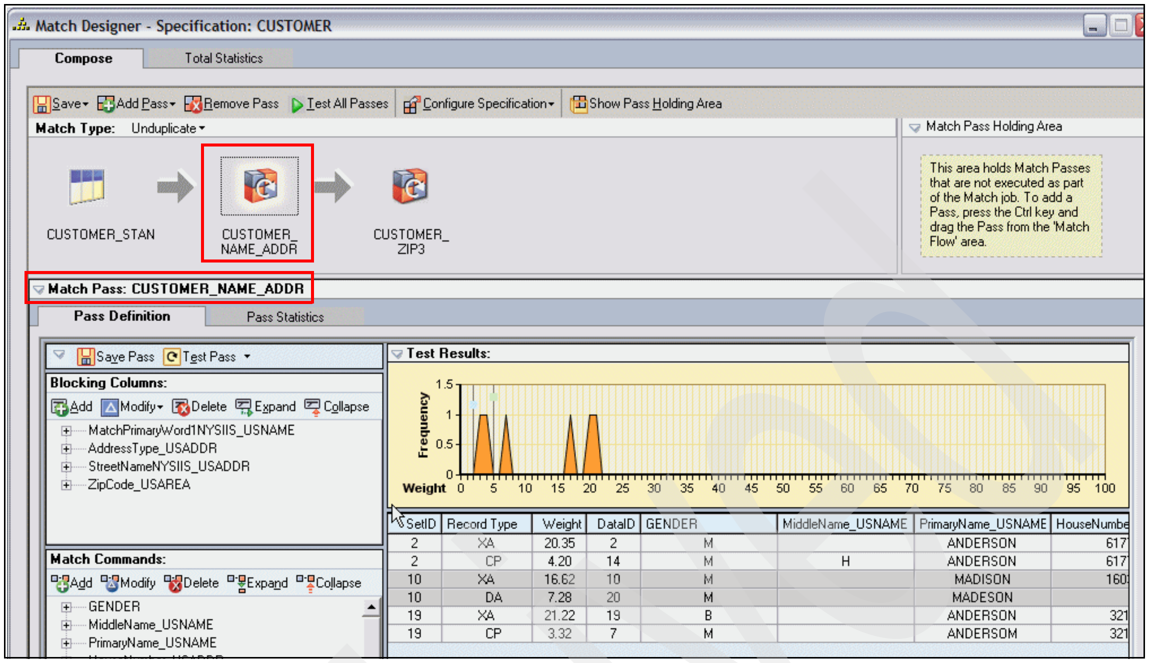
Cutoff Values: Clerical: 2. Match: 5.

Figure 1-312 Create J10_Undup_MatchSpec_STAN_CUSTOMER job 13/17



Test runs completed for all active Passes. Click on any Pass to view results.

Figure 1-313 Create J10_Undup_MatchSpec_STAN_CUSTOMER job 14/17



Match Type: Unduplicate

This area holds Match Passes that are not executed as part of the Match job. To add a Pass, press the Ctrl key and drag the Pass from the 'Match Flow' area.

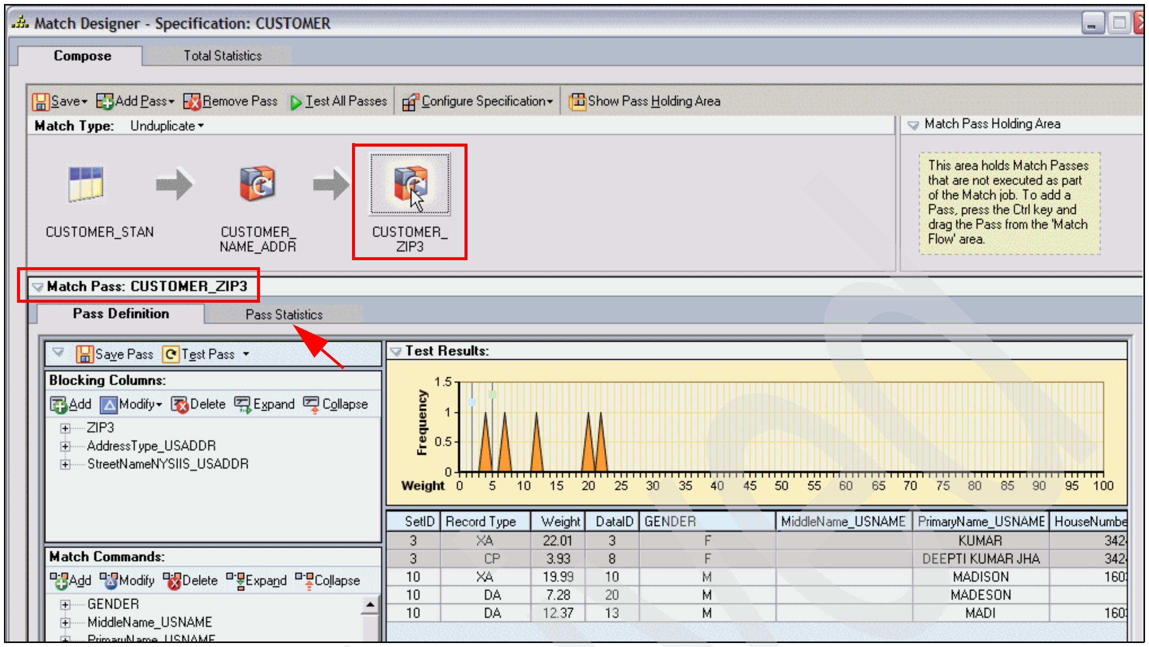
Match Pass: CUSTOMER_NAME_ADDR

Blocking Columns:
- MatchPrimaryWord1NYSIIS_USNAME
- AddressType_USADDR
- StreetNameNYSIIS_USADDR
- ZipCode_USAREA

Match Commands:
- GENDER
- MiddleName_USNAME
- PrimaryName_USNAME

SetID	Record Type	Weight	DataID	GENDER	MiddleName_USNAME	PrimaryName_USNAME	HouseNumbe
2	XA	20.35	2	M		ANDERSON	617
2	CP	4.20	14	M	H	ANDERSON	617
10	XA	16.62	10	M		MADISON	160
10	DA	7.28	20	M		MADESON	
19	XA	21.22	19	B		ANDERSON	321
19	CP	3.32	7	M		ANDERSOM	321

Figure 1-314 Create J10_Undup_MatchSpec_STAN_CUSTOMER job 15/17


Match Designer - Specification: CUSTOMER

Match Type: Unduplicate

This area holds Match Passes that are not executed as part of the Match job. To add a Pass, press the Ctrl key and drag the Pass from the 'Match Flow' area.

Match Pass: CUSTOMER_ZIP3

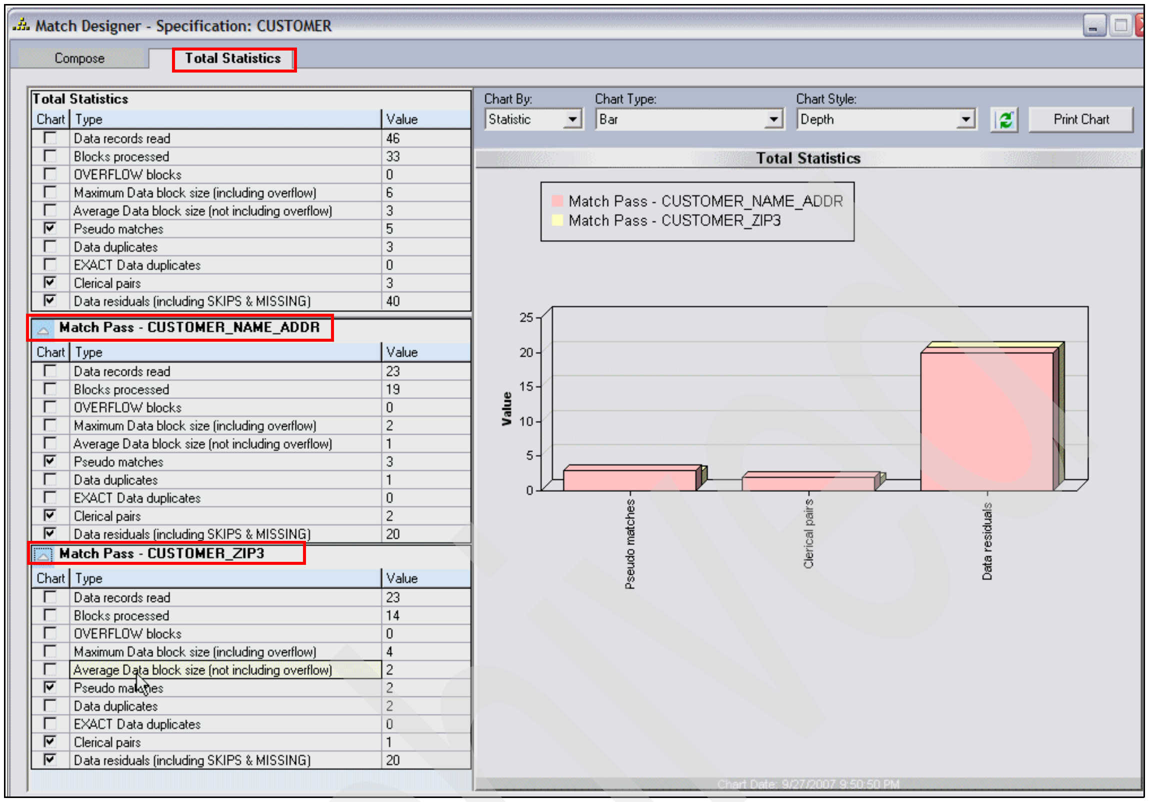
Blocking Columns:
- ZIP3
- AddressType_USADDR
- StreetNameNYSIIS_USADDR

Match Commands:
- GENDER
- MiddleName_USNAME
- PrimaryName_USNAME

Test Results:

SetID	Record Type	Weight	DataID	GENDER	MiddleName_USNAME	PrimaryName_USNAME	HouseNumbe
3	XA	22.01	3	F		KUMAR	342
3	CP	3.93	8	F		DEEPTI KUMAR JHA	342
10	XA	19.99	10	M		MADISON	160
10	DA	7.28	20	M		MADESON	
10	DA	12.37	13	M		MADI	160

Figure 1-315 Create J10_Undup_MatchSpec_STAN_CUSTOMER job 16/17


Match Designer - Specification: CUSTOMER

Compose | **Total Statistics**

Total Statistics

Chart	Type	Value
	Data records read	46
	Blocks processed	33
	OVERFLOW blocks	0
	Maximum Data block size (including overflow)	6
	Average Data block size (not including overflow)	3
✓	Pseudo matches	5
	Data duplicates	3
	EXACT Data duplicates	0
✓	Clerical pairs	3
✓	Data residuals (including SKIPS & MISSING)	40

Match Pass - CUSTOMER_NAME_ADDR

Chart	Type	Value
	Data records read	23
	Blocks processed	19
	OVERFLOW blocks	0
	Maximum Data block size (including overflow)	2
	Average Data block size (not including overflow)	1
✓	Pseudo matches	3
	Data duplicates	1
	EXACT Data duplicates	0
✓	Clerical pairs	2
✓	Data residuals (including SKIPS & MISSING)	20

Match Pass - CUSTOMER_ZIP3

Chart	Type	Value
	Data records read	23
	Blocks processed	14
	OVERFLOW blocks	0
	Maximum Data block size (including overflow)	4
	Average Data block size (not including overflow)	2
✓	Pseudo matches	2
	Data duplicates	2
	EXACT Data duplicates	0
✓	Clerical pairs	1
✓	Data residuals (including SKIPS & MISSING)	20

Chart By: Statistic
Chart Type: Bar
Chart Style: Depth

Print Chart
Total Statistics
Match Pass - CUSTOMER_NAME_ADDR
Match Pass - CUSTOMER_ZIP3
Value
25
20
15
10
5
0
Pseudo matches
Clerical pairs
Data residuals
Chart Date: 9/27/2007 9:50:50 PM

Figure 1-316 Create J10_Undup_MatchSpec_STAN_CUSTOMER job 17/17

J11_UNDUP_DEP_MATCH_CUSTOMER

In this step, we determine whether there are any duplicates in the credit card customer file by running the Unduplicate stage with the match specification and match frequency information created in job "J10_Undup_MatchSpec_STAN_CUSTOMER" on page 269 and "J10_MATCHFREQ_STAN_CUSTOMER" on page 266 respectively. The output is matched (merge of master and duplicates using a Funnel stage) records, records for clerical review, and residuals (records that do not match).

Figure 1-317 on page 285 through Figure 1-331 on page 290 describe the main steps using Designer Client to build and execute the Unduplicate Match Specification job.

The main steps are as follows:

1. Figure 1-317 on page 285 shows the various stages that are used in this job, including the data set that was created in "J08_STAN_CUSTOMER_Domain_Specific" on page 241, the match frequency Data Set created in "J10_MATCHFREQ_STAN_CUSTOMER" on page 266, an Unduplicate stage, and a Funnel stage to merge master and duplicate records. Three data sets are created:
   - One data set contains the merged master and duplicates by the Funnel stage
   - The other two data sets contain the clerical and residual records as the output of the Unduplicate stage.

   We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object created earlier.
2. Configure the input STAN_CUSTOMER data set that was created in "J08_STAN_CUSTOMER_Domain_Specific" on page 241, and the input STAN_CUSTOMER_FREQ data set that was created in "J10_MATCHFREQ_STAN_CUSTOMER" on page 266. We do not repeat this process here, because it is similar to earlier configurations.
3. Next, configure the UNDUP_DUP Unduplicate stage as follows:
   a. Double-click the **UNDUP_DUP** stage icon and click the ... Match Specification button. We do not show this process here.
   b. From the Repository window, double-click the Match Specifications folder. Then, select and right-click **CUSTOMER** and as shown in Figure 1-318 on page 285. Select **Provision All** from the menu and click **OK**. You are attaching the Unduplicate Match specification that was created in "J10_Undup_MatchSpec_STAN_CUSTOMER" on page 269.
   c. Check all the Match Outputs fields in the UNDUP_DEP - Unduplicate Match Stage shown in Figure 1-319 on page 286:
      - Match sends matched (master) records as output data.
      - Clerical separates those records that require clerical review.
      - Duplicates include duplicate records that are above the match cutoff.
      - Residuals separate records that are not duplicates as residuals.

      Keep the default Dependent Match Type as highlighted—this directs it to remove duplicates after every pass.

      Click **Stage Properties** to configure link ordering on both the input and output links by clicking the Stage tab. (We do not show this process here.)
   d. In the UNDUP_DEP - Unduplicate Match window, under the Output tab for the MASTERS link (Output name field), click **Mapping** as shown in Figure 1-320 on page 286. Copy all the columns from Columns pane to the MASTERS pane.

Repeat the process for the CLERICAL, DUPLICATE, and RESIDUAL links (Output name field). We do not repeat this process here.

4. Configure the FUNNEL stage by double-clicking the FUNNEL stage. Then, in the FUNNEL- Funnel window under the Output tab for the INMATCH link (Output name field), click **Mapping** as shown in Figure 1-321 on page 287. Copy all the columns from Columns pane to the INMATCH pane.
5. Next configure the three data set objects:
   - CUSTOMER_UNDUP_DEP_MATCHES
   - CUSTOMER_UNDUP_DEP_CLERICAL
   - CUSTOMER_UNDUP_DEP_RESIDUALS

   We do not repeat this process here.
6. After saving, compiling, and running this job j11_UNDUP_DEP_MATCH_CUSTOMER (Figure 1-322 on page 287), view the content of the three data set objects as shown in Figure 1-323 on page 288 through Figure 1-331 on page 290.

   Out of a total of 23 records in the input to this process:
   - Six (6) records are masters and duplicates
   - Three (3) records are for clerical review
   - Remaining fourteen (14) records are residuals.
7. We do not repeat the process of saving of the data set metadata to Table Definitions here.

The next step is to create a single report that contains each clerical review record followed by the master record it relates to, so that the clerical reviewer is able to view record details next to each other, as described in "J12_CLERICAL_REVIEW_CUSTOMER" on page 290.

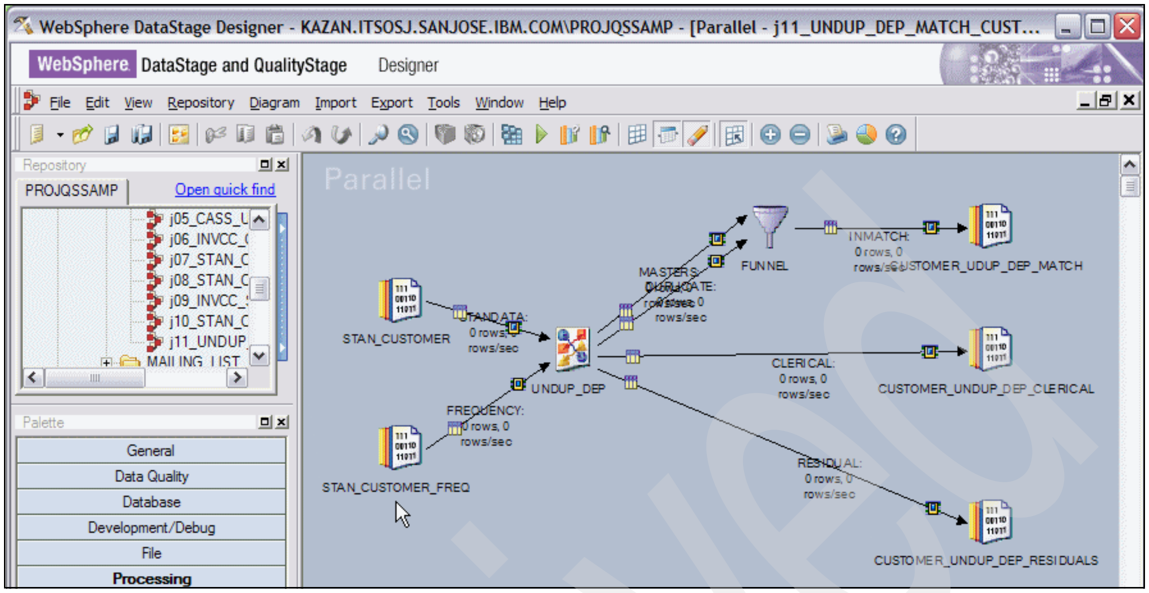

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j11_UNDUP_DEP_MATCH_CUST...
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Repository
PROJQSSAMP
Open quick find
j05_CASS_L
j06_INVCC_
j07_STAN_C
j08_STAN_C
j09_INVCC_
j10_STAN_C
j11_UNDUP
MAILING_LIST
Palette
General
Data Quality
Database
Development/Debug
File
Processing
Parallel
STAN_CUSTOMER
STANDATA: 0 rows, 0 rows/sec
UNDUP_DEP
FREQUENCY: 0 rows, 0 rows/sec
STAN_CUSTOMER_FREQ
MASTERS: 0 rows, 0 rows/sec
DUPLICATE: 0 rows, 0 rows/sec
FUNNEL
INMATCH: 0 rows, 0 rows/sec
CUSTOMER_UNDUP_DEP_MATCH
CLERICAL: 0 rows, 0 rows/sec
CUSTOMER_UNDUP_DEP_CLERICAL
RESIDUAL: 0 rows, 0 rows/sec
CUSTOMER_UNDUP_DEP_RESIDUALS

Figure 1-317 Create J11_UNDUP_DEP_MATCH_CUSTOMER job 1/15

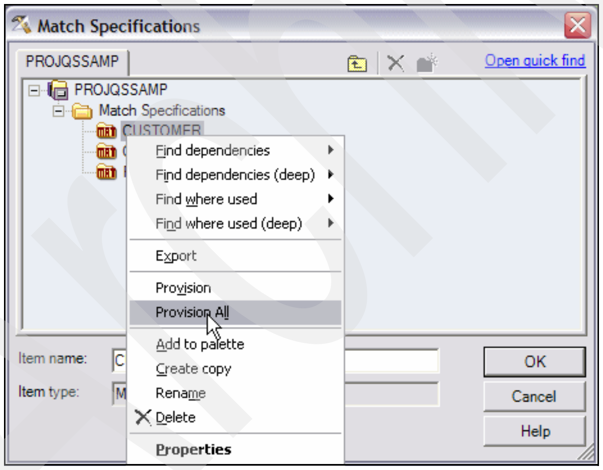
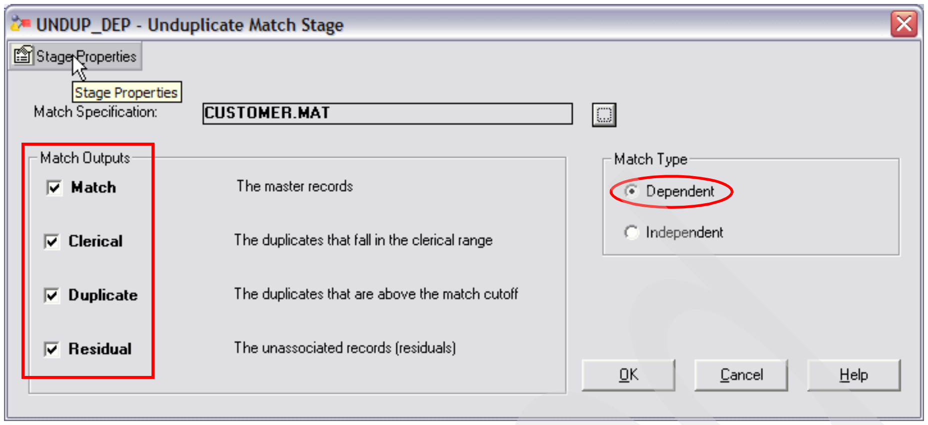


Figure 1-318 Create J11_UNDUP_DEP_MATCH_CUSTOMER job 2/15

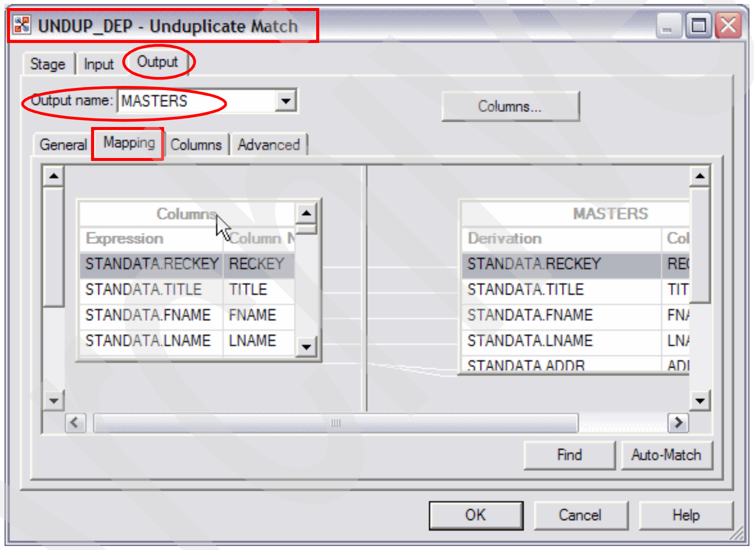


Match Specification: CUSTOMER.MAT

Match Outputs

✓ Match	The master records
✓ Clerical	The duplicates that fall in the clerical range
✓ Duplicate	The duplicates that are above the match cutoff
✓ Residual	The unassociated records (residuals)

Figure 1-319   Create J11_UNDUP_DEP_MATCH_CUSTOMER job 3/15



Output name: MASTERS

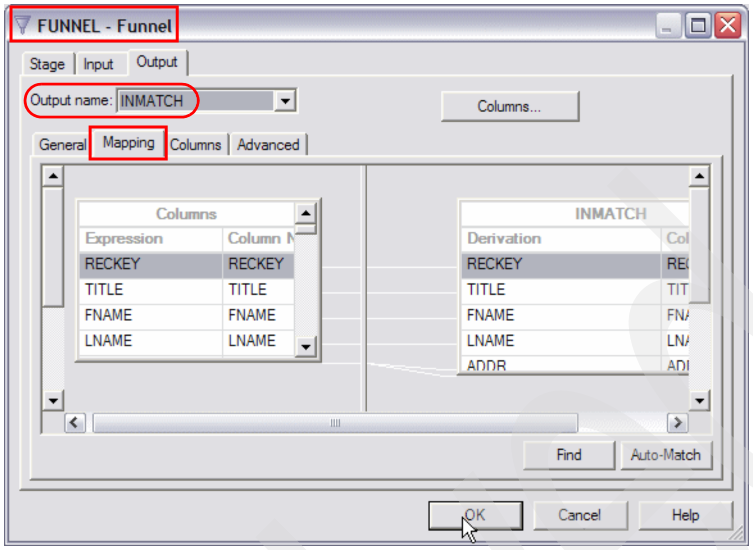
Columns

Expression	Column N
STANDATA.RECKEY	RECKEY
STANDATA.TITLE	TITLE
STANDATA.FNAME	FNAME
STANDATA.LNAME	LNAME

MASTERS

Derivation	Col
STANDATA.RECKEY	REC
STANDATA.TITLE	TIT
STANDATA.FNAME	FNA
STANDATA.LNAME	LNA
STANDATA.ADDR	ADI

Figure 1-320   Create J11_UNDUP_DEP_MATCH_CUSTOMER job 4/15



Columns	
Expression	Column N
RECKEY	RECKEY
TITLE	TITLE
FNAME	FNAME
LNAME	LNAME

INMATCH	
Derivation	Col
RECKEY	REC
TITLE	TIT
FNAME	FNA
LNAME	LNA
ADDR	ADI

Figure 1-321   Create J11_UNDUP_DEP_MATCH_CUSTOMER job 5/15

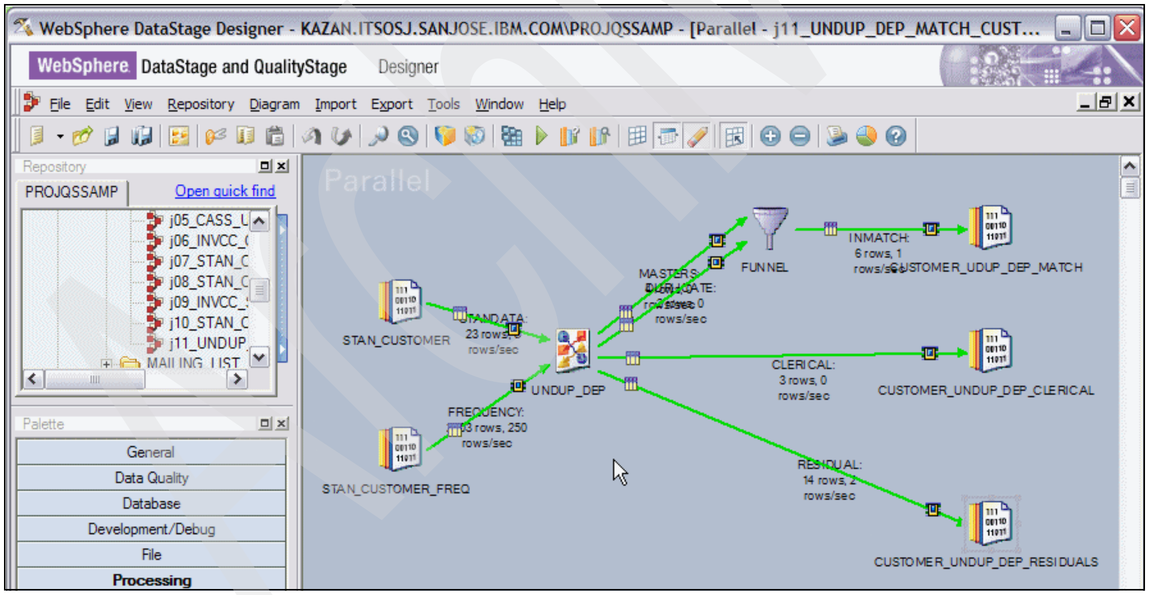
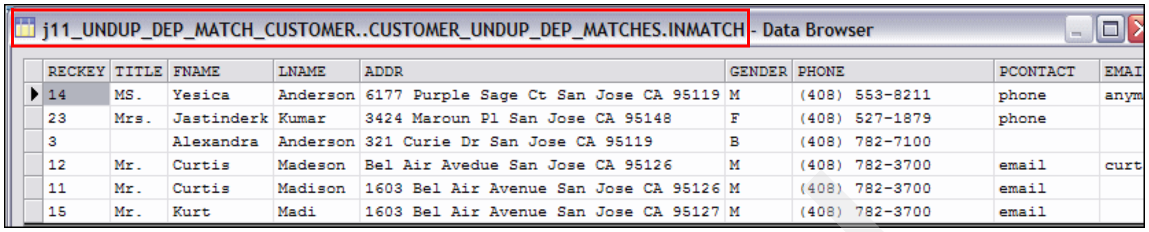

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j11_UNDUP_DEP_MATCH_CUST...
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Repository
PROJQSSAMP
Open quick find
j05_CASS_L
j06_INVCC_(
j07_STAN_C
j08_STAN_C
j09_INVCC_
j10_STAN_C
j11_UNDUP
MAILING LIST
Palette
General
Data Quality
Database
Development/Debug
File
Processing
Parallel
STAN_CUSTOMER
STANDATA: 23 rows, 0 rows/sec
UNDUP_DEP
FREQUENCY: 53 rows, 250 rows/sec
STAN_CUSTOMER_FREQ
MASTERS:
DUPLICATE: rows/sec 0 rows/sec
FUNNEL
INMATCH: 6 rows, 1 rows/sec
CUSTOMER_UNDUP_DEP_MATCH
CLERICAL: 3 rows, 0 rows/sec
CUSTOMER_UNDUP_DEP_CLERICAL
RESIDUAL: 14 rows, 2 rows/sec
CUSTOMER_UNDUP_DEP_RESIDUALS

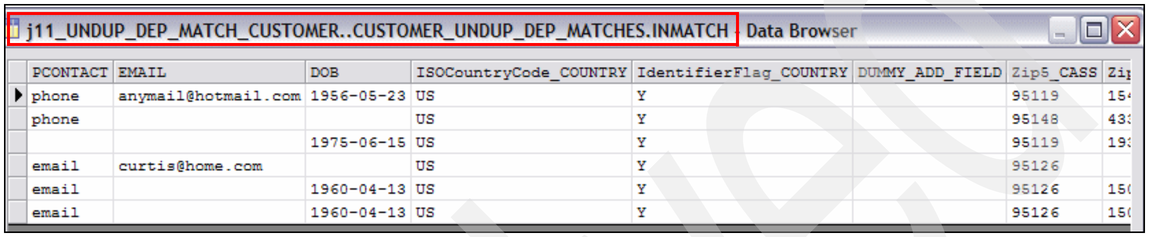
Figure 1-322   Create J11_UNDUP_DEP_MATCH_CUSTOMER job 6/15



j11_UNDUP_DEP_MATCH_CUSTOMER..CUSTOMER_UNDUP_DEP_MATCHES.INMATCH - Data Browser

RECKEY	TITLE	FNAME	LNAME	ADDR	GENDER	PHONE	PCONTACT	EMAI
14	MS.	Yesica	Anderson	6177 Purple Sage Ct San Jose CA 95119	M	(408) 553-8211	phone	anym
23	Mrs.	Jastinderk	Kumar	3424 Maroun Pl San Jose CA 95148	F	(408) 527-1879	phone	
3		Alexandra	Anderson	321 Curie Dr San Jose CA 95119	B	(408) 782-7100		
12	Mr.	Curtis	Madeson	Bel Air Avedue San Jose CA 95126	M	(408) 782-3700	email	curt
11	Mr.	Curtis	Madison	1603 Bel Air Avenue San Jose CA 95126	M	(408) 782-3700	email	
15	Mr.	Kurt	Madi	1603 Bel Air Avenue San Jose CA 95127	M	(408) 782-3700	email	

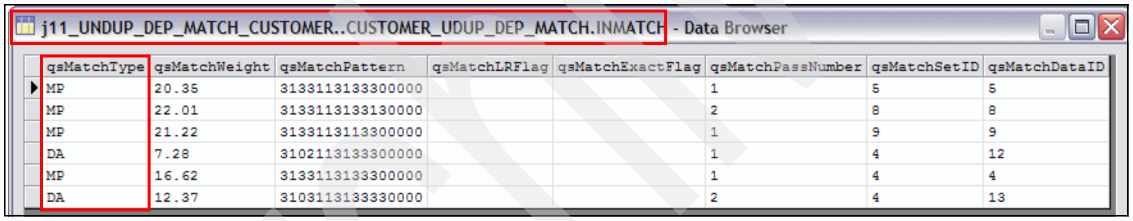
Figure 1-323 Create J11_UNDUP_DEP_MATCH_CUSTOMER job 7/15



j11_UNDUP_DEP_MATCH_CUSTOMER..CUSTOMER_UNDUP_DEP_MATCHES.INMATCH - Data Browser

PCONTACT	EMAIL	DOB	ISOCountryCode_COUNTRY	IdentifierFlag_COUNTRY	DUMMY_ADD_FIELD	Zip5_CASS	Zi
phone	anymail@hotmail.com	1956-05-23	US	Y		95119	15
phone			US	Y		95148	43
		1975-06-15	US	Y		95119	19
email	curtis@home.com		US	Y		95126	
email		1960-04-13	US	Y		95126	15
email		1960-04-13	US	Y		95126	15

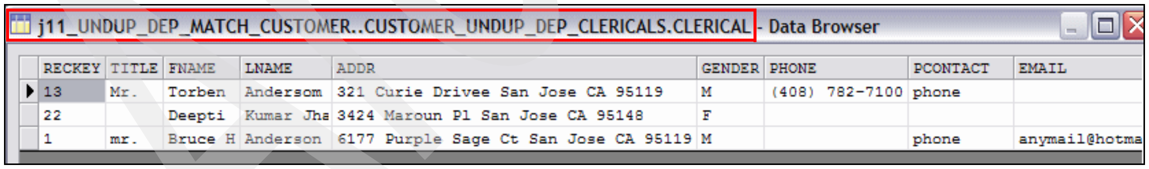
Figure 1-324 Create J11_UNDUP_DEP_MATCH_CUSTOMER job 8/15



j11_UNDUP_DEP_MATCH_CUSTOMER..CUSTOMER_UDUP_DEP_MATCH.INMATCH - Data Browser

qsMatchType	qsMatchWeight	qsMatchPattern	qsMatchLRFlag	qsMatchExactFlag	qsMatchPassNumber	qsMatchSetID	qsMatchDataID
MP	20.35	3133113133300000			1	5	5
MP	22.01	3133113133130000			2	8	8
MP	21.22	3133113113300000			1	9	9
DA	7.28	3102113133300000			1	4	12
MP	16.62	3133113133300000			1	4	4
DA	12.37	3103113133330000			2	4	13

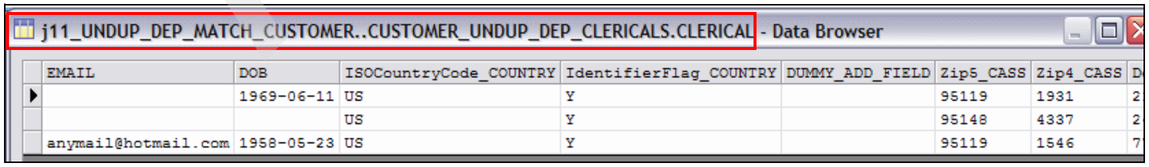
Figure 1-325 Create J11_UNDUP_DEP_MATCH_CUSTOMER job 9/15



j11_UNDUP_DEP_MATCH_CUSTOMER..CUSTOMER_UNDUP_DEP_CLERICALS.CLERICAL - Data Browser

RECKEY	TITLE	FNAME	LNAME	ADDR	GENDER	PHONE	PCONTACT	EMAIL
13	Mr.	Torben	Anderson	321 Curie Drivee San Jose CA 95119	M	(408) 782-7100	phone	
22		Deepti	Kumar Jha	3424 Maroun Pl San Jose CA 95148	F			
1	mr.	Bruce H	Anderson	6177 Purple Sage Ct San Jose CA 95119	M		phone	anymail@hotma

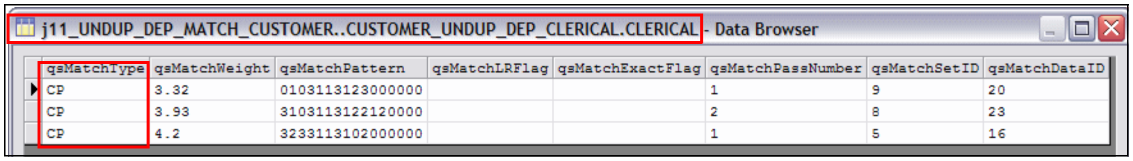
Figure 1-326 Create J11_UNDUP_DEP_MATCH_CUSTOMER job 10/15



j11_UNDUP_DEP_MATCH_CUSTOMER..CUSTOMER_UNDUP_DEP_CLERICALS.CLERICAL - Data Browser

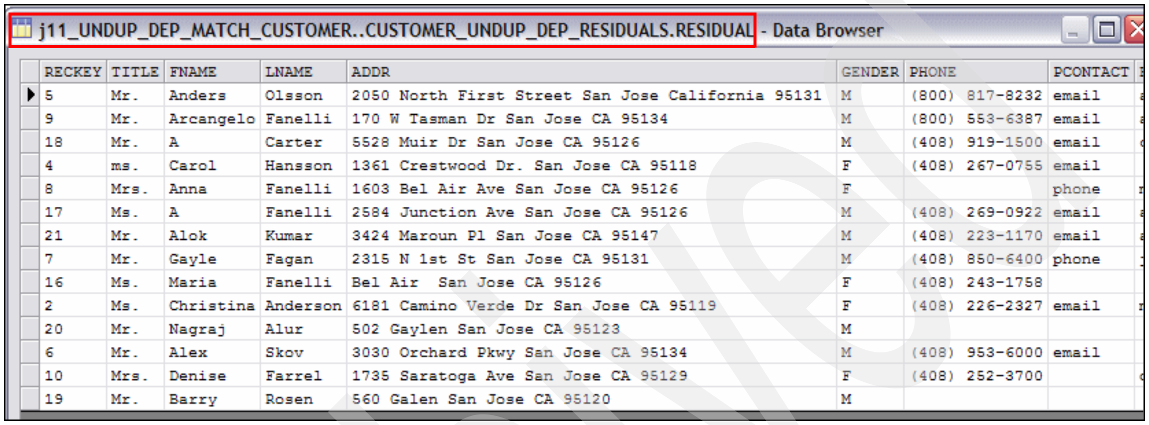
EMAIL	DOB	ISOCountryCode_COUNTRY	IdentifierFlag_COUNTRY	DUMMY_ADD_FIELD	Zip5_CASS	Zip4_CASS	D
	1969-06-11	US	Y		95119	1931	2
		US	Y		95148	4337	2
anymail@hotmail.com	1958-05-23	US	Y		95119	1546	7

Figure 1-327 Create J11_UNDUP_DEP_MATCH_CUSTOMER job 11/15


j11_UNDUP_DEP_MATCH_CUSTOMER..CUSTOMER_UNDUP_DEP_CLERICAL.CLERICAL - Data Browser

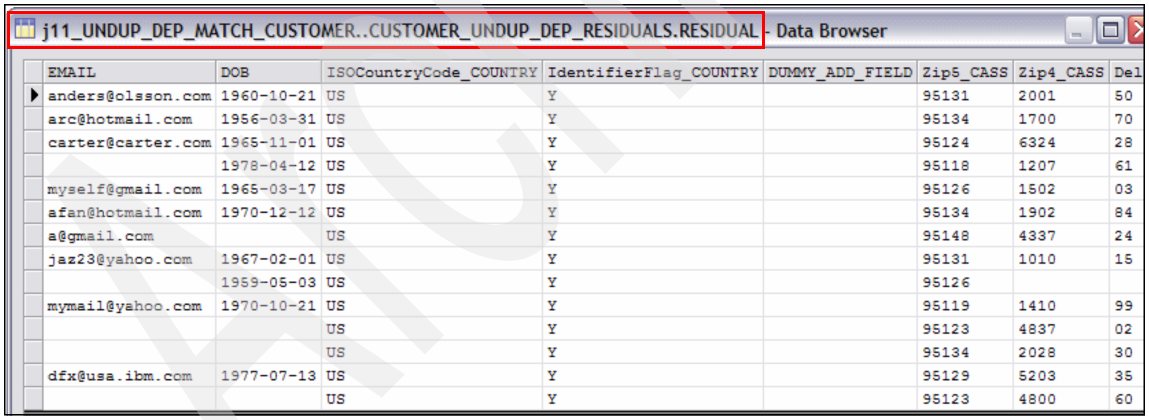
qsMatchType	qsMatchWeight	qsMatchPattern	qsMatchLRFlag	qsMatchExactFlag	qsMatchPassNumber	qsMatchSetID	qsMatchDataID
CP	3.32	0103113123000000			1	9	20
CP	3.93	3103113122120000			2	8	23
CP	4.2	3233113102000000			1	5	16

Figure 1-328   Create J11_UNDUP_DEP_MATCH_CUSTOMER job 12/15


j11_UNDUP_DEP_MATCH_CUSTOMER..CUSTOMER_UNDUP_DEP_RESIDUALS.RESIDUAL - Data Browser

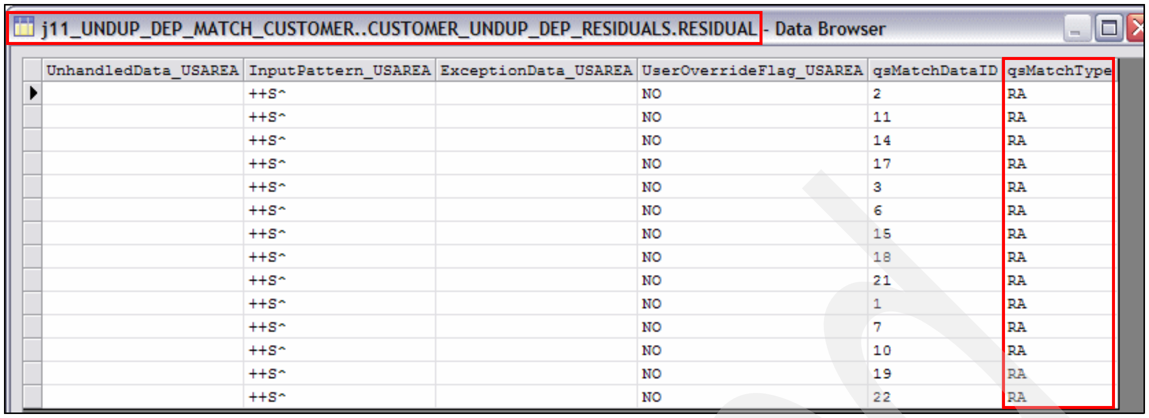
RECKEY	TITLE	FNAME	LNAME	ADDR	GENDER	PHONE	PCONTACT
5	Mr.	Anders	Olsson	2050 North First Street San Jose California 95131	M	(800) 817-8232	email
9	Mr.	Arcangelo	Fanelli	170 W Tasman Dr San Jose CA 95134	M	(800) 553-6387	email
18	Mr.	A	Carter	5528 Muir Dr San Jose CA 95126	M	(408) 919-1500	email
4	ms.	Carol	Hansson	1361 Crestwood Dr. San Jose CA 95118	F	(408) 267-0755	email
8	Mrs.	Anna	Fanelli	1603 Bel Air Ave San Jose CA 95126	F		phone
17	Ms.	A	Fanelli	2584 Junction Ave San Jose CA 95126	M	(408) 269-0922	email
21	Mr.	Alok	Kumar	3424 Maroun Pl San Jose CA 95147	M	(408) 223-1170	email
7	Mr.	Gayle	Fagan	2315 N 1st St San Jose CA 95131	M	(408) 850-6400	phone
16	Ms.	Maria	Fanelli	Bel Air San Jose CA 95126	F	(408) 243-1758	
2	Ms.	Christina	Anderson	6181 Camino Verde Dr San Jose CA 95119	F	(408) 226-2327	email
20	Mr.	Nagraj	Alur	502 Gaylen San Jose CA 95123	M		
6	Mr.	Alex	Skov	3030 Orchard Pkwy San Jose CA 95134	M	(408) 953-6000	email
10	Mrs.	Denise	Farrel	1735 Saratoga Ave San Jose CA 95129	F	(408) 252-3700	
19	Mr.	Barry	Rosen	560 Galen San Jose CA 95120	M		

Figure 1-329   Create J11_UNDUP_DEP_MATCH_CUSTOMER job 13/15


j11_UNDUP_DEP_MATCH_CUSTOMER..CUSTOMER_UNDUP_DEP_RESIDUALS.RESIDUAL - Data Browser

EMAIL	DOB	ISOCountryCode_COUNTRY	IdentifierFlag_COUNTRY	DUMMY_ADD_FIELD	Zip5_CASS	Zip4_CASS	Del
anders@olsson.com	1960-10-21	US	Y		95131	2001	50
arc@hotmail.com	1956-03-31	US	Y		95134	1700	70
carter@carter.com	1965-11-01	US	Y		95124	6324	28
	1978-04-12	US	Y		95118	1207	61
myself@gmail.com	1965-03-17	US	Y		95126	1502	03
afan@hotmail.com	1970-12-12	US	Y		95134	1902	84
a@gmail.com		US	Y		95148	4337	24
jaz23@yahoo.com	1967-02-01	US	Y		95131	1010	15
	1959-05-03	US	Y		95126		
mymail@yahoo.com	1970-10-21	US	Y		95119	1410	99
		US	Y		95123	4837	02
		US	Y		95134	2028	30
dfx@usa.ibm.com	1977-07-13	US	Y		95129	5203	35
		US	Y		95123	4800	60

Figure 1-330   Create J11_UNDUP_DEP_MATCH_CUSTOMER job 14/15


j11_UNDUP_DEP_MATCH_CUSTOMER..CUSTOMER_UNDUP_DEP_RESIDUALS.RESIDUAL - Data Browser

UnhandledData_USAREA	InputPattern_USAREA	ExceptionData_USAREA	UserOverrideFlag_USAREA	qsMatchDataID	qsMatchType
	++S^		NO	2	RA
	++S^		NO	11	RA
	++S^		NO	14	RA
	++S^		NO	17	RA
	++S^		NO	3	RA
	++S^		NO	6	RA
	++S^		NO	15	RA
	++S^		NO	18	RA
	++S^		NO	21	RA
	++S^		NO	1	RA
	++S^		NO	7	RA
	++S^		NO	10	RA
	++S^		NO	19	RA
	++S^		NO	22	RA

Figure 1-331 Create J11_UNDUP_DEP_MATCH_CUSTOMER job 15/15

J12_CLERICAL_REVIEW_CUSTOMER

In this step, we create a single report of the matched credit card customers' file and the records in the clerical review file that was created in "J11_UNDUP_DEP_MATCH_CUSTOMER" on page 282 to enable a manual review of potential match records.

We performed the following:

1. First, we merged the matched records data set (CUSTOMER_UNDUP_DEP_MATCHES) and clerical review records (CUSTOMER_CLERICAL_REVIEW) using a Funnel stage.
2. Then, we removed duplicates from the matched records data set (because we had merged them using a Funnel stage in "J11_UNDUP_DEP_MATCH_CUSTOMER" on page 282).
3. Finally, we joined the output of the previous two steps on the match set ID (qsMatchSetID column) to produce the desired report.

Figure 1-332 on page 292 through Figure 1-339 on page 295 describe the main steps using Designer Client to perform this task.

The main steps are as follows:

1. Figure 1-332 on page 292 shows the various stages that are used in this job, including three data sets that were created in "J11_UNDUP_DEP_MATCH_CUSTOMER" on page 282, a Funnel stage, a Remove Duplicates stage, and a JOIN stage. Create one data set that contains the matched and clerical records organized for convenient manual review. We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object created earlier.

2. Configure the three input data set objects that were created in "J11_UNDUP_DEP_MATCH_CUSTOMER" on page 282:
   - CUSTOMER_UNDUP_DEP_MATCHES
   - CUSTOMER_UNDUP_DEP_CLERICALS
   - CUSTOMER_UNDUP_DEP_CLERICALS_MASTER

   We do not repeat this process here because this is a similar process to earlier configurations.
3. Configure the Funnel stage to merge the two inputs from CUSTOMER_UNDUP_DEP_MATCHES and CUSTOMER_UNDUP_DEP_CLERICALS data sets. We do not repeat this process here because this is a similar process to earlier configurations.
4. Configure the Remove Duplicates stage as follows:
   a. Double-click the **Remove_Duplicates_11** stage icon to open the Remove_Duplicates_11 - Remove Duplicates window. Under the **Stage** tab provide the Keys That Define Duplicates—qsMatchSetID in this case as shown in Figure 1-333 on page 293.
   b. Click the Output tab, and for the JOINOUT link click the Mapping tab. Copy the column qsMatchSetID from the Columns pane to the JOINOUT pane as shown in Figure 1-334 on page 293. Click **OK**.
5. Configure the JOIN stage, which performs join operations on two or more data sets input to the stage and then outputs the resulting data set. The data sets input to the Join stage must be key partitioned and sorted to ensure that rows with the same key column values are located in the same partition and are processed by the same node. The main steps are as follows:
   a. Double-click the **JOIN** stage icon and in the JOIN - Join window that opens select the **Stage** tab. In the Properties tab, provide the Join Keys—qsMatchSetID in this case as shown in Figure 1-335 on page 294.
   b. Click the Output tab, and for the IN link click the Mapping tab. Copy all the columns from the Columns pane to the IN pane as shown in Figure 1-336 on page 294. Click **OK**.
6. Configure the output data set object CUSTOMER_CLERICAL_REVIEW to store the results of the join. We do not repeat this process here because this is a similar process to earlier configurations.
7. After saving this job j12_CLERICAL_REVIEW_CUSTOMER, compiling and running it, you view the content of the output data set object CUSTOMER_CLERICAL_REVIEW as shown in Figure 1-337 on page 295 through Figure 1-339 on page 295.

   This report shows three clerical review records (qsMatchType has value `CP`) immediately followed by one or more master records (qsMatchType has value

MP) that could match them. This method of organizing the records makes it more convenient for manual review.

8. Save the data set metadata to Table Definitions. We do not repeat this process here.

After clerical review is completed, you create a clean set of master and duplicate records and survive the best information into the master record using the SURVIVE stage, as described in the next step "J13_SURVIVE_CUSTOMER" on page 295.

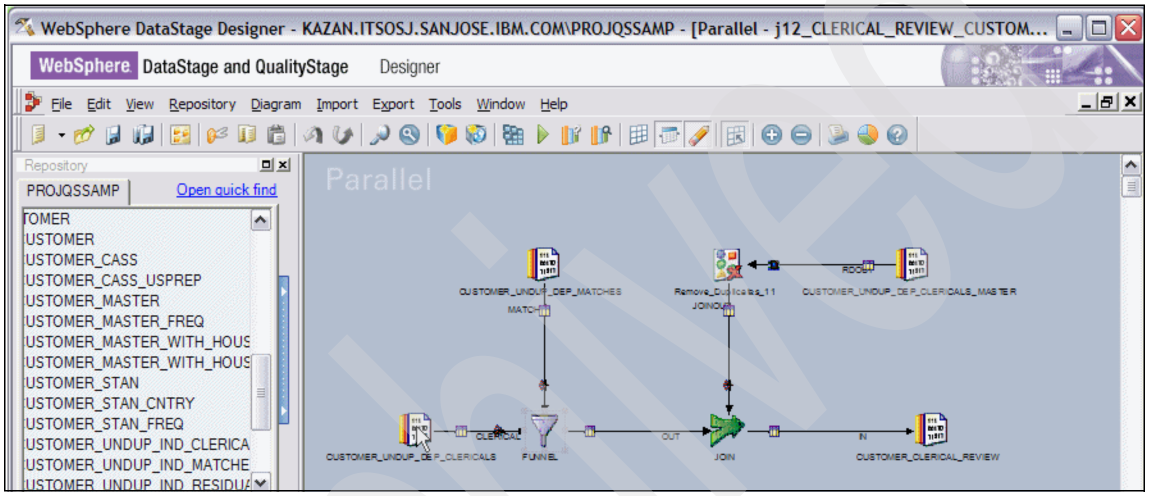
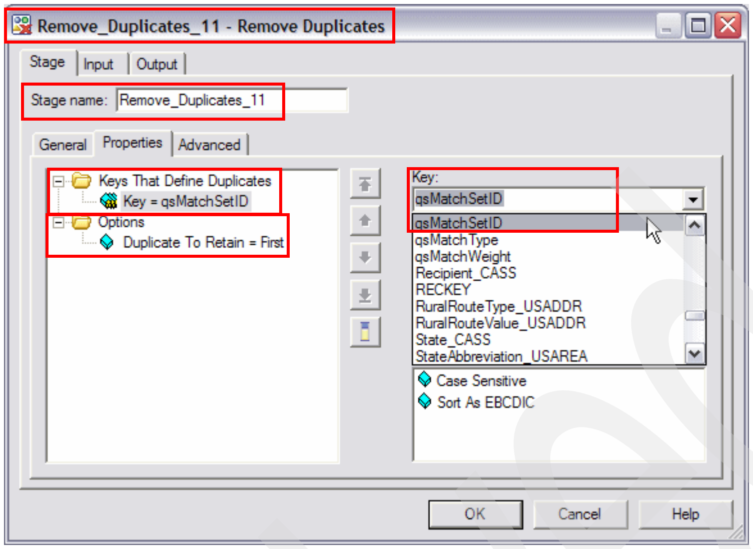

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j12_CLERICAL_REVIEW_CUSTOM...
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Repository
PROJQSSAMP
Open quick find
TOMER
USTOMER
USTOMER_CASS
USTOMER_CASS_USPREP
USTOMER_MASTER
USTOMER_MASTER_FREQ
USTOMER_MASTER_WITH_HOUS
USTOMER_MASTER_WITH_HOUS
USTOMER_STAN
USTOMER_STAN_CNTRY
USTOMER_STAN_FREQ
USTOMER_UNDUP_IND_CLERICA
USTOMER_UNDUP_IND_MATCHE
USTOMER_UNDUP_IND_RESIDUA
Parallel
CUSTOMER_UNDUP_DEP_MATCHES
MATCH
Remove_Duplicates_11
JOINOUT
RDOUT
CUSTOMER_UNDUP_DEP_CLERICALS_MASTER
CUSTOMER_UNDUP_DEP_CLERICALS
CLERICAL
FUNNEL
OUT
JOIN
IN
CUSTOMER_CLERICAL_REVIEW

Figure 1-332 Create J12_CLERICAL_REVIEW_CUSTOMER job 1/8



Remove_Duplicates_11 - Remove Duplicates

Stage name: Remove_Duplicates_11

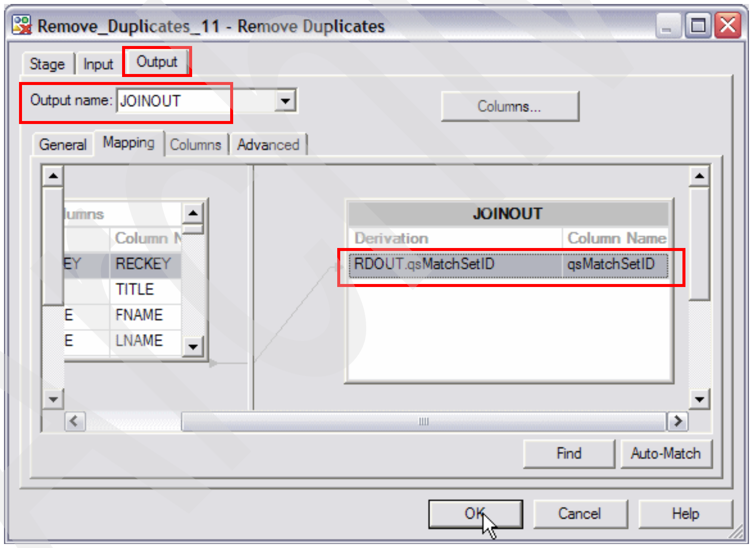
Keys That Define Duplicates
Key = qsMatchSetID
Options
Duplicate To Retain = First

Key: qsMatchSetID

qsMatchSetID
qsMatchType
qsMatchWeight
Recipient_CASS
RECKEY
RuralRouteType_USADDR
RuralRouteValue_USADDR
State_CASS
StateAbbreviation_USAREA

Case Sensitive
Sort As EBCDIC

Figure 1-333   Create J12_CLERICAL_REVIEW_CUSTOMER job 2/8



Remove_Duplicates_11 - Remove Duplicates

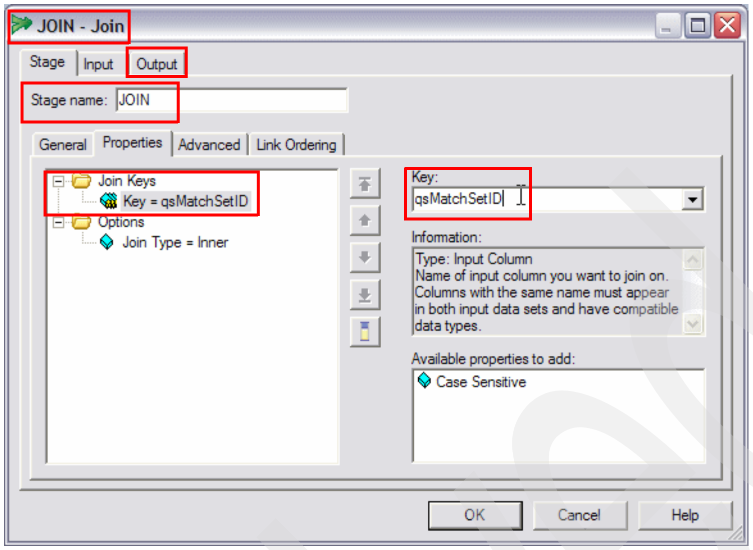
Output name: JOINOUT

lumns	Column N
EY	RECKEY
	TITLE
E	FNAME
E	LNAME

JOINOUT

Derivation	Column Name
RDOUT.qsMatchSetID	qsMatchSetID

Figure 1-334   Create J12_CLERICAL_REVIEW_CUSTOMER job 3/8



Stage name: JOIN

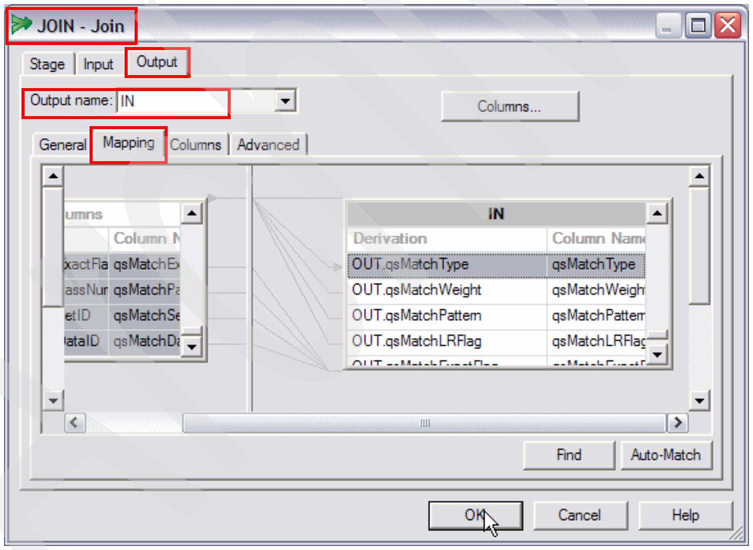
- Join Keys
  - Key = qsMatchSetID
- Options
  - Join Type = Inner

Key: qsMatchSetID

Information:
Type: Input Column
Name of input column you want to join on.
Columns with the same name must appear in both input data sets and have compatible data types.

Available properties to add:
Case Sensitive

Figure 1-335 Create J12_CLERICAL_REVIEW_CUSTOMER job 4/8

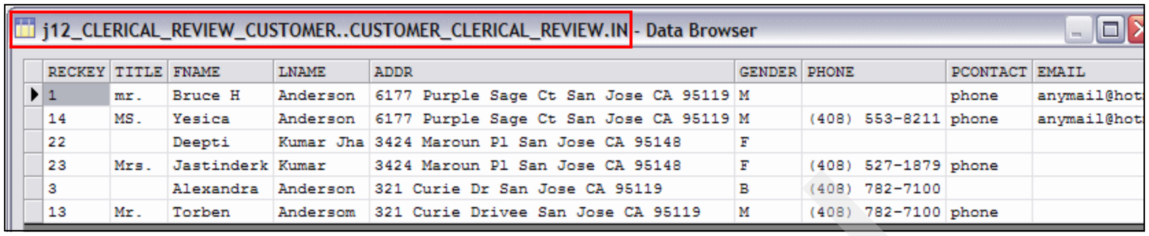


Output name: IN

umns	Column N
xactFla	qsMatchEx
assNur	qsMatchPa
etID	qsMatchSe
ataID	qsMatchDa

IN	
Derivation	Column Name
OUT.qsMatchType	qsMatchType
OUT.qsMatchWeight	qsMatchWeigh
OUT.qsMatchPattern	qsMatchPattern
OUT.qsMatchLRFlag	qsMatchLRFlag

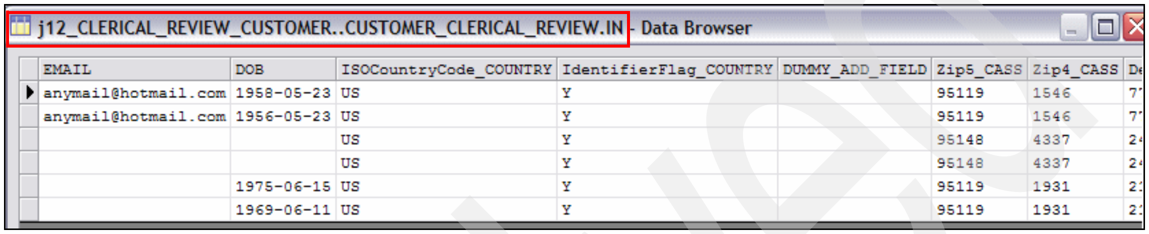
Figure 1-336 Create J12_CLERICAL_REVIEW_CUSTOMER job 5/8



j12_CLERICAL_REVIEW_CUSTOMER..CUSTOMER_CLERICAL_REVIEW.IN - Data Browser

RECKEY	TITLE	FNAME	LNAME	ADDR	GENDER	PHONE	PCONTACT	EMAIL
1	mr.	Bruce H	Anderson	6177 Purple Sage Ct San Jose CA 95119	M		phone	anymail@hot
14	MS.	Yesica	Anderson	6177 Purple Sage Ct San Jose CA 95119	M	(408) 553-8211	phone	anymail@hot
22		Deepti	Kumar Jha	3424 Maroun Pl San Jose CA 95148	F			
23	Mrs.	Jastinderk	Kumar	3424 Maroun Pl San Jose CA 95148	F	(408) 527-1879	phone	
3		Alexandra	Anderson	321 Curie Dr San Jose CA 95119	B	(408) 782-7100		
13	Mr.	Torben	Andersom	321 Curie Drivee San Jose CA 95119	M	(408) 782-7100	phone	

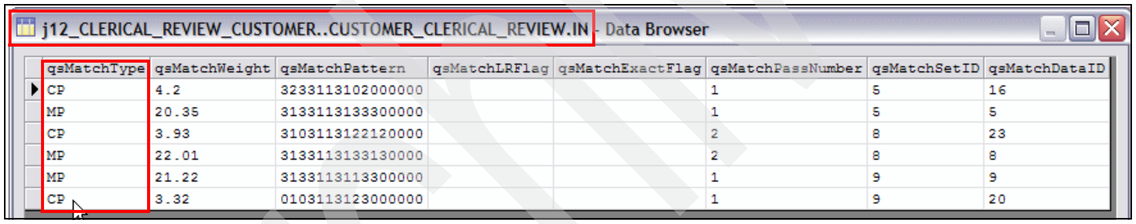
Figure 1-337 Create J11_UNDUP_DEP_MATCH_CUSTOMER job 6/8



j12_CLERICAL_REVIEW_CUSTOMER..CUSTOMER_CLERICAL_REVIEW.IN - Data Browser

EMAIL	DOB	ISOCountryCode_COUNTRY	IdentifierFlag_COUNTRY	DUMMY_ADD_FIELD	Zip5_CASS	Zip4_CASS	De
anymail@hotmail.com	1958-05-23	US	Y		95119	1546	7
anymail@hotmail.com	1956-05-23	US	Y		95119	1546	7
		US	Y		95148	4337	2
		US	Y		95148	4337	2
	1975-06-15	US	Y		95119	1931	2
	1969-06-11	US	Y		95119	1931	2

Figure 1-338 Create J11_UNDUP_DEP_MATCH_CUSTOMER job 7/8



j12_CLERICAL_REVIEW_CUSTOMER..CUSTOMER_CLERICAL_REVIEW.IN - Data Browser

qsMatchType	qsMatchWeight	qsMatchPattern	qsMatchLRFlag	qsMatchExactFlag	qsMatchPassNumber	qsMatchSetID	qsMatchDataID
CP	4.2	3233113102000000			1	5	16
MP	20.35	3133113133300000			1	5	5
CP	3.93	3103113122120000			2	8	23
MP	22.01	3133113133130000			2	8	8
MP	21.22	3133113113300000			1	9	9
CP	3.32	0103113123000000			1	9	20

Figure 1-339 Create J12_CLERICAL_REVIEW_CUSTOMER job 8/8

J13_SURVIVE_CUSTOMER

In this step, we survive the best information from the set of duplicates. The input to this step is the data set that contains matched and duplicated records that was created by the Funnel stage in "J11_UNDUP_DEP_MATCH_CUSTOMER" on page 282 or one that was modified after the clerical review step in "J12_CLERICAL_REVIEW_CUSTOMER" on page 290.

Figure 1-340 on page 297 through Figure 1-353 on page 303 describe the main steps using Designer Client to perform this task.

The main steps are as follows:

1. Figure 1-340 on page 297 shows the various stages that are used in this job, including the input data set CUSTOMER_UNDUP_DEP_MATCHES that was created in "J11_UNDUP_DEP_MATCH_CUSTOMER" on page 282, which

contains matched and duplicate records, a Survive stage, and an output data set containing the survived records. We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object that was created earlier.

2. Configure the input data set CUSTOMER_UNDUP_DEP_MATCHES. We do not repeat this process here because it is similar to earlier configurations.

3. Configure the SURVIVE stage. With the Survive stage, you test column values to ascertain which columns are the best candidates for that record. These columns are combined to become the output record for the group. In selecting a best candidate, you can specify to test the following column values:

   – Record creation data
   – Data source
   – Length of data in a column
   – Frequency of data in a group

   You configure the SURVIVE stage with rules to compare the columns against a best case.

   To configure the SURVIVE stage:

   a. Double-click the SURVIVE stage, and in the Copy_of_SURVIVE - Survive Stage window, click **New Rule** in Figure 1-341 on page 298 to open the Survive Rules Definition window.

   b. The SURVIVE stage requires a rule that contains one or more targets and a TRUE condition expression. Select **AllColumns** from Available Columns and click the > button to move AllColumns to the Target(s) column as shown in Figure 1-342 on page 298.

   c. From the Survive Rule (Pick one) area, click **Analyze Column** and select **qsMatchType** (as the target to which to compare other columns) from the drop-down menu. From the Technique field drop-down menu, click **Equals**. The rules syntax for the Equals technique is `c."column" = "DATA"`. In the Data field, type `MP` and click **OK** as shown in Figure 1-342 on page 298.

   d. Repeat the process for the MatchFirstName_USNAME with the Most Frequent (non-blank) technique, MiddleName_USNAME with the Longest technique, and PrimaryName_USNAME with the Longest technique as shown in Figure 1-343 on page 298 through Figure 1-345 on page 299.

   e. Create a complex survive expression by selecting it in the Survive Rules Definition window to open the Rule Expression Builder pane where you select columns, functions, and operations to build the expression as shown in Figure 1-346 on page 300. You can check the expression by clicking **Check Expression**. Click **OK**.

f. View the rules that you added in the Survive Stage grid shown in Figure 1-347 on page 301. From the Select the group identification data column, choose the Selected Column qsMatchSetID from the list, and click **OK**.

g. Click **Stage Properties** in Figure 1-348 on page 301, and when the SURVIVE - Survive window opens, click the **Output** tab for the link IN (Output name field) and then the Mapping tab. Select the columns from the Columns pane and copy them to the Survived pane as shown in Figure 1-349 on page 302. Click **OK**.

4. Configure the output data set object CUSTOMER_SURVIVE_MP_DA to store the results of the Survive stage. We do not repeat this process here because it is similar to earlier configurations.

5. After saving this job j13_SURVIVE, compiling and running it (Figure 1-350 on page 302), view the content of the output data set object CUSTOMER_SURVIVE_MP_DA as shown in Figure 1-351 on page 302 through Figure 1-353 on page 303.

   This report shows the master records with the survived information from the duplicates.

Now, proceed to “J14_CUSTOMER_MASTER” on page 303.

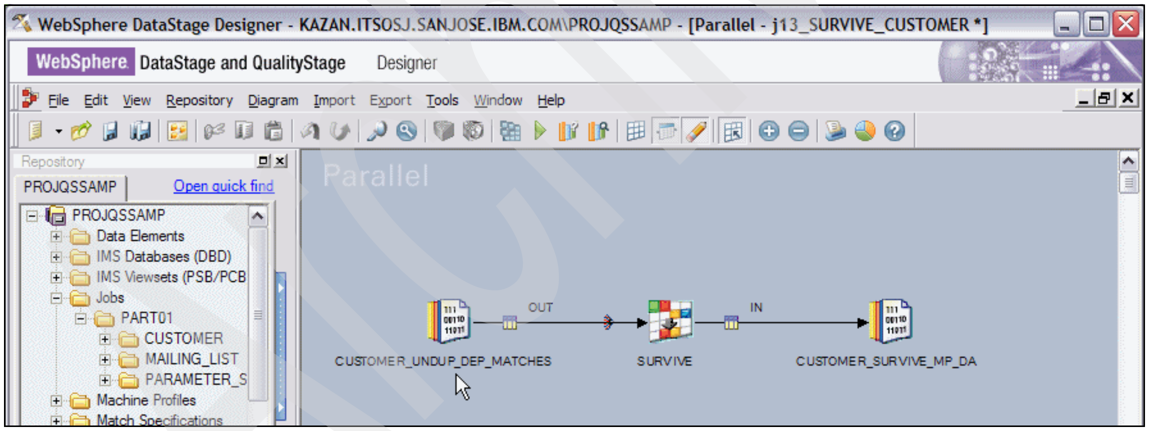
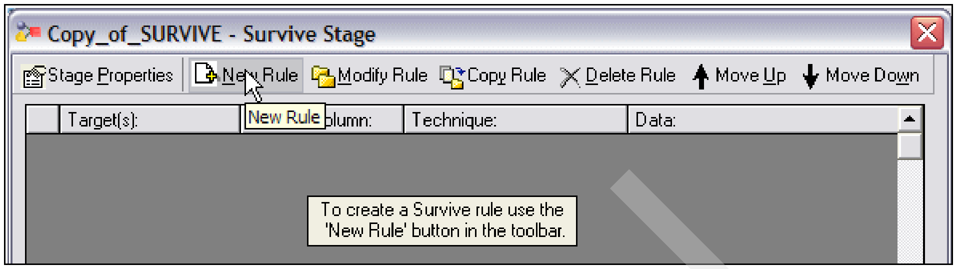
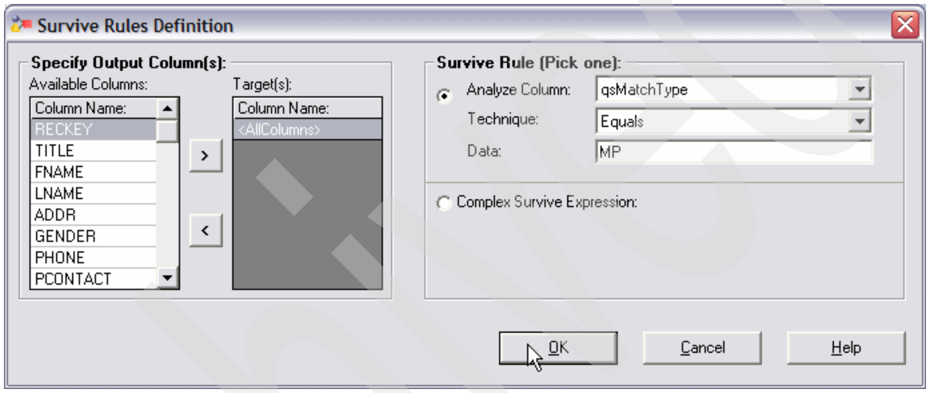

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j13_SURVIVE_CUSTOMER *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Repository
PROJQSSAMP
Open quick find
PROJQSSAMP
Data Elements
IMS Databases (DBD)
IMS Viewsets (PSB/PCB
Jobs
PART01
CUSTOMER
MAILING_LIST
PARAMETER_S
Machine Profiles
Match Specifications
Parallel
OUT
IN
CUSTOMER_UNDUP_DEP_MATCHES
SURVIVE
CUSTOMER_SURVIVE_MP_DA

Figure 1-340 Create J13_SURVIVE_CUSTOMER job 1/14



To create a Survive rule use the 'New Rule' button in the toolbar.

Figure 1-341   Create J13_SURVIVE_CUSTOMER job 2/14

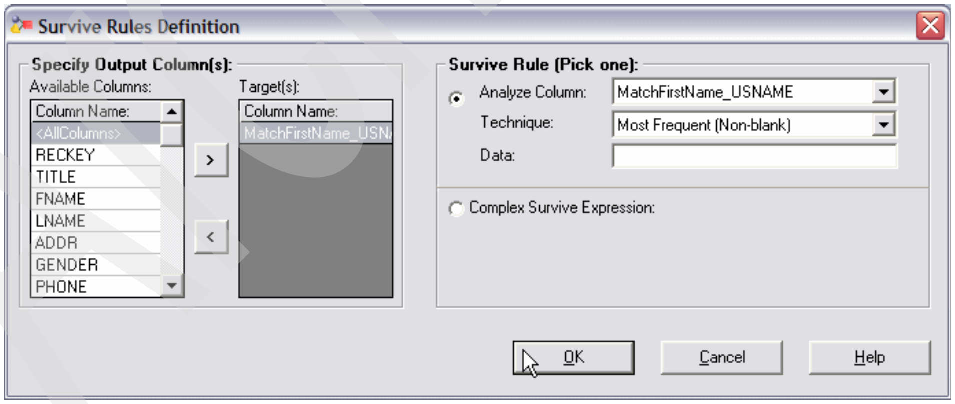


Analyze Column: qsMatchType

Technique: Equals

Data: MP

Figure 1-342   Create J13_SURVIVE_CUSTOMER job 3/14

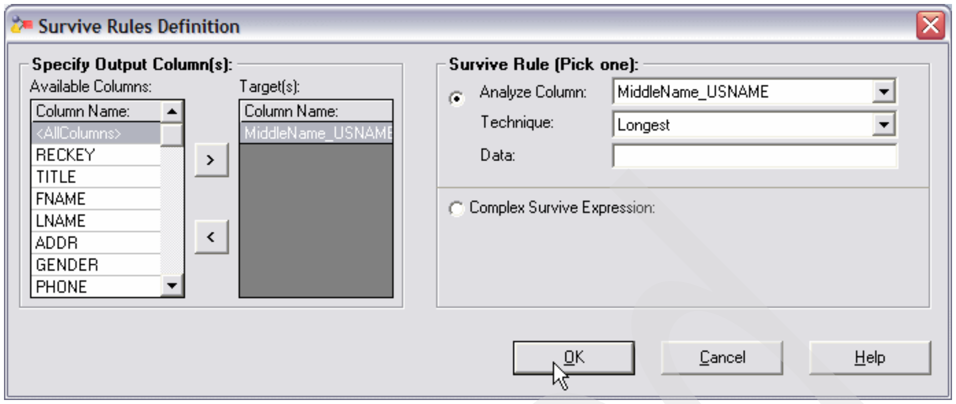


Analyze Column: MatchFirstName_USNAME

Technique: Most Frequent (Non-blank)

Data:

Figure 1-343   Create J13_SURVIVE_CUSTOMER job 4/14


Survive Rules Definition

Specify Output Column(s):

Available Columns:

Column Name:
<AllColumns>
RECKEY
TITLE
FNAME
LNAME
ADDR
GENDER
PHONE

Target(s):

Column Name:
MiddleName_USNAME

Survive Rule (Pick one):

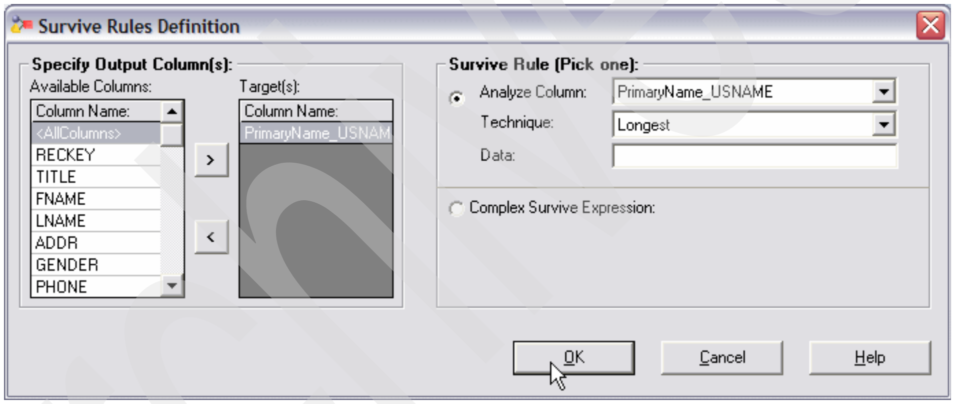
Analyze Column: MiddleName_USNAME

Technique: Longest

Data:

Complex Survive Expression:

Figure 1-344   Create J13_SURVIVE_CUSTOMER job 5/14


Survive Rules Definition

Specify Output Column(s):

Available Columns:

Column Name:
<AllColumns>
RECKEY
TITLE
FNAME
LNAME
ADDR
GENDER
PHONE

Target(s):

Column Name:
PrimaryName_USNAM

Survive Rule (Pick one):

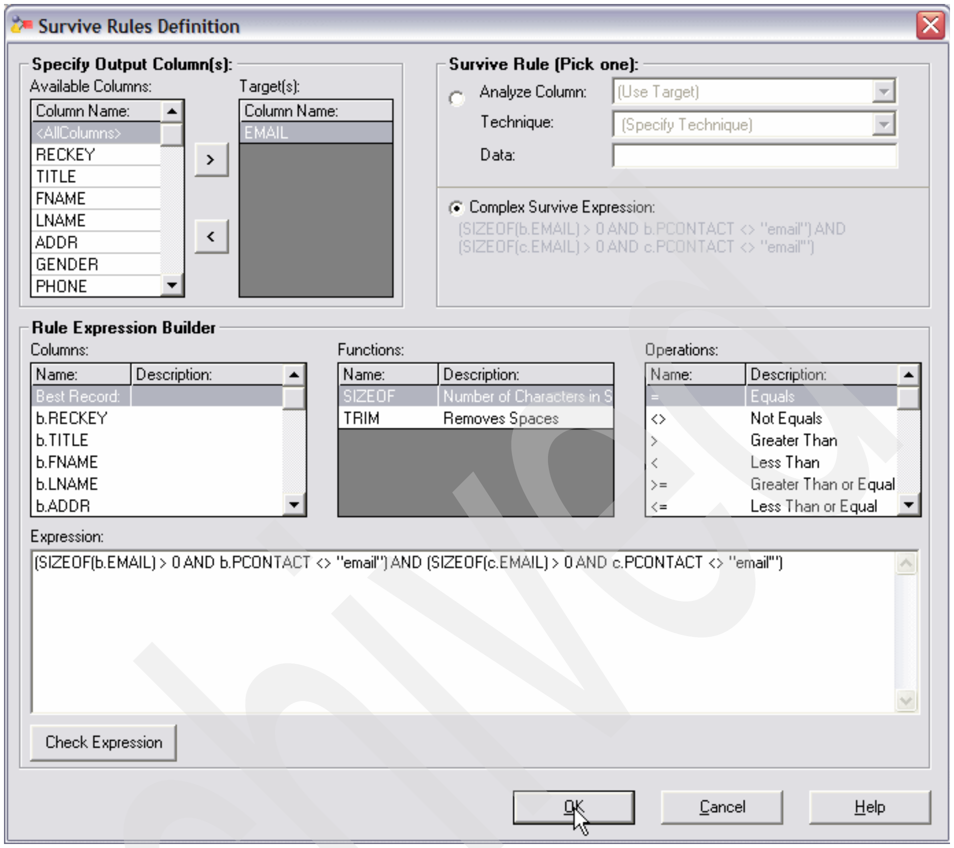
Analyze Column: PrimaryName_USNAME

Technique: Longest

Data:

Complex Survive Expression:

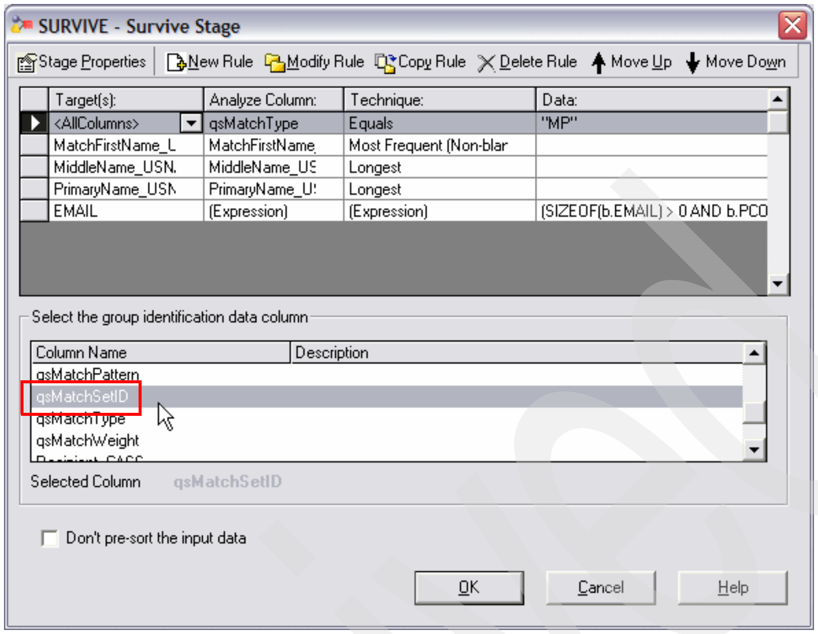
Figure 1-345   Create J13_SURVIVE_CUSTOMER job 6/14



Complex Survive Expression: (SIZEOF(b.EMAIL) > 0 AND b.PCONTACT <> "email") AND (SIZEOF(c.EMAIL) > 0 AND c.PCONTACT <> "email")

Expression: (SIZEOF(b.EMAIL) > 0 AND b.PCONTACT <> "email") AND (SIZEOF(c.EMAIL) > 0 AND c.PCONTACT <> "email")

Figure 1-346 Create J13_SURVIVE_CUSTOMER job 7/14



Target(s):	Analyze Column:	Technique:	Data:
<AllColumns>	qsMatchType	Equals	"MP"
MatchFirstName_L	MatchFirstName	Most Frequent (Non-blar	
MiddleName_USN.	MiddleName_US	Longest	
PrimaryName_USN	PrimaryName_U:	Longest	
EMAIL	(Expression)	(Expression)	(SIZEOF(b.EMAIL) > 0 AND b.PCO

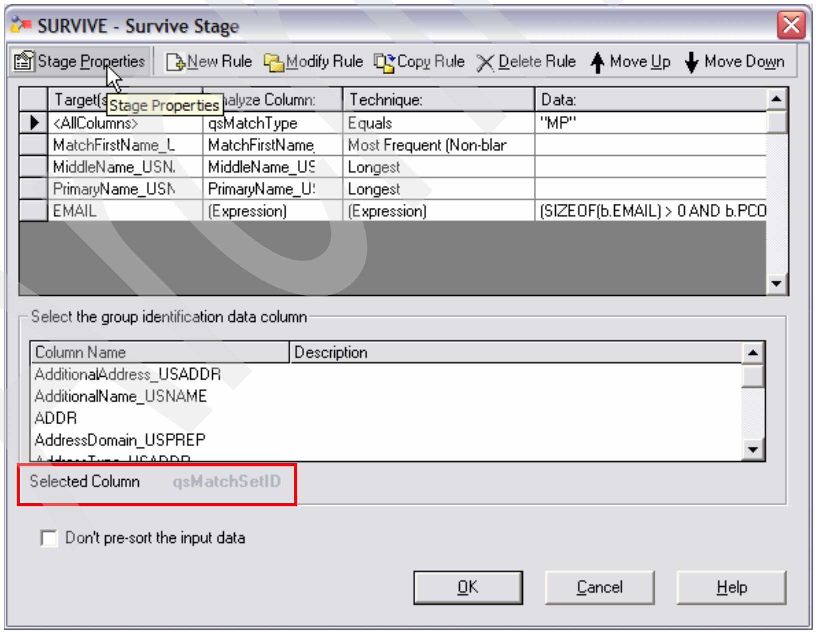
Select the group identification data column

Column Name	Description
qsMatchPattern	
qsMatchSetID	
qsMatchType	
qsMatchWeight	

Selected Column qsMatchSetID

Don't pre-sort the input data

Figure 1-347 Create J13_SURVIVE_CUSTOMER job 8/14



Target(s):	Analyze Column:	Technique:	Data:
<AllColumns>	qsMatchType	Equals	"MP"
MatchFirstName_L	MatchFirstName	Most Frequent (Non-blar	
MiddleName_USN.	MiddleName_US	Longest	
PrimaryName_USN	PrimaryName_U:	Longest	
EMAIL	(Expression)	(Expression)	(SIZEOF(b.EMAIL) > 0 AND b.PCO

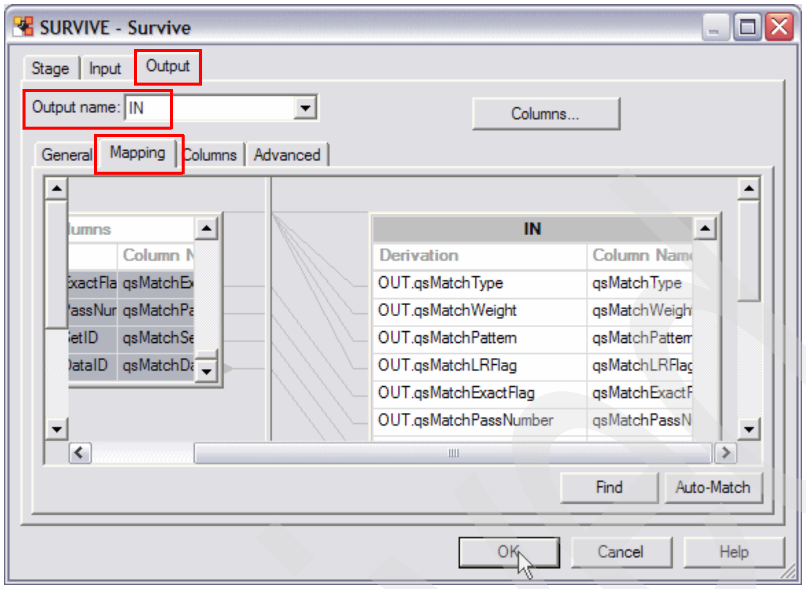
Select the group identification data column

Column Name	Description
AdditionalAddress_USADDR	
AdditionalName_USNAME	
ADDR	
AddressDomain_USPREP	

Selected Column qsMatchSetID

Don't pre-sort the input data

Figure 1-348 Create J13_SURVIVE_CUSTOMER job 9/14



Derivation	Column Name
OUT.qsMatchType	qsMatchType
OUT.qsMatchWeight	qsMatchWeigh
OUT.qsMatchPattern	qsMatchPatter
OUT.qsMatchLRFlag	qsMatchLRFlag
OUT.qsMatchExactFlag	qsMatchExactF
OUT.qsMatchPassNumber	qsMatchPassN

Figure 1-349   Create J13_SURVIVE_CUSTOMER job 10/14

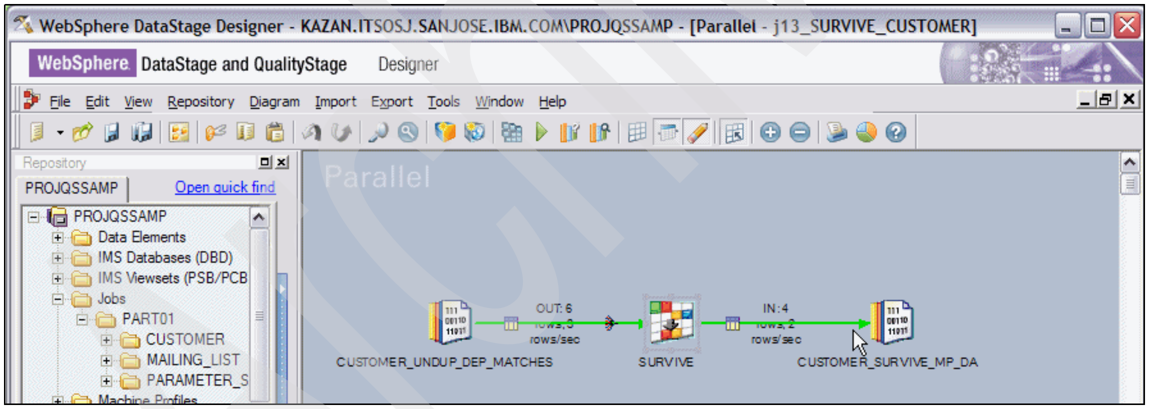
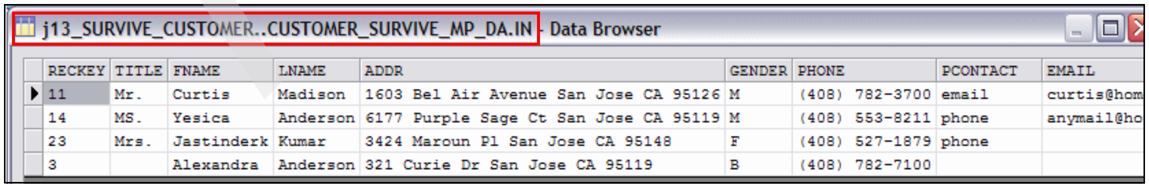
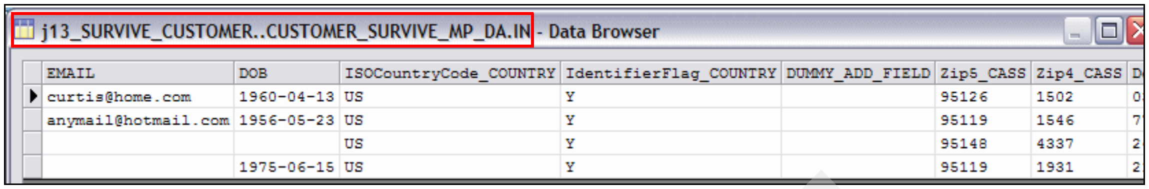

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j13_SURVIVE_CUSTOMER]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Repository
PROJQSSAMP
Open quick find
PROJQSSAMP
Data Elements
IMS Databases (DBD)
IMS Viewsets (PSB/PCB
Jobs
PART01
CUSTOMER
MAILING_LIST
PARAMETER_S
Machine Profiles
Parallel
OUT: 6 rows, 3 rows/sec
IN: 4 rows, 2 rows/sec
CUSTOMER_UNDUP_DEP_MATCHES
SURVIVE
CUSTOMER_SURVIVE_MP_DA

Figure 1-350   Create J13_SURVIVE_CUSTOMER job 11/14


j13_SURVIVE_CUSTOMER..CUSTOMER_SURVIVE_MP_DA.IN - Data Browser

RECKEY	TITLE	FNAME	LNAME	ADDR	GENDER	PHONE	PCONTACT	EMAIL
11	Mr.	Curtis	Madison	1603 Bel Air Avenue San Jose CA 95126	M	(408) 782-3700	email	curtis@hom
14	MS.	Yesica	Anderson	6177 Purple Sage Ct San Jose CA 95119	M	(408) 553-8211	phone	anymail@ho
23	Mrs.	Jastinderk	Kumar	3424 Maroun Pl San Jose CA 95148	F	(408) 527-1879	phone	
3		Alexandra	Anderson	321 Curie Dr San Jose CA 95119	B	(408) 782-7100		

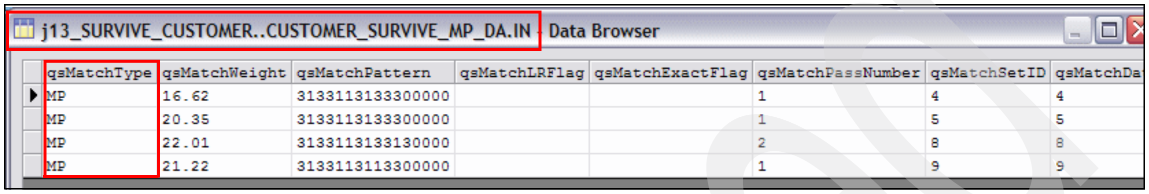
Figure 1-351   Create J13_SURVIVE_CUSTOMER job 12/14



j13_SURVIVE_CUSTOMER..CUSTOMER_SURVIVE_MP_DA.IN - Data Browser

EMAIL	DOB	ISOCountryCode_COUNTRY	IdentifierFlag_COUNTRY	DUMMY_ADD_FIELD	Zip5_CASS	Zip4_CASS	D
curtis@home.com	1960-04-13	US	Y		95126	1502	0
anymail@hotmail.com	1956-05-23	US	Y		95119	1546	7
		US	Y		95148	4337	2
	1975-06-15	US	Y		95119	1931	2

Figure 1-352 Create J13_SURVIVE_CUSTOMER job 13/14



j13_SURVIVE_CUSTOMER..CUSTOMER_SURVIVE_MP_DA.IN - Data Browser

qsMatchType	qsMatchWeight	qsMatchPattern	qsMatchLRFlag	qsMatchExactFlag	qsMatchPassNumber	qsMatchSetID	qsMatchDa
MP	16.62	3133113133300000			1	4	4
MP	20.35	3133113133300000			1	5	5
MP	22.01	3133113133130000			2	8	8
MP	21.22	3133113113300000			1	9	9

Figure 1-353 Create J13_SURVIVE_CUSTOMER job 14/14

J14_CUSTOMER_MASTER

In this step, we create a clean master of credit card customers (with duplicates eliminated) by merging the master, clerical review (with duplicates removed), and residual records into a sequential file using the FUNNEL stage. The records inserted into the sequential file are sorted in ascending order of the RECKEY column. Also, any null values in the input records are substituted with the string `NULL`.

Note: We defined a sequential file as output rather than a data set because of a bug that produced incorrect results with a data set but not with a sequential file. A Match Specification stage to which the file is input *only* accepts data sets. Therefore, we introduced another intermediate step, J14A_CUSTOMER_MASTER, that copied the contents of the sequential file that is created here into a data set.

All the inputs to a FUNNEL stage must have the same number of columns in order to do a union. The output of the Survive stage in "J13_SURVIVE_CUSTOMER" on page 295 and the outputs of the Unduplicate stage in "J11_UNDUP_DEP_MATCH_CUSTOMER" on page 282 do not have the same number of columns. Therefore, you first need to identify the metadata and save as a table definition. Then, you use the same table definition for all the inputs.

Figure 1-354 on page 305 through Figure 1-373 on page 315 describe the main steps using Designer Client to perform this task.

The main steps are as follows:

1. Figure 1-354 on page 305 shows the various stages that are used in this job, including the input data sets CUSTOMER_SURVIVE_MP_DA (that was created in “J13_SURVIVE_CUSTOMER” on page 295, which contains survived records), CUSTOMER_UNDUP_DEP_CLERICALS, and CUSTOMER_UNDUP_DEP_RESIDUALS that was created in the “J11_UNDUP_DEP_MATCH_CUSTOMER” on page 282, a FUNNEL stage, and an output data set containing the merged records from the input. We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object created earlier.
2. Configure the input data set CUSTOMER_UNDUP_DEP_MATCHES data set as shown in Figure 1-355 on page 306 through Figure 1-362 on page 309:
   a. Right-click the **CUSTOMER_SURVIVE_MP_DA** stage icon object and select **Properties** as shown in Figure 1-355 on page 306.
   b. In the CUSTOMER_SURVIVE_MP_DA - Data Set window, under the Output tab for the OUT1 link, click the Properties tab and type information about the path and name of the file as shown in Figure 1-356 on page 306. We do not repeat this process here because it is similar to earlier configurations.
   c. Click the Columns tab and then select and delete the eight (8) “qs” columns as shown in Figure 1-357 on page 307. Confirm the deletion request by clicking **Yes** in the DataStage window as shown in Figure 1-358 on page 307.
   d. Click **Save** to save a table definition with the name CUSTOMER_MASTER for the remaining columns as shown in Figure 1-359 on page 308 through Figure 1-361 on page 309.
3. Configure the input CUSTOMER_UNDUP_DEP_CLERICALS data set (link OUT2) using the table definition CUSTOMER_MASTER as shown in Figure 1-362 on page 309 through Figure 1-366 on page 311.
4. Repeat the process for the input CUSTOMER_UNDUP_DEP_RESIDUALS data set (link OUT3). We do not show this process here.
5. Configure the FUNNEL stage. Figure 1-367 on page 312 shows the FUNNEL - Funnel window with the mapping of all columns from the Columns pane to the IN pane.
6. Next, configure the output sequential file:
   a. Double-click the sequential file CUSTOMER_MASTER_SF icon object. In the CUSTOMER_MASTER_SF - Sequential File window that opens, under the Input tab for the IN link, click the Properties tab and type information about the path and name of the file as shown in Figure 1-368 on page 312.

b. Click the Partitioning tab, select **Perform sort**, select **RECKEY**, and sort in ascending sequence as shown in Figure 1-369 on page 313. This requests the data in the sequential file to be in that sequence.

c. Click the Format tab, select the Null field value to be NULL as shown in Figure 1-370 on page 314, which indicates that the string NULL is substituted in a column when that column contains a null value. Click **OK**.

7. After saving, compiling, and running this job j14_CUSTOMER_MASTER, view the content of the output data set object CUSTOMER_MASTER_SF as shown in Figure 1-371 on page 314 through Figure 1-373 on page 315.

   This report shows the cleansed credit card customer file with no duplicates.

Now, proceed to “J14A_CUSTOMER_MASTER” on page 316.

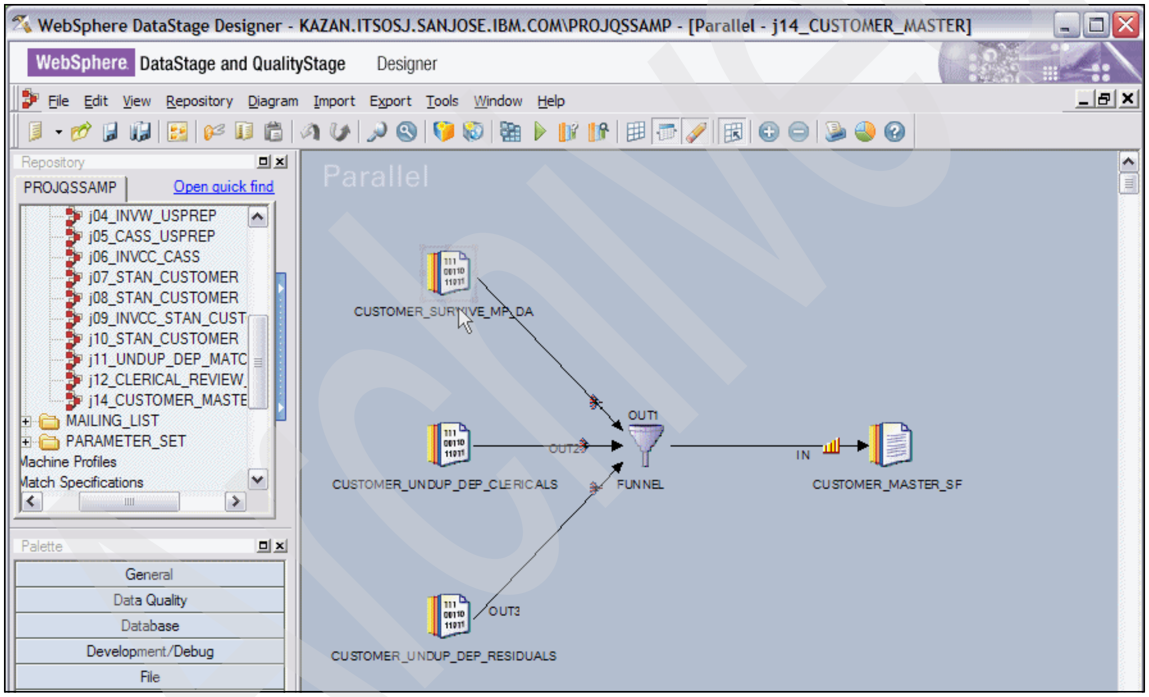


Figure 1-354 Create J14_CUSTOMER_MASTER job 1/20

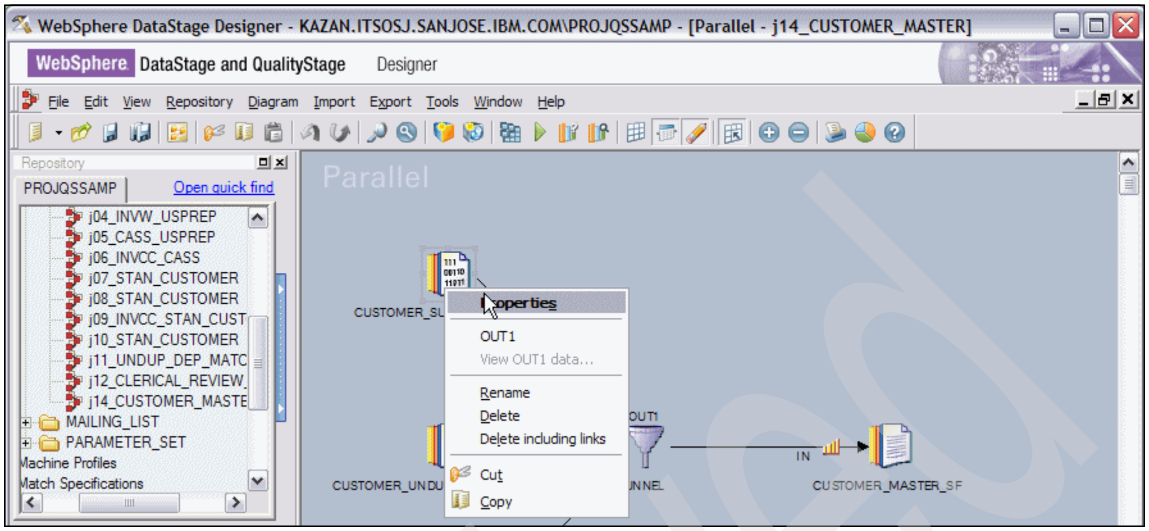
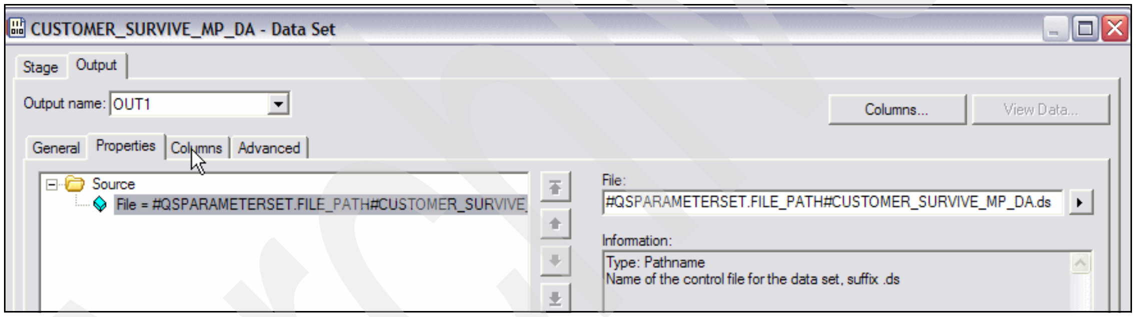


Figure 1-355   Create J14_CUSTOMER_MASTER job 2/20



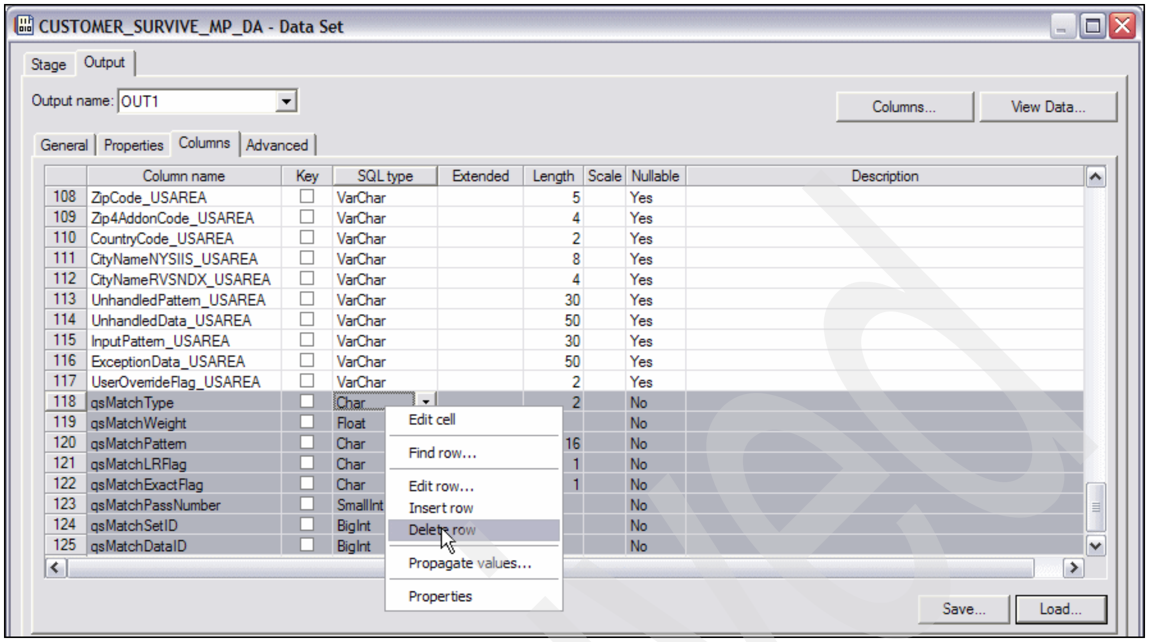
Output name: OUT1

File = #QSPARAMETERSET.FILE_PATH#CUSTOMER_SURVIVE_

File:
#QSPARAMETERSET.FILE_PATH#CUSTOMER_SURVIVE_MP_DA.ds

Information:
Type: Pathname
Name of the control file for the data set, suffix .ds

Figure 1-356   Create J14_CUSTOMER_MASTER job 3/20

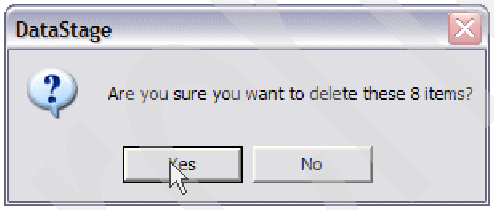


CUSTOMER_SURVIVE_MP_DA - Data Set

Output name: OUT1

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	Description
108	ZipCode_USAREA	☐	VarChar		5		Yes	
109	Zip4AddonCode_USAREA	☐	VarChar		4		Yes	
110	CountryCode_USAREA	☐	VarChar		2		Yes	
111	CityNameNYSIIS_USAREA	☐	VarChar		8		Yes	
112	CityNameRVSNDX_USAREA	☐	VarChar		4		Yes	
113	UnhandledPattern_USAREA	☐	VarChar		30		Yes	
114	UnhandledData_USAREA	☐	VarChar		50		Yes	
115	InputPattern_USAREA	☐	VarChar		30		Yes	
116	ExceptionData_USAREA	☐	VarChar		50		Yes	
117	UserOverrideFlag_USAREA	☐	VarChar		2		Yes	
118	qsMatchType	☐	Char		2		No	
119	qsMatchWeight	☐	Float				No	
120	qsMatchPattern	☐	Char		16		No	
121	qsMatchLRFlag	☐	Char		1		No	
122	qsMatchExactFlag	☐	Char		1		No	
123	qsMatchPassNumber	☐	SmallInt				No	
124	qsMatchSetID	☐	BigInt				No	
125	qsMatchDataID	☐	BigInt				No	

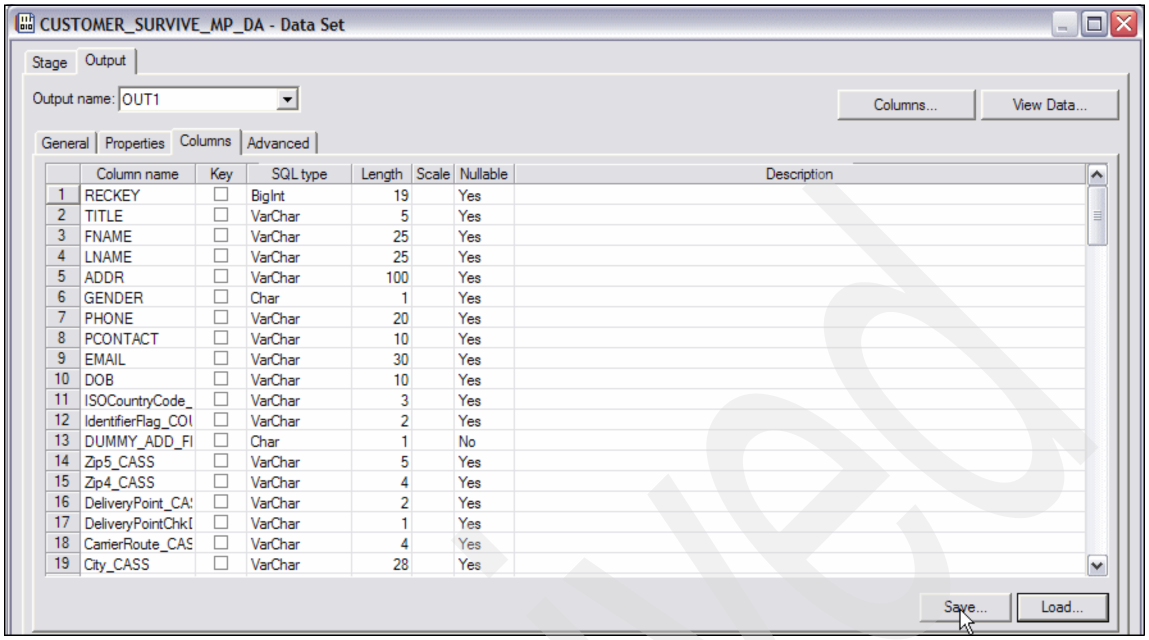
Figure 1-357 Create J14_CUSTOMER_MASTER job 4/20



DataStage

Are you sure you want to delete these 8 items?

Figure 1-358 Create J14_CUSTOMER_MASTER job 5/20

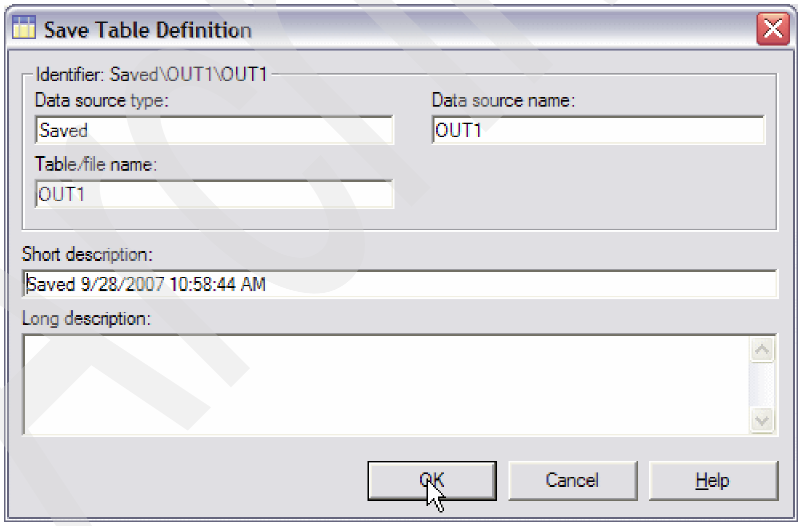


CUSTOMER_SURVIVE_MP_DA - Data Set

Output name: OUT1

	Column name	Key	SQL type	Length	Scale	Nullable	Description
1	RECKEY	☐	BigInt	19		Yes	
2	TITLE	☐	VarChar	5		Yes	
3	FNAME	☐	VarChar	25		Yes	
4	LNAME	☐	VarChar	25		Yes	
5	ADDR	☐	VarChar	100		Yes	
6	GENDER	☐	Char	1		Yes	
7	PHONE	☐	VarChar	20		Yes	
8	PCONTACT	☐	VarChar	10		Yes	
9	EMAIL	☐	VarChar	30		Yes	
10	DOB	☐	VarChar	10		Yes	
11	ISOCountryCode_	☐	VarChar	3		Yes	
12	IdentifierFlag_COI	☐	VarChar	2		Yes	
13	DUMMY_ADD_FI	☐	Char	1		No	
14	Zip5_CASS	☐	VarChar	5		Yes	
15	Zip4_CASS	☐	VarChar	4		Yes	
16	DeliveryPoint_CA	☐	VarChar	2		Yes	
17	DeliveryPointChkI	☐	VarChar	1		Yes	
18	CarrierRoute_CAS	☐	VarChar	4		Yes	
19	City_CASS	☐	VarChar	28		Yes	

Figure 1-359   Create J14_CUSTOMER_MASTER job 6/20



Save Table Definition

Identifier: Saved\OUT1\OUT1

Data source type: Saved

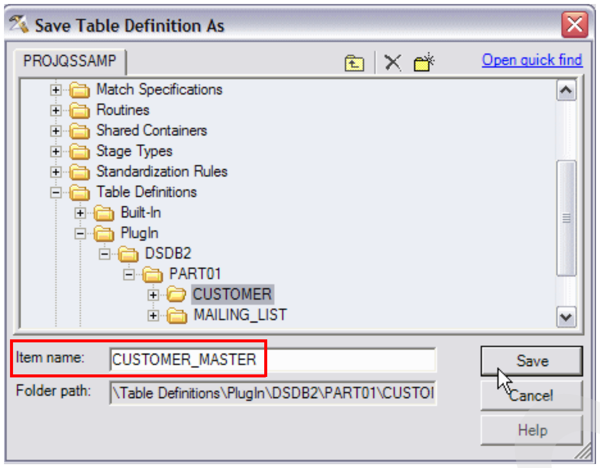
Data source name: OUT1

Table/file name: OUT1

Short description: Saved 9/28/2007 10:58:44 AM

Long description:

Figure 1-360   Create J14_CUSTOMER_MASTER job 7/20



Item name: CUSTOMER_MASTER

Folder path: \Table Definitions\PlugIn\DSDB2\PART01\CUSTOI

Figure 1-361 Create J14_CUSTOMER_MASTER job 8/20

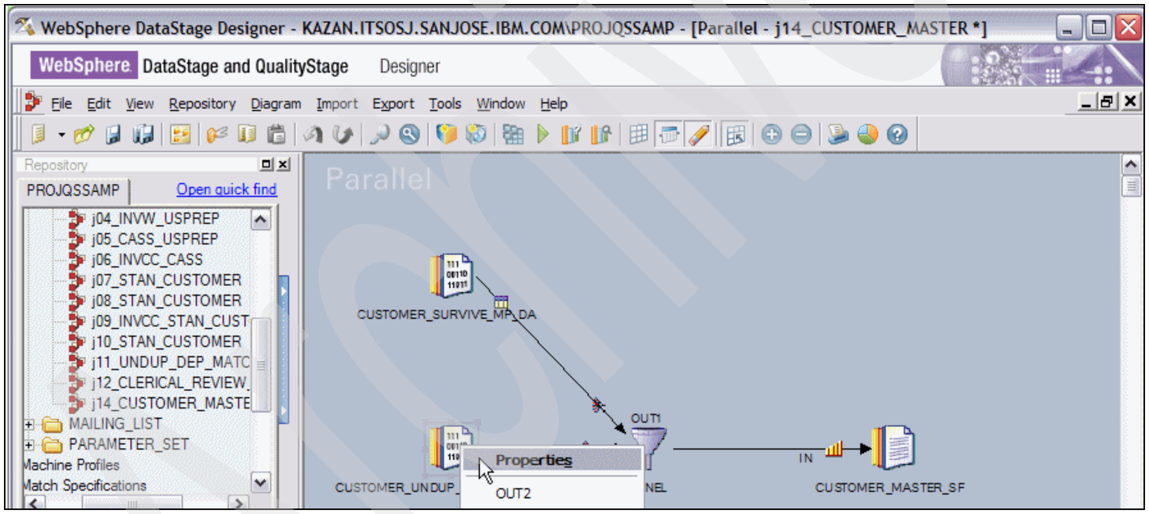


Figure 1-362 Create J14_CUSTOMER_MASTER job 9/20

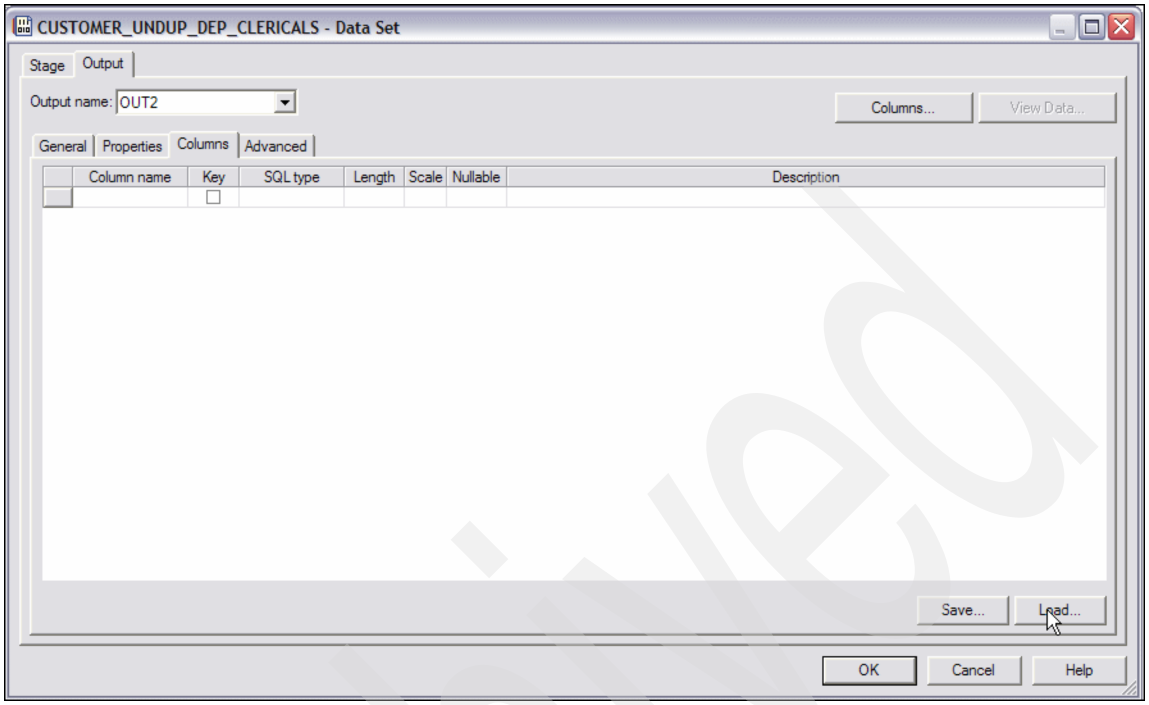


Figure 1-363 Create J14_CUSTOMER_MASTER job 10/20

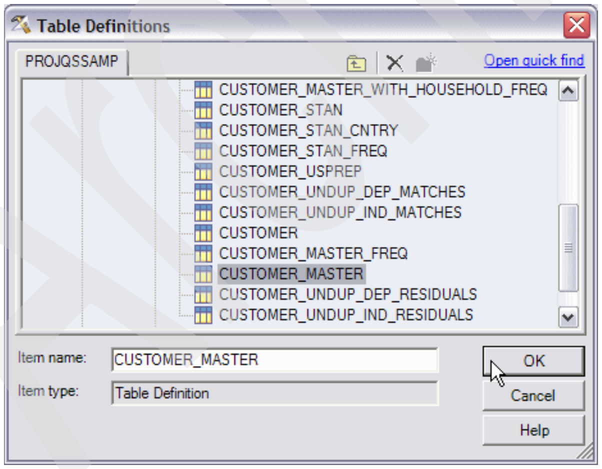
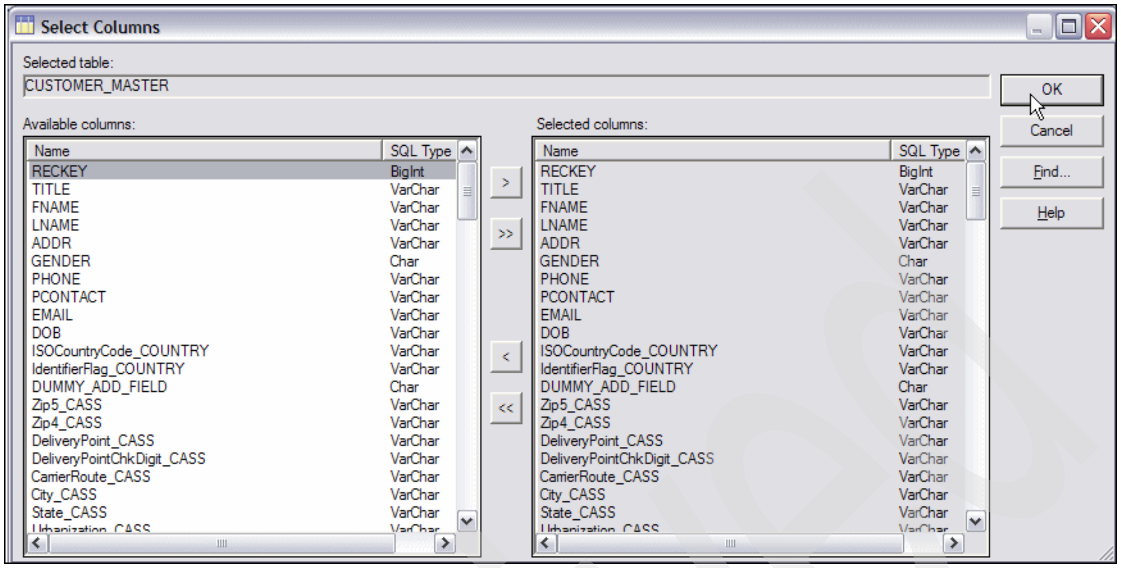


Table Definitions
PROJQSSAMP
Open quick find
CUSTOMER_MASTER_WITH_HOUSEHOLD_FREQ
CUSTOMER_STAN
CUSTOMER_STAN_CNTRY
CUSTOMER_STAN_FREQ
CUSTOMER_USPREP
CUSTOMER_UNDUP_DEP_MATCHES
CUSTOMER_UNDUP_IND_MATCHES
CUSTOMER
CUSTOMER_MASTER_FREQ
CUSTOMER_MASTER
CUSTOMER_UNDUP_DEP_RESIDUALS
CUSTOMER_UNDUP_IND_RESIDUALS
Item name: CUSTOMER_MASTER
Item type: Table Definition
OK
Cancel
Help

Figure 1-364 Create J14_CUSTOMER_MASTER job 11/20



Selected table: CUSTOMER_MASTER

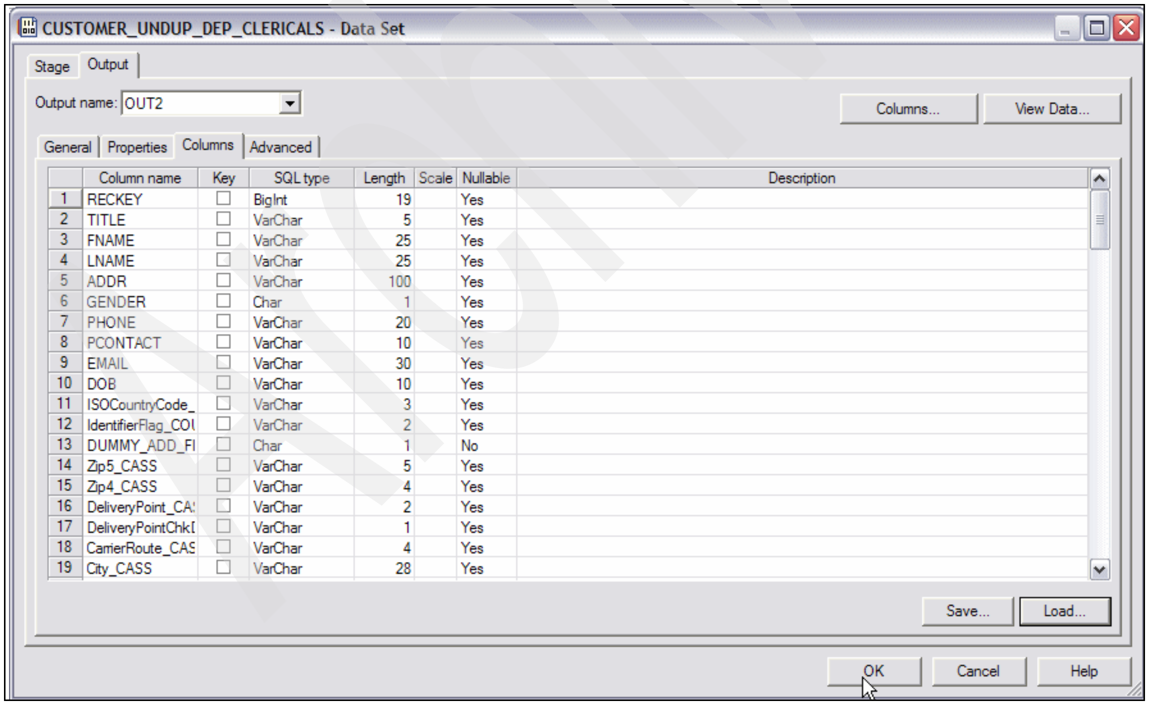
Available columns:

Name	SQL Type
RECKEY	BigInt
TITLE	VarChar
FNAME	VarChar
LNAME	VarChar
ADDR	VarChar
GENDER	Char
PHONE	VarChar
PCONTACT	VarChar
EMAIL	VarChar
DOB	VarChar
ISOCountryCode_COUNTRY	VarChar
IdentifierFlag_COUNTRY	VarChar
DUMMY_ADD_FIELD	Char
Zip5_CASS	VarChar
Zip4_CASS	VarChar
DeliveryPoint_CASS	VarChar
DeliveryPointChkDigit_CASS	VarChar
CarrierRoute_CASS	VarChar
City_CASS	VarChar
State_CASS	VarChar

Selected columns:

Name	SQL Type
RECKEY	BigInt
TITLE	VarChar
FNAME	VarChar
LNAME	VarChar
ADDR	VarChar
GENDER	Char
PHONE	VarChar
PCONTACT	VarChar
EMAIL	VarChar
DOB	VarChar
ISOCountryCode_COUNTRY	VarChar
IdentifierFlag_COUNTRY	VarChar
DUMMY_ADD_FIELD	Char
Zip5_CASS	VarChar
Zip4_CASS	VarChar
DeliveryPoint_CASS	VarChar
DeliveryPointChkDigit_CASS	VarChar
CarrierRoute_CASS	VarChar
City_CASS	VarChar
State_CASS	VarChar

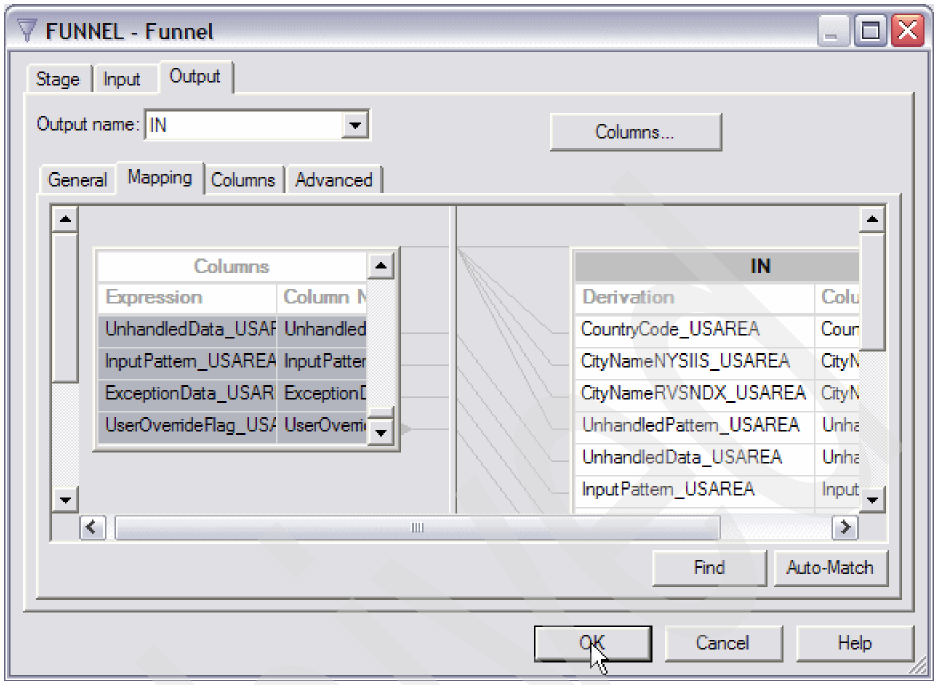
Figure 1-365 Create J14_CUSTOMER_MASTER job 12/20



Output name: OUT2

	Column name	Key	SQL type	Length	Scale	Nullable	Description
1	RECKEY	☐	BigInt	19		Yes	
2	TITLE	☐	VarChar	5		Yes	
3	FNAME	☐	VarChar	25		Yes	
4	LNAME	☐	VarChar	25		Yes	
5	ADDR	☐	VarChar	100		Yes	
6	GENDER	☐	Char	1		Yes	
7	PHONE	☐	VarChar	20		Yes	
8	PCONTACT	☐	VarChar	10		Yes	
9	EMAIL	☐	VarChar	30		Yes	
10	DOB	☐	VarChar	10		Yes	
11	ISOCountryCode_	☐	VarChar	3		Yes	
12	IdentifierFlag_COI	☐	VarChar	2		Yes	
13	DUMMY_ADD_FI	☐	Char	1		No	
14	Zip5_CASS	☐	VarChar	5		Yes	
15	Zip4_CASS	☐	VarChar	4		Yes	
16	DeliveryPoint_CA!	☐	VarChar	2		Yes	
17	DeliveryPointChkI	☐	VarChar	1		Yes	
18	CarrierRoute_CAS	☐	VarChar	4		Yes	
19	City_CASS	☐	VarChar	28		Yes	

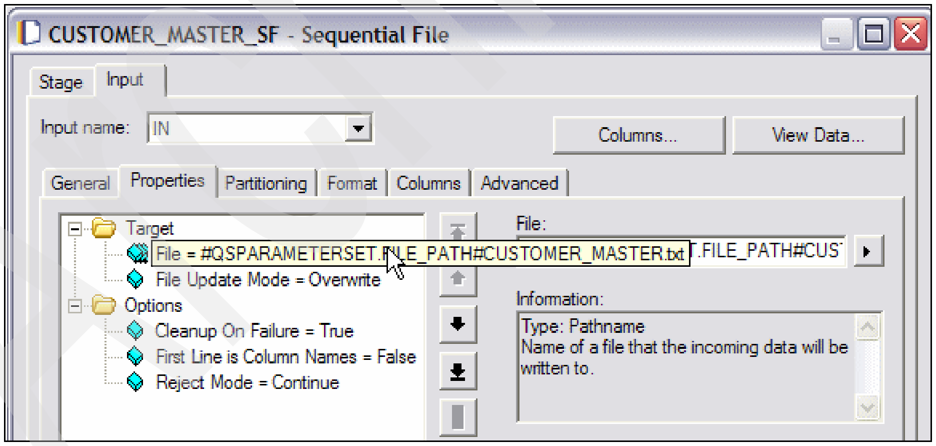
Figure 1-366 Create J14_CUSTOMER_MASTER job 13/20



Expression	Column N
UnhandledData_USAF	Unhandled
InputPattern_USAREA	InputPatter
ExceptionData_USAR	ExceptionD
UserOverrideFlag_US/	UserOverri

Derivation	Colu
CountryCode_USAREA	Coun
CityNameNYSIIS_USAREA	CityN
CityNameRVSNDX_USAREA	CityN
UnhandledPattern_USAREA	Unha
UnhandledData_USAREA	Unha
InputPattern_USAREA	Input

Figure 1-367 Create J14_CUSTOMER_MASTER job 14/20

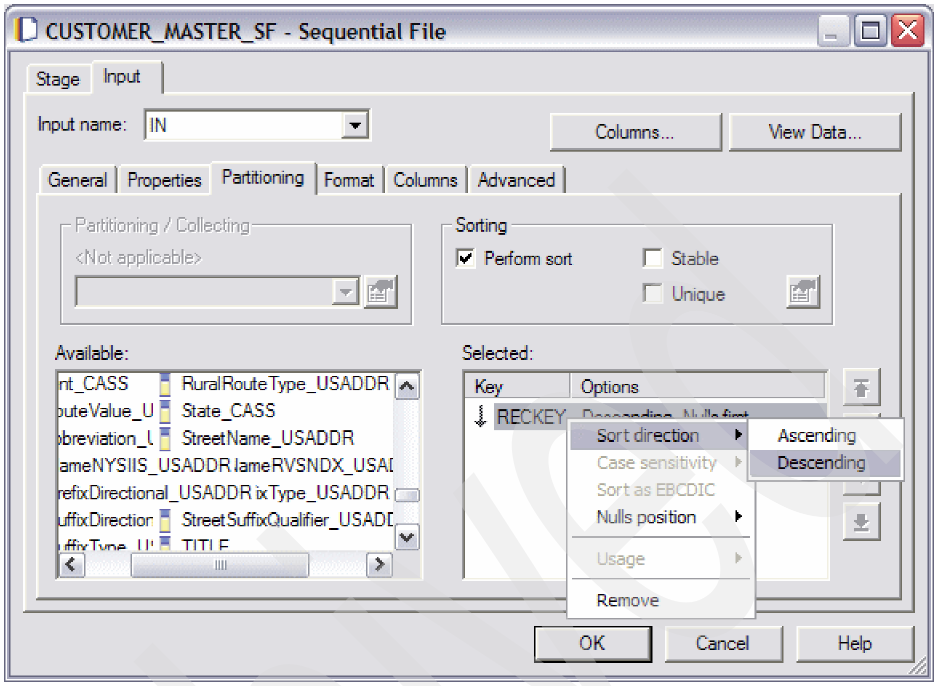


Target
File = #QSPARAMETERSET.FILE_PATH#CUSTOMER_MASTER.txt
File Update Mode = Overwrite
Options
Cleanup On Failure = True
First Line is Column Names = False
Reject Mode = Continue

File:
T.FILE_PATH#CUS

Information:
Type: Pathname
Name of a file that the incoming data will be written to.

Figure 1-368 Create J14_CUSTOMER_MASTER job 15/20


CUSTOMER_MASTER_SF - Sequential File

Stage | Input

Input name: IN

Columns... | View Data...

General | Properties | Partitioning | Format | Columns | Advanced

Partitioning / Collecting: <Not applicable>

Sorting: [✓ Perform sort] Stable Unique

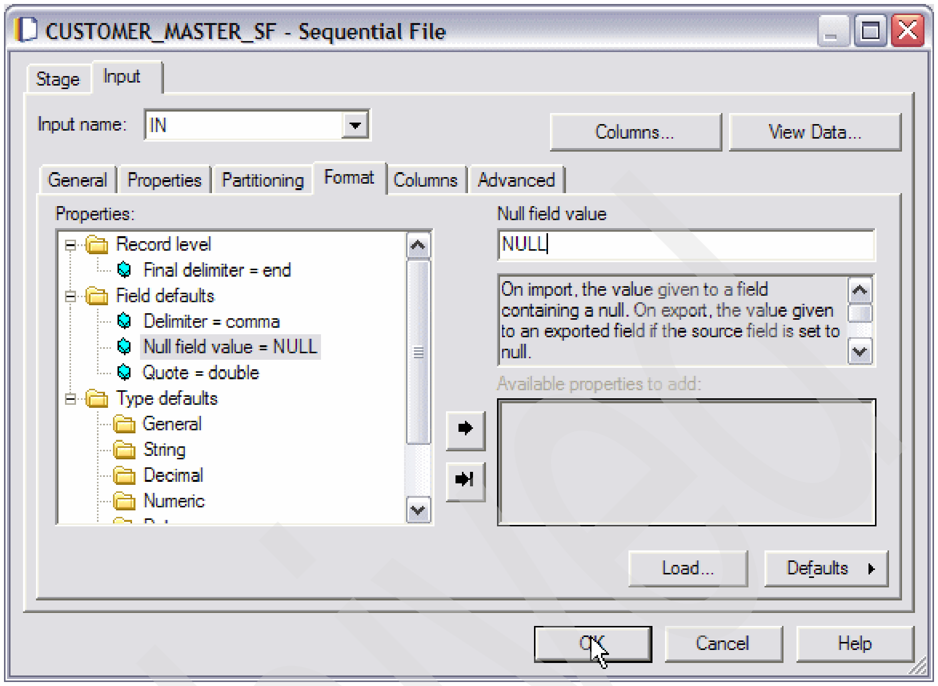
Available:

```
nt_CASS            RuralRouteType_USADDR
buteValue_U        State_CASS
bbreviation_U      StreetName_USADDR
ameNYSIIS_USADDR   lameRVSNDX_USAI
refixDirectional_USADDR   ixType_USADDR
uffixDirection     StreetSuffixQualifier_USADI
uffixType_U        TITLE
```

Selected:

Key	Options
RECKEY	Descending, Nulls first

Figure 1-369 Create J14_CUSTOMER_MASTER job 16/20


CUSTOMER_MASTER_SF - Sequential File

Input name: IN

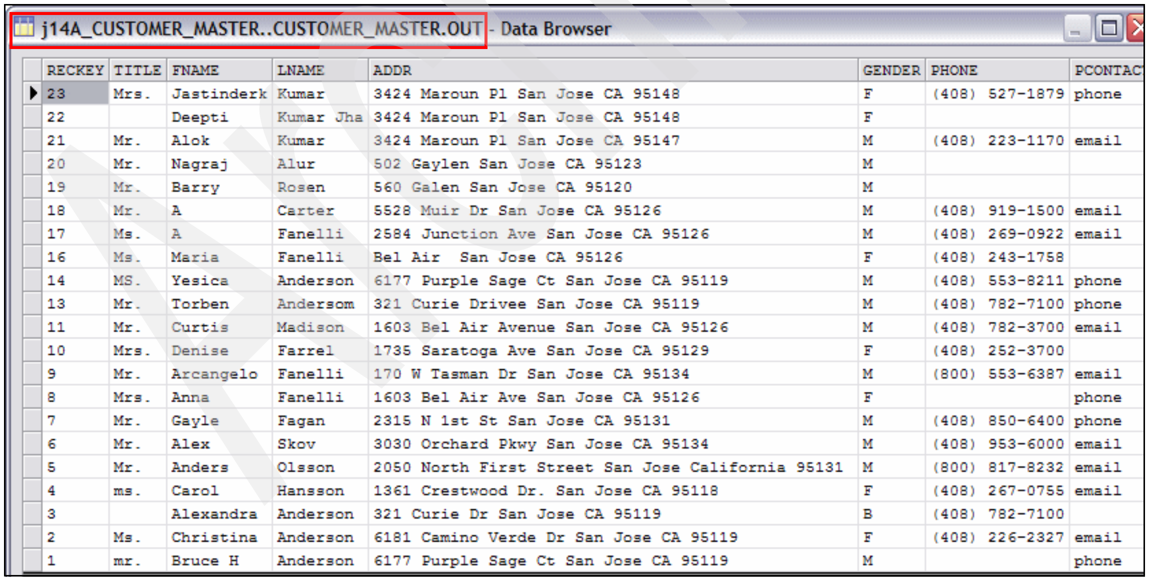
Properties:

- Record level
  - Final delimiter = end
- Field defaults
  - Delimiter = comma
  - Null field value = NULL
  - Quote = double
- Type defaults
  - General
  - String
  - Decimal
  - Numeric

Null field value: NULL

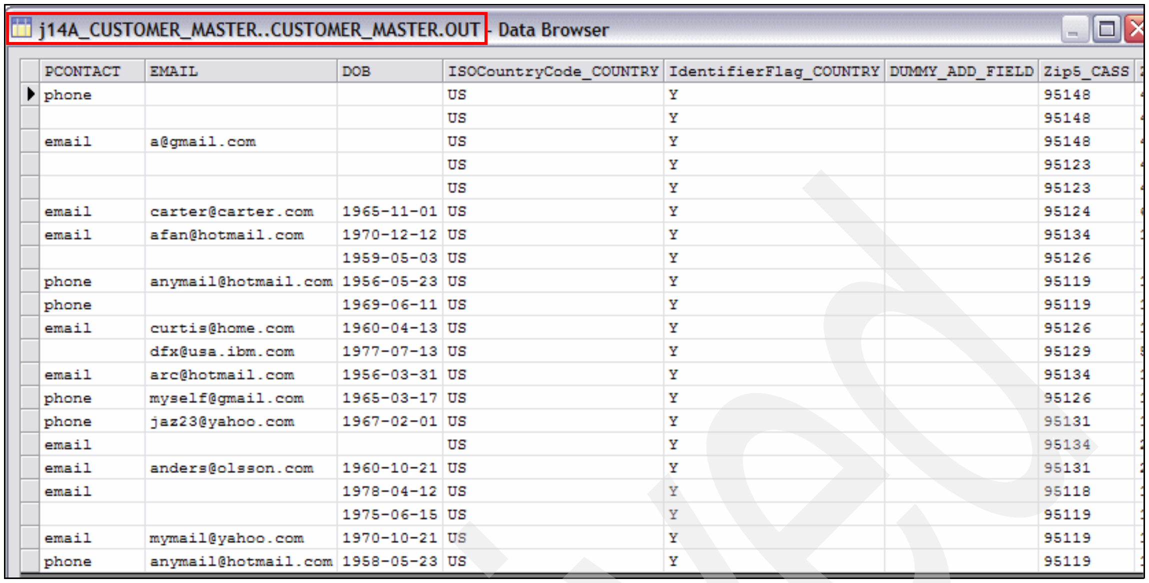
On import, the value given to a field containing a null. On export, the value given to an exported field if the source field is set to null.

Figure 1-370   Create J14_CUSTOMER_MASTER job 17/20


j14A_CUSTOMER_MASTER..CUSTOMER_MASTER.OUT - Data Browser

RECKEY	TITLE	FNAME	LNAME	ADDR	GENDER	PHONE	PCONTAC
23	Mrs.	Jastinderk	Kumar	3424 Maroun Pl San Jose CA 95148	F	(408) 527-1879	phone
22		Deepti	Kumar Jha	3424 Maroun Pl San Jose CA 95148	F		
21	Mr.	Alok	Kumar	3424 Maroun Pl San Jose CA 95147	M	(408) 223-1170	email
20	Mr.	Nagraj	Alur	502 Gaylen San Jose CA 95123	M		
19	Mr.	Barry	Rosen	560 Galen San Jose CA 95120	M		
18	Mr.	A	Carter	5528 Muir Dr San Jose CA 95126	M	(408) 919-1500	email
17	Ms.	A	Fanelli	2584 Junction Ave San Jose CA 95126	M	(408) 269-0922	email
16	Ms.	Maria	Fanelli	Bel Air San Jose CA 95126	F	(408) 243-1758	
14	MS.	Yesica	Anderson	6177 Purple Sage Ct San Jose CA 95119	M	(408) 553-8211	phone
13	Mr.	Torben	Andersom	321 Curie Drivee San Jose CA 95119	M	(408) 782-7100	phone
11	Mr.	Curtis	Madison	1603 Bel Air Avenue San Jose CA 95126	M	(408) 782-3700	email
10	Mrs.	Denise	Farrel	1735 Saratoga Ave San Jose CA 95129	F	(408) 252-3700	
9	Mr.	Arcangelo	Fanelli	170 W Tasman Dr San Jose CA 95134	M	(800) 553-6387	email
8	Mrs.	Anna	Fanelli	1603 Bel Air Ave San Jose CA 95126	F		phone
7	Mr.	Gayle	Fagan	2315 N 1st St San Jose CA 95131	M	(408) 850-6400	phone
6	Mr.	Alex	Skov	3030 Orchard Pkwy San Jose CA 95134	M	(408) 953-6000	email
5	Mr.	Anders	Olsson	2050 North First Street San Jose California 95131	M	(800) 817-8232	email
4	ms.	Carol	Hansson	1361 Crestwood Dr. San Jose CA 95118	F	(408) 267-0755	email
3		Alexandra	Anderson	321 Curie Dr San Jose CA 95119	B	(408) 782-7100	
2	Ms.	Christina	Anderson	6181 Camino Verde Dr San Jose CA 95119	F	(408) 226-2327	email
1	mr.	Bruce H	Anderson	6177 Purple Sage Ct San Jose CA 95119	M		phone

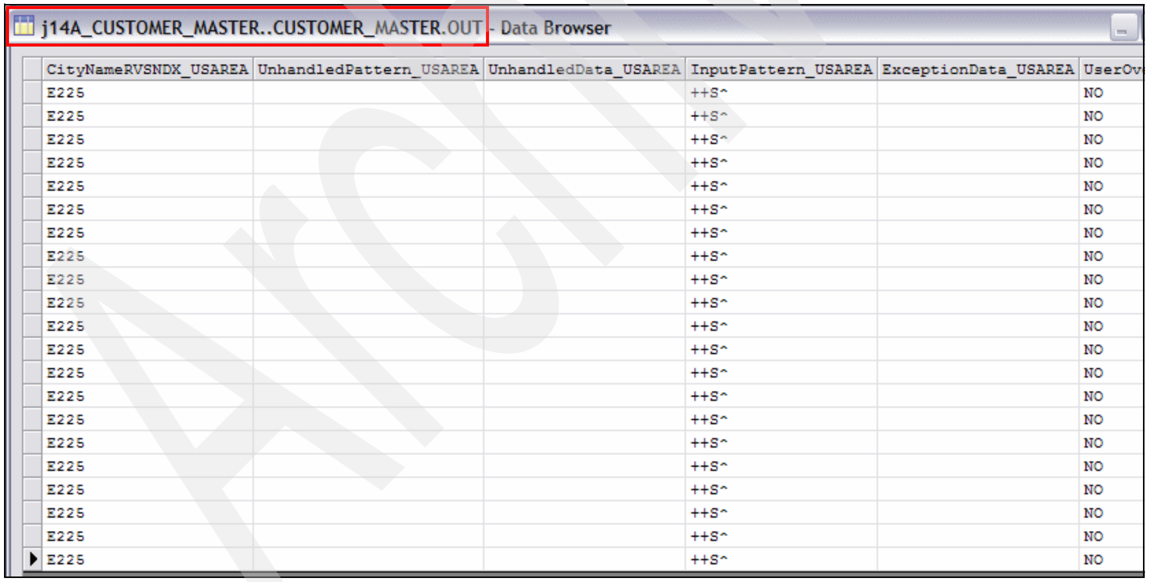
Figure 1-371   Create J14_CUSTOMER_MASTER job 18/20



j14A_CUSTOMER_MASTER..CUSTOMER_MASTER.OUT - Data Browser

PCONTACT	EMAIL	DOB	ISOCountryCode_COUNTRY	IdentifierFlag_COUNTRY	DUMMY_ADD_FIELD	Zip5_CASS
phone			US	Y		95148
			US	Y		95148
email	a@gmail.com		US	Y		95148
			US	Y		95123
			US	Y		95123
email	carter@carter.com	1965-11-01	US	Y		95124
email	afan@hotmail.com	1970-12-12	US	Y		95134
		1959-05-03	US	Y		95126
phone	anymail@hotmail.com	1956-05-23	US	Y		95119
phone		1969-06-11	US	Y		95119
email	curtis@home.com	1960-04-13	US	Y		95126
	dfx@usa.ibm.com	1977-07-13	US	Y		95129
email	arc@hotmail.com	1956-03-31	US	Y		95134
phone	myself@gmail.com	1965-03-17	US	Y		95126
phone	jaz23@yahoo.com	1967-02-01	US	Y		95131
email			US	Y		95134
email	anders@olsson.com	1960-10-21	US	Y		95131
email		1978-04-12	US	Y		95118
		1975-06-15	US	Y		95119
email	mymail@yahoo.com	1970-10-21	US	Y		95119
phone	anymail@hotmail.com	1958-05-23	US	Y		95119

Figure 1-372 Create J14_CUSTOMER_MASTER job 19/20



j14A_CUSTOMER_MASTER..CUSTOMER_MASTER.OUT - Data Browser

CityNameRVSNDX_USAREA	UnhandledPattern_USAREA	UnhandledData_USAREA	InputPattern_USAREA	ExceptionData_USAREA	UserOv
E225			++S^		NO
E225			++S^		NO
E225			++S^		NO
E225			++S^		NO
E225			++S^		NO
E225			++S^		NO
E225			++S^		NO
E225			++S^		NO
E225			++S^		NO
E225			++S^		NO
E225			++S^		NO
E225			++S^		NO
E225			++S^		NO
E225			++S^		NO
E225			++S^		NO
E225			++S^		NO
E225			++S^		NO
E225			++S^		NO
E225			++S^		NO
E225			++S^		NO
E225			++S^		NO

Figure 1-373 Create J14_CUSTOMER_MASTER job 20/20

J14A_CUSTOMER_MASTER

In this step, we copy the contents of the sequential file that was created in the “J14_CUSTOMER_MASTER” on page 303 job into a data set because a subsequent match specification that reads this data only accepts a data set as input.

Figure 1-374 shows the job that consists of an input sequential file CUSTOMER _MASTER_SF and an output data set CUSTOMER_MASTER—with no intervening stages.

We do not show the configuration of the input sequential file and the output data set here because it is similar to earlier configurations.

Now, proceed to “J15_FREQ_CUSTOMER_MASTER” on page 316.

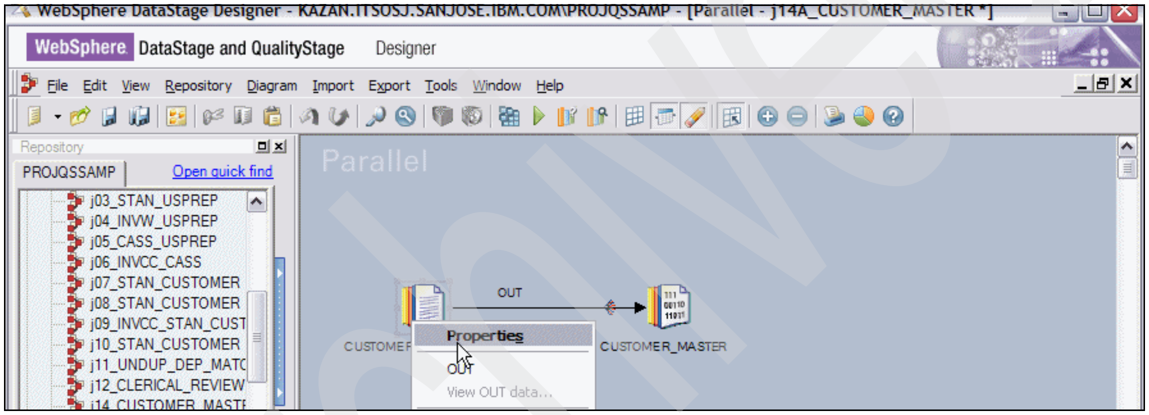


Figure 1-374 Create J14A_CUSTOMER_MASTER

J15_FREQ_CUSTOMER_MASTER

In this next step, we generate frequency distribution on all the columns in the cleansed credit card customer file of the job “J14A_CUSTOMER_MASTER” on page 316 using the Match Frequency stage. Here again, the idea is to generate match frequency for all the columns so that it can be used with any match specification. The output of this job is used in defining match specifications that are used for matching.

Figure 1-375 on page 317 shows the various stages that are used in this job, including the data set that was created in “J14A_CUSTOMER_MASTER” on page 316, a Match Frequency stage, and a Sequential File stage. We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object that was created earlier.

Because the configuration of this job is very similar to that described in "J10_MATCHFREQ_STAN_CUSTOMER" on page 266, we do not repeat it here.

The contents of the output of this stage (CUSTOMER_MASTER_FREQ) are listed in Figure 1-376 on page 318. We do not document the interpretation of the format and content of this file.

Now, proceed to "J15_Undup_MatchSpec_CUSTOMER" on page 318.

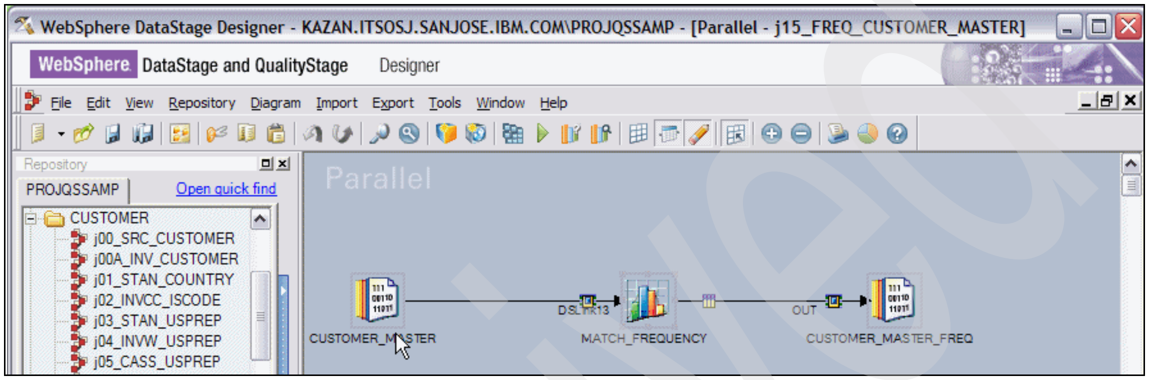
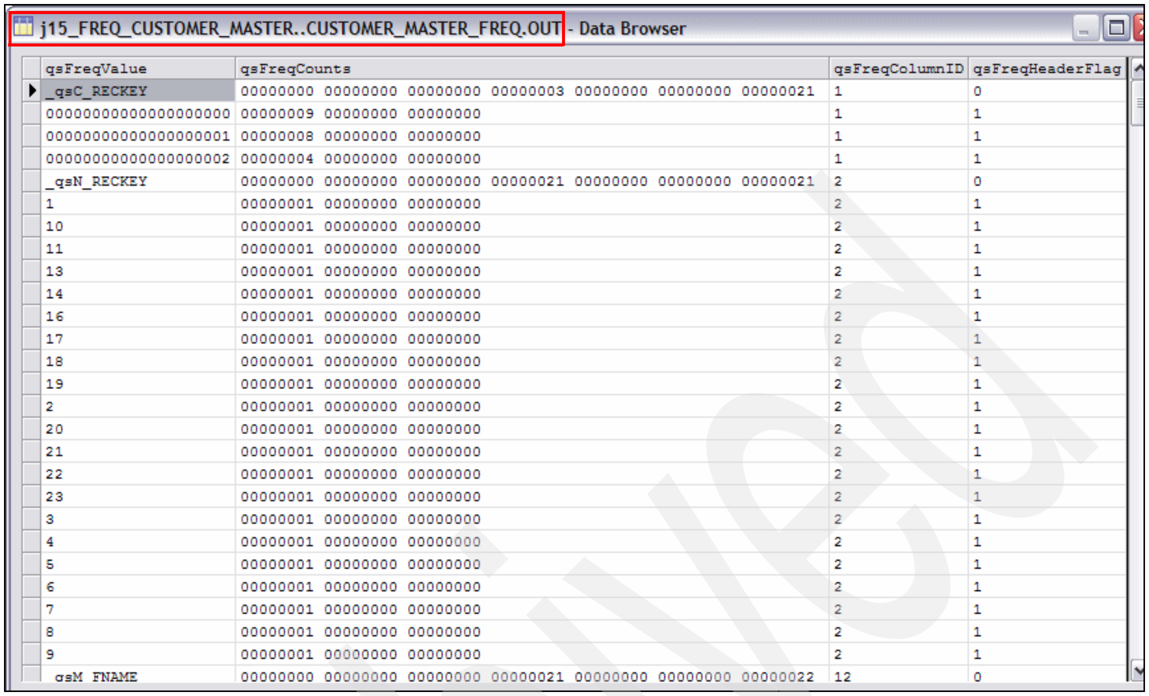


Figure 1-375 Create J15_FREQ_CUSTOMER_MASTER 1/2


j15_FREQ_CUSTOMER_MASTER..CUSTOMER_MASTER_FREQ.OUT - Data Browser

qsFreqValue	qsFreqCounts	qsFreqColumnID	qsFreqHeaderFlag
_qsC_RECKEY	00000000 00000000 00000000 00000003 00000000 00000000 00000021	1	0
0000000000000000000	00000009 00000000 00000000	1	1
0000000000000000001	00000008 00000000 00000000	1	1
0000000000000000002	00000004 00000000 00000000	1	1
_qsN_RECKEY	00000000 00000000 00000000 0000021 00000000 00000000 00000021	2	0
1	00000001 00000000 00000000	2	1
10	00000001 00000000 00000000	2	1
11	00000001 00000000 00000000	2	1
13	00000001 00000000 00000000	2	1
14	00000001 00000000 00000000	2	1
16	00000001 00000000 00000000	2	1
17	00000001 00000000 00000000	2	1
18	00000001 00000000 00000000	2	1
19	00000001 00000000 00000000	2	1
2	00000001 00000000 00000000	2	1
20	00000001 00000000 00000000	2	1
21	00000001 00000000 00000000	2	1
22	00000001 00000000 00000000	2	1
23	00000001 00000000 00000000	2	1
3	00000001 00000000 00000000	2	1
4	00000001 00000000 00000000	2	1
5	00000001 00000000 00000000	2	1
6	00000001 00000000 00000000	2	1
7	00000001 00000000 00000000	2	1
8	00000001 00000000 00000000	2	1
9	00000001 00000000 00000000	2	1
_qsM_FNAME	00000000 00000000 00000000 00000021 00000000 00000000 00000022	12	0

Figure 1-376 Create J15_FREQ_CUSTOMER_MASTER 2/2

J15_Undup_MatchSpec_CUSTOMER

In this step, we generate a match specification for an Unduplicate match stage using as input the match frequency data that was created in the job "J15_FREQ_CUSTOMER_MASTER" on page 316. The specification included two passes:

- The first pass blocked on the primary (last) name and a phonetic encoding (NYSIIS) of the address.
- The second pass blocked on the five digit ZIP code and a phonetic encoding (NYSIIS) of the address.

Note: We repeated the phonetic encoding (NYSIIS) of the address as a blocking variable in the second pass to reduce the potentially large block size that could arise out of choosing a blocking variable only involving the ZIP code. While it was not an issue in our data, it is likely to be an issue in a real-world environment.

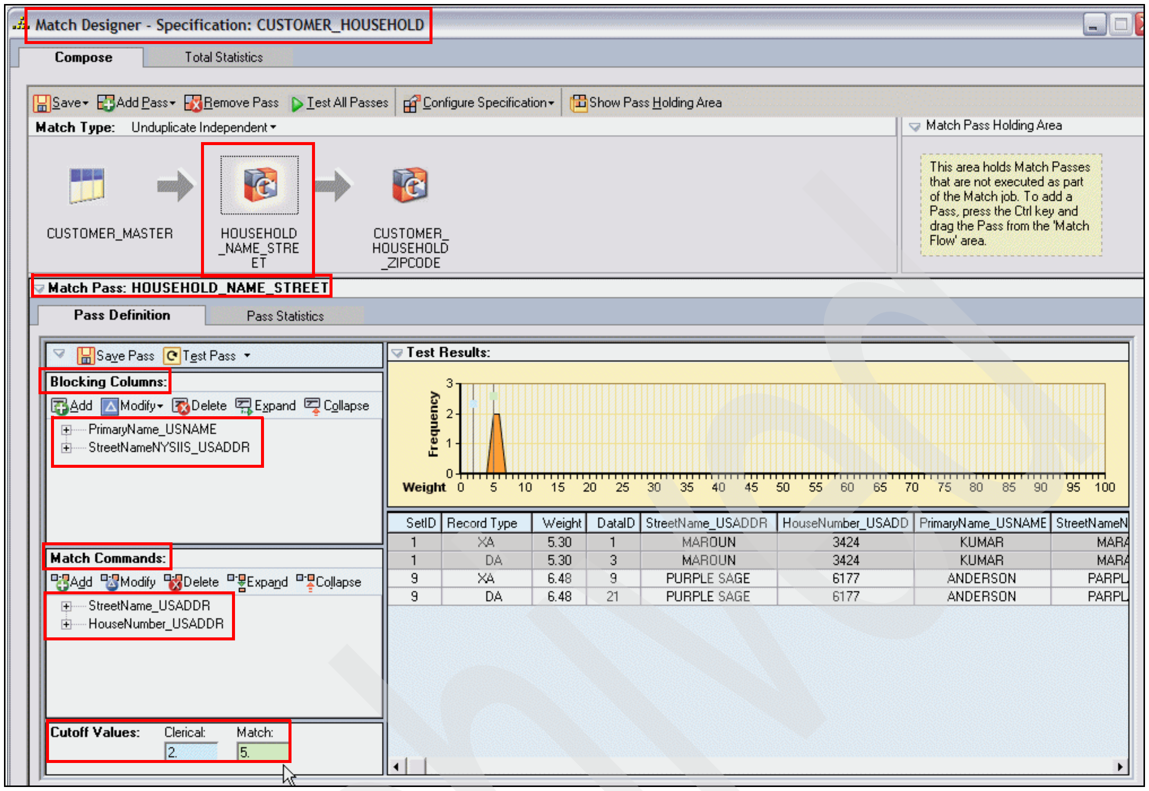
Because the creation of this match specification is very similar to that described in “J10_Undup_MatchSpec_STAN_CUSTOMER” on page 269, we do not repeat the steps that are involved here.

However, a summary of the passes are as follows:

- HOUSEHOLD_NAME_STREET pass (Figure 1-377 on page 320) with the blocking columns, match commands, cutoff values, and test results.
- CUSTOMER_HOUSEHOLD_ZIPCODE pass (Figure 1-378 on page 321) with the blocking columns, match commands, cutoff values, and test results.
- Total Statistics of the two passes in Figure 1-379 on page 322.

  Of particular interest is the statistic OVERFLOW blocks—a non-zero value indicates the need to increase the block size or define more restrictive blocking columns.

The test of the CUSTOMER_HOUSEHOLD match specification with representative sample data appears to deliver results that are accurate. This specification can then be used in an Unduplicate match stage as described in “J16_UNDUP_IND_MATCH_CUSTOMER” on page 322.



Match Type: Unduplicate Independent

This area holds Match Passes that are not executed as part of the Match job. To add a Pass, press the Ctrl key and drag the Pass from the 'Match Flow' area.

Match Pass: HOUSEHOLD_NAME_STREET

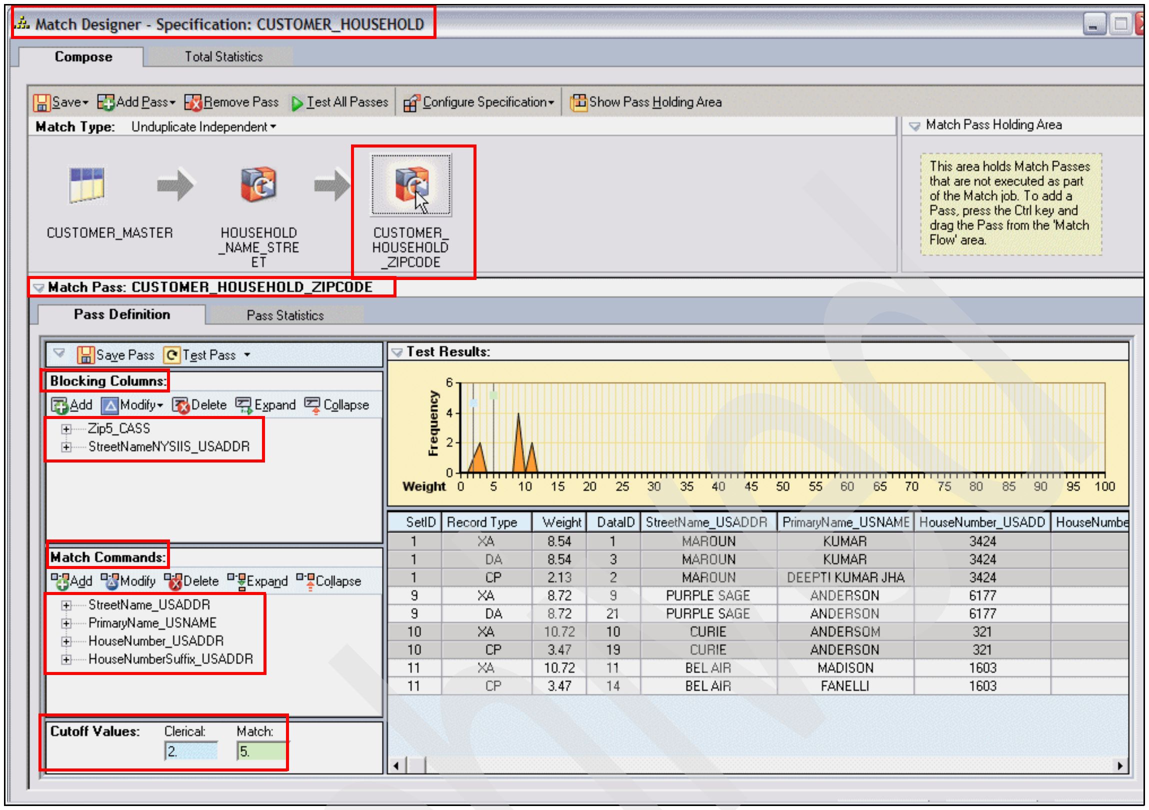
Blocking Columns:
- PrimaryName_USNAME
- StreetNameNYSIIS_USADDR

Match Commands:
- StreetName_USADDR
- HouseNumber_USADDR

Cutoff Values: Clerical: 2. Match: 5.

SetID	Record Type	Weight	DataID	StreetName_USADDR	HouseNumber_USADD	PrimaryName_USNAME	StreetNameN
1	XA	5.30	1	MAROUN	3424	KUMAR	MARA
1	DA	5.30	3	MAROUN	3424	KUMAR	MARA
9	XA	6.48	9	PURPLE SAGE	6177	ANDERSON	PARPL
9	DA	6.48	21	PURPLE SAGE	6177	ANDERSON	PARPL

Figure 1-377 Create J15_Undup_MatchSpec_CUSTOMER job 1/3



Match Type: Unduplicate Independent

This area holds Match Passes that are not executed as part of the Match job. To add a Pass, press the Ctrl key and drag the Pass from the 'Match Flow' area.

Match Pass: CUSTOMER_HOUSEHOLD_ZIPCODE

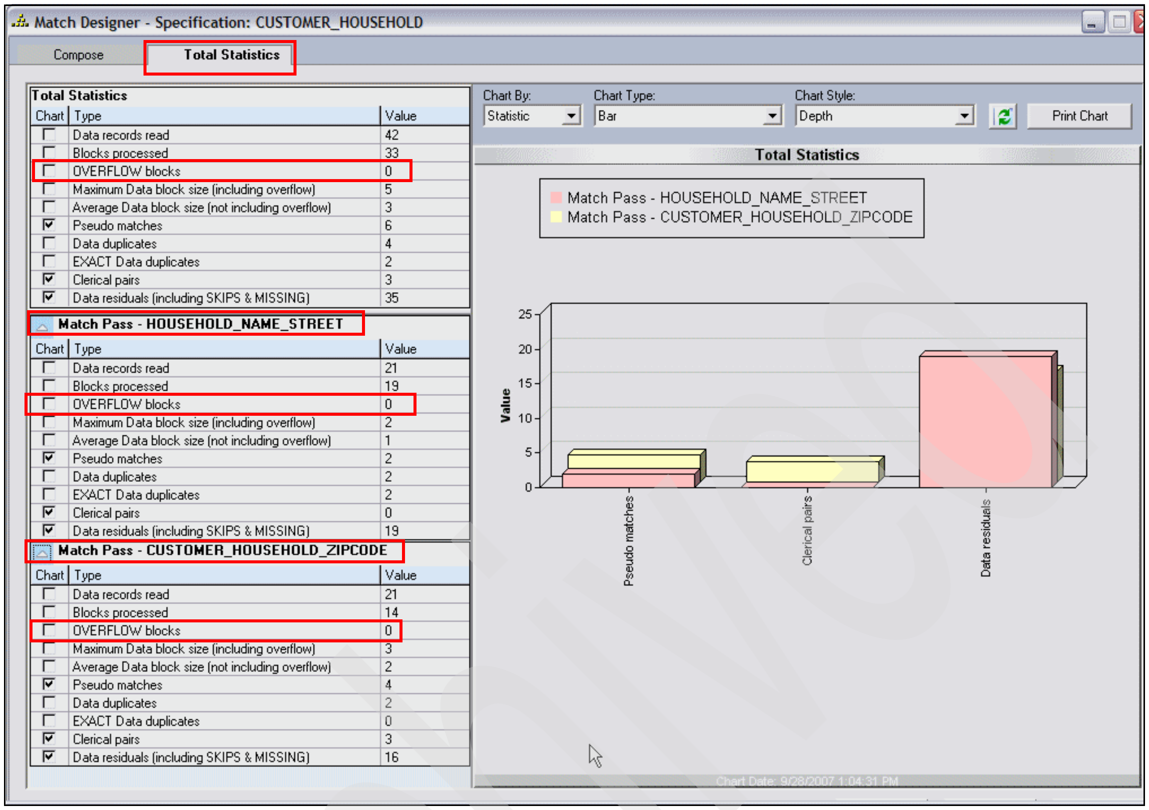
Blocking Columns:
- Zip5_CASS
- StreetNameNYSIIS_USADDR

Match Commands:
- StreetName_USADDR
- PrimaryName_USNAME
- HouseNumber_USADDR
- HouseNumberSuffix_USADDR

Cutoff Values: Clerical: 2. Match: 5.

SetID	Record Type	Weight	DataID	StreetName_USADDR	PrimaryName_USNAME	HouseNumber_USADD	HouseNumbe
1	XA	8.54	1	MAROUN	KUMAR	3424	
1	DA	8.54	3	MAROUN	KUMAR	3424	
1	CP	2.13	2	MAROUN	DEEPTI KUMAR JHA	3424	
9	XA	8.72	9	PURPLE SAGE	ANDERSON	6177	
9	DA	8.72	21	PURPLE SAGE	ANDERSON	6177	
10	XA	10.72	10	CURIE	ANDERSOM	321	
10	CP	3.47	19	CURIE	ANDERSON	321	
11	XA	10.72	11	BEL AIR	MADISON	1603	
11	CP	3.47	14	BEL AIR	FANELLI	1603	

Figure 1-378 Create J15_Undup_MatchSpec_CUSTOMER job 2/3



Total Statistics

Chart	Type	Value
☐	Data records read	42
☐	Blocks processed	33
☐	OVERFLOW blocks	0
☐	Maximum Data block size (including overflow)	5
☐	Average Data block size (not including overflow)	3
☑	Pseudo matches	6
☐	Data duplicates	4
☐	EXACT Data duplicates	2
☑	Clerical pairs	3
☑	Data residuals (including SKIPS & MISSING)	35

Match Pass - HOUSEHOLD_NAME_STREET

Chart	Type	Value
☐	Data records read	21
☐	Blocks processed	19
☐	OVERFLOW blocks	0
☐	Maximum Data block size (including overflow)	2
☐	Average Data block size (not including overflow)	1
☑	Pseudo matches	2
☐	Data duplicates	2
☐	EXACT Data duplicates	2
☑	Clerical pairs	0
☑	Data residuals (including SKIPS & MISSING)	19

Match Pass - CUSTOMER_HOUSEHOLD_ZIPCODE

Chart	Type	Value
☐	Data records read	21
☐	Blocks processed	14
☐	OVERFLOW blocks	0
☐	Maximum Data block size (including overflow)	3
☐	Average Data block size (not including overflow)	2
☑	Pseudo matches	4
☐	Data duplicates	2
☐	EXACT Data duplicates	0
☑	Clerical pairs	3
☑	Data residuals (including SKIPS & MISSING)	16

Figure 1-379 Create J15_Undup_MatchSpec_CUSTOMER job 3/3

J16_UNDUP_IND_MATCH_CUSTOMER

In this step, we determine whether there were duplicates in the credit card file using the Unduplicate stage with the match specification and match frequency information that was created in steps “J15_Undup_MatchSpec_CUSTOMER” on page 318 and “J15_FREQ_CUSTOMER_MASTER” on page 316 respectively. As before, the output is matched records (merge of master and duplicates using a Funnel stage), records for clerical review, and residuals (records that do not match).

Because the creation of this Unduplicate stage is very similar to that described in “J11_UNDUP_DEP_MATCH_CUSTOMER” on page 282, we do not repeat the steps that are involved here. However, the key differences are as follows:

- Figure 1-380 on page 324 shows the various stages that are used in this job, including the data set that was created in “J14A_CUSTOMER_MASTER” on page 316, the match frequency data set that was created in “J15_FREQ_CUSTOMER_MASTER” on page 316, an Unduplicate stage, a

Funnel stage to merge master and duplicate records. Three data sets are created:

  – One data set contain the merged master and duplicates by the Funnel stage
  – The other data sets contain the clerical and residual records as the output of the Unduplicate stage.

  We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object created earlier.

- In the UNDUP_IND - Unduplicate Match Stage window shown in Figure 1-381 on page 324, select the Match Specification CUSTOMER_HOUSEHOLD. Select the Match Type Independent in order to find groups of all records (for household information) across multiple match passes as described in Table 1-11 on page 106.
- After saving, compiling, and running this job j16_UNDUP_IND_MATCH_CUSTOMER (Figure 1-382 on page 325 shows zero rows written to the CUSTOMER_UNDUP_IND_CLERICALS), view the content of the two data set objects as shown in Figure 1-383 on page 325 through Figure 1-388 on page 327.

  Out of a total of 21 records in the input to this process:

  – Nine (9) records are masters and duplicates
  – Zero records (0) are for clerical review
  – Remaining twelve (12) records are residuals

The next step is to create a single credit card customer file with household information, as described in “J17_CUSTOMER_MASTER_WITH_HOUSEHOLD” on page 327.

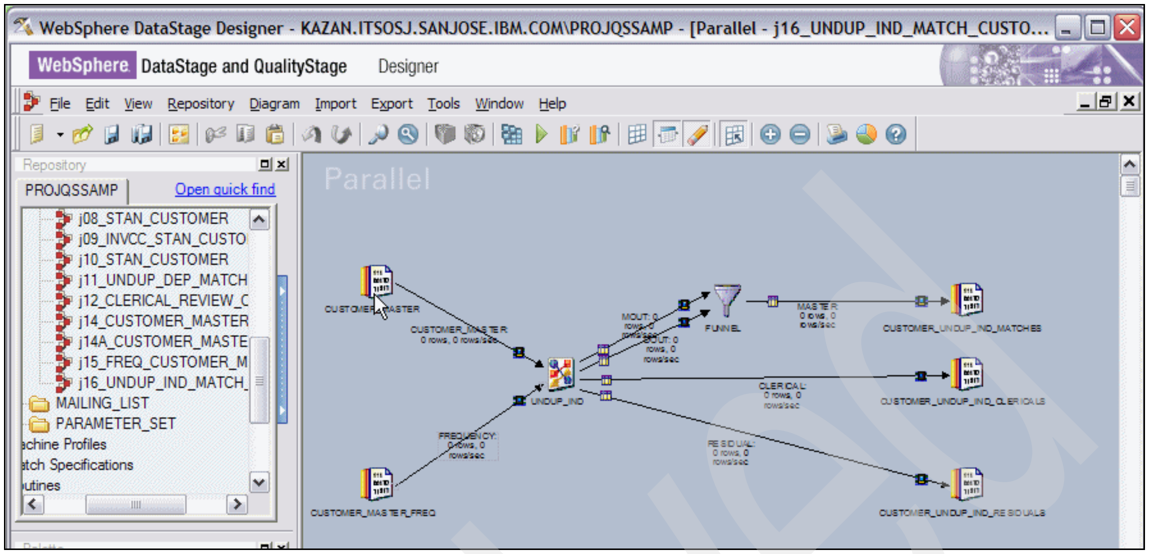
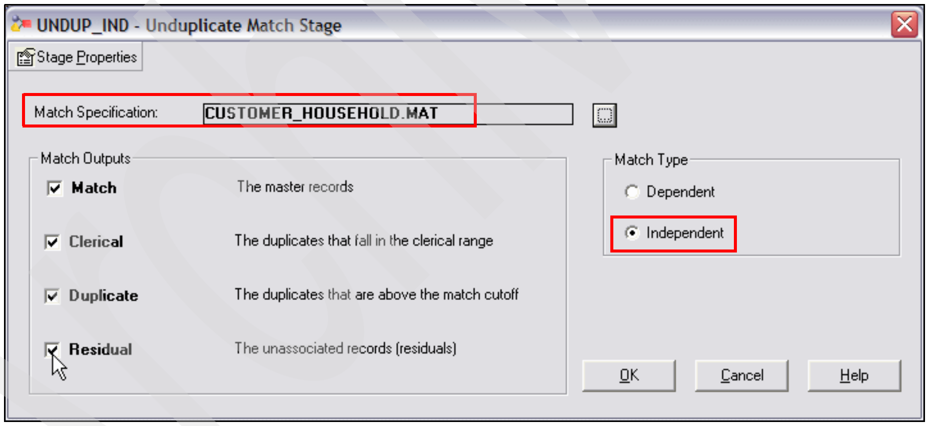

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j16_UNDUP_IND_MATCH_CUSTO...
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Repository
PROJQSSAMP
Open quick find
j08_STAN_CUSTOMER
j09_INVCC_STAN_CUSTO
j10_STAN_CUSTOMER
j11_UNDUP_DEP_MATCH
j12_CLERICAL_REVIEW_C
j14_CUSTOMER_MASTER
j14A_CUSTOMER_MASTE
j15_FREQ_CUSTOMER_M
j16_UNDUP_IND_MATCH_
MAILING_LIST
PARAMETER_SET
achine Profiles
atch Specifications
utines
Parallel
CUSTOMER_MASTER
CUSTOMER_MASTER 0 rows, 0 rows/sec
UNDUP_IND
FREQUENCY 0 rows, 0 rows/sec
CUSTOMER_MASTER_FREQ
MOUT 0 rows, 0 rows/sec
DOUT 0 rows, 0 rows/sec
FUNNEL
MASTER 0 rows, 0 rows/sec
CUSTOMER_UNDUP_IND_MATCHES
CLERICAL 0 rows, 0 rows/sec
CUSTOMER_UNDUP_IND_CLERICALS
RESIDUAL 0 rows, 0 rows/sec
CUSTOMER_UNDUP_IND_RESIDUALS

Figure 1-380   Create J16_UNDUP_IND_MATCH_CUSTOMER job 1/9


UNDUP_IND - Unduplicate Match Stage

Stage Properties

Match Specification: CUSTOMER_HOUSEHOLD.MAT

Match Outputs

☑ **Match**	The master records
☑ **Clerical**	The duplicates that fall in the clerical range
☑ **Duplicate**	The duplicates that are above the match cutoff
☑ **Residual**	The unassociated records (residuals)

Match Type

Dependent [✓ Independent]

Figure 1-381   Create J16_UNDUP_IND_MATCH_CUSTOMER job 2/9

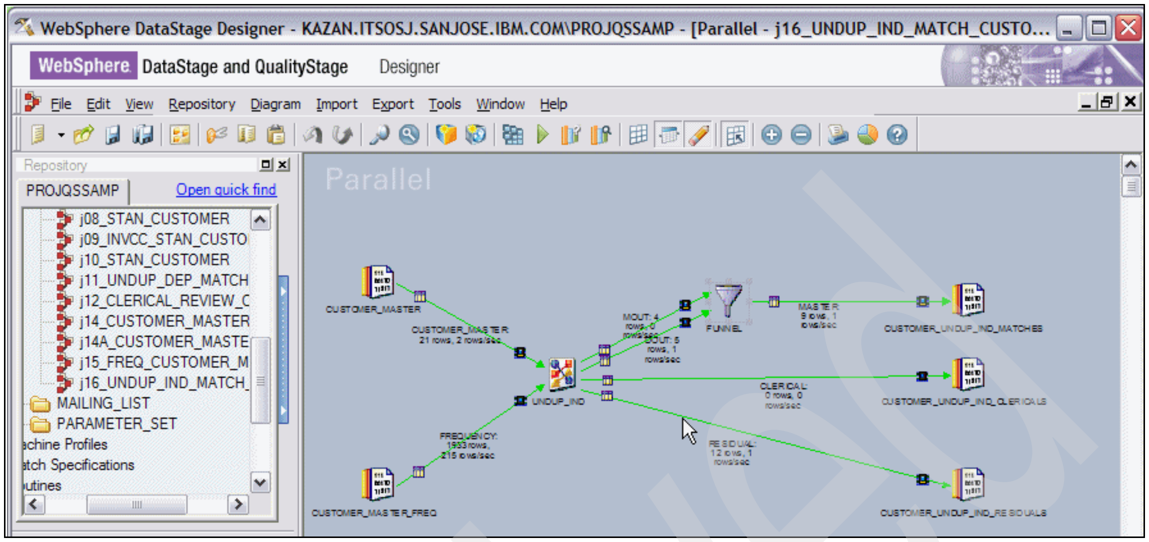
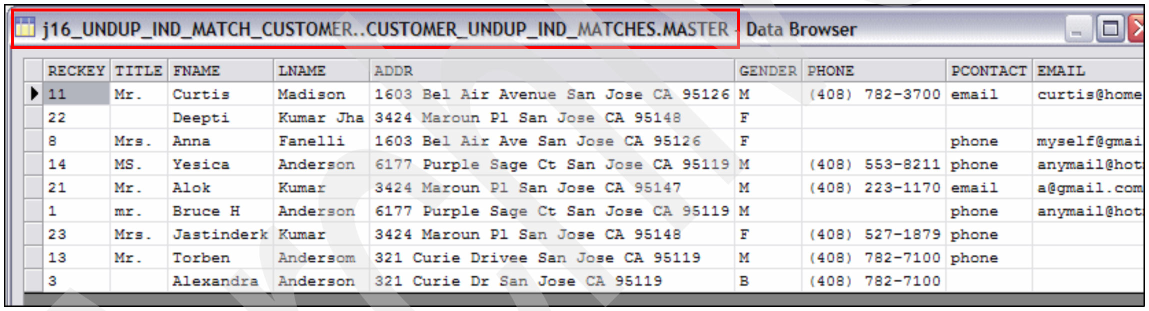
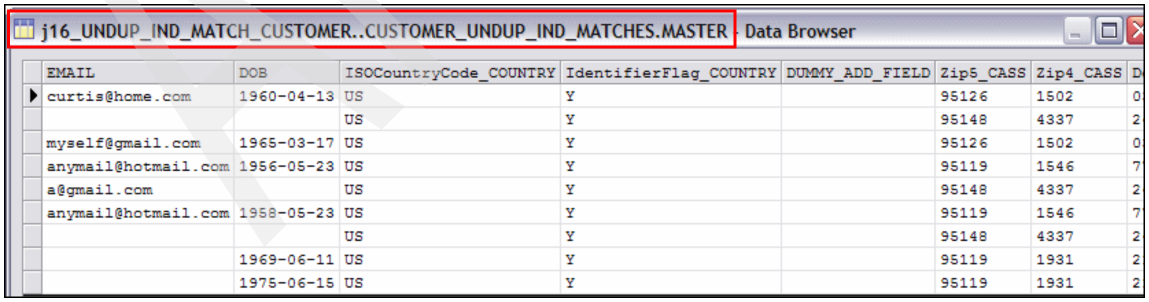

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j16_UNDUP_IND_MATCH_CUSTO...
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Repository
PROJQSSAMP
Open quick find
j08_STAN_CUSTOMER
j09_INVCC_STAN_CUSTO
j10_STAN_CUSTOMER
j11_UNDUP_DEP_MATCH
j12_CLERICAL_REVIEW_C
j14_CUSTOMER_MASTER
j14A_CUSTOMER_MASTE
j15_FREQ_CUSTOMER_M
j16_UNDUP_IND_MATCH_
MAILING_LIST
PARAMETER_SET
achine Profiles
atch Specifications
utines
Parallel
CUSTOMER_MASTER
CUSTOMER_MASTER: 21 rows, 2 rows/sec
UNDUP_IND
FREQUENCY: 1933 rows, 215 rows/sec
CUSTOMER_MASTER_FREQ
MOUT: 4 rows, 0 rows/sec
DOUT: 5 rows, 1 rows/sec
FUNNEL
MASTER: 9 rows, 1 rows/sec
CUSTOMER_UNDUP_IND_MATCHES
CLERICAL: 0 rows, 0 rows/sec
CUSTOMER_UNDUP_IND_CLERICALS
RESIDUAL: 12 rows, 1 rows/sec
CUSTOMER_UNDUP_IND_RESIDUALS

Figure 1-382 Create J16_UNDUP_IND_MATCH_CUSTOMER job 3/9


j16_UNDUP_IND_MATCH_CUSTOMER..CUSTOMER_UNDUP_IND_MATCHES.MASTER - Data Browser

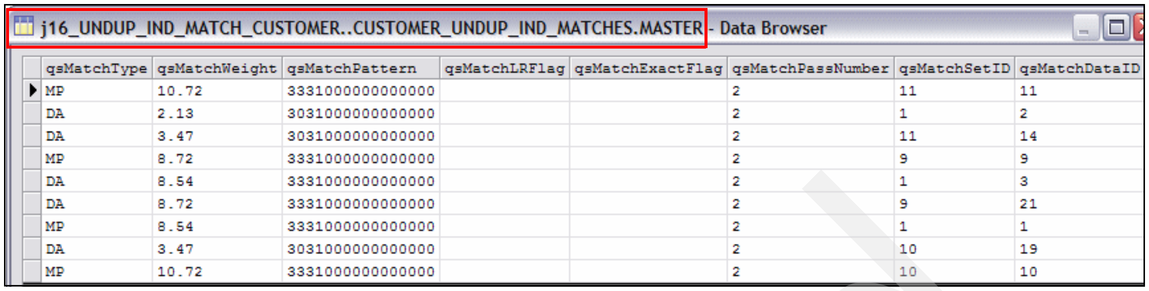
RECKEY	TITLE	FNAME	LNAME	ADDR	GENDER	PHONE	PCONTACT	EMAIL
11	Mr.	Curtis	Madison	1603 Bel Air Avenue San Jose CA 95126	M	(408) 782-3700	email	curtis@home
22		Deepti	Kumar Jha	3424 Maroun Pl San Jose CA 95148	F			
8	Mrs.	Anna	Fanelli	1603 Bel Air Ave San Jose CA 95126	F		phone	myself@gmai
14	MS.	Yesica	Anderson	6177 Purple Sage Ct San Jose CA 95119	M	(408) 553-8211	phone	anymail@hot
21	Mr.	Alok	Kumar	3424 Maroun Pl San Jose CA 95147	M	(408) 223-1170	email	a@gmail.com
1	mr.	Bruce H	Anderson	6177 Purple Sage Ct San Jose CA 95119	M		phone	anymail@hot
23	Mrs.	Jastinderk	Kumar	3424 Maroun Pl San Jose CA 95148	F	(408) 527-1879	phone	
13	Mr.	Torben	Andersom	321 Curie Drivee San Jose CA 95119	M	(408) 782-7100	phone	
3		Alexandra	Anderson	321 Curie Dr San Jose CA 95119	B	(408) 782-7100		

Figure 1-383 Create J16_UNDUP_IND_MATCH_CUSTOMER job 4/9


j16_UNDUP_IND_MATCH_CUSTOMER..CUSTOMER_UNDUP_IND_MATCHES.MASTER - Data Browser

EMAIL	DOB	ISOCountryCode_COUNTRY	IdentifierFlag_COUNTRY	DUMMY_ADD_FIELD	Zip5_CASS	Zip4_CASS	D
curtis@home.com	1960-04-13	US	Y		95126	1502	0
		US	Y		95148	4337	2
myself@gmail.com	1965-03-17	US	Y		95126	1502	0
anymail@hotmail.com	1956-05-23	US	Y		95119	1546	7
a@gmail.com		US	Y		95148	4337	2
anymail@hotmail.com	1958-05-23	US	Y		95119	1546	7
		US	Y		95148	4337	2
	1969-06-11	US	Y		95119	1931	2
	1975-06-15	US	Y		95119	1931	2

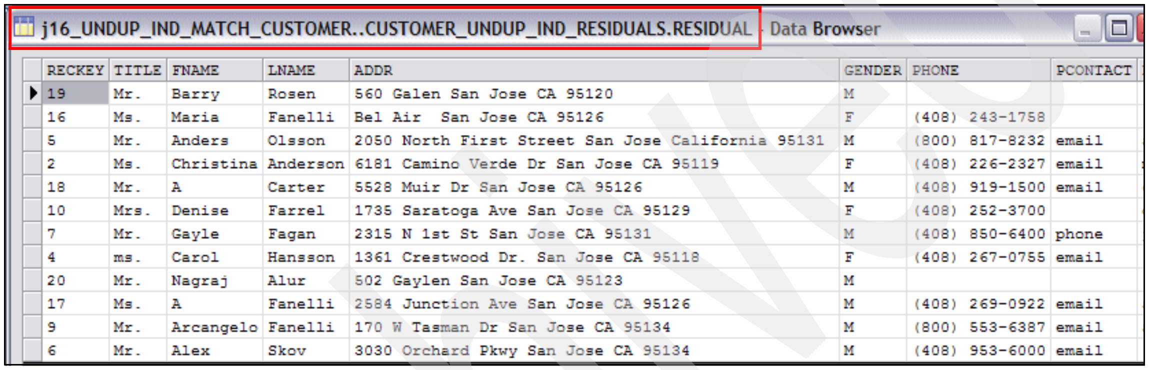
Figure 1-384 Create J16_UNDUP_IND_MATCH_CUSTOMER job 5/9



j16_UNDUP_IND_MATCH_CUSTOMER..CUSTOMER_UNDUP_IND_MATCHES.MASTER - Data Browser

qsMatchType	qsMatchWeight	qsMatchPattern	qsMatchLRFlag	qsMatchExactFlag	qsMatchPassNumber	qsMatchSetID	qsMatchDataID
MP	10.72	3331000000000000			2	11	11
DA	2.13	3031000000000000			2	1	2
DA	3.47	3031000000000000			2	11	14
MP	8.72	3331000000000000			2	9	9
DA	8.54	3331000000000000			2	1	3
DA	8.72	3331000000000000			2	9	21
MP	8.54	3331000000000000			2	1	1
DA	3.47	3031000000000000			2	10	19
MP	10.72	3331000000000000			2	10	10

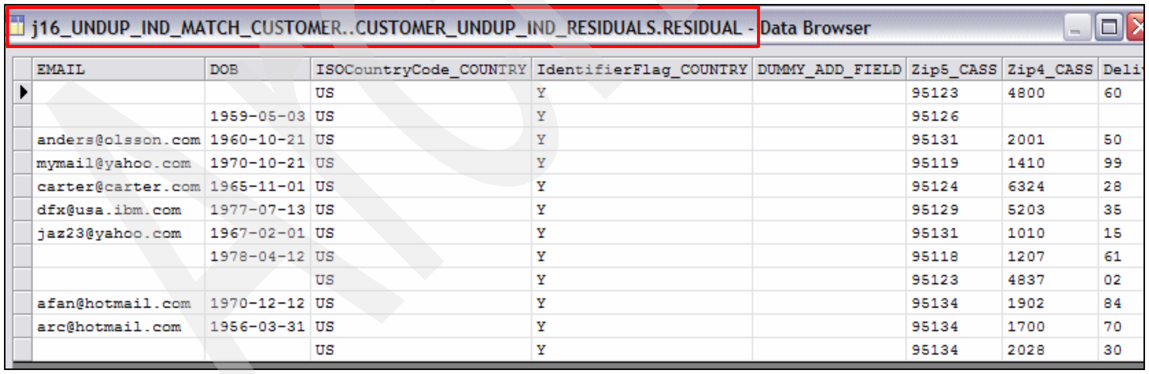
Figure 1-385   Create J16_UNDUP_IND_MATCH_CUSTOMER job 6/9



j16_UNDUP_IND_MATCH_CUSTOMER..CUSTOMER_UNDUP_IND_RESIDUALS.RESIDUAL - Data Browser

RECKEY	TITLE	FNAME	LNAME	ADDR	GENDER	PHONE	PCONTACT
19	Mr.	Barry	Rosen	560 Galen San Jose CA 95120	M		
16	Ms.	Maria	Fanelli	Bel Air San Jose CA 95126	F	(408) 243-1758	
5	Mr.	Anders	Olsson	2050 North First Street San Jose California 95131	M	(800) 817-8232	email
2	Ms.	Christina	Anderson	6181 Camino Verde Dr San Jose CA 95119	F	(408) 226-2327	email
18	Mr.	A	Carter	5528 Muir Dr San Jose CA 95126	M	(408) 919-1500	email
10	Mrs.	Denise	Farrel	1735 Saratoga Ave San Jose CA 95129	F	(408) 252-3700	
7	Mr.	Gayle	Fagan	2315 N 1st St San Jose CA 95131	M	(408) 850-6400	phone
4	ms.	Carol	Hansson	1361 Crestwood Dr. San Jose CA 95118	F	(408) 267-0755	email
20	Mr.	Nagraj	Alur	502 Gaylen San Jose CA 95123	M		
17	Ms.	A	Fanelli	2584 Junction Ave San Jose CA 95126	M	(408) 269-0922	email
9	Mr.	Arcangelo	Fanelli	170 W Tasman Dr San Jose CA 95134	M	(800) 553-6387	email
6	Mr.	Alex	Skov	3030 Orchard Pkwy San Jose CA 95134	M	(408) 953-6000	email

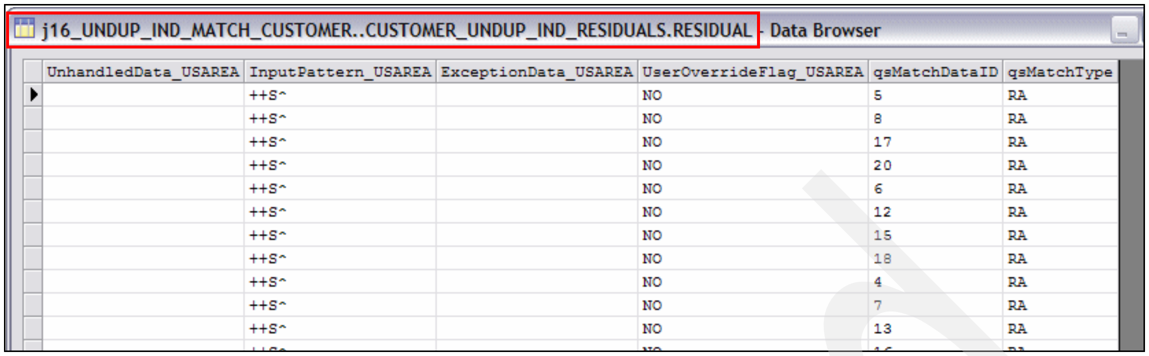
Figure 1-386   Create J16_UNDUP_IND_MATCH_CUSTOMER job 7/9



j16_UNDUP_IND_MATCH_CUSTOMER..CUSTOMER_UNDUP_IND_RESIDUALS.RESIDUAL - Data Browser

EMAIL	DOB	ISOCountryCode_COUNTRY	IdentifierFlag_COUNTRY	DUMMY_ADD_FIELD	Zip5_CASS	Zip4_CASS	Deli
		US	Y		95123	4800	60
	1959-05-03	US	Y		95126		
anders@olsson.com	1960-10-21	US	Y		95131	2001	50
mymail@yahoo.com	1970-10-21	US	Y		95119	1410	99
carter@carter.com	1965-11-01	US	Y		95124	6324	28
dfx@usa.ibm.com	1977-07-13	US	Y		95129	5203	35
jaz23@yahoo.com	1967-02-01	US	Y		95131	1010	15
	1978-04-12	US	Y		95118	1207	61
		US	Y		95123	4837	02
afan@hotmail.com	1970-12-12	US	Y		95134	1902	84
arc@hotmail.com	1956-03-31	US	Y		95134	1700	70
		US	Y		95134	2028	30

Figure 1-387   Create J16_UNDUP_IND_MATCH_CUSTOMER job 8/9


j16_UNDUP_IND_MATCH_CUSTOMER..CUSTOMER_UNDUP_IND_RESIDUALS.RESIDUAL - Data Browser

UnhandledData_USAREA	InputPattern_USAREA	ExceptionData_USAREA	UserOverrideFlag_USAREA	qsMatchDataID	qsMatchType
	++S^		NO	5	RA
	++S^		NO	8	RA
	++S^		NO	17	RA
	++S^		NO	20	RA
	++S^		NO	6	RA
	++S^		NO	12	RA
	++S^		NO	15	RA
	++S^		NO	18	RA
	++S^		NO	4	RA
	++S^		NO	7	RA
	++S^		NO	13	RA

Figure 1-388 Create J16_UNDUP_IND_MATCH_CUSTOMER job 9/9

J17_CUSTOMER_MASTER_WITH_HOUSEHOLD

In this step, we create a clean master of credit card customers by merging the master and residual records into a data set using the FUNNEL stage. We add a household ID (using a Transformer stage) to these records in the output with a value (from the qsMatchSetID) when a record belongs to a household and a zero when it does not belong to a household.

Because the steps involved using the FUNNEL and Transformer stages that were covered earlier, we do not repeat the steps that are involved here. However, some of the key points of interest are as follows:

1. Figure 1-389 on page 328 shows the various stages that are used in this job, including the three input data sets (CUSTOMER_UNDUP_IND_MATCHES that was created in “J16_UNDUP_IND_MATCH_CUSTOMER” on page 322, CUSTOMER_UNDUP_IND_CLERICAL that was created in “J16_UNDUP_IND_MATCH_CUSTOMER” on page 322, and CUSTOMER_UNDUP_IND_RESIDUALS that was created in “J16_UNDUP_IND_MATCH_CUSTOMER” on page 322), two FUNNEL stages, two Transformer stages (each adds a household identifier), and an output Data Set stage. We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object created earlier.
2. The ADD_HOUSEHOLD_ID - Transformer Stage shows the addition of a column HOUSEHOLD_ID (Figure 1-390 on page 329), which has the value stored in the column qsMatchSetID in the Derivation Substitution column.
3. The ADD_BLANK_HOUSEHOLD - Transformer Stage (Figure 1-391 on page 330) shows the addition of a column HOUSEHOLD_ID, which has the value zero in the Derivation Substitution column.
4. After saving, compiling, and running this jobJ17_CUSTOMER_MASTER_WITH_HOUSEHOLD (Figure 1-392 on page 331), view the content of the

CUSTOMER_MASTER_WITH_HOUSEHOLD data set object as shown in Figure 1-393 on page 331 through Figure 1-395 on page 332.

This report shows the cleansed credit card customer file with household information added.

Now, proceed to obtain match frequency information about all the columns of this file for use in subsequent matching stages, as described in "J18_MATCHFREQ_CUSTOMER_WITH_HOUSEHOLD" on page 332.

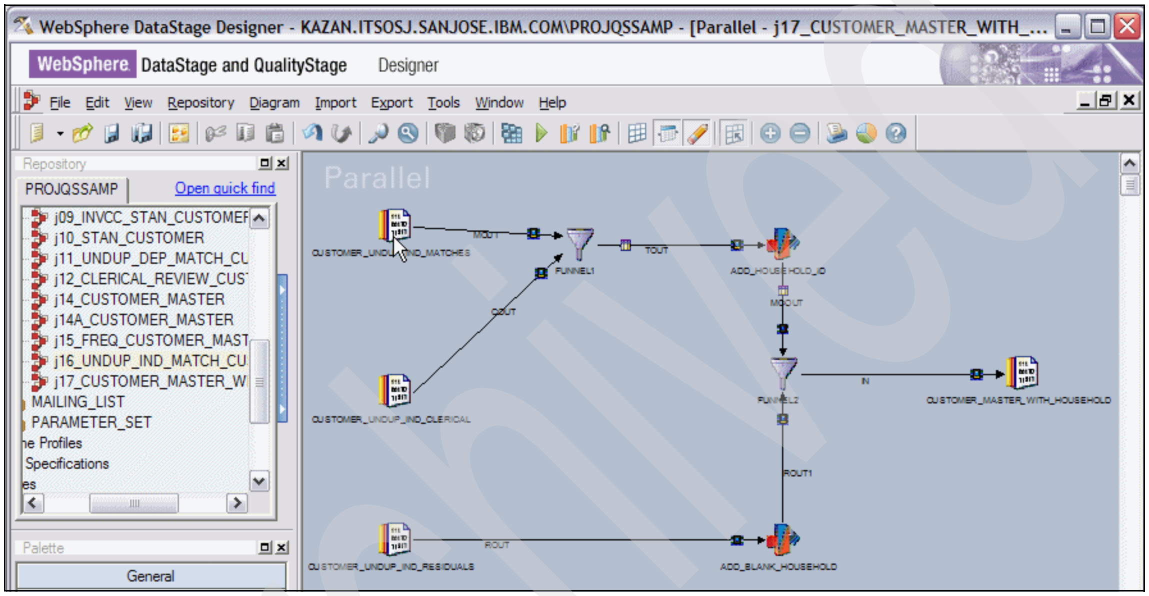
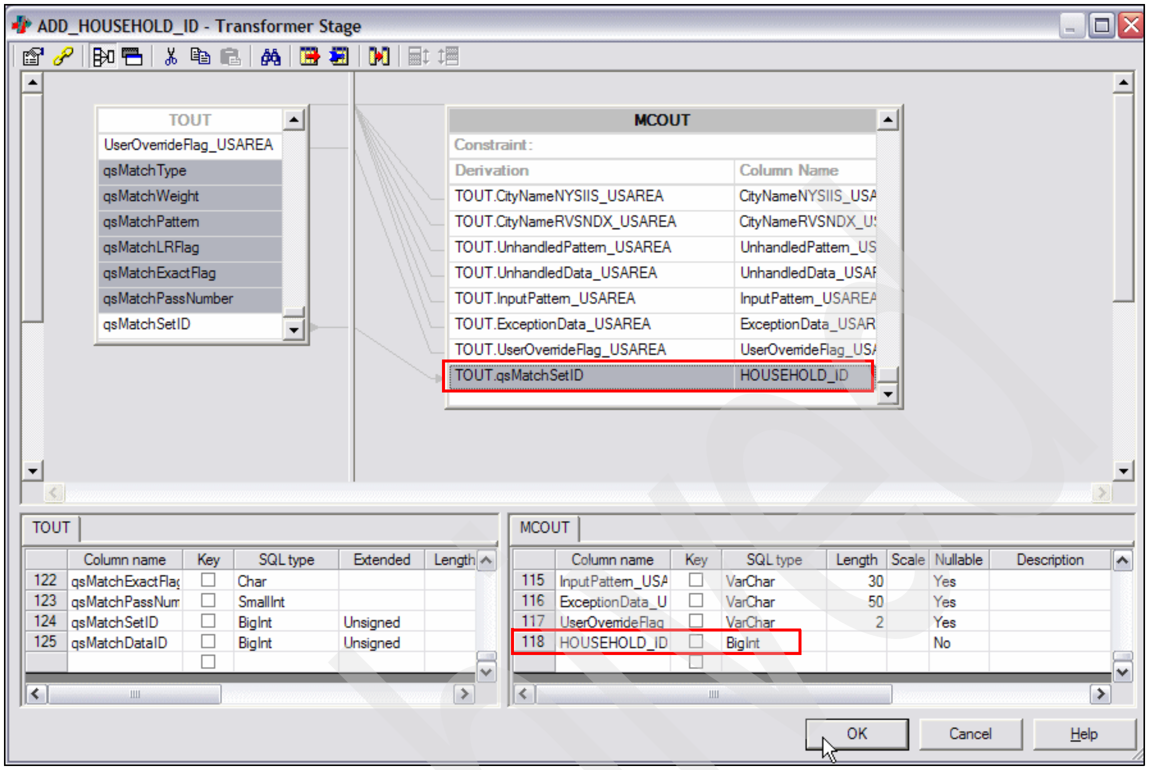


Figure 1-389 Create J17_CUSTOMER_MASTER_WITH_HOUSEHOLD job 1/7


ADD_HOUSEHOLD_ID - Transformer Stage

TOUT
UserOverrideFlag_USAREA
qsMatchType
qsMatchWeight
qsMatchPattern
qsMatchLRFlag
qsMatchExactFlag
qsMatchPassNumber
qsMatchSetID

MCOUT

Constraint:

Derivation	Column Name
TOUT.CityNameNYSIIS_USAREA	CityNameNYSIIS_USA
TOUT.CityNameRVSNDX_USAREA	CityNameRVSNDX_U
TOUT.UnhandledPattern_USAREA	UnhandledPattern_US
TOUT.UnhandledData_USAREA	UnhandledData_USAR
TOUT.InputPattern_USAREA	InputPattern_USAREA
TOUT.ExceptionData_USAREA	ExceptionData_USAR
TOUT.UserOverrideFlag_USAREA	UserOverrideFlag_US
TOUT.qsMatchSetID	HOUSEHOLD_ID

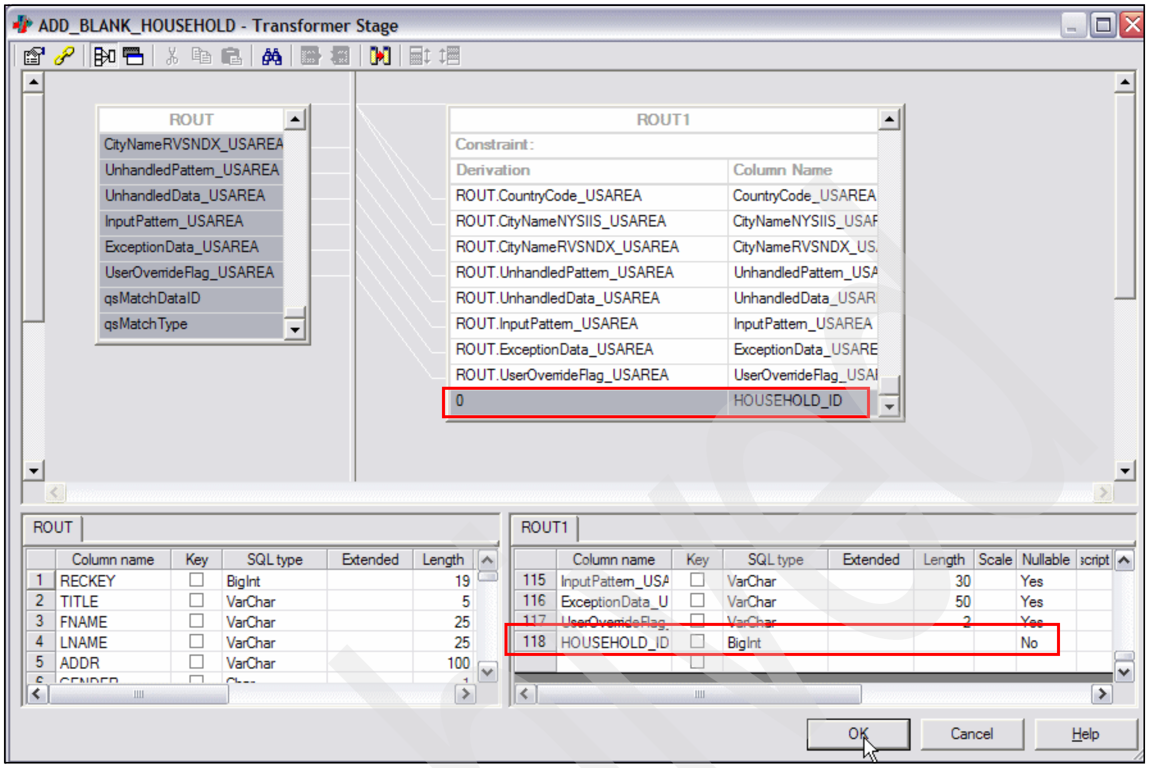
TOUT

	Column name	Key	SQL type	Extended	Length
122	qsMatchExactFlag	☐	Char		
123	qsMatchPassNum	☐	SmallInt		
124	qsMatchSetID	☐	BigInt	Unsigned	
125	qsMatchDataID	☐	BigInt	Unsigned	
		☐			

MCOUT

	Column name	Key	SQL type	Length	Scale	Nullable	Description
115	InputPattern_USA	☐	VarChar	30		Yes	
116	ExceptionData_U	☐	VarChar	50		Yes	
117	UserOverrideFlag	☐	VarChar	2		Yes	
118	HOUSEHOLD_ID	☐	BigInt			No	
		☐					

Figure 1-390 Create J17_CUSTOMER_MASTER_WITH_HOUSEHOLD job 2/7


ADD_BLANK_HOUSEHOLD - Transformer Stage

ROUT

- CityNameRVSNDX_USAREA
- UnhandledPattern_USAREA
- UnhandledData_USAREA
- InputPattern_USAREA
- ExceptionData_USAREA
- UserOverrideFlag_USAREA
- qsMatchDataID
- qsMatchType

ROUT1

Constraint:

Derivation	Column Name
ROUT.CountryCode_USAREA	CountryCode_USAREA
ROUT.CityNameNYSIIS_USAREA	CityNameNYSIIS_USAF
ROUT.CityNameRVSNDX_USAREA	CityNameRVSNDX_US.
ROUT.UnhandledPattern_USAREA	UnhandledPattern_USA
ROUT.UnhandledData_USAREA	UnhandledData_USAR
ROUT.InputPattern_USAREA	InputPattern_USAREA
ROUT.ExceptionData_USAREA	ExceptionData_USARE
ROUT.UserOverrideFlag_USAREA	UserOverrideFlag_USAI
0	HOUSEHOLD_ID

ROUT

	Column name	Key	SQL type	Extended	Length
1	RECKEY	☐	BigInt		19
2	TITLE	☐	VarChar		5
3	FNAME	☐	VarChar		25
4	LNAME	☐	VarChar		25
5	ADDR	☐	VarChar		100

ROUT1

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	script
115	InputPattern_USA	☐	VarChar		30		Yes	
116	ExceptionData_U	☐	VarChar		50		Yes	
117	UserOverrideFlag	☐	VarChar		2		Yes	
118	HOUSEHOLD_ID	☐	BigInt				No	

Figure 1-391 Create J17_CUSTOMER_MASTER_WITH_HOUSEHOLD job 3/7

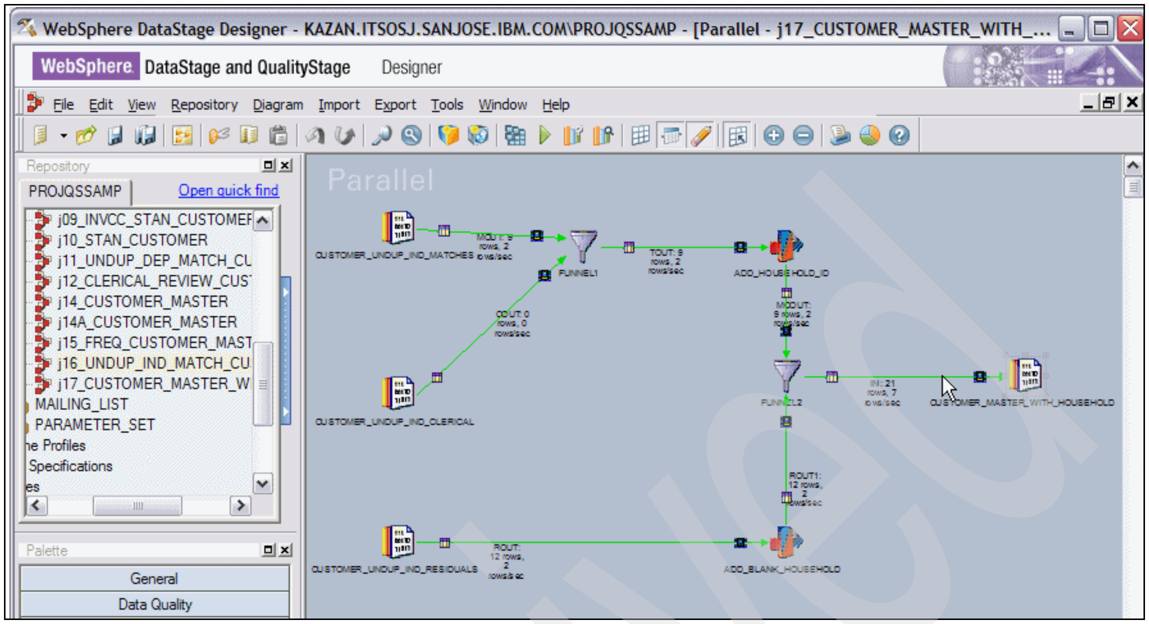
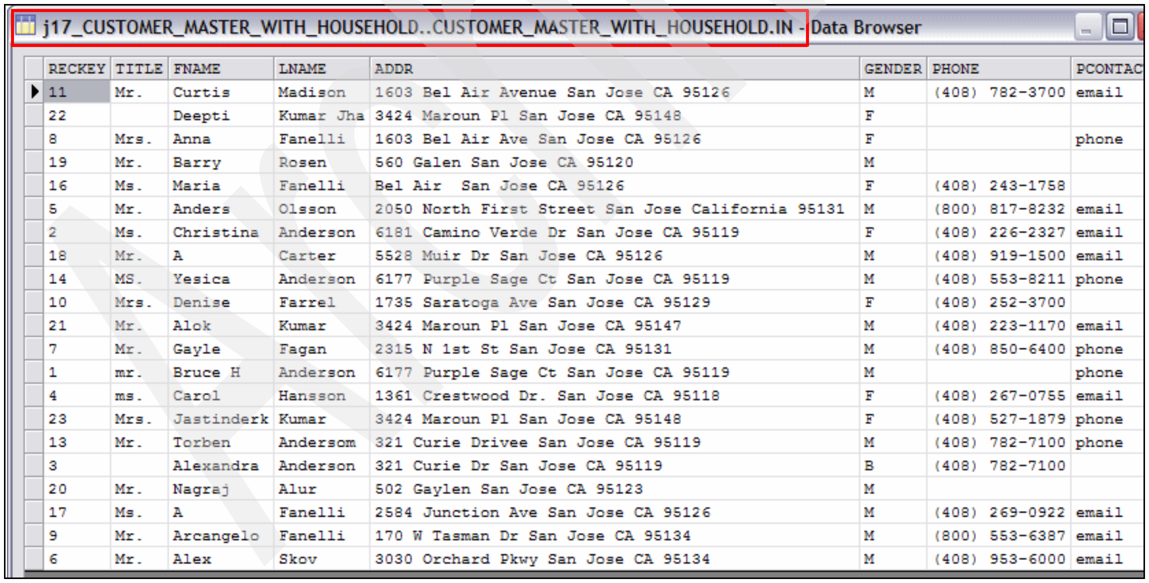
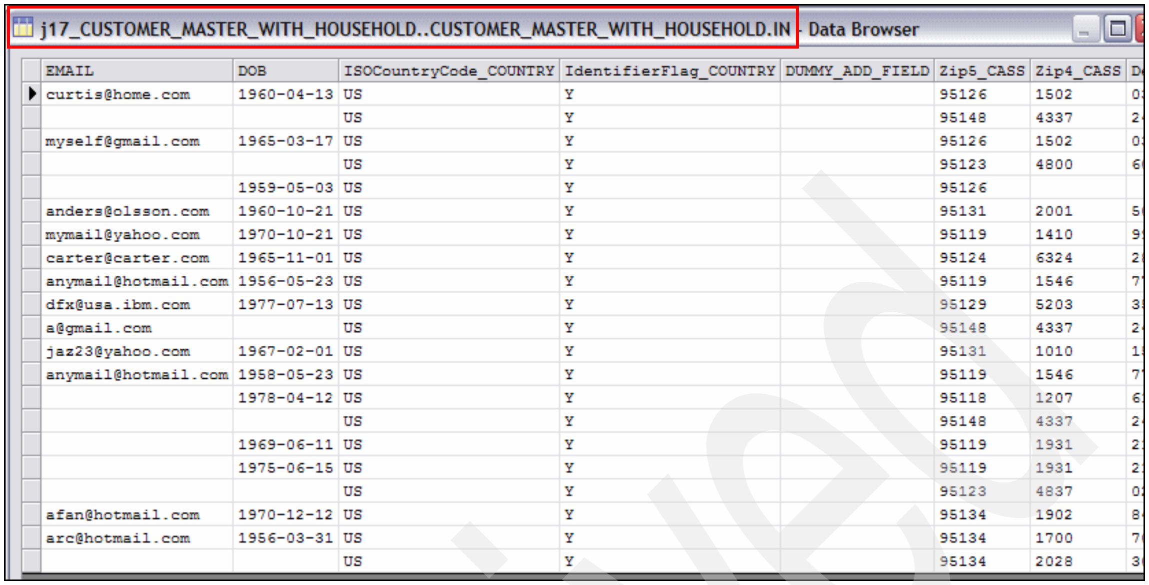


Figure 1-392 Create J17_CUSTOMER_MASTER_WITH_HOUSEHOLD job 4/7


j17_CUSTOMER_MASTER_WITH_HOUSEHOLD..CUSTOMER_MASTER_WITH_HOUSEHOLD.IN - Data Browser

RECKEY	TITLE	FNAME	LNAME	ADDR	GENDER	PHONE	PCONTAC
11	Mr.	Curtis	Madison	1603 Bel Air Avenue San Jose CA 95126	M	(408) 782-3700	email
22		Deepti	Kumar Jha	3424 Maroun Pl San Jose CA 95148	F		
8	Mrs.	Anna	Fanelli	1603 Bel Air Ave San Jose CA 95126	F		phone
19	Mr.	Barry	Rosen	560 Galen San Jose CA 95120	M		
16	Ms.	Maria	Fanelli	Bel Air San Jose CA 95126	F	(408) 243-1758	
5	Mr.	Anders	Olsson	2050 North First Street San Jose California 95131	M	(800) 817-8232	email
2	Ms.	Christina	Anderson	6181 Camino Verde Dr San Jose CA 95119	F	(408) 226-2327	email
18	Mr.	A	Carter	5528 Muir Dr San Jose CA 95126	M	(408) 919-1500	email
14	MS.	Yesica	Anderson	6177 Purple Sage Ct San Jose CA 95119	M	(408) 553-8211	phone
10	Mrs.	Denise	Farrel	1735 Saratoga Ave San Jose CA 95129	F	(408) 252-3700	
21	Mr.	Alok	Kumar	3424 Maroun Pl San Jose CA 95147	M	(408) 223-1170	email
7	Mr.	Gayle	Fagan	2315 N 1st St San Jose CA 95131	M	(408) 850-6400	phone
1	mr.	Bruce H	Anderson	6177 Purple Sage Ct San Jose CA 95119	M		phone
4	ms.	Carol	Hansson	1361 Crestwood Dr. San Jose CA 95118	F	(408) 267-0755	email
23	Mrs.	Jastinderk	Kumar	3424 Maroun Pl San Jose CA 95148	F	(408) 527-1879	phone
13	Mr.	Torben	Andersom	321 Curie Drivee San Jose CA 95119	M	(408) 782-7100	phone
3		Alexandra	Anderson	321 Curie Dr San Jose CA 95119	B	(408) 782-7100	
20	Mr.	Nagraj	Alur	502 Gaylen San Jose CA 95123	M		
17	Ms.	A	Fanelli	2584 Junction Ave San Jose CA 95126	M	(408) 269-0922	email
9	Mr.	Arcangelo	Fanelli	170 W Tasman Dr San Jose CA 95134	M	(800) 553-6387	email
6	Mr.	Alex	Skov	3030 Orchard Pkwy San Jose CA 95134	M	(408) 953-6000	email

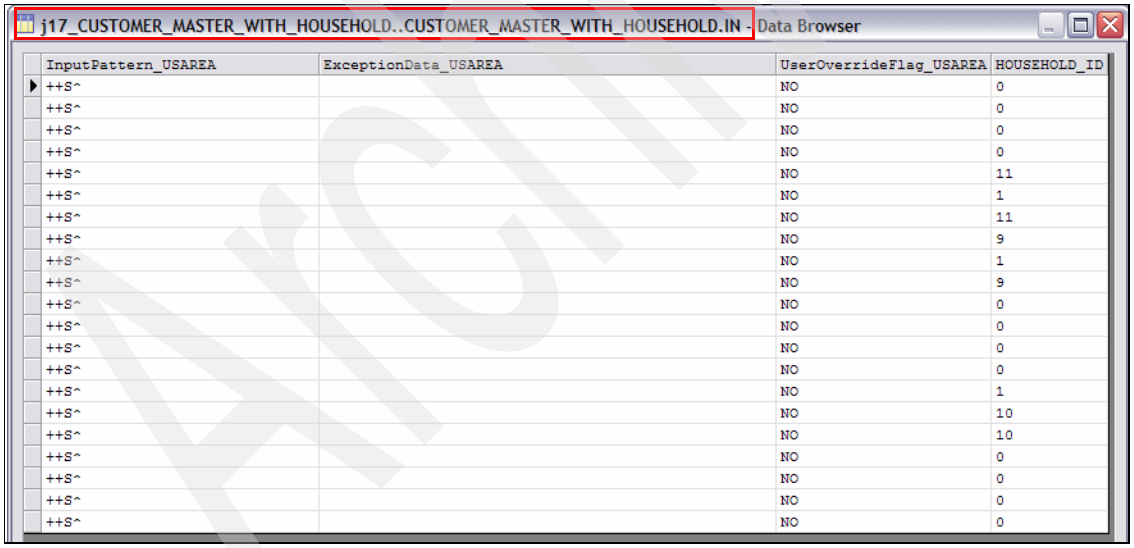
Figure 1-393 Create J17_CUSTOMER_MASTER_WITH_HOUSEHOLD job 5/7



j17_CUSTOMER_MASTER_WITH_HOUSEHOLD..CUSTOMER_MASTER_WITH_HOUSEHOLD.IN - Data Browser

EMAIL	DOB	ISOCountryCode_COUNTRY	IdentifierFlag_COUNTRY	DUMMY_ADD_FIELD	Zip5_CASS	Zip4_CASS	D
curtis@home.com	1960-04-13	US	Y		95126	1502	0
		US	Y		95148	4337	2
myself@gmail.com	1965-03-17	US	Y		95126	1502	0
		US	Y		95123	4800	6
	1959-05-03	US	Y		95126		
anders@olsson.com	1960-10-21	US	Y		95131	2001	5
mymail@yahoo.com	1970-10-21	US	Y		95119	1410	9
carter@carter.com	1965-11-01	US	Y		95124	6324	2
anymail@hotmail.com	1956-05-23	US	Y		95119	1546	7
dfx@usa.ibm.com	1977-07-13	US	Y		95129	5203	3
a@gmail.com		US	Y		95148	4337	2
jaz23@yahoo.com	1967-02-01	US	Y		95131	1010	1
anymail@hotmail.com	1958-05-23	US	Y		95119	1546	7
	1978-04-12	US	Y		95118	1207	6
		US	Y		95148	4337	2
	1969-06-11	US	Y		95119	1931	2
	1975-06-15	US	Y		95119	1931	2
		US	Y		95123	4837	0
afan@hotmail.com	1970-12-12	US	Y		95134	1902	8
arc@hotmail.com	1956-03-31	US	Y		95134	1700	7
		US	Y		95134	2028	3

Figure 1-394 Create J17_CUSTOMER_MASTER_WITH_HOUSEHOLD job 6/7



j17_CUSTOMER_MASTER_WITH_HOUSEHOLD..CUSTOMER_MASTER_WITH_HOUSEHOLD.IN - Data Browser

InputPattern_USAREA	ExceptionData_USAREA	UserOverrideFlag_USAREA	HOUSEHOLD_ID
++S^		NO	0
++S^		NO	0
++S^		NO	0
++S^		NO	0
++S^		NO	11
++S^		NO	1
++S^		NO	11
++S^		NO	9
++S^		NO	1
++S^		NO	9
++S^		NO	0
++S^		NO	0
++S^		NO	0
++S^		NO	0
++S^		NO	1
++S^		NO	10
++S^		NO	10
++S^		NO	0
++S^		NO	0
++S^		NO	0
++S^		NO	0

Figure 1-395 Create J17_CUSTOMER_MASTER_WITH_HOUSEHOLD job 7/7

J18_MATCHFREQ_CUSTOMER_WITH_HOUSEHOLD

In this step, we generate frequency distribution on all the columns in the cleansed credit card customer file, including household information of the

"J17_CUSTOMER_MASTER_WITH_HOUSEHOLD" on page 327 job using the Match Frequency stage. Here again, the idea is to generate match frequency for all the columns so that it can be used with any match specification. This is used in a later matching stage with mailing lists as described in 1.10.6, "Mailing list cleansing" on page 334.

Figure 1-396 shows the various stages that are used in this job, including the data set that was created in "J17_CUSTOMER_MASTER_WITH_HOUSEHOLD" on page 327, a Match Frequency stage, and a Sequential File stage. We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object created earlier.

Because the configuration of this job is very similar to that described in "J10_MATCHFREQ_STAN_CUSTOMER" on page 266, we do not repeat it here.

The contents of the output of this stage (CUSTOMER_MASTER_WITH_HOUSEHOLD_FREQ) are listed in Figure 1-397 on page 334. We do not document the interpretation of the format and content of this file.

Now, proceed to "J10_REFERENCE_MatchSpec_MAILING_LIST" on page 413.

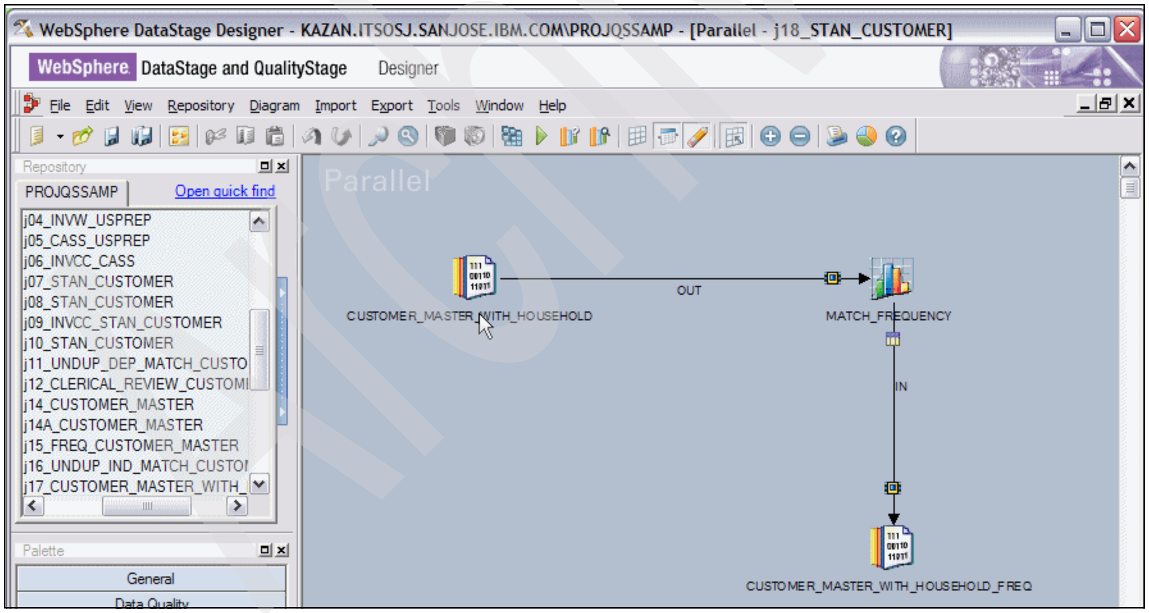
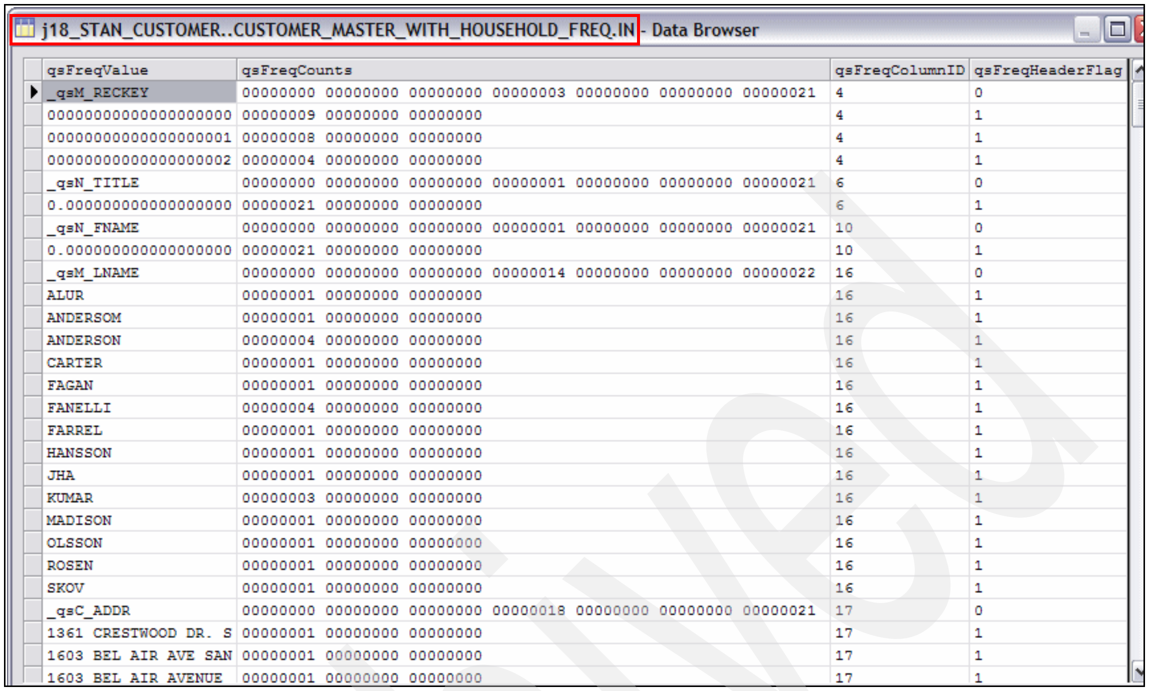

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j18_STAN_CUSTOMER]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Repository
PROJQSSAMP
Open quick find
j04_INVW_USPREP
j05_CASS_USPREP
j06_INVCC_CASS
j07_STAN_CUSTOMER
j08_STAN_CUSTOMER
j09_INVCC_STAN_CUSTOMER
j10_STAN_CUSTOMER
j11_UNDUP_DEP_MATCH_CUSTO
j12_CLERICAL_REVIEW_CUSTOMI
j14_CUSTOMER_MASTER
j14A_CUSTOMER_MASTER
j15_FREQ_CUSTOMER_MASTER
j16_UNDUP_IND_MATCH_CUSTOI
j17_CUSTOMER_MASTER_WITH_
Palette
General
Data Quality
Parallel
CUSTOMER_MASTER_WITH_HOUSEHOLD
OUT
MATCH_FREQUENCY
IN
CUSTOMER_MASTER_WITH_HOUSEHOLD_FREQ

Figure 1-396 Create J18_MATCHFREQ_CUSTOMER_WITH_HOUSEHOLD 1/2


j18_STAN_CUSTOMER..CUSTOMER_MASTER_WITH_HOUSEHOLD_FREQ.IN - Data Browser

qsFreqValue	qsFreqCounts	qsFreqColumnID	qsFreqHeaderFlag
_qsM_RECKEY	00000000 00000000 00000000 00000003 00000000 00000000 00000021	4	0
00000000000000000000	00000009 00000000 00000000	4	1
00000000000000000001	00000008 00000000 00000000	4	1
00000000000000000002	00000004 00000000 00000000	4	1
_qsN_TITLE	00000000 00000000 00000000 00000001 00000000 00000000 00000021	6	0
0.000000000000000000	00000021 00000000 00000000	6	1
_qsN_FNAME	00000000 00000000 00000000 00000001 00000000 00000000 00000021	10	0
0.000000000000000000	00000021 00000000 00000000	10	1
_qsM_LNAME	00000000 00000000 00000000 00000014 00000000 00000000 00000022	16	0
ALUR	00000001 00000000 00000000	16	1
ANDERSOM	00000001 00000000 00000000	16	1
ANDERSON	00000004 00000000 00000000	16	1
CARTER	00000001 00000000 00000000	16	1
FAGAN	00000001 00000000 00000000	16	1
FANELLI	00000004 00000000 00000000	16	1
FARREL	00000001 00000000 00000000	16	1
HANSSON	00000001 00000000 00000000	16	1
JHA	00000001 00000000 00000000	16	1
KUMAR	00000003 00000000 00000000	16	1
MADISON	00000001 00000000 00000000	16	1
OLSSON	00000001 00000000 00000000	16	1
ROSEN	00000001 00000000 00000000	16	1
SKOV	00000001 00000000 00000000	16	1
_qsC_ADDR	00000000 00000000 00000000 00000018 00000000 00000000 00000021	17	0
1361 CRESTWOOD DR. S	00000001 00000000 00000000	17	1
1603 BEL AIR AVE SAN	00000001 00000000 00000000	17	1
1603 BEL AIR AVENUE	00000001 00000000 00000000	17	1

Figure 1-397 Create J18_MATCHFREQ_CUSTOMER_WITH_HOUSEHOLD 2/2

1.10.6 Mailing list cleansing

Figure 1-398 on page 341 shows the processing flow and jobs used for:

- Cleansing the merged mailing lists purchased from three different sources.
- Matching the cleansed mailing list with the cleansed credit card customer file to enhance it, as well as eliminate those records in the mailing list that already exist in the cleansed credit card customer file.
- Determining mailing list persons who belong to the same household to reduce mailing costs.

The steps briefly are:

1. We begin by extracting all the mailing list data from a DB2 database[26] and loading it into a data set to isolate it from changes during analysis. We also pre-processed it for analysis, which involved changing default values to nulls for columns such as the telephone number.[27]

[26] We had merged and loaded the purchased mailing lists into a DB2 database previously, and do not show this process here.

[27] If the content was (999) 999-9999 (user default), then insert a null in the output.

Job “J00_SRC_MAILING_LIST” on page 342 performs this step.

2. We then split the mailing list records into two files: one with name and address fields and the another for single domain fields such as telephone numbers and e-mail addresses. The purpose of splitting the mailing list records is to allow processing of the single domain and text data in parallel by the QualityStage administrators and appropriate subject matter experts.

   Note: In this case, for convenience, we chose to use all the columns in the input data for the both streams. Job “J01_STAN_COUNTRY_M” on page 353 for the name and address fields and job “J00A_INV_MAILING_LIST” on page 347.

   In the job “J00A_INV_MAILING_LIST” on page 347, we use the Investigate stage on non-text columns such as source (of the mailing list), home and cell phone numbers, and e-mail addresses. We used character concatenate and character discrete with combinations of *C*, *T*, and *X* masks. The objective of this step is to validate some of the main non-text columns. Any errors that are detected can be resolved by modifying the source directly or by using IBM WebSphere QualityStage jobs to cleanse and modify the targets.

3. Next, we analyze the mailing list’s addresses to determine the (ISO code) country using the COUNTRY rule set in the Standardize stage.

   Job “J01_STAN_COUNTRY_M” on page 353 performs this step.

4. The ISO codes that are generated by the previous step are analyzed by the Investigate stage using character discrete with the *C* mask to obtain frequency distribution. This step identifies whether the addresses in the mailing list file belong to more than one country and identifies the codes of the countries in the addresses. In this case, all the addresses were U.S. addresses.

   Job “J02_INVCC_ISOCODE_M” on page 358 performs this step.

5. We then use the Standardize stage with the domain-preprocessor rule set USPREP to move name and address data into Name, Address, and Area domains.

   Job “J03_STAN_USPREP_M” on page 361 performs this step.

6. We then use the Investigate stage using word investigate on the Name, Address, and Area domains to determine whether the domain-preprocessor USPREP rule set parsed the tokens in the name and address fields into the correct domains successfully.

   Job “J04_INVW_USPREP_M” on page 366 performs this step.

7. A visual analysis of the token and pattern reports of the “J04_INVW_USPREP_M” on page 366 job is performed to determine if the parsing was successful.

   We found certain errors with the parsing, and so chose to perform a classification override for the USPREP rule set to fix the errors.

   Job “J04_Z_After_Override” on page 373 performs this step.

8. The “J03_STAN_USPREP_M” on page 361, “J04_INVW_USPREP_M” on page 366, and “J04_Z_After_Override” on page 373 steps is repeated until the name and address data is moved into the correct domain columns.

   Attention: Generally, the overrides in the job “J04_Z_After_Override” on page 373 steps should be limited to simple classification overrides that move name and address data to the correct name/address/area domain buckets, rather than more complex pattern overrides.

9. After all the name and address data is moved to the correct domain buckets, we use the CASS stage to validate, correct, and standardize the U.S. addresses in the Address domain. We also include a Transformer stage to add a second address line column to the mailing list file because CASS requires two address lines as input for its processing.

   Job “J05_CASS_USPREP_M” on page 374 performs this step.

10. We then run the Investigate stage with character concatenate on the (address related columns) results of the job “J05_CASS_USPREP_M” on page 374 step to determine investigate addresses that are not recognized by CASS (delivery point verification or DPV).

    Note: Due to a bug with handling nulls, we introduced a Transformer stage to convert nulls to a space using column derivation.

    We also investigate using character concatenate (on CASS generated columns DPVMATCHFLAG_CASS and DPVCODE1_CASS) using a *C* mask, the output of CASS. A value of `A1` in the DPVCODE1_CASS field indicates a potential problem.

    Job “J06_INVCC_CASS_M” on page 379 performs this step.

11. The next step is to standardize the name and address contents of the output of job “J05_CASS_USPREP_M” on page 374 using the domain-preprocessor USPREP rule set. We also add a column to the output that only had the first three characters of the ZIP code using a Transformer stage. This new column (ZIP3) is used as a blocking variable in the following matching stage.

Job “J07_STAN_MAILING_LIST_Domain_Preprocessor” on page 383 performs this step.

12. After reviewing the patterns that were generated in the “J07_STAN_MAILING_LIST_Domain_Preprocessor” on page 383 job, we find certain names not classified as first names (F). Therefore, we add them to the domain-specific rule set USNAME. We do not show this process here, it is similar to the process described in “Selecting override object types to modify rule sets” on page 64.

13. After the classification overrides for the names, in the next step, we standardize the name and address contents of the output of job “J07_STAN_MAILING_LIST_Domain_Preprocessor” on page 383 using the domain-specific USNAME (with column NameDomain_USPREP), USADDR (with column AddressDomain_USPREP) and USAREA (with column AreaDomain_USPREP) rule sets. Three separate processes are defined—one for each rule set.

    Note: Here again, we had to introduce a null handling Transformer stage to circumvent a bug.

    Job “J08_STAN_MAILING_LIST_Domain_Specific” on page 390 performs this step.

14. The next step identifies unhandled patterns in the “J08_STAN_MAILING_LIST_Domain_Specific” on page 390 job. We run the Investigate stage with character concatenate using the *C* mask on the unhandled pattern column from the results of “J08_STAN_MAILING_LIST_Domain_Specific” on page 390—the columns investigated correspond to the name, address, and area domains.

    Job “J09_INVCC_STAN_MAILING_LIST” on page 399 performs this step.

15. Because we find unhandled patterns (such as F,I) in the name domain, we create pattern overrides in the domain-specific USNAME rule set. We also find unhandled patterns in the address domain and create overrides in the domain-specific USADDR rule set. We then re-run the “J08_STAN_MAILING_LIST_Domain_Specific” on page 390 step.

    Job “J09_INVCC_STAN_MAILING_LIST” on page 399 performs this step.

16. After the unhandled patterns are handled, we generate frequency distribution on all the columns in the mailing list file using the Match Frequency stage. The idea is to generate match frequency for all the columns so that it can be used with any match specification.

    Job “J10_MATCHFREQ_STAN_MAILING_LIST” on page 411 performs this step.

17. The next step is to generate a match specification for a Reference match stage using as input the match frequency data that was created in the “J10_MATCHFREQ_STAN_MAILING_LIST” on page 411 job. The specification included a single pass based on the three digits ZIP code and the street name.

    Job “J10_REFERENCE_MatchSpec_MAILING_LIST” on page 413 performs this step.

18. The next step is to determine if there were any matches between the records in the credit card customer file and mailing list file by running the Reference stage, which required standardized data (from “J17_CUSTOMER_MASTER_WITH_HOUSEHOLD” on page 327) and reference data (from “J08_STAN_MAILING_LIST_Domain_Specific” on page 390) as source data, a reference match specification (from “J10_REFERENCE_MatchSpec_MAILING_LIST” on page 413), and frequency information for both sources (from “J18_MATCHFREQ_CUSTOMER_WITH_HOUSEHOLD” on page 332 and “J10_MATCHFREQ_STAN_MAILING_LIST” on page 411). The output is matched records, records for clerical review, mailing list duplicates, credit card customer duplicates, mailing list residuals, and credit card customer residuals.

    Job “J11_REFMATCH” on page 416 performs this step.

19. We then create a single report of the matched mailing list persons and the records in the clerical review records to enable a manual review of potential match records.

    Job “J12_CLERICAL_REPORT_MAILING_LIST” on page 430 performs this step.

20. If a manual review of the report finds duplicates, theses duplicates must be considered as part of the file that contains matched records. We do not show this process here.

21. The next step is to enhance the information in the credit card customer file with information in the mailing list if appropriate, for example the middle name.

    Job “J13_ENHANCE_CUSTOMER” on page 437 performs this step.

22. The next step is to generate frequency distribution on all the columns in the residual mailing list file that was created in the “J11_REFMATCH” on page 416 step using the Match Frequency stage. Here again, the idea is to generate match frequency for all the columns so that it can be used with any match specification.

    Job “J14_MAILING_LIST_RESIDUAL_FREQ” on page 442 performs this step.

23. The next step is to generate a match specification for an Unduplicate match stage using as input the match frequency data that was created in the "J14_MAILING_LIST_RESIDUAL_FREQ" on page 442 job. The specification included two passes:

    - The first pass blocks on a phonetic encoding (NYSIIS) of the primary (last) name, the address, phonetic encoding (NYSIIS) of the street name, and the ZIP code.
    - The second pass blocks on the three digit ZIP code.

    Job "J14_UNDUP_DEP_MATCHSPEC_MAILING" on page 445 performs this step.

24. Next, we determine whether there were duplicates in the mailing list persons file using the Unduplicate stage with the match specification and match frequency information that was created in steps "J14_UNDUP_DEP_MATCHSPEC_MAILING" on page 445 and "J14_MAILING_LIST_RESIDUAL_FREQ" on page 442 respectively. As before, the output is matched records (merge of master and duplicates using a Funnel stage), records for clerical review, and residuals (records that do not match).

    Job "J15_UNDUP_DEP_MATCH_MAILING" on page 449 performs this step.

25. We find no duplicates for clerical review. Therefore, the next step is to survive the best information from the set of duplicates in the matched records that were created in the "J15_UNDUP_DEP_MATCH_MAILING" on page 449 step using the SURVIVE stage.

    Job "J16_SURVIVE_MAILING" on page 454 performs this step.

26. A clean master of mailing list persons is created by merging the master that was created in the "J16_SURVIVE_MAILING" on page 454 step and residual records that were created in the "J15_UNDUP_DEP_MATCH_MAILING" on page 449 step into a data set using the FUNNEL stage.

    Job "J17_MAILING_MASTER" on page 458 performs this step.

27. In order to generate household information for the master mailing list persons that was created in the "J17_MAILING_MASTER" on page 458 step, we generate frequency distribution on all the columns in the master mailing list file using the Match Frequency stage. Here again, the idea is to generate match frequency for all the columns so that it can be used with any match specification.

    Job "J18_FREQ_MAILING_MASTER" on page 461 performs this step.

28. The next step is to generate a match specification for an Unduplicate match stage using as input the match frequency data that was created in the "J18_FREQ_MAILING_MASTER" on page 461 job. The specification included two passes:

- The first pass blocks on the primary (last) name and a phonetic encoding (NYSIIS) of the street name, and the ZIP code.
- The second pass blocks on the five digit ZIP code.

Job “J18_UNDUP_IND_MATCHSPEC_MAILING” on page 464 performs this step.

29. Next, we determine whether there were duplicates in the mailing list persons file using the Unduplicate stage with the match specification and match frequency information that was created in steps “J18_UNDUP_IND_MATCHSPEC_MAILING” on page 464 and “J18_FREQ_MAILING_MASTER” on page 461 respectively. As before, the output is matched records (merge of master and duplicates using a Funnel stage), records for clerical review, and residuals (records that do not match).

    Job “J19_UNDUP_IND_MATCH_MAILING” on page 468 performs this step.

30. A clean master of mailing list persons was created by merging the master and residual records that were generated in the “J19_UNDUP_IND_MATCH_MAILING” on page 468 step into a data set using the FUNNEL stage. We add a household ID (using a Transformer stage) to these records in the output with a value (qsMatchSetID) when a record belonged to a household and a zero when it did not belong to a household.

    Job “J20_MAILING_MASTER_WITH_HOUSEHOLD” on page 473 performs this step.

Note: After these steps, we now have a clean mailing list persons file with household information that can be used for promotional mailing at reduced mailing costs.

These jobs are described in more detail in the following sections.

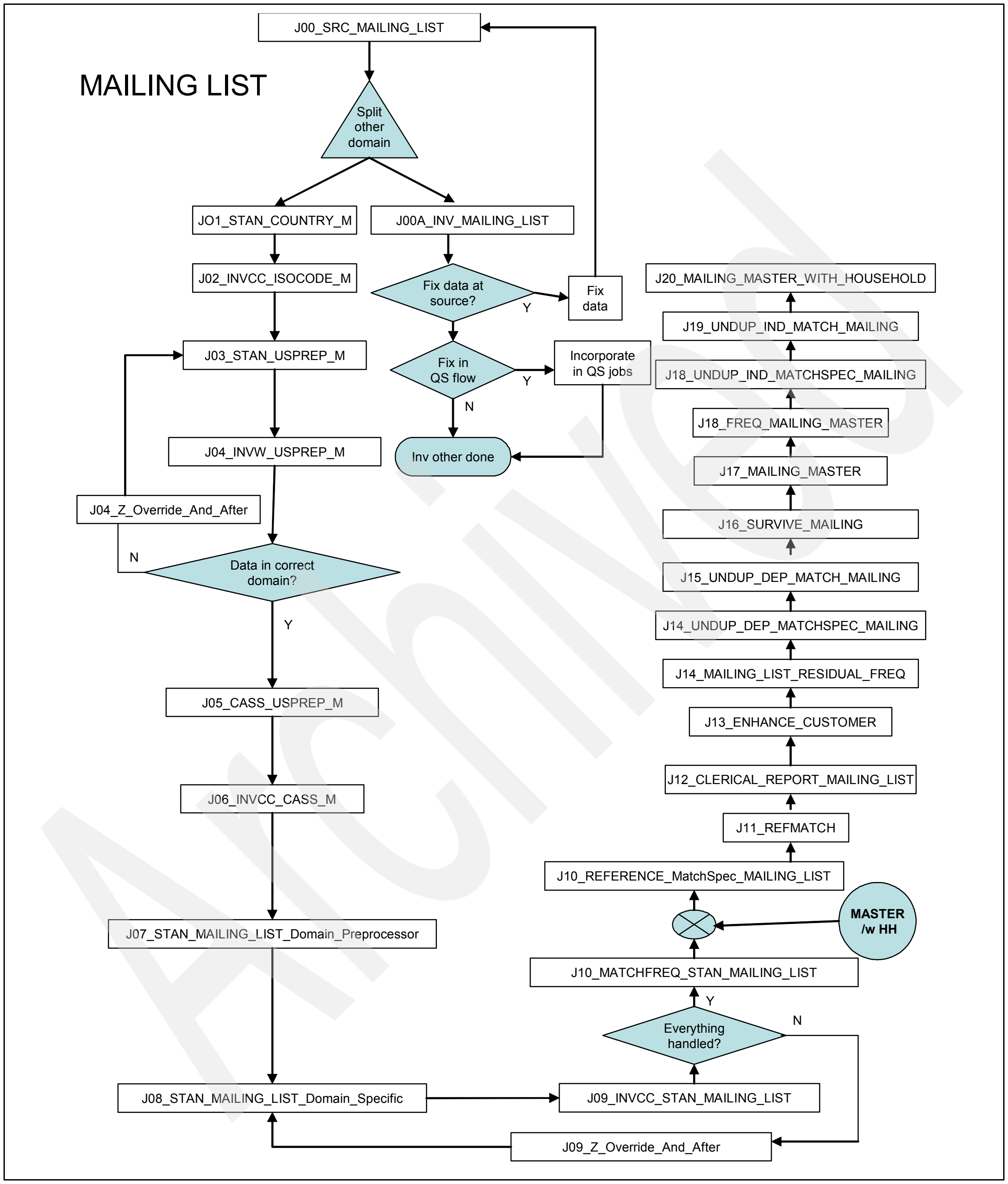


Figure 1-398 Mailing list cleanup and merge with Customer data process flow

J00_SRC_MAILING_LIST

As mentioned earlier, we begin by extracting all the mailing list data from a DB2 database and loading it into a data set to isolate it from changes during analysis. We also pre-processed it for analysis, which involved changing default values to nulls for columns such as the telephone number.

Figure 1-399 shows the various stages that are used in this job, including a DB2 UDB API stage that is used to access the data in the MAILING_LIST table. Its rows are pre-processed to convert default values (such as (999) 999-9999) in the PHONE column to nulls using the Transformer stage. The transformed data is written to a data set. We modify the names of the stages as shown, and this job uses the same QSPARAMETERSET object created earlier.

Because the configuration of this job is very similar to that described in “J00_SRC_CUSTOMER” on page 142, we do not repeat it here. However, some of the configurations of interest are as follows:

- Figure 1-400 on page 343 shows the generated SQL to retrieve all the rows in the MAILING_LIST table.
- Figure 1-401 on page 343 through Figure 1-404 on page 345 show the configuration of the Transformer Stage that pre-processes data that is retrieved from the MAILING_LIST table to replace user-defined default values in the CELL_PHONE, HOME_PHONE, and EMAIL columns to null.
- After saving, compiling, and running the job, Figure 1-405 on page 345 shows the results of the execution.
- Figure 1-406 on page 346 and Figure 1-407 on page 347 show the pre-processed rows in the MAILING_LIST table.

Now, proceed to “J01_STAN_COUNTRY_M” on page 353.

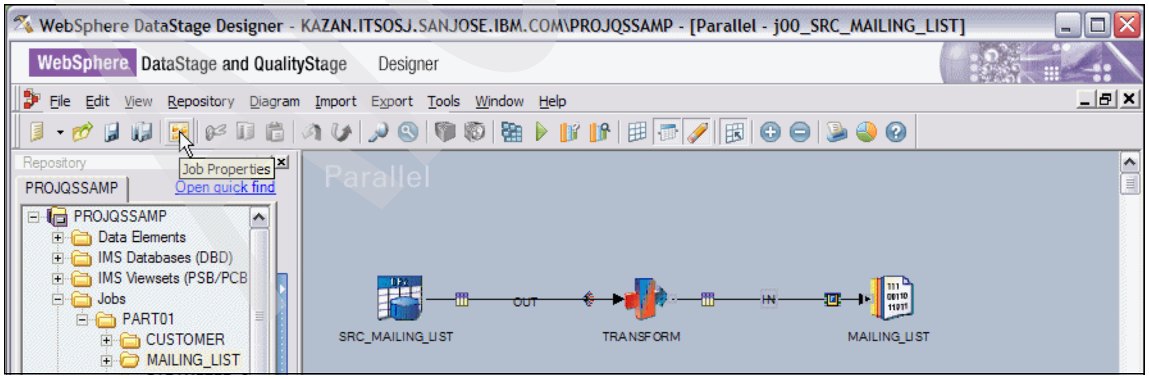
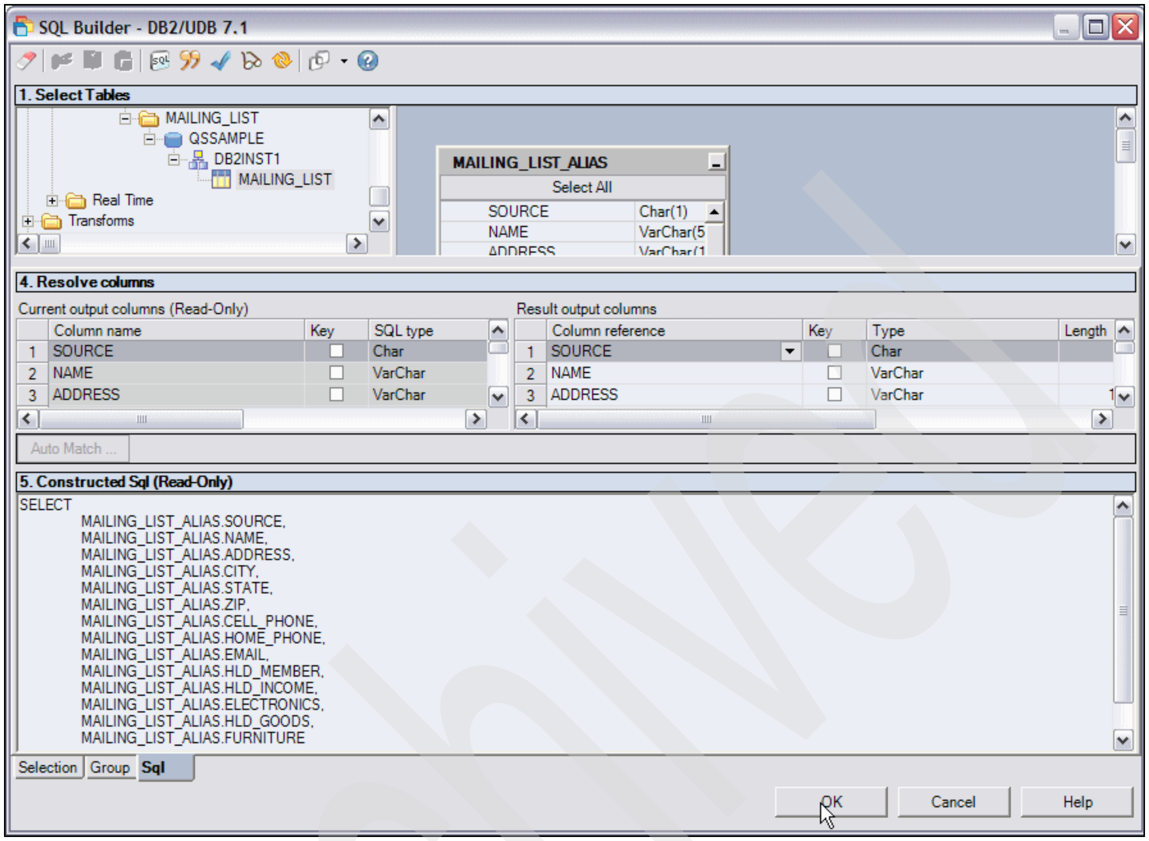

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j00_SRC_MAILING_LIST]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Repository
Job Properties
PROJQSSAMP
Open quick find
PROJQSSAMP
Data Elements
IMS Databases (DBD)
IMS Viewsets (PSB/PCB
Jobs
PART01
CUSTOMER
MAILING_LIST
Parallel
OUT
IN
SRC_MAILING_LIST
TRANSFORM
MAILING_LIST

Figure 1-399 Create J00_SRC_MAILING_LIST job 1/9



SQL Builder - DB2/UDB 7.1

1. Select Tables

MAILING_LIST_ALIAS

SOURCE	Char(1)
NAME	VarChar(5
ADDRESS	VarChar(1

4. Resolve columns

Current output columns (Read-Only)

	Column name	Key	SQL type
1	SOURCE	☐	Char
2	NAME	☐	VarChar
3	ADDRESS	☐	VarChar

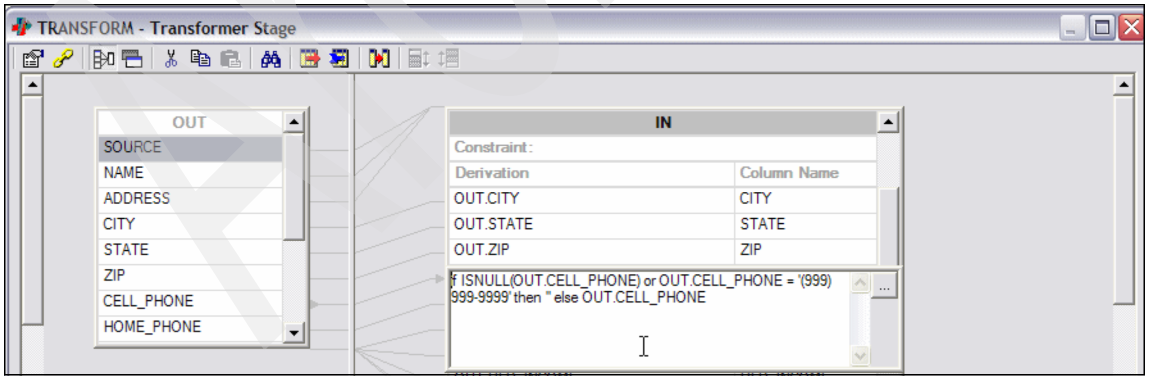
Result output columns

	Column reference	Key	Type	Length
1	SOURCE	☐	Char	
2	NAME	☐	VarChar	
3	ADDRESS	☐	VarChar	1

5. Constructed Sql (Read-Only)

```
SELECT
        MAILING_LIST_ALIAS.SOURCE,
        MAILING_LIST_ALIAS.NAME,
        MAILING_LIST_ALIAS.ADDRESS,
        MAILING_LIST_ALIAS.CITY,
        MAILING_LIST_ALIAS.STATE,
        MAILING_LIST_ALIAS.ZIP,
        MAILING_LIST_ALIAS.CELL_PHONE,
        MAILING_LIST_ALIAS.HOME_PHONE,
        MAILING_LIST_ALIAS.EMAIL,
        MAILING_LIST_ALIAS.HLD_MEMBER,
        MAILING_LIST_ALIAS.HLD_INCOME,
        MAILING_LIST_ALIAS.ELECTRONICS,
        MAILING_LIST_ALIAS.HLD_GOODS,
        MAILING_LIST_ALIAS.FURNITURE
```

Figure 1-400 Create J00_SRC_MAILING_LIST job 2/9



TRANSFORM - Transformer Stage

OUT: SOURCE, NAME, ADDRESS, CITY, STATE, ZIP, CELL_PHONE, HOME_PHONE

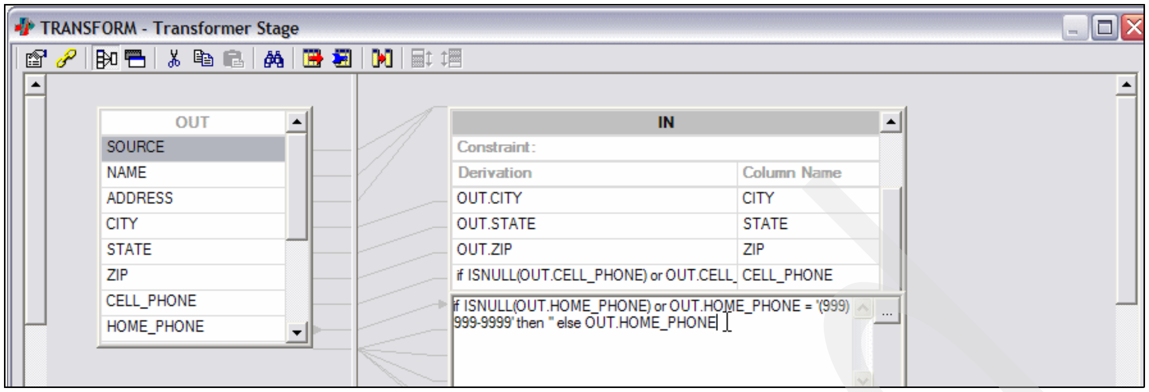
IN

Constraint:

Derivation	Column Name
OUT.CITY	CITY
OUT.STATE	STATE
OUT.ZIP	ZIP

```
f ISNULL(OUT.CELL_PHONE) or OUT.CELL_PHONE = '(999)
999-9999' then '' else OUT.CELL_PHONE
```

Figure 1-401 Create J00_SRC_MAILING_LIST job 3/9



OUT: SOURCE, NAME, ADDRESS, CITY, STATE, ZIP, CELL_PHONE, HOME_PHONE

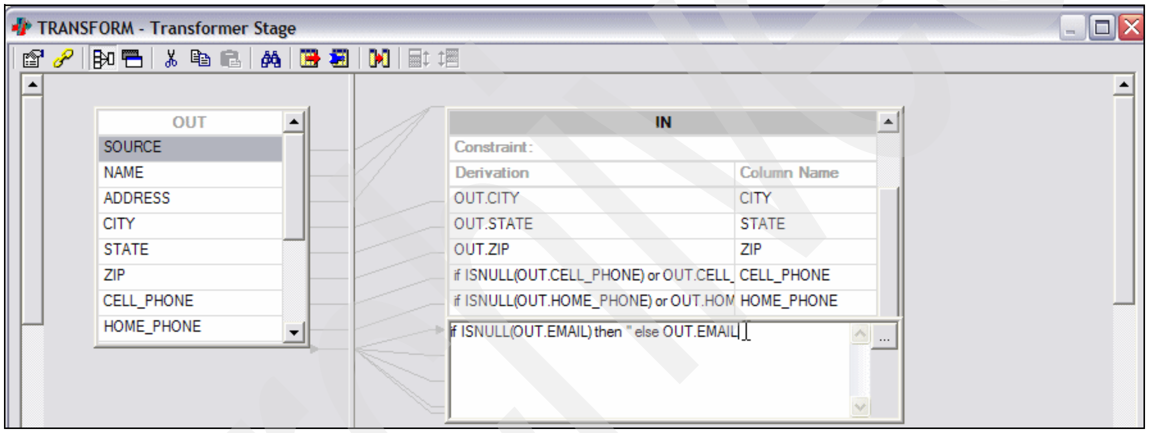
IN

Constraint:

Derivation	Column Name
OUT.CITY	CITY
OUT.STATE	STATE
OUT.ZIP	ZIP
if ISNULL(OUT.CELL_PHONE) or OUT.CELL_	CELL_PHONE

```
if ISNULL(OUT.HOME_PHONE) or OUT.HOME_PHONE = '(999)
999-9999' then '' else OUT.HOME_PHONE
```

Figure 1-402   Create J00_SRC_MAILING_LIST job 4/9



OUT: SOURCE, NAME, ADDRESS, CITY, STATE, ZIP, CELL_PHONE, HOME_PHONE

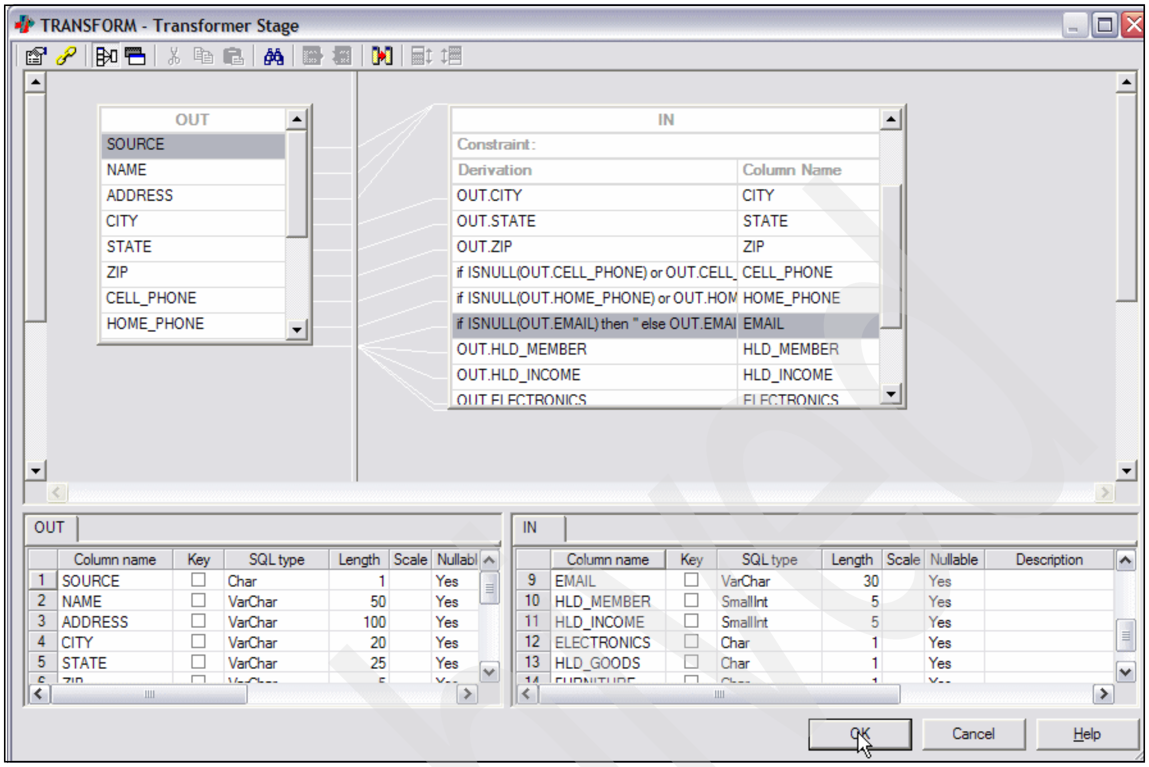
IN

Constraint:

Derivation	Column Name
OUT.CITY	CITY
OUT.STATE	STATE
OUT.ZIP	ZIP
if ISNULL(OUT.CELL_PHONE) or OUT.CELL_	CELL_PHONE
if ISNULL(OUT.HOME_PHONE) or OUT.HOM	HOME_PHONE

```
if ISNULL(OUT.EMAIL) then '' else OUT.EMAIL
```

Figure 1-403   Create J00_SRC_MAILING_LIST job 5/9



OUT
SOURCE
NAME
ADDRESS
CITY
STATE
ZIP
CELL_PHONE
HOME_PHONE

IN

Constraint:

Derivation	Column Name
OUT.CITY	CITY
OUT.STATE	STATE
OUT.ZIP	ZIP
if ISNULL(OUT.CELL_PHONE) or OUT.CELL_	CELL_PHONE
if ISNULL(OUT.HOME_PHONE) or OUT.HOM	HOME_PHONE
if ISNULL(OUT.EMAIL) then '' else OUT.EMAI	EMAIL
OUT.HLD_MEMBER	HLD_MEMBER
OUT.HLD_INCOME	HLD_INCOME
OUT.ELECTRONICS	ELECTRONICS

OUT

	Column name	Key	SQL type	Length	Scale	Nullabl
1	SOURCE	☐	Char	1		Yes
2	NAME	☐	VarChar	50		Yes
3	ADDRESS	☐	VarChar	100		Yes
4	CITY	☐	VarChar	20		Yes
5	STATE	☐	VarChar	25		Yes

IN

	Column name	Key	SQL type	Length	Scale	Nullable	Description
9	EMAIL	☐	VarChar	30		Yes	
10	HLD_MEMBER	☐	SmallInt	5		Yes	
11	HLD_INCOME	☐	SmallInt	5		Yes	
12	ELECTRONICS	☐	Char	1		Yes	
13	HLD_GOODS	☐	Char	1		Yes	

Figure 1-404 Create J00_SRC_MAILING_LIST job 6/9

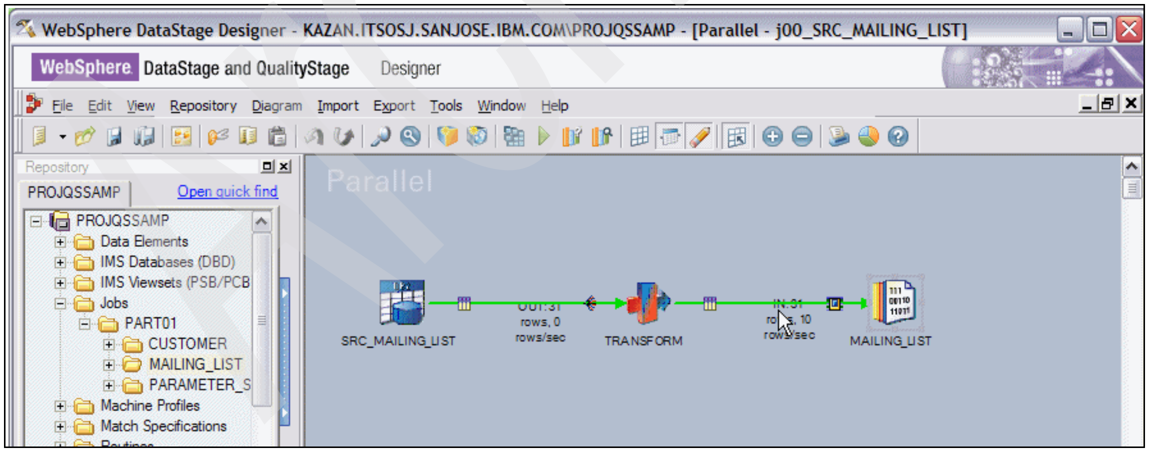
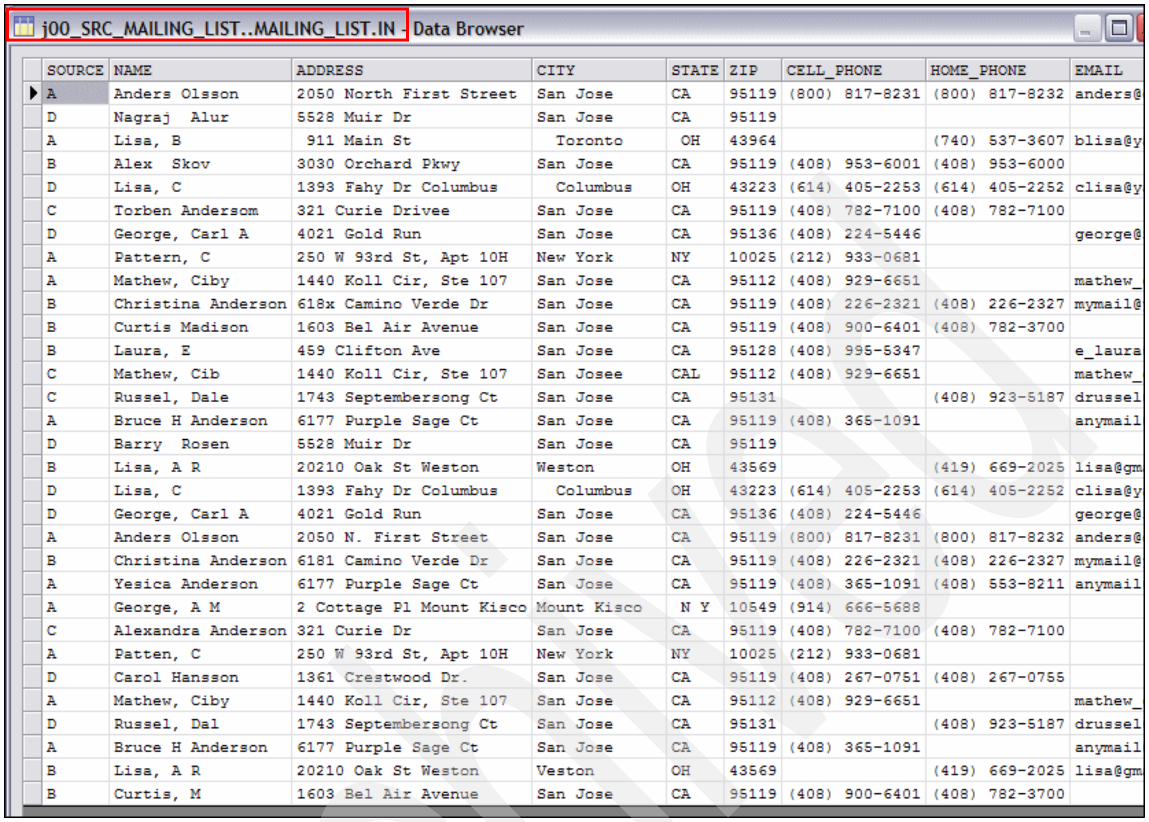
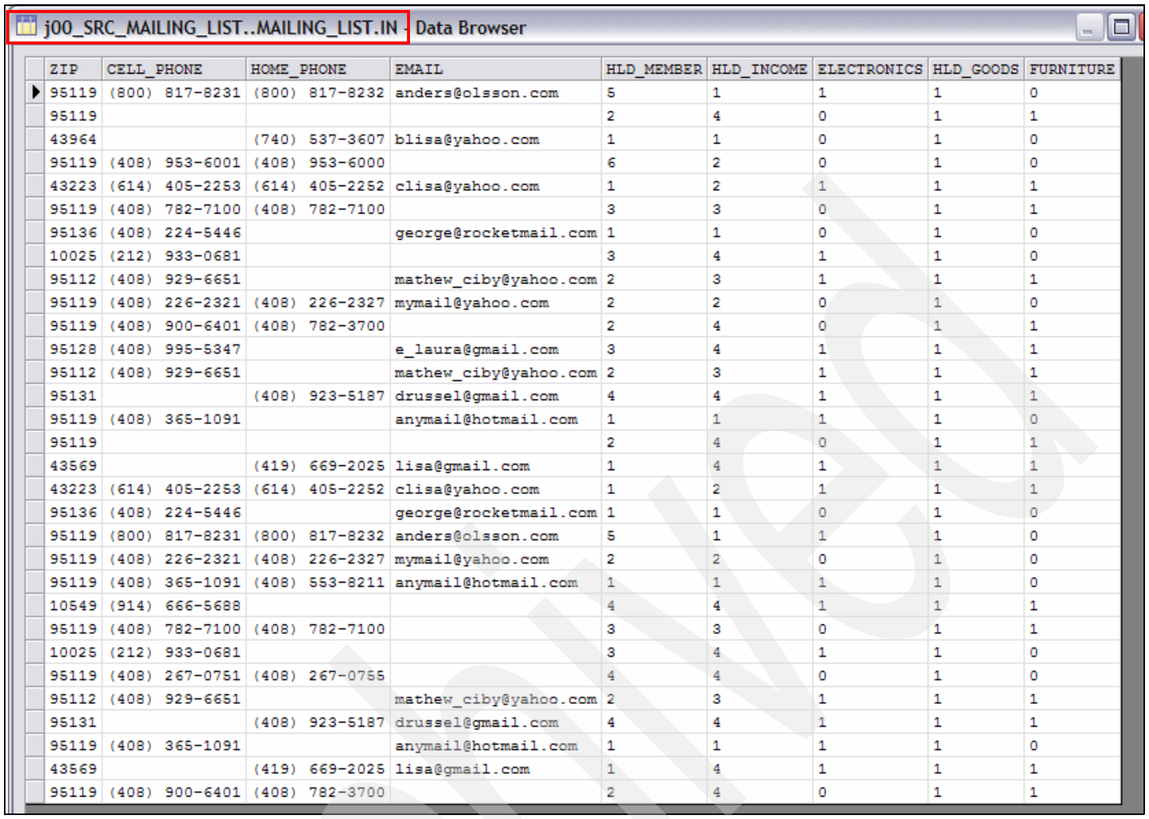


Figure 1-405 Create J00_SRC_MAILING_LIST job 7/9


j00_SRC_MAILING_LIST..MAILING_LIST.IN - Data Browser

SOURCE	NAME	ADDRESS	CITY	STATE	ZIP	CELL_PHONE	HOME_PHONE	EMAIL
A	Anders Olsson	2050 North First Street	San Jose	CA	95119	(800) 817-8231	(800) 817-8232	anders@
D	Nagraj Alur	5528 Muir Dr	San Jose	CA	95119			
A	Lisa, B	911 Main St	Toronto	OH	43964		(740) 537-3607	blisa@y
B	Alex Skov	3030 Orchard Pkwy	San Jose	CA	95119	(408) 953-6001	(408) 953-6000	
D	Lisa, C	1393 Fahy Dr Columbus	Columbus	OH	43223	(614) 405-2253	(614) 405-2252	clisa@y
C	Torben Andersom	321 Curie Drivee	San Jose	CA	95119	(408) 782-7100	(408) 782-7100	
D	George, Carl A	4021 Gold Run	San Jose	CA	95136	(408) 224-5446		george@
A	Pattern, C	250 W 93rd St, Apt 10H	New York	NY	10025	(212) 933-0681		
A	Mathew, Ciby	1440 Koll Cir, Ste 107	San Jose	CA	95112	(408) 929-6651		mathew_
B	Christina Anderson	618x Camino Verde Dr	San Jose	CA	95119	(408) 226-2321	(408) 226-2327	mymail@
B	Curtis Madison	1603 Bel Air Avenue	San Jose	CA	95119	(408) 900-6401	(408) 782-3700	
B	Laura, E	459 Clifton Ave	San Jose	CA	95128	(408) 995-5347		e_laura
C	Mathew, Cib	1440 Koll Cir, Ste 107	San Josee	CAL	95112	(408) 929-6651		mathew_
C	Russel, Dale	1743 Septembersong Ct	San Jose	CA	95131		(408) 923-5187	drussel
A	Bruce H Anderson	6177 Purple Sage Ct	San Jose	CA	95119	(408) 365-1091		anymail
D	Barry Rosen	5528 Muir Dr	San Jose	CA	95119			
B	Lisa, A R	20210 Oak St Weston	Weston	OH	43569		(419) 669-2025	lisa@gm
D	Lisa, C	1393 Fahy Dr Columbus	Columbus	OH	43223	(614) 405-2253	(614) 405-2252	clisa@y
D	George, Carl A	4021 Gold Run	San Jose	CA	95136	(408) 224-5446		george@
A	Anders Olsson	2050 N. First Street	San Jose	CA	95119	(800) 817-8231	(800) 817-8232	anders@
B	Christina Anderson	6181 Camino Verde Dr	San Jose	CA	95119	(408) 226-2321	(408) 226-2327	mymail@
A	Yesica Anderson	6177 Purple Sage Ct	San Jose	CA	95119	(408) 365-1091	(408) 553-8211	anymail
A	George, A M	2 Cottage Pl Mount Kisco	Mount Kisco	N Y	10549	(914) 666-5688		
C	Alexandra Anderson	321 Curie Dr	San Jose	CA	95119	(408) 782-7100	(408) 782-7100	
A	Patten, C	250 W 93rd St, Apt 10H	New York	NY	10025	(212) 933-0681		
D	Carol Hansson	1361 Crestwood Dr.	San Jose	CA	95119	(408) 267-0751	(408) 267-0755	
A	Mathew, Ciby	1440 Koll Cir, Ste 107	San Jose	CA	95112	(408) 929-6651		mathew_
D	Russel, Dal	1743 Septembersong Ct	San Jose	CA	95131		(408) 923-5187	drussel
A	Bruce H Anderson	6177 Purple Sage Ct	San Jose	CA	95119	(408) 365-1091		anymail
B	Lisa, A R	20210 Oak St Weston	Veston	OH	43569		(419) 669-2025	lisa@gm
B	Curtis, M	1603 Bel Air Avenue	San Jose	CA	95119	(408) 900-6401	(408) 782-3700	

Figure 1-406   Create J00_SRC_MAILING_LIST job 8/9



j00_SRC_MAILING_LIST..MAILING_LIST.IN - Data Browser

ZIP	CELL_PHONE	HOME_PHONE	EMAIL	HLD_MEMBER	HLD_INCOME	ELECTRONICS	HLD_GOODS	FURNITURE
95119	(800) 817-8231	(800) 817-8232	anders@olsson.com	5	1	1	1	0
95119				2	4	0	1	1
43964		(740) 537-3607	blisa@yahoo.com	1	1	0	1	0
95119	(408) 953-6001	(408) 953-6000		6	2	0	1	0
43223	(614) 405-2253	(614) 405-2252	clisa@yahoo.com	1	2	1	1	1
95119	(408) 782-7100	(408) 782-7100		3	3	0	1	1
95136	(408) 224-5446		george@rocketmail.com	1	1	0	1	0
10025	(212) 933-0681			3	4	1	1	0
95112	(408) 929-6651		mathew_ciby@yahoo.com	2	3	1	1	1
95119	(408) 226-2321	(408) 226-2327	mymail@yahoo.com	2	2	0	1	0
95119	(408) 900-6401	(408) 782-3700		2	4	0	1	1
95128	(408) 995-5347		e_laura@gmail.com	3	4	1	1	1
95112	(408) 929-6651		mathew_ciby@yahoo.com	2	3	1	1	1
95131		(408) 923-5187	drussel@gmail.com	4	4	1	1	1
95119	(408) 365-1091		anymail@hotmail.com	1	1	1	1	0
95119				2	4	0	1	1
43569		(419) 669-2025	lisa@gmail.com	1	4	1	1	1
43223	(614) 405-2253	(614) 405-2252	clisa@yahoo.com	1	2	1	1	1
95136	(408) 224-5446		george@rocketmail.com	1	1	0	1	0
95119	(800) 817-8231	(800) 817-8232	anders@olsson.com	5	1	1	1	0
95119	(408) 226-2321	(408) 226-2327	mymail@yahoo.com	2	2	0	1	0
95119	(408) 365-1091	(408) 553-8211	anymail@hotmail.com	1	1	1	1	0
10549	(914) 666-5688			4	4	1	1	1
95119	(408) 782-7100	(408) 782-7100		3	3	0	1	1
10025	(212) 933-0681			3	4	1	1	0
95119	(408) 267-0751	(408) 267-0755		4	4	0	1	0
95112	(408) 929-6651		mathew_ciby@yahoo.com	2	3	1	1	1
95131		(408) 923-5187	drussel@gmail.com	4	4	1	1	1
95119	(408) 365-1091		anymail@hotmail.com	1	1	1	1	0
43569		(419) 669-2025	lisa@gmail.com	1	4	1	1	1
95119	(408) 900-6401	(408) 782-3700		2	4	0	1	1

Figure 1-407 Create J00_SRC_MAILING_LIST job 9/9

J00A_INV_MAILING_LIST

In this step, we use the Investigate stage on non-text columns such as source (of the mailing list), home and cell phone numbers, and e-mail addresses. We used character concatenate and character discrete with combinations of C, T, and X masks. The objective of this step is to validate some of the main non-text columns. Any errors that are detected can be resolved by modifying the source directly or using IBM WebSphere QualityStage jobs to cleanse and modify the targets.

Figure 1-408 on page 349 shows the various stages that are used in this job, including the data set that was created in "J00_SRC_MAILING_LIST" on page 342, one COPY stage, three Investigate stages, and one Sequential File stage for each Investigate stage. We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object that was created earlier.

Because the configuration of this job is very similar to that described in "J02_INVCC_ISCODE" on page 186, we do not repeat it here. However, some of the configurations of interest are as follows:

- Figure 1-409 on page 349 shows the character discrete investigate columns CELL_PHONE, EMAIL, and HOME_PHONE configured with the *C* mask.

  Figure 1-413 on page 352 shows the report that is generated for this Investigate stage after compiling and running (Figure 1-412 on page 351) this job.

- Figure 1-410 on page 350 shows the character discrete investigate columns CELL_PHONE, EMAIL, and HOME_PHONE configured with the *T* mask.

  Figure 1-414 on page 352 shows the report that is generated for this Investigate stage after compiling and running this job.

- Figure 1-411 on page 350 shows the character concatenate investigate columns SOURCE, CELL_PHONE, EMAIL, and HOME_PHONE configured with a combination of *C*, *T*, and *X* masks. This investigate stage reveals the source (of the mailing list purchase) and the telephone and e-mail addresses that are provided by them.

  Figure 1-415 on page 353 shows the report that is generated for this Investigate stage after compiling and running this job.

Note: You need to review these reports for validity and then correct them using data cleansing techniques. Time constraints during our testing prevented us from creating data cleansing jobs to do so.

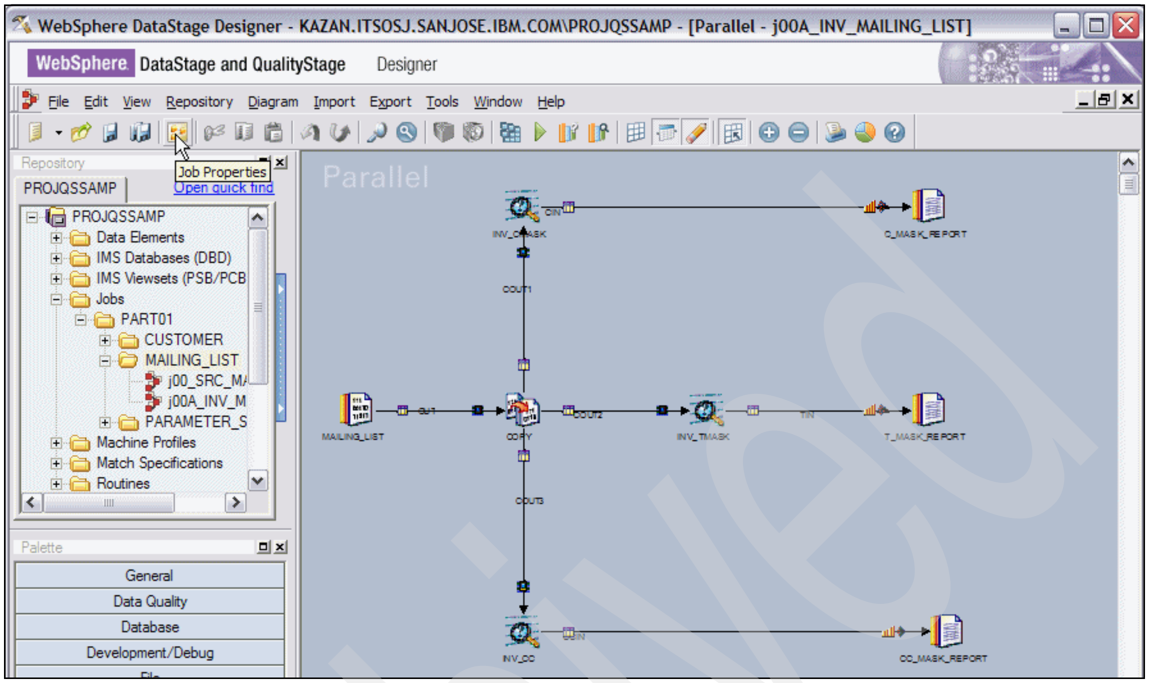
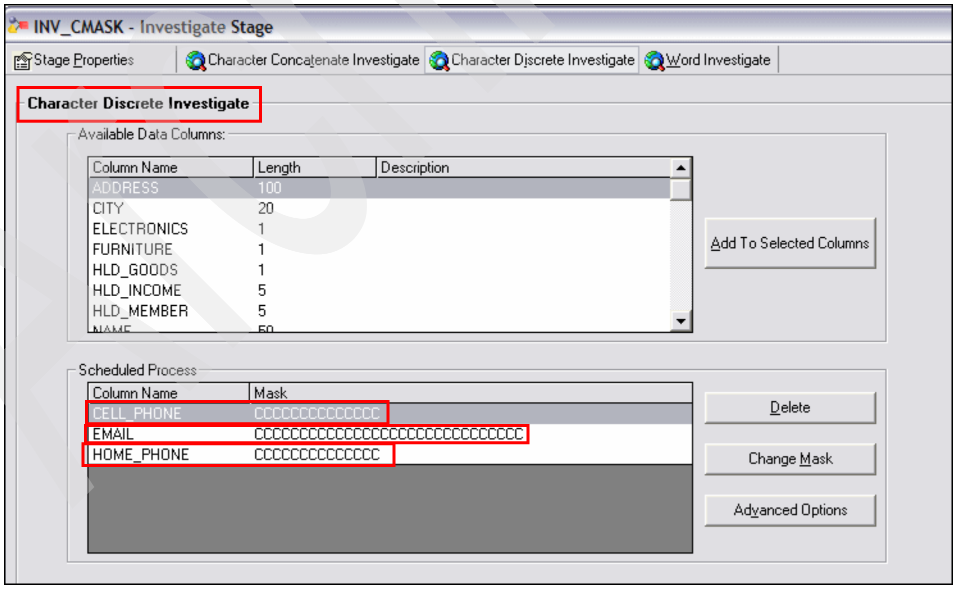


Figure 1-408 Create J00A_INV_MAILING_LIST job 1/8


INV_CMASK - Investigate Stage
Stage Properties
Character Concatenate Investigate
Character Discrete Investigate
Word Investigate
Character Discrete Investigate
Add To Selected Columns
Delete
Change Mask
Advanced Options

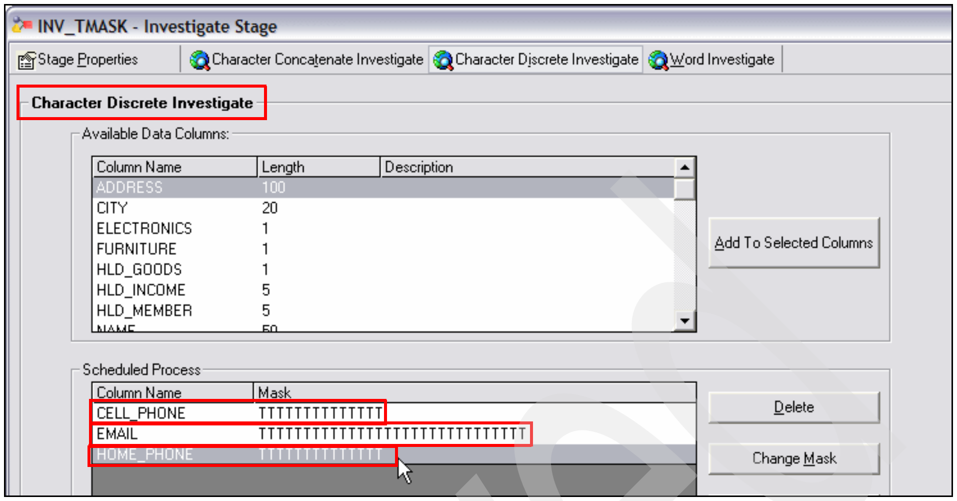
Available Data Columns:

Column Name	Length	Description
ADDRESS	100	
CITY	20	
ELECTRONICS	1	
FURNITURE	1	
HLD_GOODS	1	
HLD_INCOME	5	
HLD_MEMBER	5	
NAME	50	

Scheduled Process

Column Name	Mask
CELL_PHONE	CCCCCCCCCCCCCC
EMAIL	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
HOME_PHONE	CCCCCCCCCCCCCC

Figure 1-409 Create J00A_INV_MAILING_LIST job 2/8


INV_TMASK - Investigate Stage
Stage Properties
Character Concatenate Investigate
Character Discrete Investigate
Word Investigate
Character Discrete Investigate
Add To Selected Columns
Delete
Change Mask

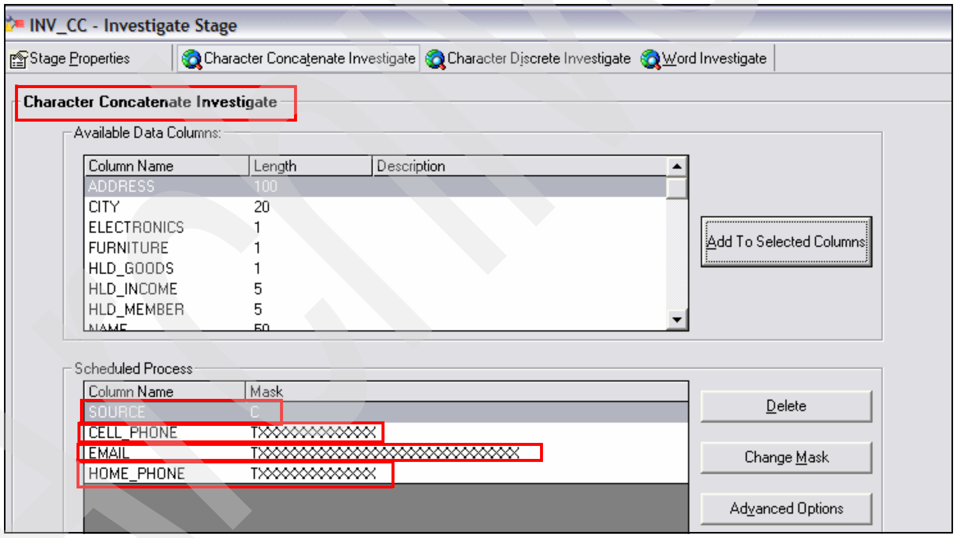
Available Data Columns:

Column Name	Length	Description
ADDRESS	100	
CITY	20	
ELECTRONICS	1	
FURNITURE	1	
HLD_GOODS	1	
HLD_INCOME	5	
HLD_MEMBER	5	
NAME	50	

Scheduled Process

Column Name	Mask
CELL_PHONE	TTTTTTTTTTTTTT
EMAIL	TTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
HOME_PHONE	TTTTTTTTTTTTTT

Figure 1-410 Create J00A_INV_MAILING_LIST job 3/8


INV_CC - Investigate Stage
Stage Properties
Character Concatenate Investigate
Character Discrete Investigate
Word Investigate
Character Concatenate Investigate
Add To Selected Columns
Delete
Change Mask
Advanced Options

Available Data Columns:

Column Name	Length	Description
ADDRESS	100	
CITY	20	
ELECTRONICS	1	
FURNITURE	1	
HLD_GOODS	1	
HLD_INCOME	5	
HLD_MEMBER	5	
NAME	50	

Scheduled Process

Column Name	Mask
SOURCE	C
CELL_PHONE	TXXXXXXXXXXXXX
EMAIL	TXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
HOME_PHONE	TXXXXXXXXXXXXX

Figure 1-411 Create J00A_INV_MAILING_LIST job 4/8

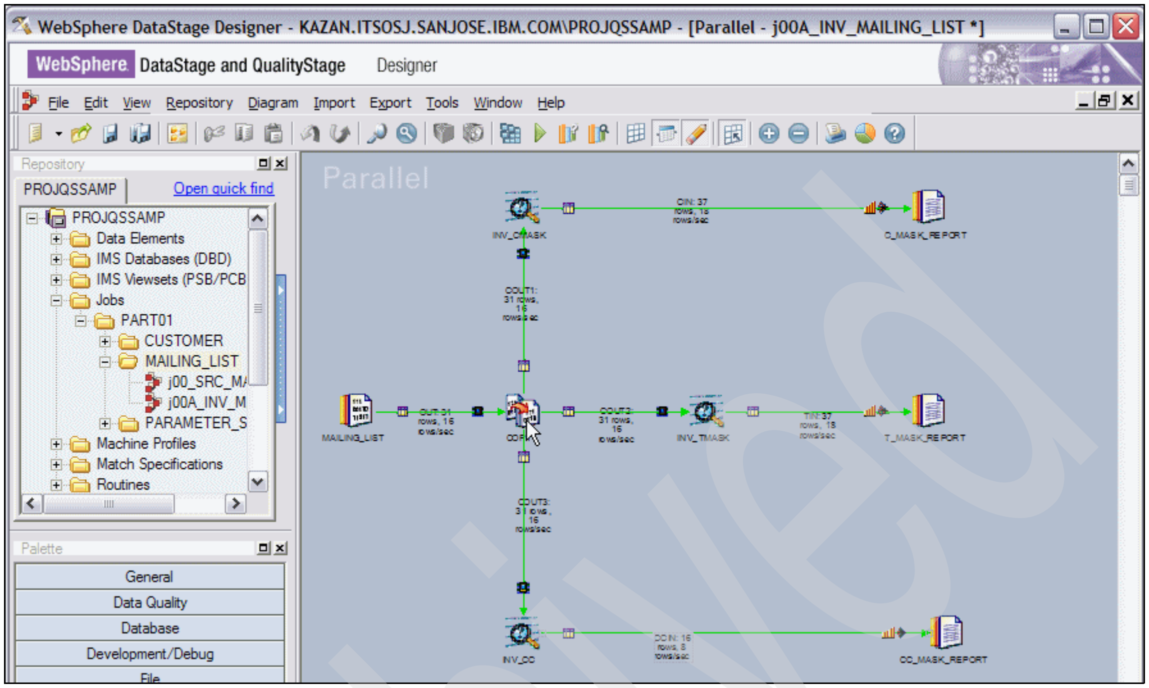
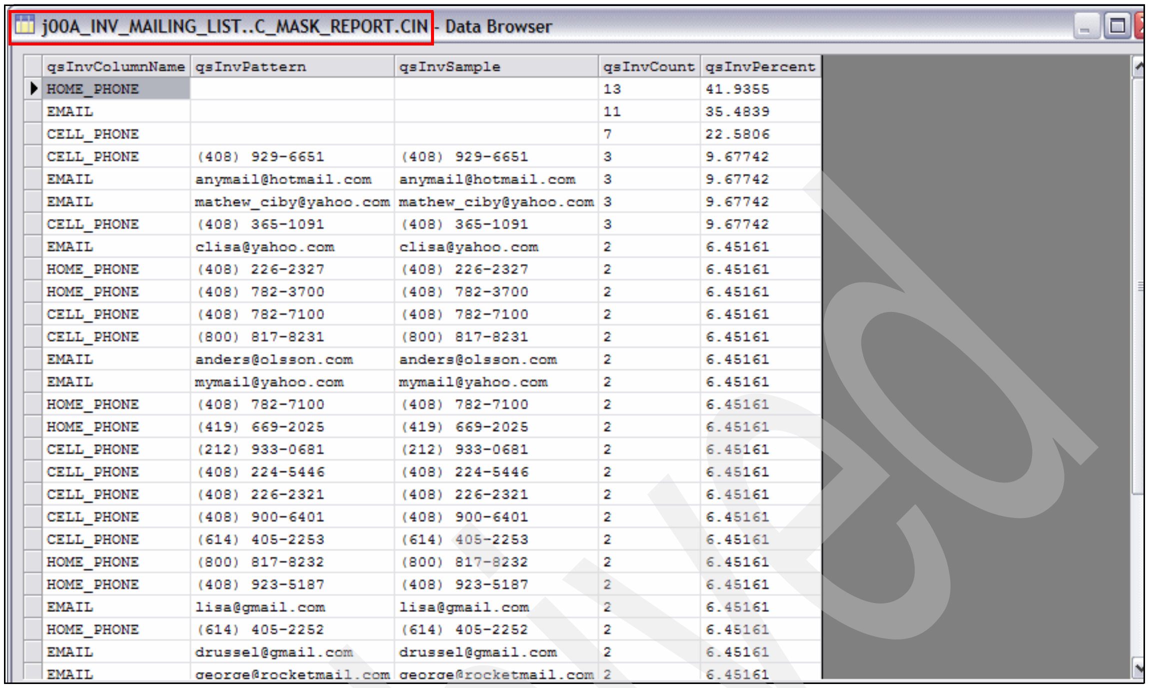

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j00A_INV_MAILING_LIST *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Repository
PROJQSSAMP
Open quick find
PROJQSSAMP
Data Elements
IMS Databases (DBD)
IMS Viewsets (PSB/PCB
Jobs
PART01
CUSTOMER
MAILING_LIST
j00_SRC_MA
j00A_INV_M
PARAMETER_S
Machine Profiles
Match Specifications
Routines
Palette
General
Data Quality
Database
Development/Debug
File
Parallel
INV_CMASK
OIN: 37 rows, 18 rows/sec
C_MASK_REPORT
COUT1: 31 rows, 16 rows/sec
MAILING_LIST
OUT: 31 rows, 16 rows/sec
COPY
COUT2: 31 rows, 16 rows/sec
INV_TMASK
TIN: 37 rows, 18 rows/sec
T_MASK_REPORT
COUT3: 31 rows, 16 rows/sec
INV_CC
CCIN: 16 rows, 8 rows/sec
CC_MASK_REPORT

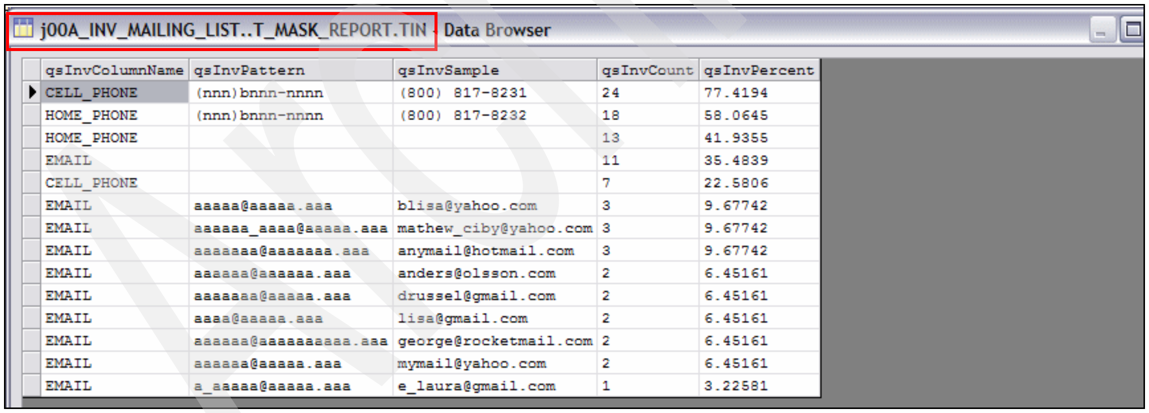
Figure 1-412   Create J00A_INV_MAILING_LIST job 5/8



j00A_INV_MAILING_LIST..C_MASK_REPORT.CIN - Data Browser

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
HOME_PHONE			13	41.9355
EMAIL			11	35.4839
CELL_PHONE			7	22.5806
CELL_PHONE	(408) 929-6651	(408) 929-6651	3	9.67742
EMAIL	anymail@hotmail.com	anymail@hotmail.com	3	9.67742
EMAIL	mathew_ciby@yahoo.com	mathew_ciby@yahoo.com	3	9.67742
CELL_PHONE	(408) 365-1091	(408) 365-1091	3	9.67742
EMAIL	clisa@yahoo.com	clisa@yahoo.com	2	6.45161
HOME_PHONE	(408) 226-2327	(408) 226-2327	2	6.45161
HOME_PHONE	(408) 782-3700	(408) 782-3700	2	6.45161
CELL_PHONE	(408) 782-7100	(408) 782-7100	2	6.45161
CELL_PHONE	(800) 817-8231	(800) 817-8231	2	6.45161
EMAIL	anders@olsson.com	anders@olsson.com	2	6.45161
EMAIL	mymail@yahoo.com	mymail@yahoo.com	2	6.45161
HOME_PHONE	(408) 782-7100	(408) 782-7100	2	6.45161
HOME_PHONE	(419) 669-2025	(419) 669-2025	2	6.45161
CELL_PHONE	(212) 933-0681	(212) 933-0681	2	6.45161
CELL_PHONE	(408) 224-5446	(408) 224-5446	2	6.45161
CELL_PHONE	(408) 226-2321	(408) 226-2321	2	6.45161
CELL_PHONE	(408) 900-6401	(408) 900-6401	2	6.45161
CELL_PHONE	(614) 405-2253	(614) 405-2253	2	6.45161
HOME_PHONE	(800) 817-8232	(800) 817-8232	2	6.45161
HOME_PHONE	(408) 923-5187	(408) 923-5187	2	6.45161
EMAIL	lisa@gmail.com	lisa@gmail.com	2	6.45161
HOME_PHONE	(614) 405-2252	(614) 405-2252	2	6.45161
EMAIL	drussel@gmail.com	drussel@gmail.com	2	6.45161
EMAIL	george@rocketmail.com	george@rocketmail.com	2	6.45161

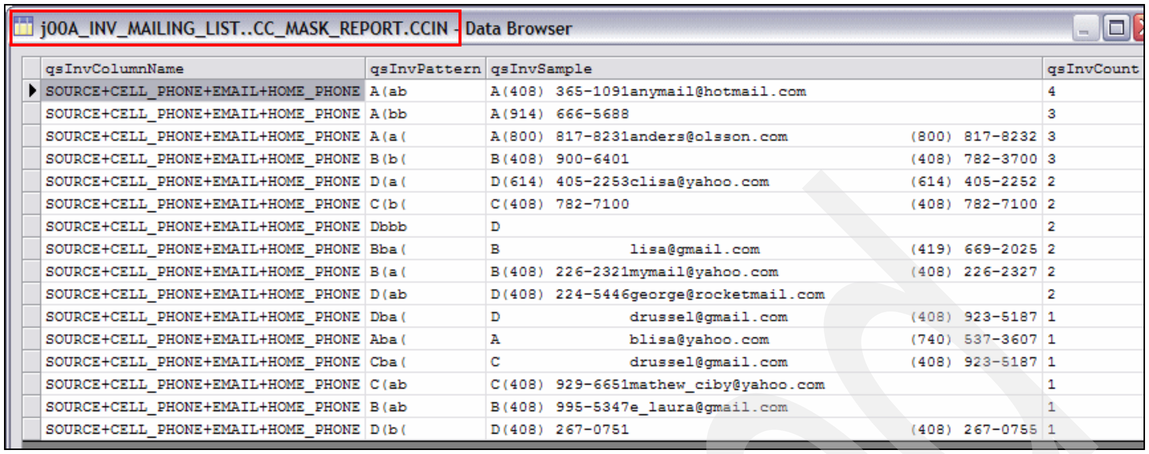
Figure 1-413 Create J00A_INV_MAILING_LIST job 6/8



j00A_INV_MAILING_LIST..T_MASK_REPORT.TIN - Data Browser

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
CELL_PHONE	(nnn)bnnn-nnnn	(800) 817-8231	24	77.4194
HOME_PHONE	(nnn)bnnn-nnnn	(800) 817-8232	18	58.0645
HOME_PHONE			13	41.9355
EMAIL			11	35.4839
CELL_PHONE			7	22.5806
EMAIL	aaaaa@aaaaa.aaa	blisa@yahoo.com	3	9.67742
EMAIL	aaaaaa_aaaa@aaaaa.aaa	mathew_ciby@yahoo.com	3	9.67742
EMAIL	aaaaaaa@aaaaaaa.aaa	anymail@hotmail.com	3	9.67742
EMAIL	aaaaaa@aaaaaa.aaa	anders@olsson.com	2	6.45161
EMAIL	aaaaaaa@aaaaa.aaa	drussel@gmail.com	2	6.45161
EMAIL	aaaa@aaaaa.aaa	lisa@gmail.com	2	6.45161
EMAIL	aaaaaa@aaaaaaaaaa.aaa	george@rocketmail.com	2	6.45161
EMAIL	aaaaaa@aaaaa.aaa	mymail@yahoo.com	2	6.45161
EMAIL	a_aaaaa@aaaaa.aaa	e_laura@gmail.com	1	3.22581

Figure 1-414 Create J00A_INV_MAILING_LIST job 7/8



qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount
SOURCE+CELL_PHONE+EMAIL+HOME_PHONE	A(ab	A(408) 365-1091anymail@hotmail.com	4
SOURCE+CELL_PHONE+EMAIL+HOME_PHONE	A(bb	A(914) 666-5688	3
SOURCE+CELL_PHONE+EMAIL+HOME_PHONE	A(a(	A(800) 817-8231anders@olsson.com (800) 817-8232	3
SOURCE+CELL_PHONE+EMAIL+HOME_PHONE	B(b(	B(408) 900-6401 (408) 782-3700	3
SOURCE+CELL_PHONE+EMAIL+HOME_PHONE	D(a(	D(614) 405-2253clisa@yahoo.com (614) 405-2252	2
SOURCE+CELL_PHONE+EMAIL+HOME_PHONE	C(b(	C(408) 782-7100 (408) 782-7100	2
SOURCE+CELL_PHONE+EMAIL+HOME_PHONE	Dbbb	D	2
SOURCE+CELL_PHONE+EMAIL+HOME_PHONE	Bba(	B lisa@gmail.com (419) 669-2025	2
SOURCE+CELL_PHONE+EMAIL+HOME_PHONE	B(a(	B(408) 226-2321mymail@yahoo.com (408) 226-2327	2
SOURCE+CELL_PHONE+EMAIL+HOME_PHONE	D(ab	D(408) 224-5446george@rocketmail.com	2
SOURCE+CELL_PHONE+EMAIL+HOME_PHONE	Dba(	D drussel@gmail.com (408) 923-5187	1
SOURCE+CELL_PHONE+EMAIL+HOME_PHONE	Aba(	A blisa@yahoo.com (740) 537-3607	1
SOURCE+CELL_PHONE+EMAIL+HOME_PHONE	Cba(	C drussel@gmail.com (408) 923-5187	1
SOURCE+CELL_PHONE+EMAIL+HOME_PHONE	C(ab	C(408) 929-6651mathew_ciby@yahoo.com	1
SOURCE+CELL_PHONE+EMAIL+HOME_PHONE	B(ab	B(408) 995-5347e_laura@gmail.com	1
SOURCE+CELL_PHONE+EMAIL+HOME_PHONE	D(b(	D(408) 267-0751 (408) 267-0755	1

Figure 1-415 Create J00A_INV_MAILING_LIST job 8/8

J01_STAN_COUNTRY_M

Next, we split the mailing list records that were created in the "J00_SRC_MAILING_LIST" on page 342 job into two files—one with name and address fields and another for single domain fields such as telephone numbers and e-mail addresses. As in the case of the credit card customer file cleansing, we do this to allow processing of the single domain and text data in parallel by the QualityStage administrators and appropriate subject matter experts.

Note: In this case, for convenience, we chose to use all the columns in the input data for the both streams. Job J01_STAN_COUNTRY_M (described here) for the name and address fields, and job "J00A_INV_MAILING_LIST" on page 347 for the source, home and cell phone numbers, and e-mail addresses.

In this step, we analyze the mailing list's addresses to determine their (ISO code) country using the COUNTRY rule set in the Standardize stage.

Figure 1-416 on page 354 shows the various stages that are used in this job, including the data set that was created in "J00_SRC_MAILING_LIST" on page 342, a Standardize stage, and an output Data Set stage. This job uses a QSPARAMETERSET object that is similar to the one described in 1.10.4, "Create a parameter set object" on page 130. We modified the names of the stages as shown, and this job uses the QSPARAMETERSET object that was created earlier.

Because the configuration of this job is very similar to that described in “J01_STAN_COUNTRY” on page 168, we do not repeat it here. However, some of the configurations of interest are as follows:

- Figure 1-417 on page 355 and Figure 1-418 on page 355 show the Standardize Rule Process window with the configured COUNTRY rule set and the literal ZQUSZQ followed by the CITY, STATE and ZIP columns in the Selected Columns list.
- After saving, compiling, and running the job, Figure 1-419 on page 356 shows the results of the execution.
- Figure 1-420 on page 357 and Figure 1-421 on page 358 shows a report with the two columns ISOCountryCode_COUNTRY and IdentifierFlag_COUNTRY added by the Standardize stage. It shows `US` for each row with the identifier flag of `Y` in all but two cases that had invalid state codes. It defaulted the country code to `US` because of the inclusion of the literal ZQUSZQ.

Now, proceed to “J02_INVCC_ISOCODE_M” on page 358.

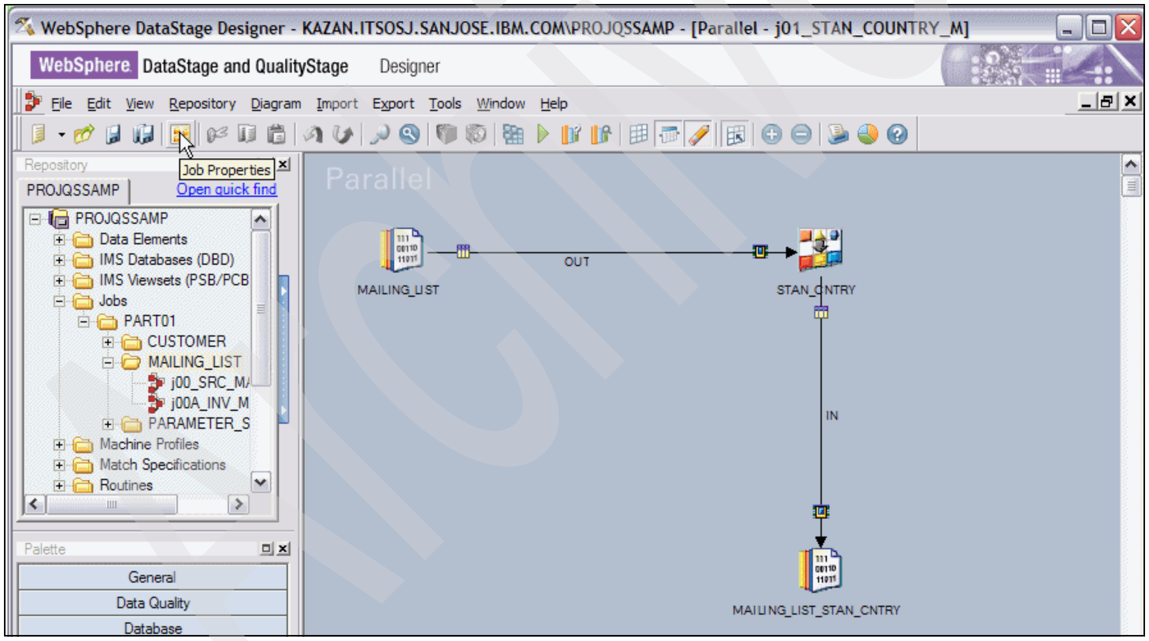
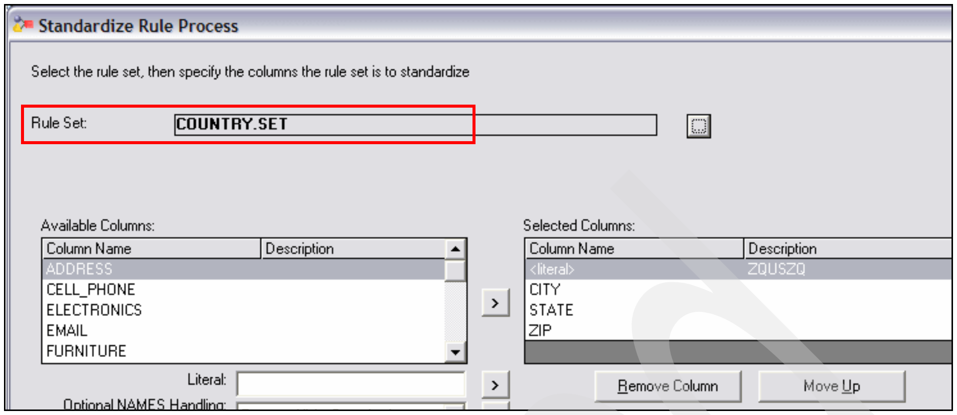


Figure 1-416 Create J01_STAN_COUNTRY_M job 1/6


Standardize Rule Process

Select the rule set, then specify the columns the rule set is to standardize

Rule Set: COUNTRY.SET

Available Columns:

Column Name	Description
ADDRESS	
CELL_PHONE	
ELECTRONICS	
EMAIL	
FURNITURE	

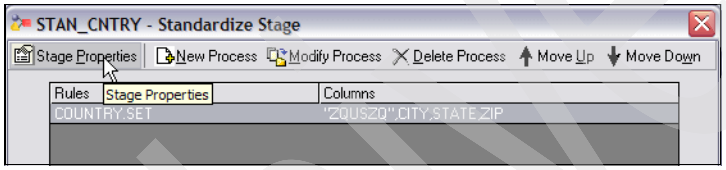
Selected Columns:

Column Name	Description
<literal>	ZQUSZQ
CITY	
STATE	
ZIP	

Literal:

Optional NAMES Handling:

Figure 1-417 Create J01_STAN_COUNTRY_M job 2/6


STAN_CNTRY - Standardize Stage

Rules	Columns
COUNTRY.SET	"ZQUSZQ",CITY,STATE,ZIP

Figure 1-418 Create J01_STAN_COUNTRY_M job 3/6

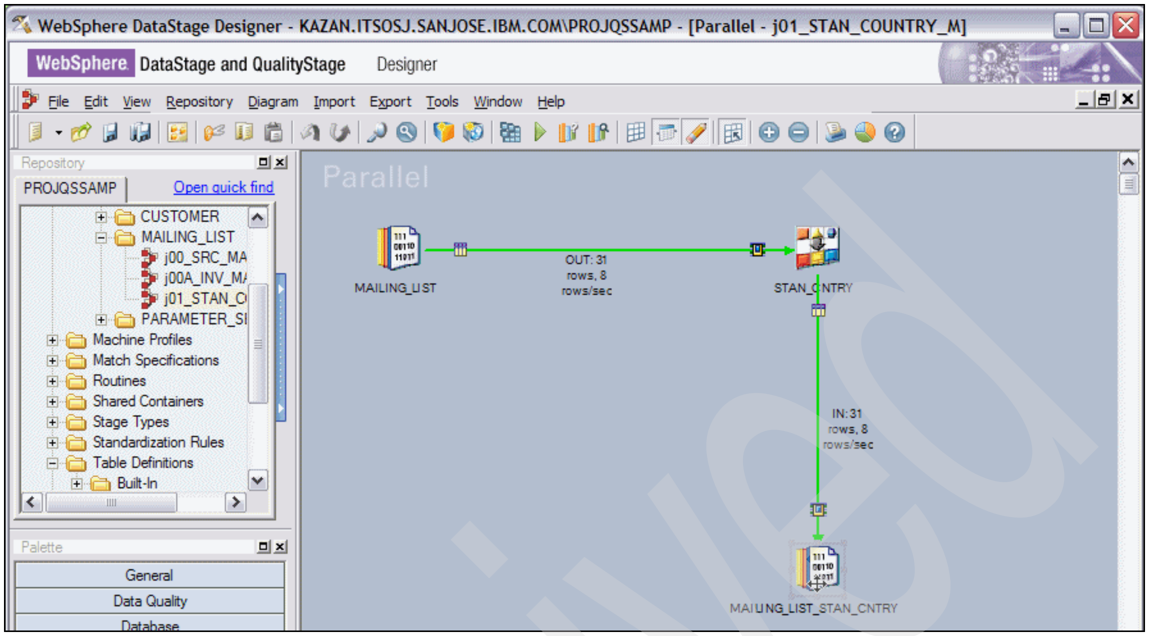
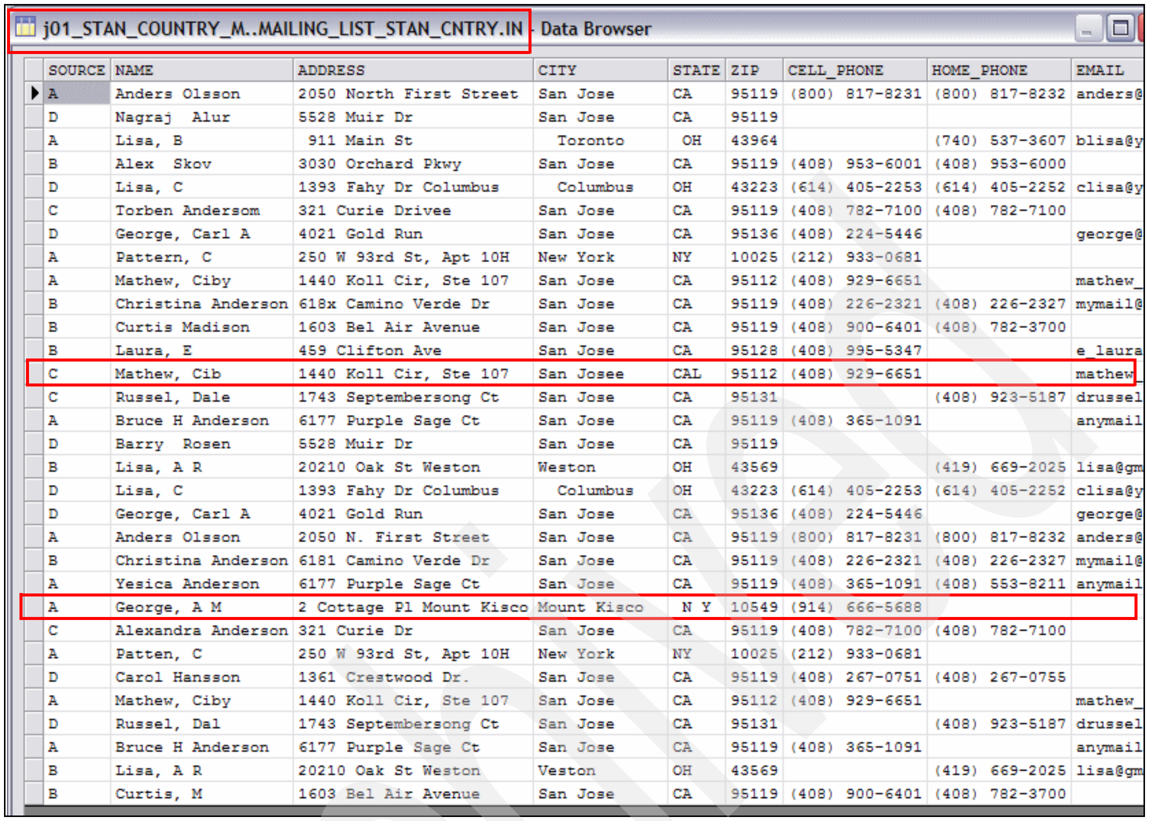


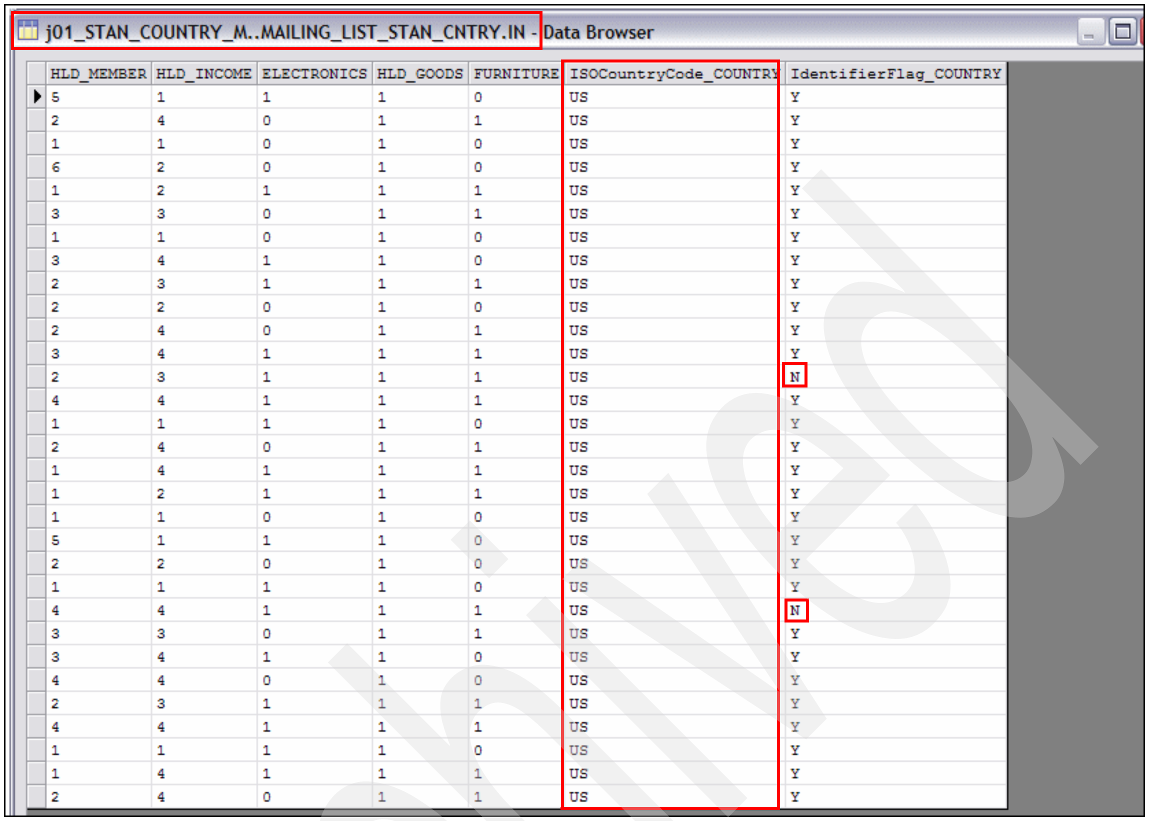
Figure 1-419 Create J01_STAN_COUNTRY_M job 4/6



j01_STAN_COUNTRY_M..MAILING_LIST_STAN_CNTRY.IN - Data Browser

SOURCE	NAME	ADDRESS	CITY	STATE	ZIP	CELL_PHONE	HOME_PHONE	EMAIL
A	Anders Olsson	2050 North First Street	San Jose	CA	95119	(800) 817-8231	(800) 817-8232	anders@
D	Nagraj Alur	5528 Muir Dr	San Jose	CA	95119			
A	Lisa, B	911 Main St	Toronto	OH	43964		(740) 537-3607	blisa@y
B	Alex Skov	3030 Orchard Pkwy	San Jose	CA	95119	(408) 953-6001	(408) 953-6000	
D	Lisa, C	1393 Fahy Dr Columbus	Columbus	OH	43223	(614) 405-2253	(614) 405-2252	clisa@y
C	Torben Andersom	321 Curie Drivee	San Jose	CA	95119	(408) 782-7100	(408) 782-7100	
D	George, Carl A	4021 Gold Run	San Jose	CA	95136	(408) 224-5446		george@
A	Pattern, C	250 W 93rd St, Apt 10H	New York	NY	10025	(212) 933-0681		
A	Mathew, Ciby	1440 Koll Cir, Ste 107	San Jose	CA	95112	(408) 929-6651		mathew_
B	Christina Anderson	618x Camino Verde Dr	San Jose	CA	95119	(408) 226-2321	(408) 226-2327	mymail@
B	Curtis Madison	1603 Bel Air Avenue	San Jose	CA	95119	(408) 900-6401	(408) 782-3700	
B	Laura, E	459 Clifton Ave	San Jose	CA	95128	(408) 995-5347		e_laura
C	Mathew, Cib	1440 Koll Cir, Ste 107	San Josee	CAL	95112	(408) 929-6651		mathew_
C	Russel, Dale	1743 Septembersong Ct	San Jose	CA	95131		(408) 923-5187	drussel
A	Bruce H Anderson	6177 Purple Sage Ct	San Jose	CA	95119	(408) 365-1091		anymail
D	Barry Rosen	5528 Muir Dr	San Jose	CA	95119			
B	Lisa, A R	20210 Oak St Weston	Weston	OH	43569		(419) 669-2025	lisa@gm
D	Lisa, C	1393 Fahy Dr Columbus	Columbus	OH	43223	(614) 405-2253	(614) 405-2252	clisa@y
D	George, Carl A	4021 Gold Run	San Jose	CA	95136	(408) 224-5446		george@
A	Anders Olsson	2050 N. First Street	San Jose	CA	95119	(800) 817-8231	(800) 817-8232	anders@
B	Christina Anderson	6181 Camino Verde Dr	San Jose	CA	95119	(408) 226-2321	(408) 226-2327	mymail@
A	Yesica Anderson	6177 Purple Sage Ct	San Jose	CA	95119	(408) 365-1091	(408) 553-8211	anymail
A	George, A M	2 Cottage Pl Mount Kisco	Mount Kisco	N Y	10549	(914) 666-5688		
C	Alexandra Anderson	321 Curie Dr	San Jose	CA	95119	(408) 782-7100	(408) 782-7100	
A	Patten, C	250 W 93rd St, Apt 10H	New York	NY	10025	(212) 933-0681		
D	Carol Hansson	1361 Crestwood Dr.	San Jose	CA	95119	(408) 267-0751	(408) 267-0755	
A	Mathew, Ciby	1440 Koll Cir, Ste 107	San Jose	CA	95112	(408) 929-6651		mathew_
D	Russel, Dal	1743 Septembersong Ct	San Jose	CA	95131		(408) 923-5187	drussel
A	Bruce H Anderson	6177 Purple Sage Ct	San Jose	CA	95119	(408) 365-1091		anymail
B	Lisa, A R	20210 Oak St Weston	Veston	OH	43569		(419) 669-2025	lisa@gm
B	Curtis, M	1603 Bel Air Avenue	San Jose	CA	95119	(408) 900-6401	(408) 782-3700	

Figure 1-420 Create J01_STAN_COUNTRY_M job 5/6



j01_STAN_COUNTRY_M..MAILING_LIST_STAN_CNTRY.IN - Data Browser

HLD_MEMBER	HLD_INCOME	ELECTRONICS	HLD_GOODS	FURNITURE	ISOCountryCode_COUNTRY	IdentifierFlag_COUNTRY
5	1	1	1	0	US	Y
2	4	0	1	1	US	Y
1	1	0	1	0	US	Y
6	2	0	1	0	US	Y
1	2	1	1	1	US	Y
3	3	0	1	1	US	Y
1	1	0	1	0	US	Y
3	4	1	1	0	US	Y
2	3	1	1	1	US	Y
2	2	0	1	0	US	Y
2	4	0	1	1	US	Y
3	4	1	1	1	US	Y
2	3	1	1	1	US	N
4	4	1	1	1	US	Y
1	1	1	1	0	US	Y
2	4	0	1	1	US	Y
1	4	1	1	1	US	Y
1	2	1	1	1	US	Y
1	1	0	1	0	US	Y
5	1	1	1	0	US	Y
2	2	0	1	0	US	Y
1	1	1	1	0	US	Y
4	4	1	1	1	US	N
3	3	0	1	1	US	Y
3	4	1	1	0	US	Y
4	4	0	1	0	US	Y
2	3	1	1	1	US	Y
4	4	1	1	1	US	Y
1	1	1	1	0	US	Y
1	4	1	1	1	US	Y
2	4	0	1	1	US	Y

Figure 1-421 Create J01_STAN_COUNTRY_M job 6/6

J02_INVCC_ISOCODE_M

In this step, we analyze the ISO codes that were generated in the "J01_STAN_COUNTRY_M" on page 353 job by the Investigate stage using character discrete with the *C* mask to obtain frequency distribution. This step identifies whether the addresses in the mailing list file belong to more than one country and identifies the codes of the countries in the addresses. In this case, all the addresses are U.S. addresses.

Figure 1-422 on page 359 shows the various stages that are used in this job, including the data set that was created in "J01_STAN_COUNTRY_M" on page 353, an Investigate stage, and an output Sequential File stage. We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object that was created earlier.

Because the configuration of this job is very similar to that described in "J02_INVCC_ISCODE" on page 186, we do not repeat it here. However, some of the configurations of interest are as follows:

- Figure 1-423 on page 360 shows the INV_CC_ISOCODE - Investigate Stage window with the Character Concatenate Investigate option and columns ISOCountryCode_Country and IdentifierFlag_COUNTRY selected with the *C* masks for each character.
- After saving, compiling, and running this job, the job statistics are shown in Figure 1-424 on page 360.
- The output of the Investigate stage written to the sequential file is shown in Figure 1-425 on page 361. It shows two records:
  - One record with a concatenated value of `US Y` in 93.5484% of the records
  - Another record with a concatenated value of `US N` in 6.45161% of the records

Now, proceed to "J03_STAN_USPREP_M" on page 361.

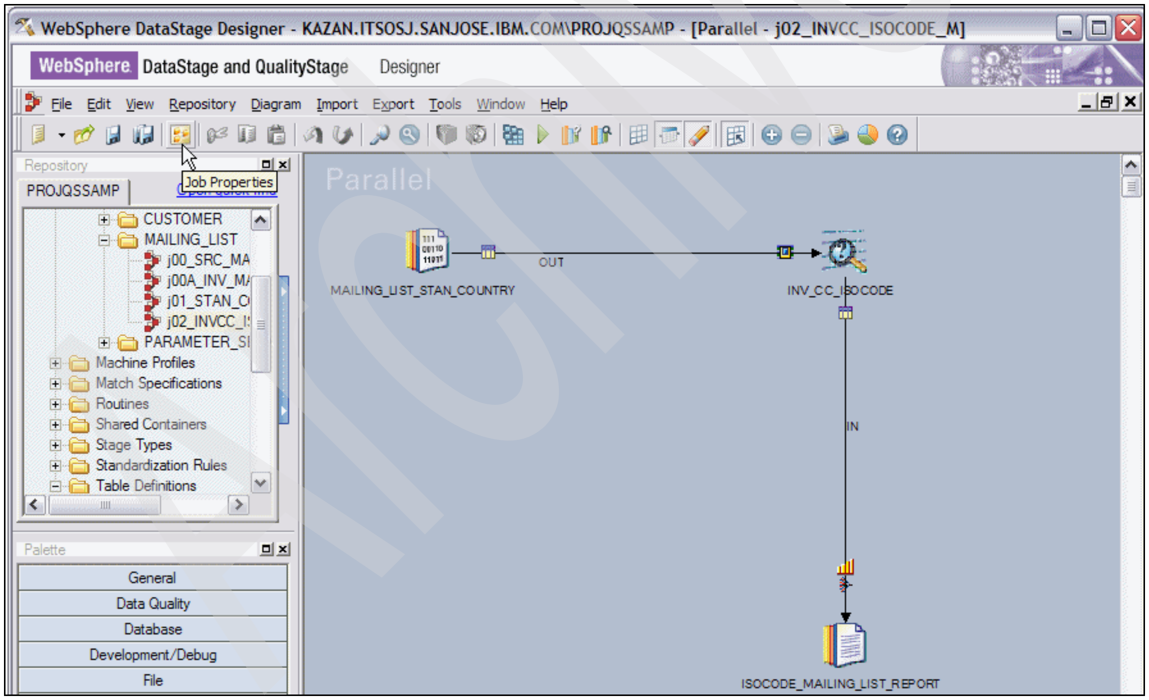
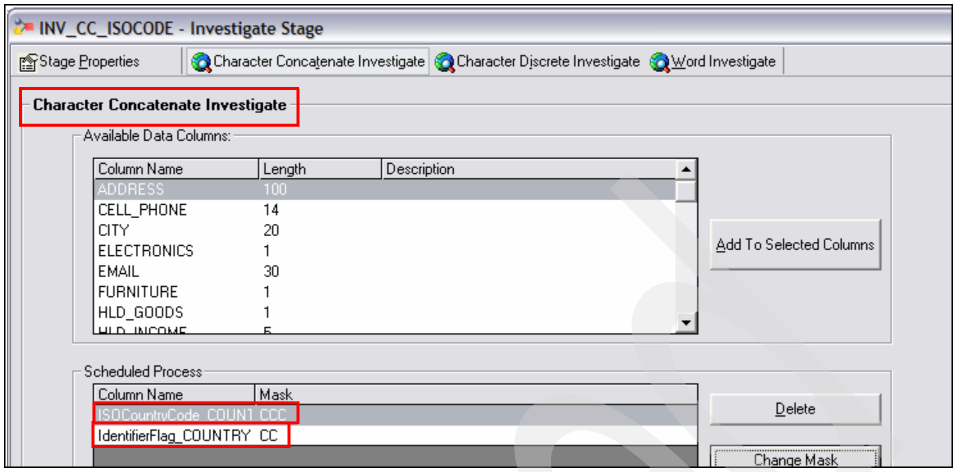


Figure 1-422 Create J02_INVCC_ISOCODE_M job 1/4


INV_CC_ISOCODE - Investigate Stage

Stage Properties | Character Concatenate Investigate | Character Discrete Investigate | Word Investigate

Character Concatenate Investigate

Available Data Columns:

Column Name	Length	Description
ADDRESS	100	
CELL_PHONE	14	
CITY	20	
ELECTRONICS	1	
EMAIL	30	
FURNITURE	1	
HLD_GOODS	1	
HLD_INCOME	5	

Add To Selected Columns

Scheduled Process

Column Name	Mask
ISOCountryCode_COUNT CCC	
IdentifierFlag_COUNTRY CC	

Delete

Change Mask

Figure 1-423 Create J02_INVCC_ISOCODE_M job 2/4

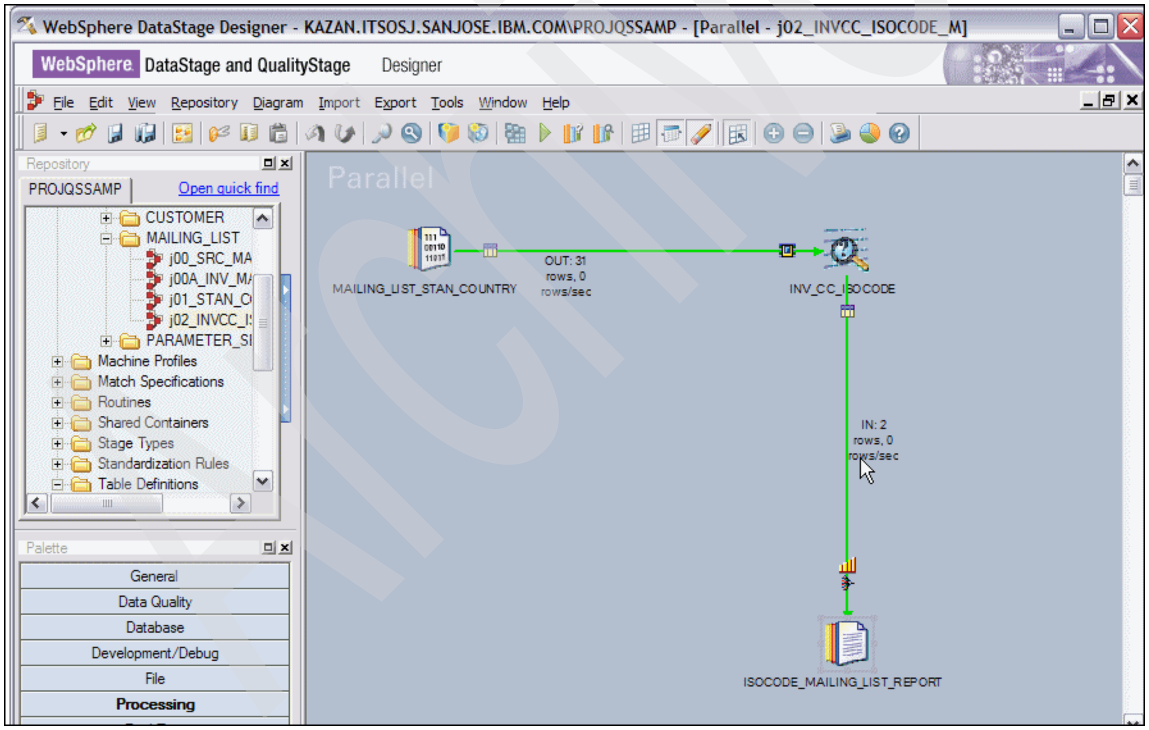
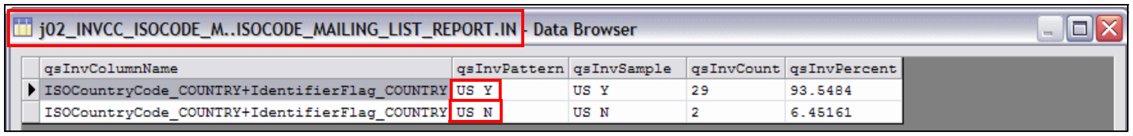

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j02_INVCC_ISOCODE_M]

Figure 1-424 Create J02_INVCC_ISOCODE_M job 3/4


j02_INVCC_ISOCODE_M..ISOCODE_MAILING_LIST_REPORT.IN - Data Browser

	qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
▶	ISOCountryCode_COUNTRY+IdentifierFlag_COUNTRY	US Y	US Y	29	93.5484
	ISOCountryCode_COUNTRY+IdentifierFlag_COUNTRY	US N	US N	2	6.45161

Figure 1-425 Create J02_INVCC_ISOCODE_M job 4/4

J03_STAN_USPREP_M

In this step, we use the Standardize stage with the domain-preprocessor rule set USPREP to move name and address data into Name, Address, and Area domains.

Figure 1-426 on page 362 shows the various stages that are used in this job, including the data set that was created in “J01_STAN_COUNTRY_M” on page 353, a Standardize stage, and an output Data Set stage. We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object created earlier.

Because the configuration of this job is very similar to that described in “J03_STAN_USPREP” on page 196, we do not repeat it here. However, some of the configurations of interest are as follows:

- The Standardize Rule Process windows of Figure 1-427 on page 362 and Figure 1-428 on page 363 show the USPREP rule set and the literals and columns selected for analysis. It shows the following selected literals and columns:

  ```
  ZQNAMEZQ NAME ZQADDRZQ ADDRESS ZQAREAZQ CITY STATE ZIP
  ```

- After saving, compiling, and running this job, view the results as shown in Figure 1-429 on page 363.
- The content of the MAILING_LIST_STAN_USPREP data set is shown in Figure 1-430 on page 364 through Figure 1-432 on page 366.

  The report shows the following:

  – Columns NameDomain_USPREP (contains prefix, first name, last name, suffix tokens), AddressDomain_USPREP (contains apartment, street name and street type tokens), and AreaDomain_USPREP (contains city, state, ZIP code tokens) that was parsed from the input columns.

  – InputPattern_USPREP and OutboundPattern_USPREP columns that contain the patterns generated after processing the name and address columns in the input file.

Now, proceed to “J04_INVW_USPREP_M” on page 366.

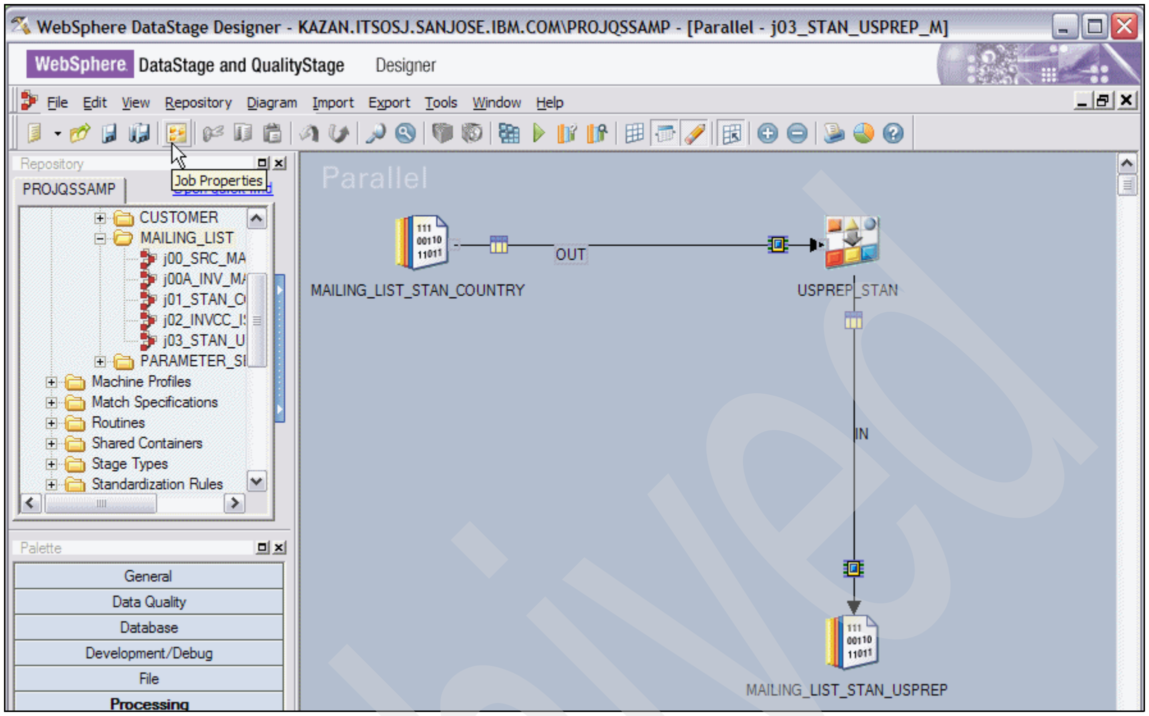
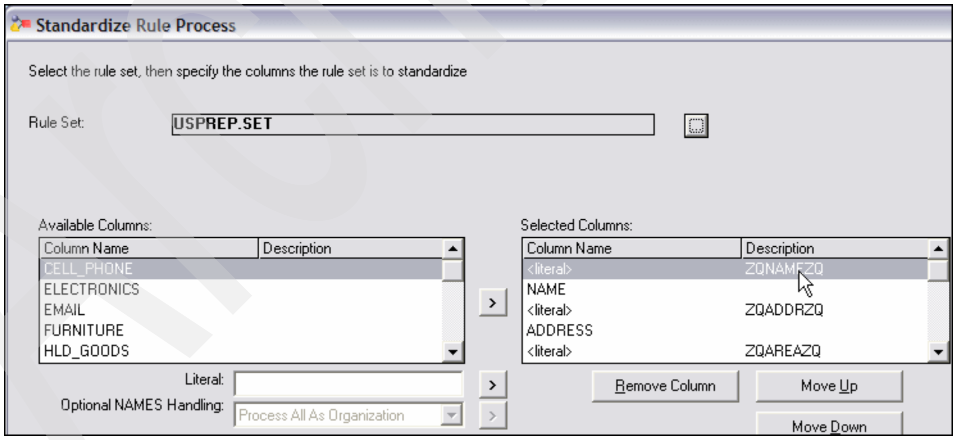


Figure 1-426 Create J03_STAN_USPREP_M job 1/7



Rule Set: USPREP.SET

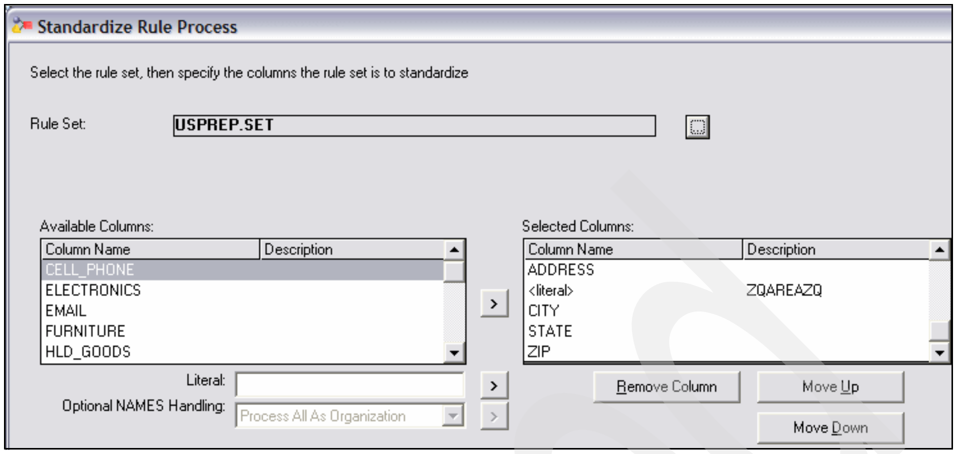
Available Columns:

Column Name	Description
CELL_PHONE	
ELECTRONICS	
EMAIL	
FURNITURE	
HLD_GOODS	

Selected Columns:

Column Name	Description
<literal>	ZQNAMEZQ
NAME	
<literal>	ZQADDRZQ
ADDRESS	
<literal>	ZQAREAZQ

Figure 1-427 Create J03_STAN_USPREP_M job 2/7


Standardize Rule Process

Select the rule set, then specify the columns the rule set is to standardize

Rule Set: USPREP.SET

Available Columns:

Column Name	Description
CELL_PHONE	
ELECTRONICS	
EMAIL	
FURNITURE	
HLD_GOODS	

Selected Columns:

Column Name	Description
ADDRESS	
<literal>	ZQAREAZQ
CITY	
STATE	
ZIP	

Literal:

Optional NAMES Handling: Process All As Organization

Figure 1-428 Create J03_STAN_USPREP_M job 3/7

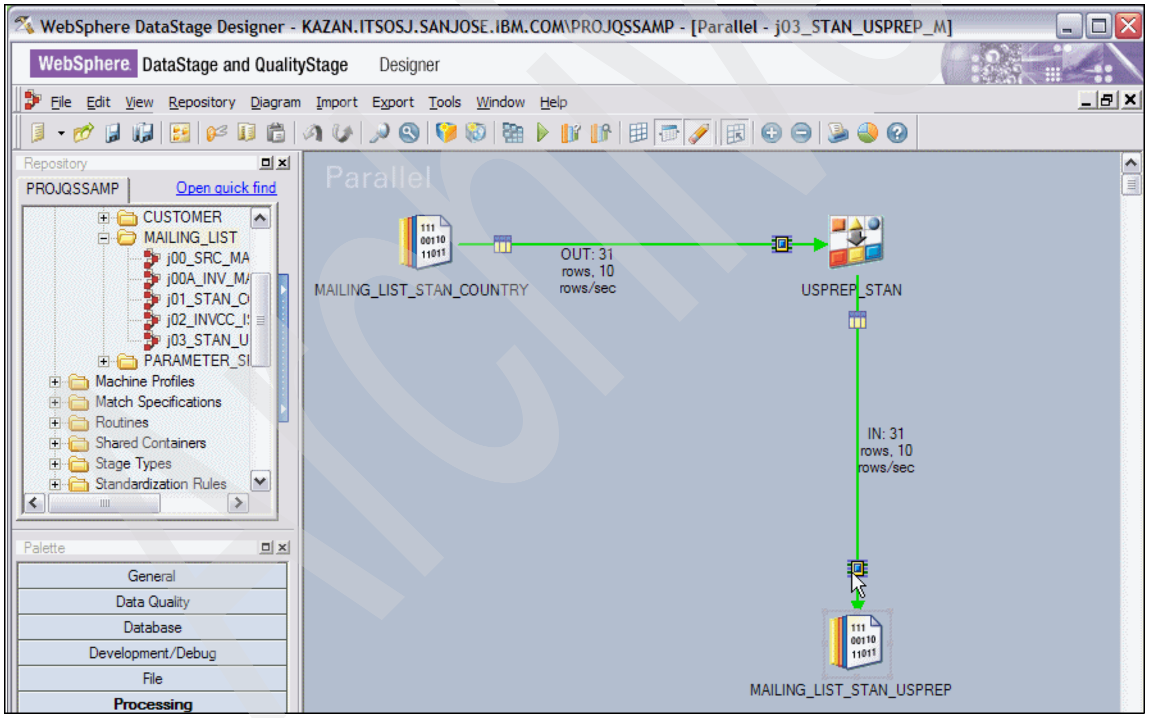
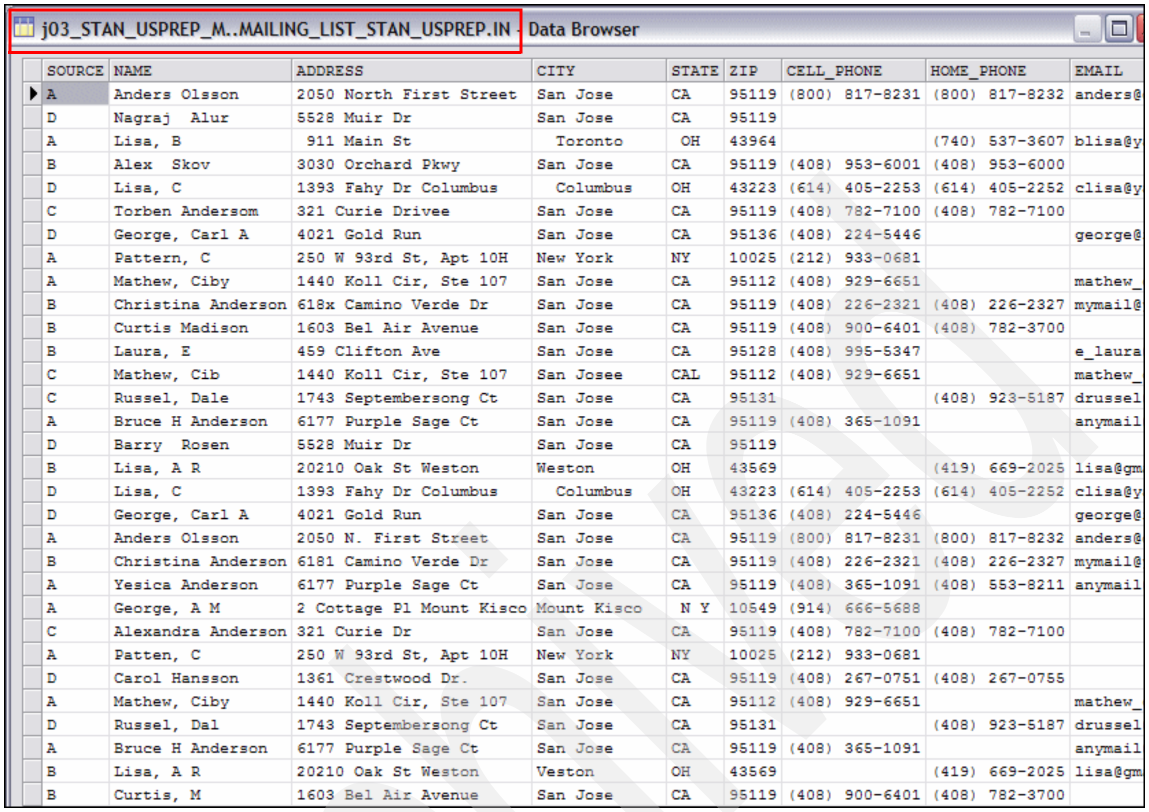


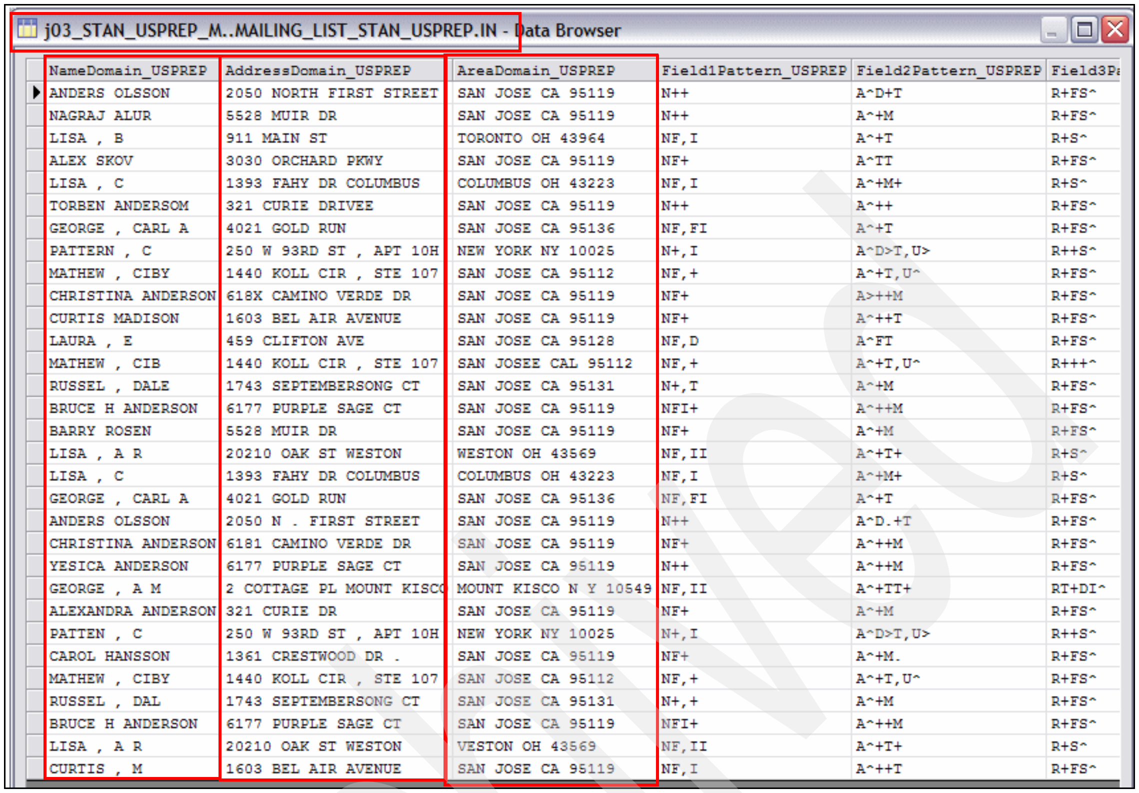
Figure 1-429 Create J03_STAN_USPREP_M job 4/7



j03_STAN_USPREP_M..MAILING_LIST_STAN_USPREP.IN - Data Browser

SOURCE	NAME	ADDRESS	CITY	STATE	ZIP	CELL_PHONE	HOME_PHONE	EMAIL
A	Anders Olsson	2050 North First Street	San Jose	CA	95119	(800) 817-8231	(800) 817-8232	anders@
D	Nagraj Alur	5528 Muir Dr	San Jose	CA	95119			
A	Lisa, B	911 Main St	Toronto	OH	43964		(740) 537-3607	blisa@y
B	Alex Skov	3030 Orchard Pkwy	San Jose	CA	95119	(408) 953-6001	(408) 953-6000	
D	Lisa, C	1393 Fahy Dr Columbus	Columbus	OH	43223	(614) 405-2253	(614) 405-2252	clisa@y
C	Torben Andersom	321 Curie Drivee	San Jose	CA	95119	(408) 782-7100	(408) 782-7100	
D	George, Carl A	4021 Gold Run	San Jose	CA	95136	(408) 224-5446		george@
A	Pattern, C	250 W 93rd St, Apt 10H	New York	NY	10025	(212) 933-0681		
A	Mathew, Ciby	1440 Koll Cir, Ste 107	San Jose	CA	95112	(408) 929-6651		mathew_
B	Christina Anderson	618x Camino Verde Dr	San Jose	CA	95119	(408) 226-2321	(408) 226-2327	mymail@
B	Curtis Madison	1603 Bel Air Avenue	San Jose	CA	95119	(408) 900-6401	(408) 782-3700	
B	Laura, E	459 Clifton Ave	San Jose	CA	95128	(408) 995-5347		e_laura
C	Mathew, Cib	1440 Koll Cir, Ste 107	San Josee	CAL	95112	(408) 929-6651		mathew_
C	Russel, Dale	1743 Septembersong Ct	San Jose	CA	95131		(408) 923-5187	drussel
A	Bruce H Anderson	6177 Purple Sage Ct	San Jose	CA	95119	(408) 365-1091		anymail
D	Barry Rosen	5528 Muir Dr	San Jose	CA	95119			
B	Lisa, A R	20210 Oak St Weston	Weston	OH	43569		(419) 669-2025	lisa@gm
D	Lisa, C	1393 Fahy Dr Columbus	Columbus	OH	43223	(614) 405-2253	(614) 405-2252	clisa@y
D	George, Carl A	4021 Gold Run	San Jose	CA	95136	(408) 224-5446		george@
A	Anders Olsson	2050 N. First Street	San Jose	CA	95119	(800) 817-8231	(800) 817-8232	anders@
B	Christina Anderson	6181 Camino Verde Dr	San Jose	CA	95119	(408) 226-2321	(408) 226-2327	mymail@
A	Yesica Anderson	6177 Purple Sage Ct	San Jose	CA	95119	(408) 365-1091	(408) 553-8211	anymail
A	George, A M	2 Cottage Pl Mount Kisco	Mount Kisco	N Y	10549	(914) 666-5688		
C	Alexandra Anderson	321 Curie Dr	San Jose	CA	95119	(408) 782-7100	(408) 782-7100	
A	Patten, C	250 W 93rd St, Apt 10H	New York	NY	10025	(212) 933-0681		
D	Carol Hansson	1361 Crestwood Dr.	San Jose	CA	95119	(408) 267-0751	(408) 267-0755	
A	Mathew, Ciby	1440 Koll Cir, Ste 107	San Jose	CA	95112	(408) 929-6651		mathew_
D	Russel, Dal	1743 Septembersong Ct	San Jose	CA	95131		(408) 923-5187	drussel
A	Bruce H Anderson	6177 Purple Sage Ct	San Jose	CA	95119	(408) 365-1091		anymail
B	Lisa, A R	20210 Oak St Weston	Veston	OH	43569		(419) 669-2025	lisa@gm
B	Curtis, M	1603 Bel Air Avenue	San Jose	CA	95119	(408) 900-6401	(408) 782-3700	

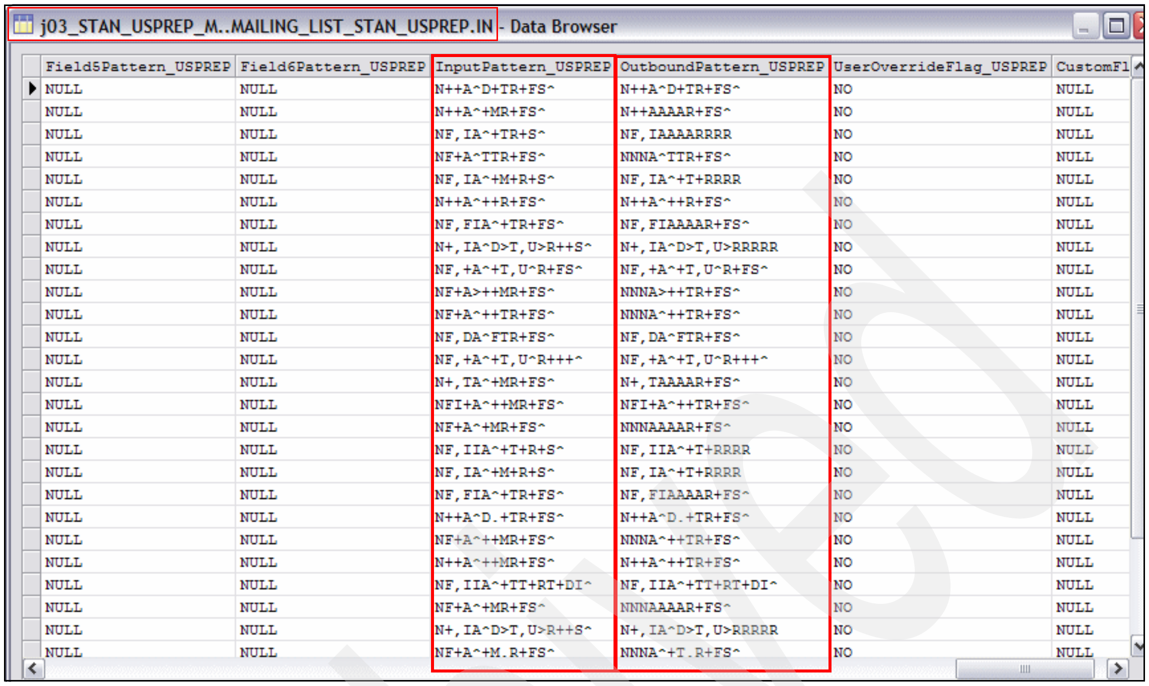
Figure 1-430 Create J03_STAN_USPREP_M job 5/7



j03_STAN_USPREP_M..MAILING_LIST_STAN_USPREP.IN - Data Browser

NameDomain_USPREP	AddressDomain_USPREP	AreaDomain_USPREP	Field1Pattern_USPREP	Field2Pattern_USPREP	Field3Pa
ANDERS OLSSON	2050 NORTH FIRST STREET	SAN JOSE CA 95119	N++	A^D+T	R+FS^
NAGRAJ ALUR	5528 MUIR DR	SAN JOSE CA 95119	N++	A^+M	R+FS^
LISA , B	911 MAIN ST	TORONTO OH 43964	NF,I	A^+T	R+S^
ALEX SKOV	3030 ORCHARD PKWY	SAN JOSE CA 95119	NF+	A^TT	R+FS^
LISA , C	1393 FAHY DR COLUMBUS	COLUMBUS OH 43223	NF,I	A^+M+	R+S^
TORBEN ANDERSOM	321 CURIE DRIVEE	SAN JOSE CA 95119	N++	A^++	R+FS^
GEORGE , CARL A	4021 GOLD RUN	SAN JOSE CA 95136	NF,FI	A^+T	R+FS^
PATTERN , C	250 W 93RD ST , APT 10H	NEW YORK NY 10025	N+,I	A^D>T,U>	R++S^
MATHEW , CIBY	1440 KOLL CIR , STE 107	SAN JOSE CA 95112	NF,+	A^+T,U^	R+FS^
CHRISTINA ANDERSON	618X CAMINO VERDE DR	SAN JOSE CA 95119	NF+	A>++M	R+FS^
CURTIS MADISON	1603 BEL AIR AVENUE	SAN JOSE CA 95119	NF+	A^++T	R+FS^
LAURA , E	459 CLIFTON AVE	SAN JOSE CA 95128	NF,D	A^FT	R+FS^
MATHEW , CIB	1440 KOLL CIR , STE 107	SAN JOSEE CAL 95112	NF,+	A^+T,U^	R+++^
RUSSEL , DALE	1743 SEPTEMBERSONG CT	SAN JOSE CA 95131	N+,T	A^+M	R+FS^
BRUCE H ANDERSON	6177 PURPLE SAGE CT	SAN JOSE CA 95119	NFI+	A^++M	R+FS^
BARRY ROSEN	5528 MUIR DR	SAN JOSE CA 95119	NF+	A^+M	R+FS^
LISA , A R	20210 OAK ST WESTON	WESTON OH 43569	NF,II	A^+T+	R+S^
LISA , C	1393 FAHY DR COLUMBUS	COLUMBUS OH 43223	NF,I	A^+M+	R+S^
GEORGE , CARL A	4021 GOLD RUN	SAN JOSE CA 95136	NF,FI	A^+T	R+FS^
ANDERS OLSSON	2050 N . FIRST STREET	SAN JOSE CA 95119	N++	A^D.+T	R+FS^
CHRISTINA ANDERSON	6181 CAMINO VERDE DR	SAN JOSE CA 95119	NF+	A^++M	R+FS^
YESICA ANDERSON	6177 PURPLE SAGE CT	SAN JOSE CA 95119	N++	A^++M	R+FS^
GEORGE , A M	2 COTTAGE PL MOUNT KISCO	MOUNT KISCO N Y 10549	NF,II	A^+TT+	RT+DI^
ALEXANDRA ANDERSON	321 CURIE DR	SAN JOSE CA 95119	NF+	A^+M	R+FS^
PATTEN , C	250 W 93RD ST , APT 10H	NEW YORK NY 10025	N+,I	A^D>T,U>	R++S^
CAROL HANSSON	1361 CRESTWOOD DR .	SAN JOSE CA 95119	NF+	A^+M.	R+FS^
MATHEW , CIBY	1440 KOLL CIR , STE 107	SAN JOSE CA 95112	NF,+	A^+T,U^	R+FS^
RUSSEL , DAL	1743 SEPTEMBERSONG CT	SAN JOSE CA 95131	N+,+	A^+M	R+FS^
BRUCE H ANDERSON	6177 PURPLE SAGE CT	SAN JOSE CA 95119	NFI+	A^++M	R+FS^
LISA , A R	20210 OAK ST WESTON	VESTON OH 43569	NF,II	A^+T+	R+S^
CURTIS , M	1603 BEL AIR AVENUE	SAN JOSE CA 95119	NF,I	A^++T	R+FS^

Figure 1-431 Create J03_STAN_USPREP_M job 6/7


j03_STAN_USPREP_M..MAILING_LIST_STAN_USPREP.IN - Data Browser

Field5Pattern_USPREP	Field6Pattern_USPREP	InputPattern_USPREP	OutboundPattern_USPREP	UserOverrideFlag_USPREP	CustomFl
NULL	NULL	N++A^D+TR+FS^	N++A^D+TR+FS^	NO	NULL
NULL	NULL	N++A^+MR+FS^	N++AAAAR+FS^	NO	NULL
NULL	NULL	NF,IA^+TR+S^	NF,IAAAARRRR	NO	NULL
NULL	NULL	NF+A^TTR+FS^	NNNA^TTR+FS^	NO	NULL
NULL	NULL	NF,IA^+M+R+S^	NF,IA^+T+RRRR	NO	NULL
NULL	NULL	N++A^++R+FS^	N++A^++R+FS^	NO	NULL
NULL	NULL	NF,FIA^+TR+FS^	NF,FIAAAAR+FS^	NO	NULL
NULL	NULL	N+,IA^D>T,U>R++S^	N+,IA^D>T,U>RRRRR	NO	NULL
NULL	NULL	NF,+A^+T,U^R+FS^	NF,+A^+T,U^R+FS^	NO	NULL
NULL	NULL	NF+A>++MR+FS^	NNNA>++TR+FS^	NO	NULL
NULL	NULL	NF+A^++TR+FS^	NNNA^++TR+FS^	NO	NULL
NULL	NULL	NF,DA^FTR+FS^	NF,DA^FTR+FS^	NO	NULL
NULL	NULL	NF,+A^+T,U^R+++^	NF,+A^+T,U^R+++^	NO	NULL
NULL	NULL	N+,TA^+MR+FS^	N+,TAAAAR+FS^	NO	NULL
NULL	NULL	NFI+A^++MR+FS^	NFI+A^++TR+FS^	NO	NULL
NULL	NULL	NF+A^+MR+FS^	NNNAAAAR+FS^	NO	NULL
NULL	NULL	NF,IIA^+T+R+S^	NF,IIA^+T+RRRR	NO	NULL
NULL	NULL	NF,IA^+M+R+S^	NF,IA^+T+RRRR	NO	NULL
NULL	NULL	NF,FIA^+TR+FS^	NF,FIAAAAR+FS^	NO	NULL
NULL	NULL	N++A^D.+TR+FS^	N++A^D.+TR+FS^	NO	NULL
NULL	NULL	NF+A^++MR+FS^	NNNA^++TR+FS^	NO	NULL
NULL	NULL	N++A^++MR+FS^	N++A^++TR+FS^	NO	NULL
NULL	NULL	NF,IIA^+TT+RT+DI^	NF,IIA^+TT+RT+DI^	NO	NULL
NULL	NULL	NF+A^+MR+FS^	NNNAAAAR+FS^	NO	NULL
NULL	NULL	N+,IA^D>T,U>R++S^	N+,IA^D>T,U>RRRRR	NO	NULL
NULL	NULL	NF+A^+M.R+FS^	NNNA^+T.R+FS^	NO	NULL

Figure 1-432 Create J03_STAN_USPREP_M job 7/7

J04_INVW_USPREP_M

In this step, we use the Investigate stage using word investigate on the Name, Address, and Area domains to determine whether the domain-preprocessor USPREP rule set parsed the tokens in the name and address fields into the correct domains successfully.

We analyze the mailing list file containing the name, address, and area buckets that were generated by the job “J03_STAN_USPREP_M” on page 361 by the Investigate stage using word investigate and the domain-specific USNAME, USADDR, and USAREA rule sets to determine the degree of success that is achieved by the Standardize stage to moving the tokens to the right buckets.

Because a single Investigate stage can only have a single rule set associated with it, we need to split the mailing list file (using a Copy stage) and process it by three independent Investigate stages, with each stage using a particular domain-specific rule set. Both the pattern and token reports are generated in each Investigate stage.

Figure 1-433 on page 368 shows the various stages that are used in this job, including the data set that was created in “J03_STAN_USPREP_M” on page 361, a Copy stage, three Investigate stages that each use a different domain-specific rule set, and two sequential file stages (one each for the token report and pattern report) for each Investigate stage. We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object created earlier.

Because the configuration of this job is very similar to that described in “J04_INVW_USPREP” on page 203, we do not repeat it here. However, some of the configurations of interest are as follows:

- Figure 1-434 on page 368 shows the INVNAME - Investigate Stage window using the USNAME rule set with the Word Investigate option. The selected column is NameDomain_USPREP.
- Figure 1-435 on page 369 shows the INVADDR - Investigate Stage window using the USADDR rule set with the Word Investigate option. The selected column is AddressDomain_USPREP.
- Figure 1-436 on page 369 shows the INVAREA - Investigate Stage window using the USAREA rule set with the Word Investigate option. The selected column is AreaDomain_USPREP.
- After saving, compiling, and running this job, the contents of the output of each investigate stage is listed here:
- The outputs of the Investigate stage written to the sequential files are shown in Figure 1-438 on page 370 through Figure 1-443 on page 373.

  The reports show a number of tokens in the various reports not being recognized with proper classifications (code “?”). The patterns also appear to need overrides, but you can only be certain of this after the pattern action language has been invoked.

Now, proceed to “J03_Z_Override_And_After” on page 215.

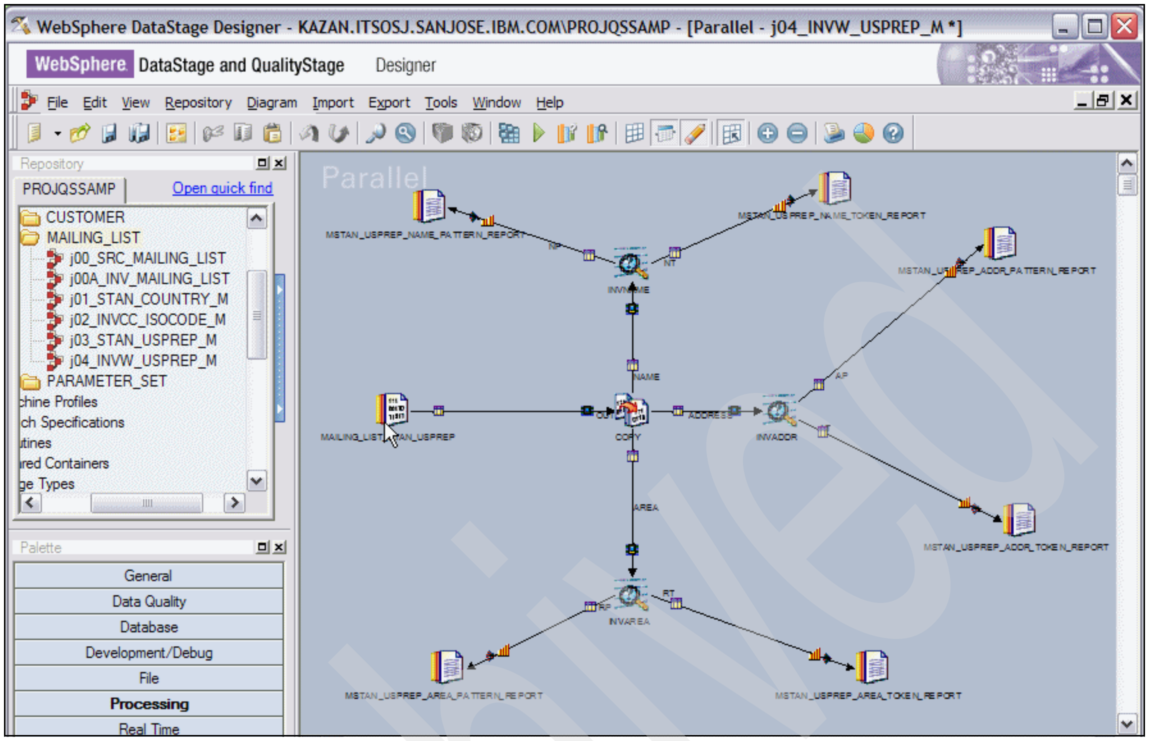
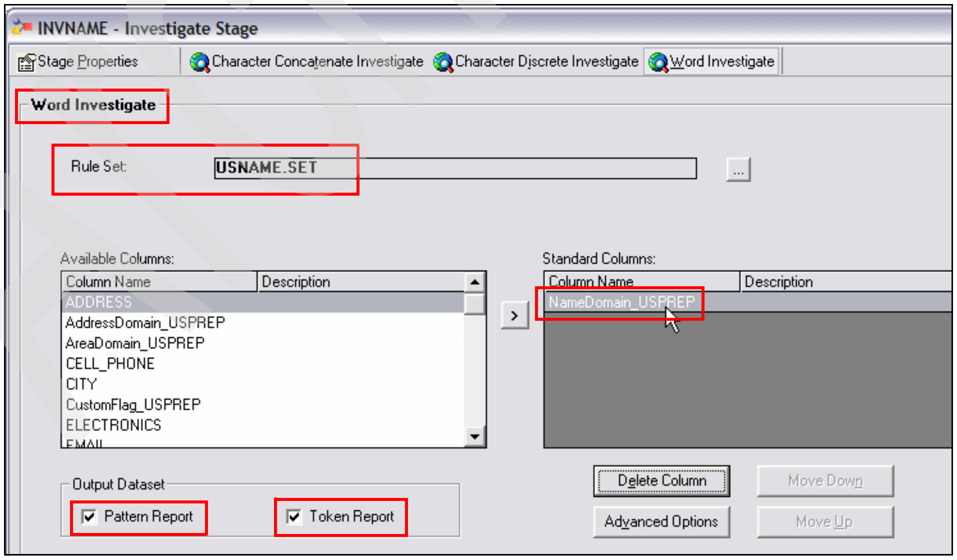

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j04_INVW_USPREP_M *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Repository
PROJQSSAMP
Open quick find
CUSTOMER
MAILING_LIST
j00_SRC_MAILING_LIST
j00A_INV_MAILING_LIST
j01_STAN_COUNTRY_M
j02_INVCC_ISOCODE_M
j03_STAN_USPREP_M
j04_INVW_USPREP_M
PARAMETER_SET
chine Profiles
ch Specifications
itines
ired Containers
ge Types
Palette
General
Data Quality
Database
Development/Debug
File
Processing
Real Time
Parallel
MSTAN_USPREP_NAME_PATTERN_REPORT
MSTAN_USPREP_NAME_TOKEN_REPORT
MSTAN_USPREP_ADDR_PATTERN_REPORT
MSTAN_USPREP_ADDR_TOKEN_REPORT
MSTAN_USPREP_AREA_PATTERN_REPORT
MSTAN_USPREP_AREA_TOKEN_REPORT
MAILING_LIST_STAN_USPREP
INVNAME
INVADDR
INVAREA
COPY
NAME
ADDRESS
AREA
NP
NT
AP
RP
RT

Figure 1-433 Create J04_INVW_USPREP_M job 1/11



Rule Set: USNAME.SET

Available Columns:

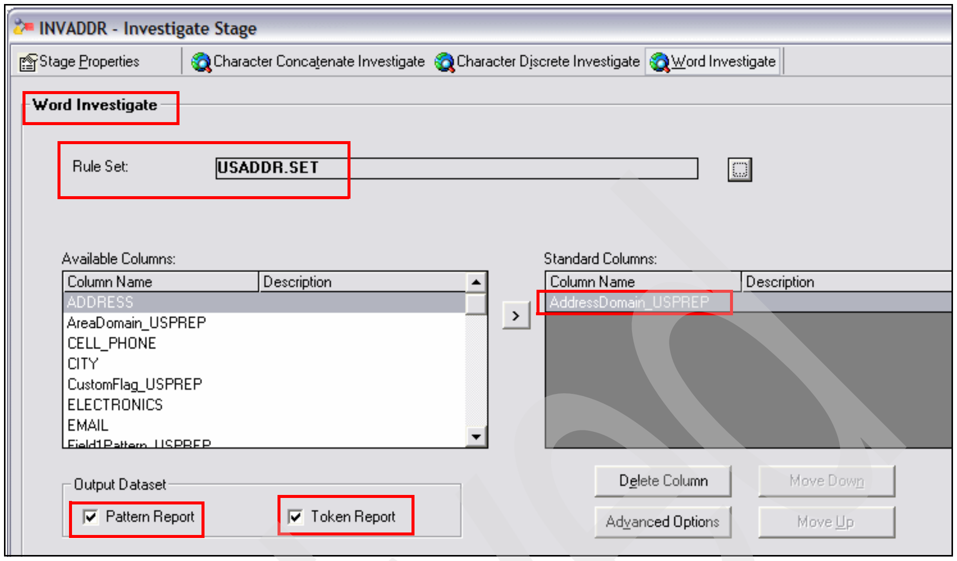
Column Name	Description
ADDRESS	
AddressDomain_USPREP	
AreaDomain_USPREP	
CELL_PHONE	
CITY	
CustomFlag_USPREP	
ELECTRONICS	
EMAIL	

Standard Columns:

Column Name	Description
NameDomain_USPREP	

Output Dataset
[✓] Pattern Report
[✓] Token Report

Figure 1-434 Create J04_INVW_USPREP_M job 2/11


INVADDR - Investigate Stage

Stage Properties | Character Concatenate Investigate | Character Discrete Investigate | Word Investigate

Word Investigate

Rule Set: USADDR.SET

Available Columns:

Column Name	Description
ADDRESS	
AreaDomain_USPREP	
CELL_PHONE	
CITY	
CustomFlag_USPREP	
ELECTRONICS	
EMAIL	
Field1Pattern_USPREP	

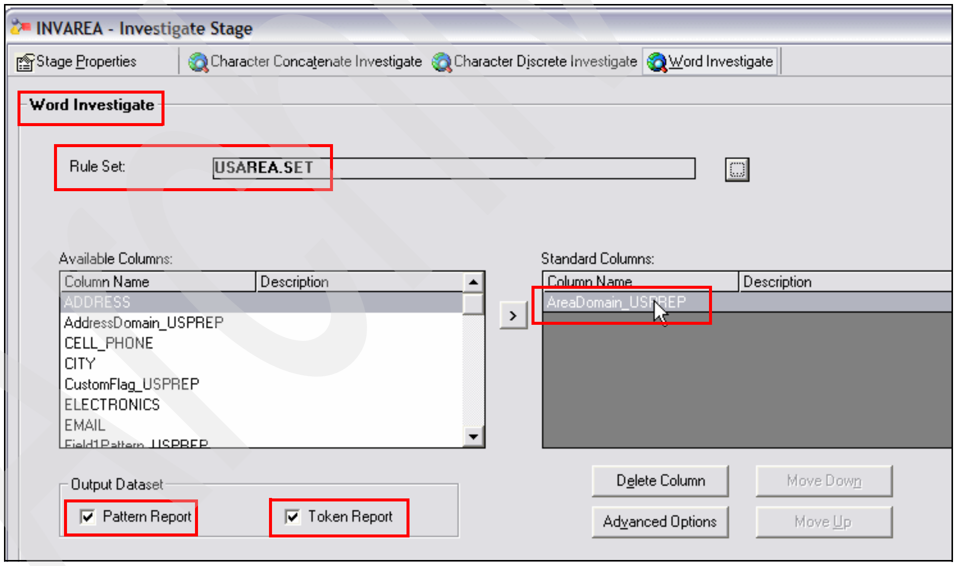
Standard Columns:

Column Name	Description
AddressDomain_USPREP	

Output Dataset

☑ Pattern Report ☑ Token Report

Figure 1-435 Create J04_INVW_USPREP_M job 3/11


INVAREA - Investigate Stage

Stage Properties | Character Concatenate Investigate | Character Discrete Investigate | Word Investigate

Word Investigate

Rule Set: USAREA.SET

Available Columns:

Column Name	Description
ADDRESS	
AddressDomain_USPREP	
CELL_PHONE	
CITY	
CustomFlag_USPREP	
ELECTRONICS	
EMAIL	
Field1Pattern_USPREP	

Standard Columns:

Column Name	Description
AreaDomain_USPREP	

Output Dataset

☑ Pattern Report ☑ Token Report

Figure 1-436 Create J04_INVW_USPREP_M job 4/11

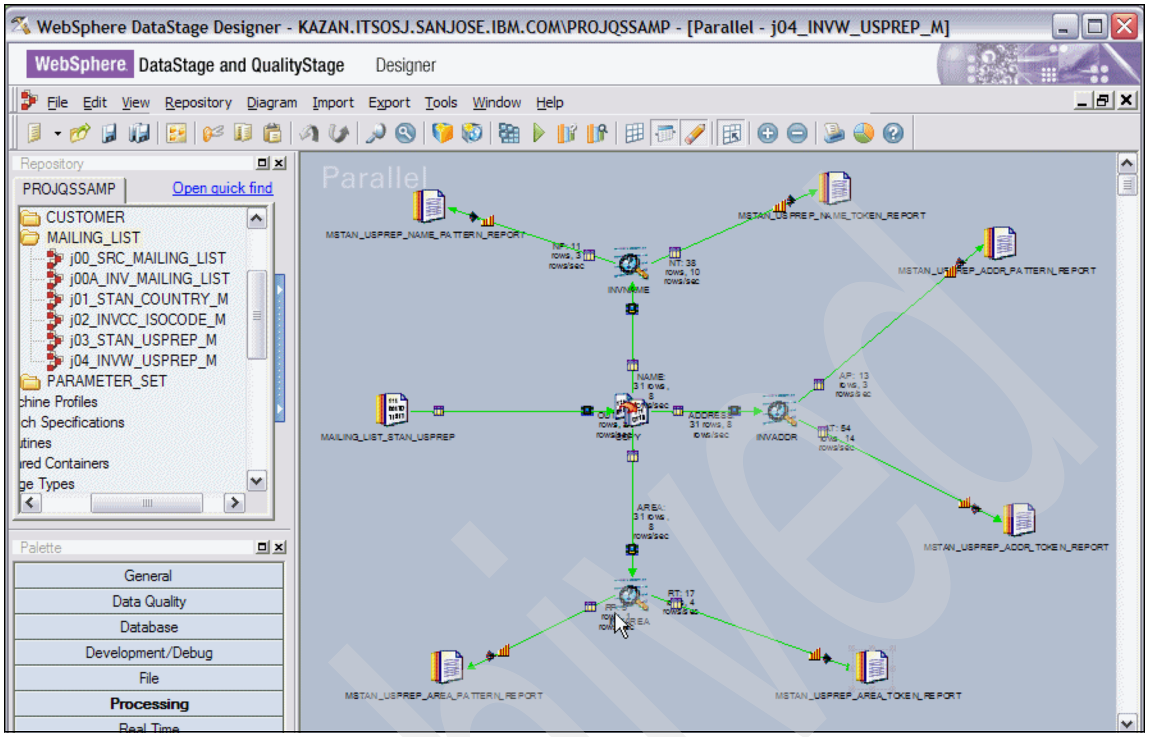
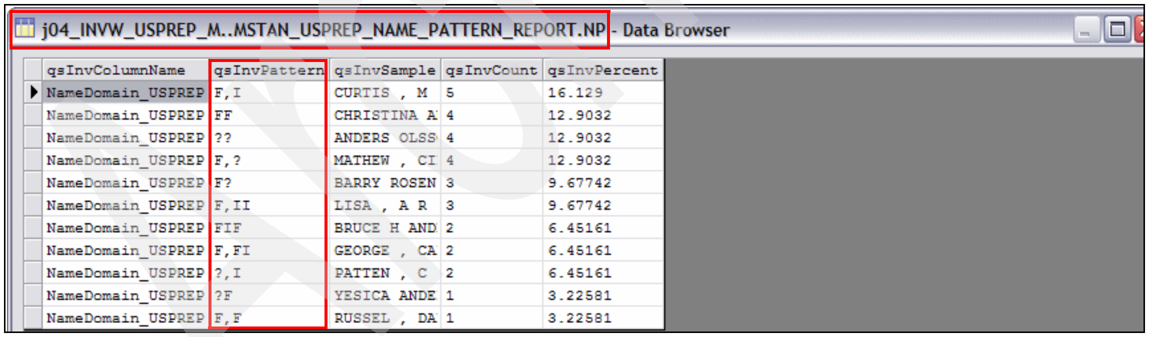
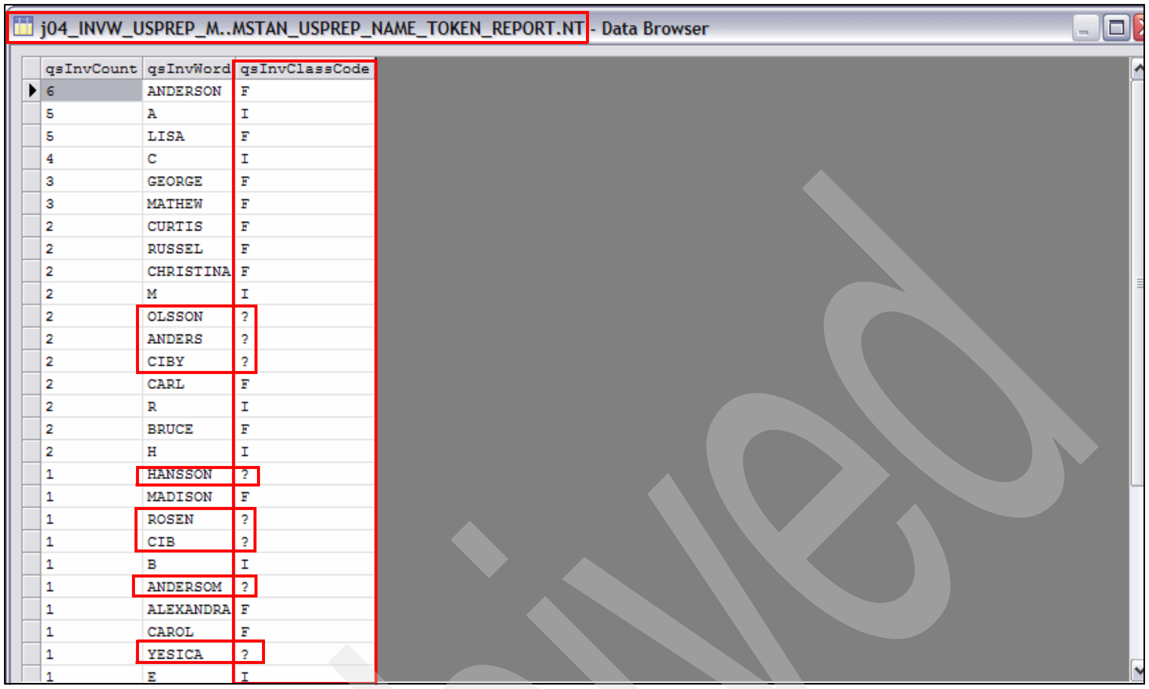

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j04_INVW_USPREP_M]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Repository
PROJQSSAMP
Open quick find
CUSTOMER
MAILING_LIST
j00_SRC_MAILING_LIST
j00A_INV_MAILING_LIST
j01_STAN_COUNTRY_M
j02_INVCC_ISOCODE_M
j03_STAN_USPREP_M
j04_INVW_USPREP_M
PARAMETER_SET
chine Profiles
ch Specifications
utines
red Containers
ge Types
Palette
General
Data Quality
Database
Development/Debug
File
Processing
Real Time
Parallel
MSTAN_USPREP_NAME_PATTERN_REPORT
MSTAN_USPREP_NAME_TOKEN_REPORT
MSTAN_USPREP_ADDR_PATTERN_REPORT
MAILING_LIST_STAN_USPREP
INVNAME
INVADDR
MSTAN_USPREP_ADDR_TOKEN_REPORT
MSTAN_USPREP_AREA_PATTERN_REPORT
MSTAN_USPREP_AREA_TOKEN_REPORT

Figure 1-437   Create J04_INVW_USPREP_M job 5/11


j04_INVW_USPREP_M..MSTAN_USPREP_NAME_PATTERN_REPORT.NP - Data Browser

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
NameDomain_USPREP	F,I	CURTIS , M	5	16.129
NameDomain_USPREP	FF	CHRISTINA A	4	12.9032
NameDomain_USPREP	??	ANDERS OLSS	4	12.9032
NameDomain_USPREP	F,?	MATHEW , CI	4	12.9032
NameDomain_USPREP	F?	BARRY ROSEN	3	9.67742
NameDomain_USPREP	F,II	LISA , A R	3	9.67742
NameDomain_USPREP	FIF	BRUCE H AND	2	6.45161
NameDomain_USPREP	F,FI	GEORGE , CA	2	6.45161
NameDomain_USPREP	?,I	PATTEN , C	2	6.45161
NameDomain_USPREP	?F	YESICA ANDE	1	3.22581
NameDomain_USPREP	F,F	RUSSEL , DA	1	3.22581

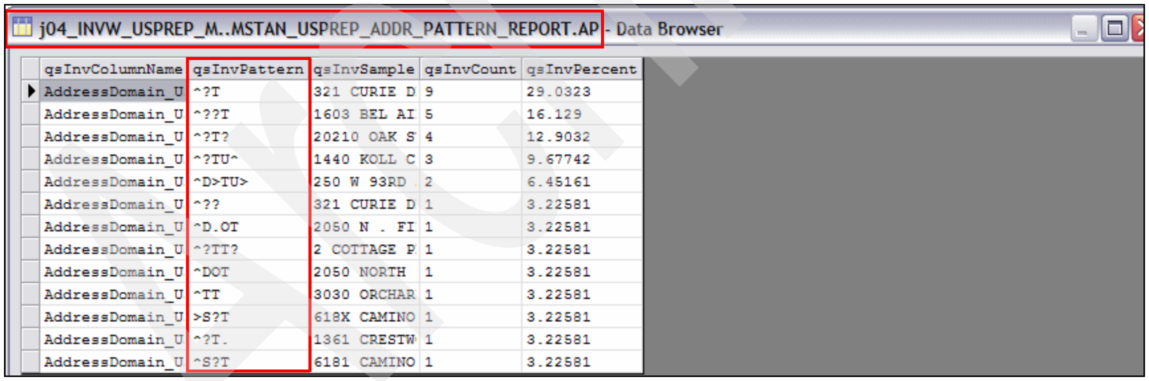
Figure 1-438   Create J04_INVW_USPREP_M job 6/11



j04_INVW_USPREP_M..MSTAN_USPREP_NAME_TOKEN_REPORT.NT - Data Browser

qsInvCount	qsInvWord	qsInvClassCode
6	ANDERSON	F
5	A	I
5	LISA	F
4	C	I
3	GEORGE	F
3	MATHEW	F
2	CURTIS	F
2	RUSSEL	F
2	CHRISTINA	F
2	M	I
2	OLSSON	?
2	ANDERS	?
2	CIBY	?
2	CARL	F
2	R	I
2	BRUCE	F
2	H	I
1	HANSSON	?
1	MADISON	F
1	ROSEN	?
1	CIB	?
1	B	I
1	ANDERSOM	?
1	ALEXANDRA	F
1	CAROL	F
1	YESICA	?
1	E	I

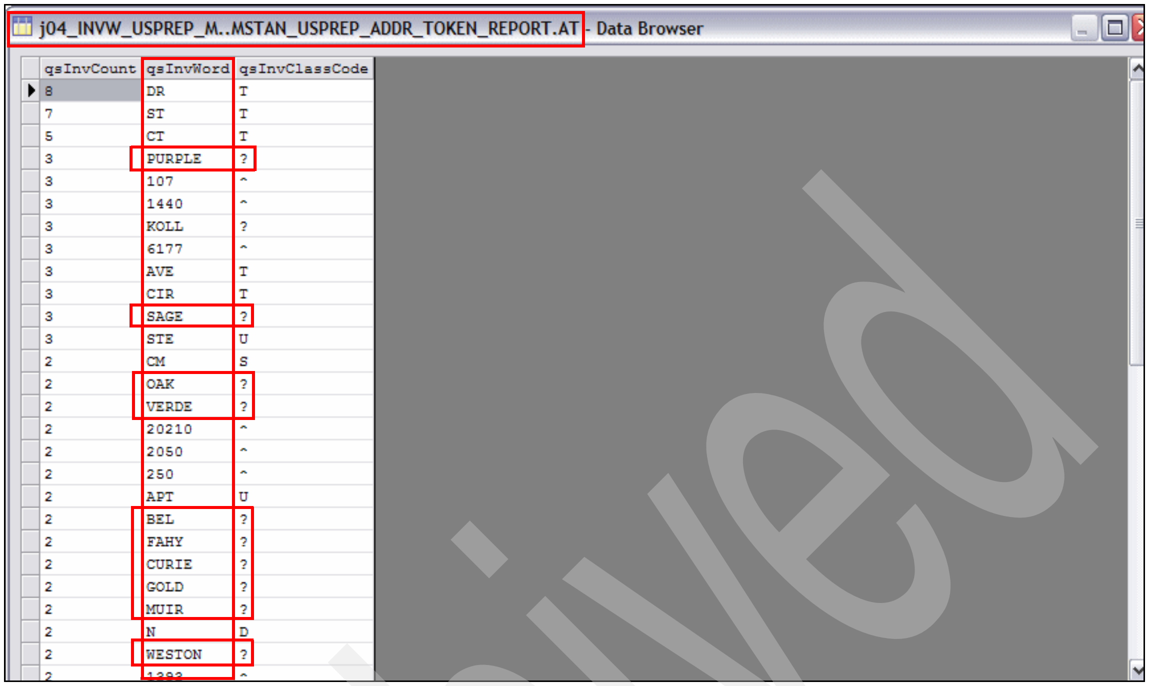
Figure 1-439   Create J04_INVW_USPREP_M job 7/11



j04_INVW_USPREP_M..MSTAN_USPREP_ADDR_PATTERN_REPORT.AP - Data Browser

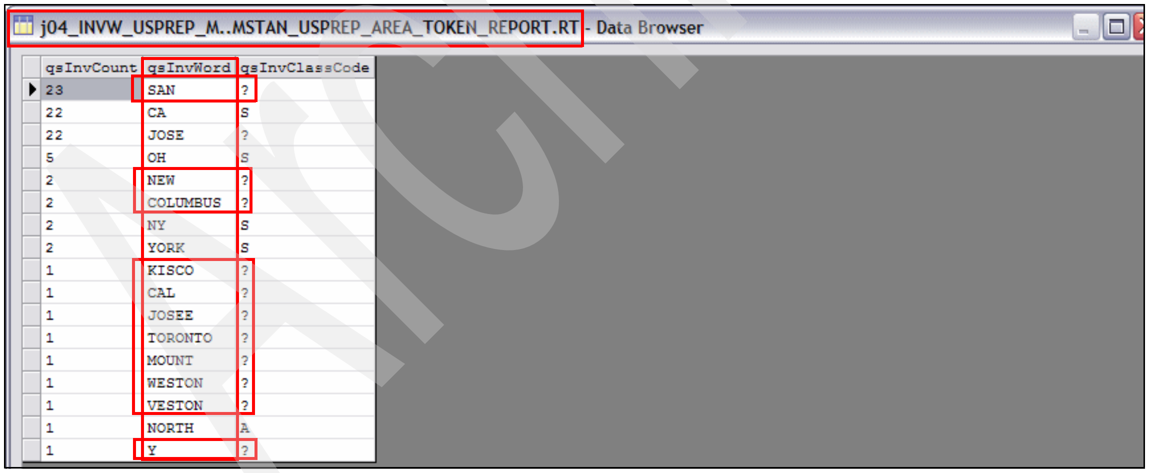
qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
AddressDomain_U	^?T	321 CURIE D	9	29.0323
AddressDomain_U	^??T	1603 BEL AI	5	16.129
AddressDomain_U	^?T?	20210 OAK S	4	12.9032
AddressDomain_U	^?TU^	1440 KOLL C	3	9.67742
AddressDomain_U	^D>TU>	250 W 93RD	2	6.45161
AddressDomain_U	^??	321 CURIE D	1	3.22581
AddressDomain_U	^D.OT	2050 N . FI	1	3.22581
AddressDomain_U	^?TT?	2 COTTAGE P	1	3.22581
AddressDomain_U	^DOT	2050 NORTH	1	3.22581
AddressDomain_U	^TT	3030 ORCHAR	1	3.22581
AddressDomain_U	>S?T	618X CAMINO	1	3.22581
AddressDomain_U	^?T.	1361 CRESTW	1	3.22581
AddressDomain_U	^S?T	6181 CAMINO	1	3.22581

Figure 1-440   Create J04_INVW_USPREP_M job 8/11


j04_INVW_USPREP_M..MSTAN_USPREP_ADDR_TOKEN_REPORT.AT - Data Browser

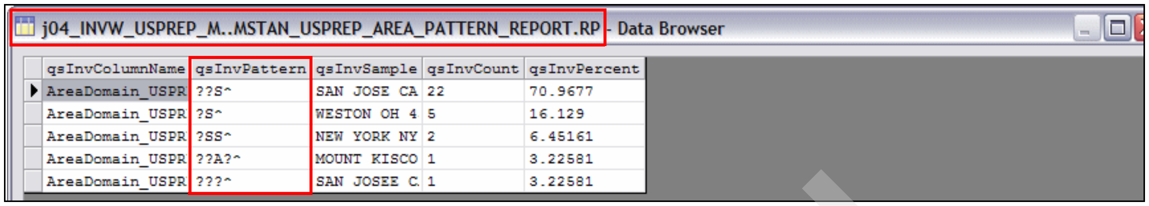
qsInvCount	qsInvWord	qsInvClassCode
8	DR	T
7	ST	T
5	CT	T
3	PURPLE	?
3	107	^
3	1440	^
3	KOLL	?
3	6177	^
3	AVE	T
3	CIR	T
3	SAGE	?
3	STE	U
2	CM	S
2	OAK	?
2	VERDE	?
2	20210	^
2	2050	^
2	250	^
2	APT	U
2	BEL	?
2	FAHY	?
2	CURIE	?
2	GOLD	?
2	MUIR	?
2	N	D
2	WESTON	?

Figure 1-441 Create J04_INVW_USPREP_M job 9/11


j04_INVW_USPREP_M..MSTAN_USPREP_AREA_TOKEN_REPORT.RT - Data Browser

qsInvCount	qsInvWord	qsInvClassCode
23	SAN	?
22	CA	S
22	JOSE	?
5	OH	S
2	NEW	?
2	COLUMBUS	?
2	NY	S
2	YORK	S
1	KISCO	?
1	CAL	?
1	JOSEE	?
1	TORONTO	?
1	MOUNT	?
1	WESTON	?
1	VESTON	?
1	NORTH	A
1	Y	?

Figure 1-442 Create J04_INVW_USPREP_M job 10/11



j04_INVW_USPREP_M..MSTAN_USPREP_AREA_PATTERN_REPORT.RP - Data Browser

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
AreaDomain_USPR	??S^	SAN JOSE CA	22	70.9677
AreaDomain_USPR	?S^	WESTON OH 4	5	16.129
AreaDomain_USPR	?SS^	NEW YORK NY	2	6.45161
AreaDomain_USPR	??A?^	MOUNT KISCO	1	3.22581
AreaDomain_USPR	???^	SAN JOSEE C.	1	3.22581

Figure 1-443 Create J04_INVW_USPREP_M job 11/11

J04_Z_After_Override

We perform a visual analysis of the token and pattern reports of the "J04_INVW_USPREP_M" on page 366 job to determine if the parsing was successful.

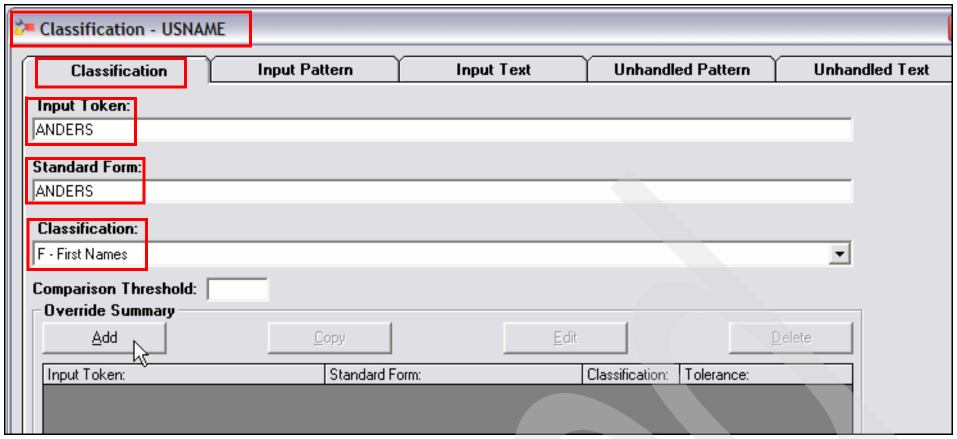
In doing so, we find certain errors with the parsing, and so chose to perform a classification override for the USNAME rule set to fix the errors.

Because the override is similar to that described in "J03_Z_Override_And_After" on page 215, we do not repeat it here. Figure 1-444 on page 374 shows a Classification override of the USNAME rule set for the token ANDERS with a classification of F.

We repeat the steps described in jobs "J03_STAN_USPREP_M" on page 361, "J04_INVW_USPREP_M" on page 366, and "J04_Z_After_Override" on page 373 until the name and address data is moved into the correct domain columns.

Attention: As mentioned earlier in "J03_Z_Override_And_After" on page 215, overrides in this job generally should be limited to simple classification overrides that move name and address data to the correct name/address/area domain buckets, rather than more complex pattern overrides.

Now, proceed to "J05_CASS_USPREP_M" on page 374.


Classification - USNAME

Input Token: ANDERS

Standard Form: ANDERS

Classification: F - First Names

Comparison Threshold:

Override Summary

Input Token:	Standard Form:	Classification:	Tolerance:

Figure 1-444 Create J04_Z_After_Override

J05_CASS_USPREP_M

After all the name and address data is moved to the correct domain buckets, we use the CASS stage to validate, correct, and standardize the U.S. addresses in the Address domain and to write a Postal Service form 3553 to a file. We also include a Transformer stage to add a second address line column to the mailing list file because CASS requires two address lines as input for its processing.

Figure 1-445 on page 375 shows the various stages that are used in this job, including the data set that was created in “J03_STAN_USPREP_M” on page 361, a Transformer stage, a CASS stage, and a Data Set stage. We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object that was created earlier.

Because the configuration of this job is very similar to that described in “J05_CASS_USPREP” on page 219, we do not repeat it here. However, some of the configurations of interest are as follows:

- Figure 1-446 on page 376 shows the ADD_SAPCE_FIELD - Transformer Stage window that describes the addition of the DUMMY_ADDRESS_FIELD of character length 1 to serve as the second address line that is required by the CASS stage.
- Figure 1-447 on page 376 shows the CASS stage configuration with details such as the assignment of columns to Address Line 1 and Address Line 2, and location of the output file and parameter set object.
- After saving, compiling, and running this job (Figure 1-448 on page 377), review the contents of the output of the CASS stage as shown in Figure 1-449 on page 377 through Figure 1-451 on page 379. It shows validation errors with some addresses, as described in “J05_CASS_USPREP” on page 219.

Note: Generally, any data error should be presented for review by appropriate personnel as early as possible in the process. An `A1` should be investigated and appropriate action taken.

Now, proceed to “J06_INVCC_CASS_M” on page 379.

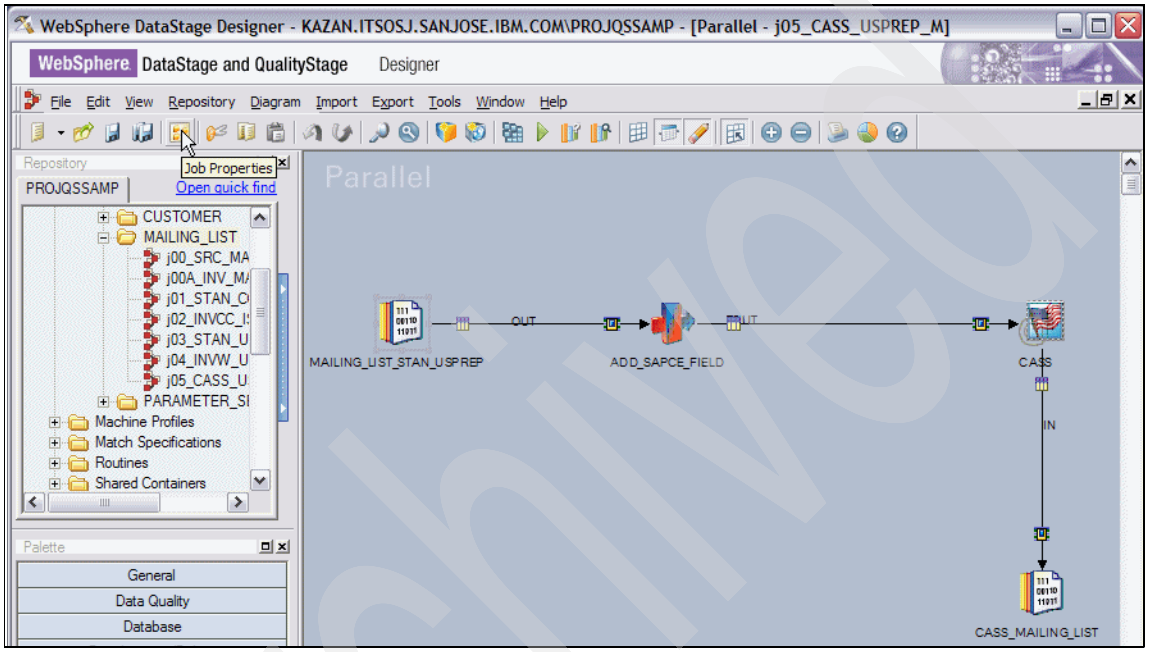
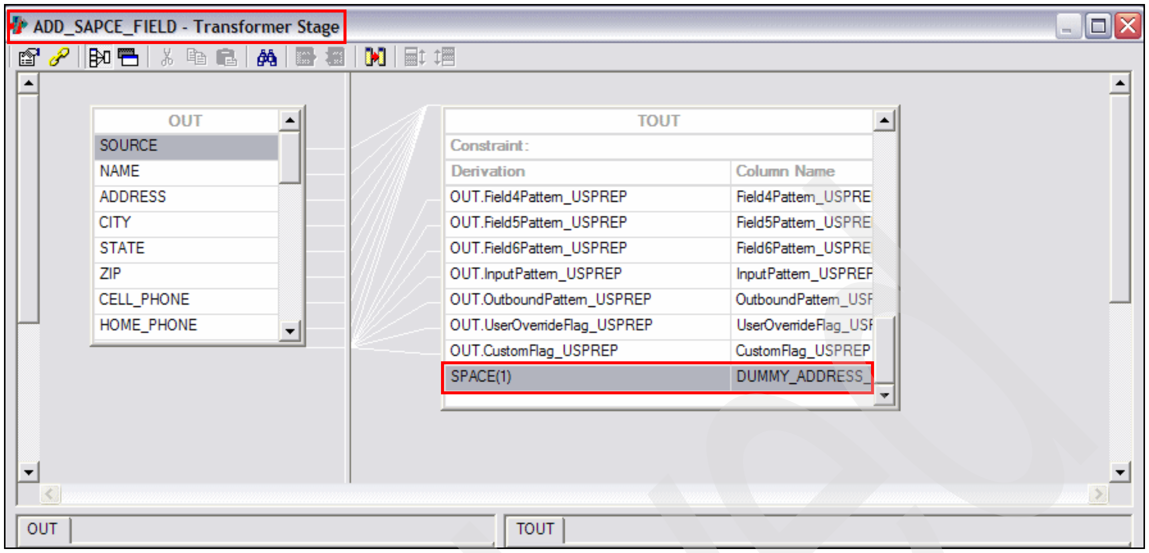


Figure 1-445 Create J05_CASS_USPREP_M job 1/7


ADD_SAPCE_FIELD - Transformer Stage

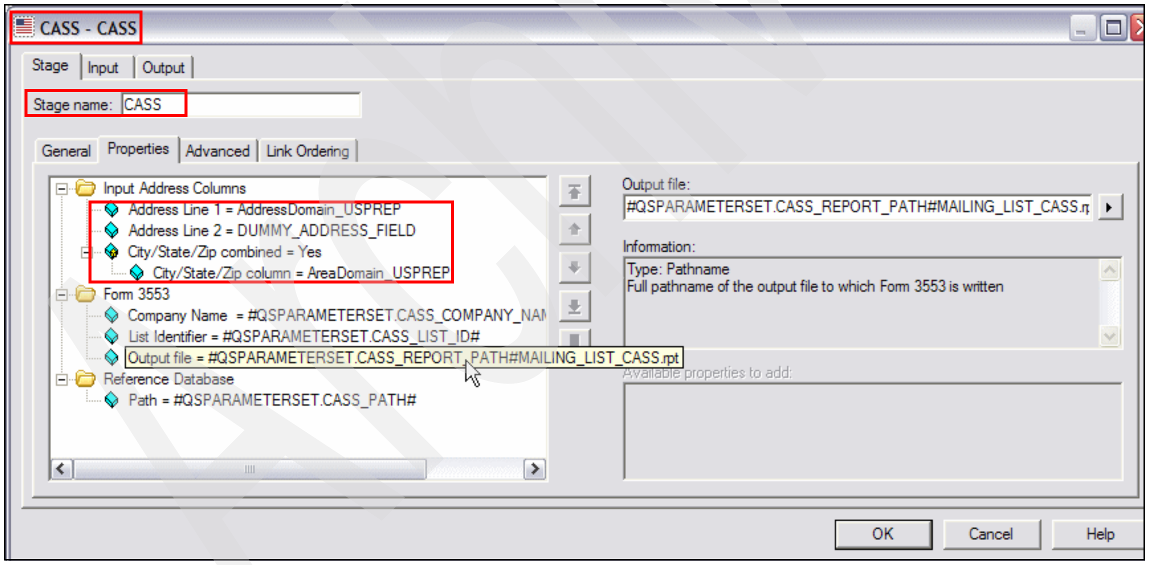
OUT
SOURCE
NAME
ADDRESS
CITY
STATE
ZIP
CELL_PHONE
HOME_PHONE

TOUT

Constraint:

Derivation	Column Name
OUT.Field4Pattern_USPREP	Field4Pattern_USPRE
OUT.Field5Pattern_USPREP	Field5Pattern_USPRE
OUT.Field6Pattern_USPREP	Field6Pattern_USPRE
OUT.InputPattern_USPREP	InputPattern_USPREF
OUT.OutboundPattern_USPREP	OutboundPattern_USF
OUT.UserOverrideFlag_USPREP	UserOverrideFlag_USF
OUT.CustomFlag_USPREP	CustomFlag_USPREP
SPACE(1)	DUMMY_ADDRESS_

Figure 1-446   Create J05_CASS_USPREP_M job 2/7


CASS - CASS

Stage name: CASS

- Input Address Columns
  - Address Line 1 = AddressDomain_USPREP
  - Address Line 2 = DUMMY_ADDRESS_FIELD
  - City/State/Zip combined = Yes
    - City/State/Zip column = AreaDomain_USPREP
- Form 3553
  - Company Name = #QSPARAMETERSET.CASS_COMPANY_NAI
  - List Identifier = #QSPARAMETERSET.CASS_LIST_ID#
  - Output file = #QSPARAMETERSET.CASS_REPORT_PATH#MAILING_LIST_CASS.rpt
- Reference Database
  - Path = #QSPARAMETERSET.CASS_PATH#

Output file:
#QSPARAMETERSET.CASS_REPORT_PATH#MAILING_LIST_CASS.r

Information:
Type: Pathname
Full pathname of the output file to which Form 3553 is written

Available properties to add:

Figure 1-447   Create J05_CASS_USPREP_M job 3/7

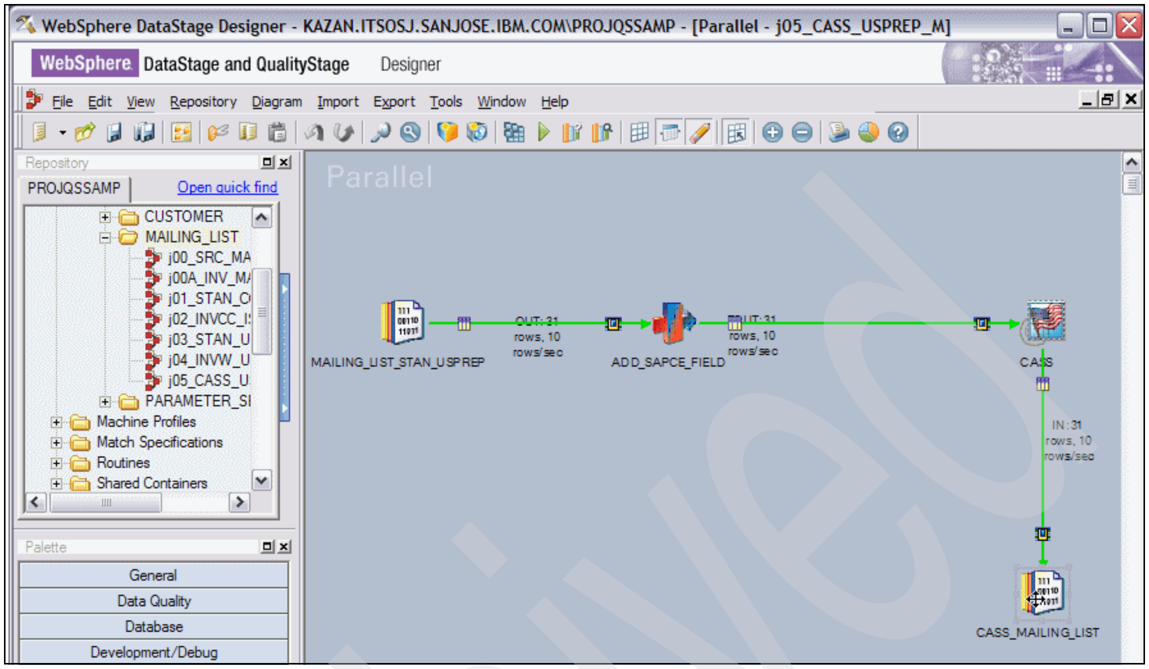
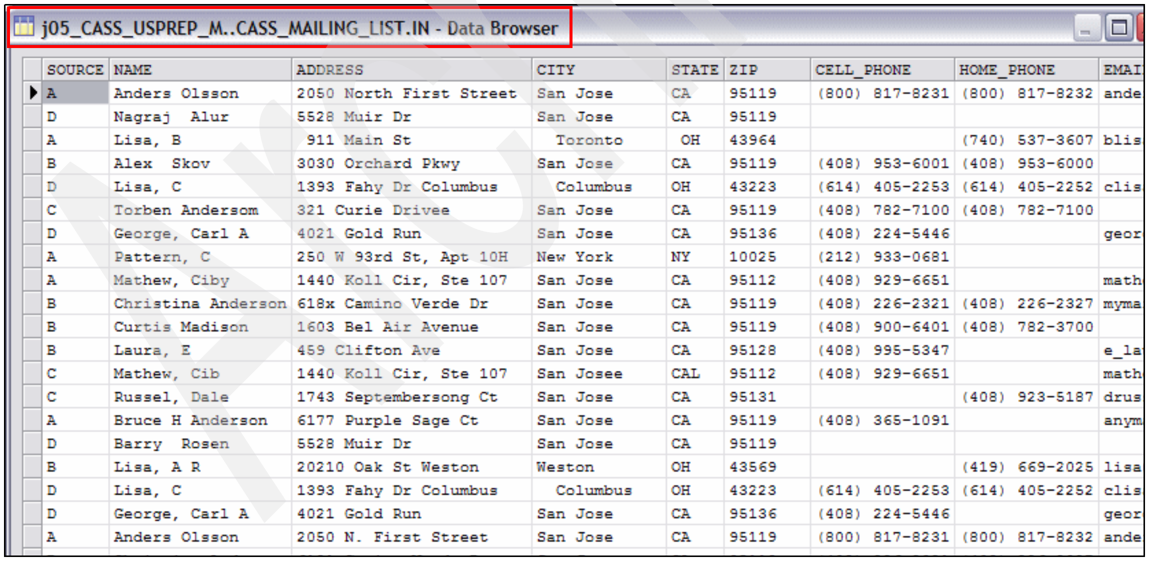


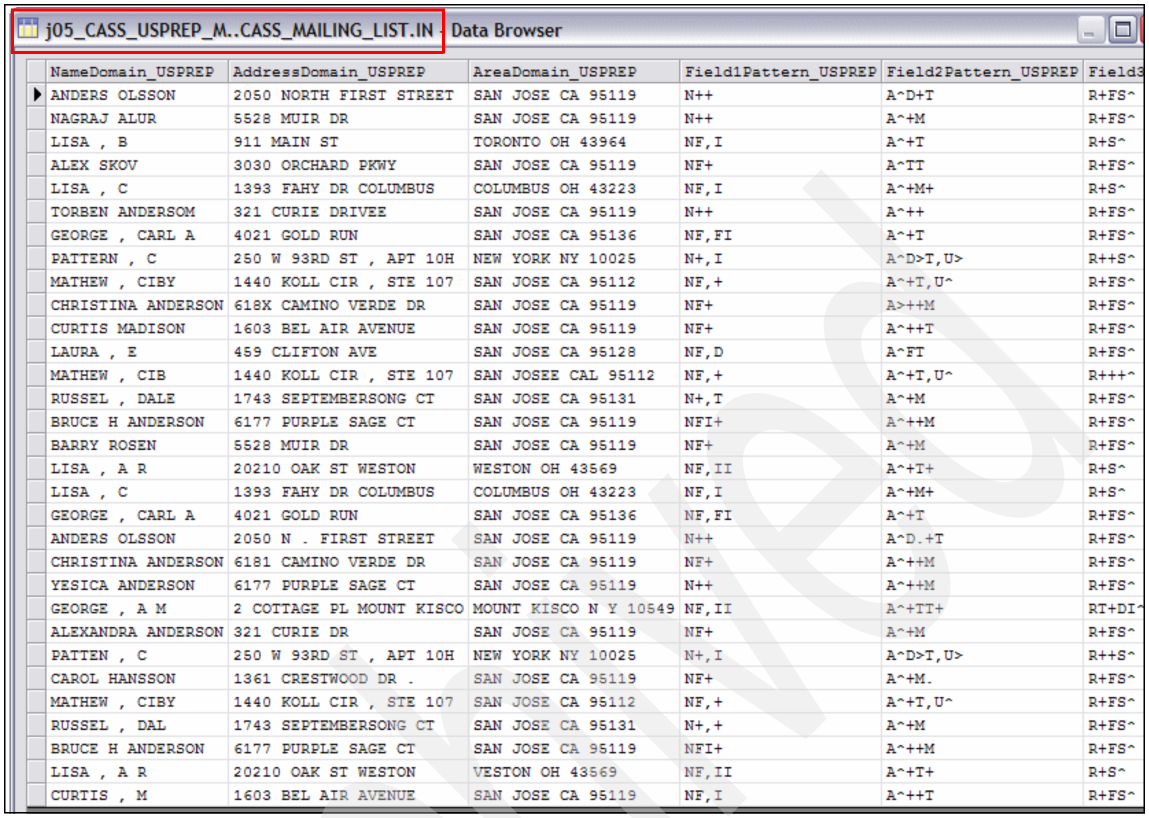
Figure 1-448 Create J05_CASS_USPREP_M job 4/7



j05_CASS_USPREP_M..CASS_MAILING_LIST.IN - Data Browser

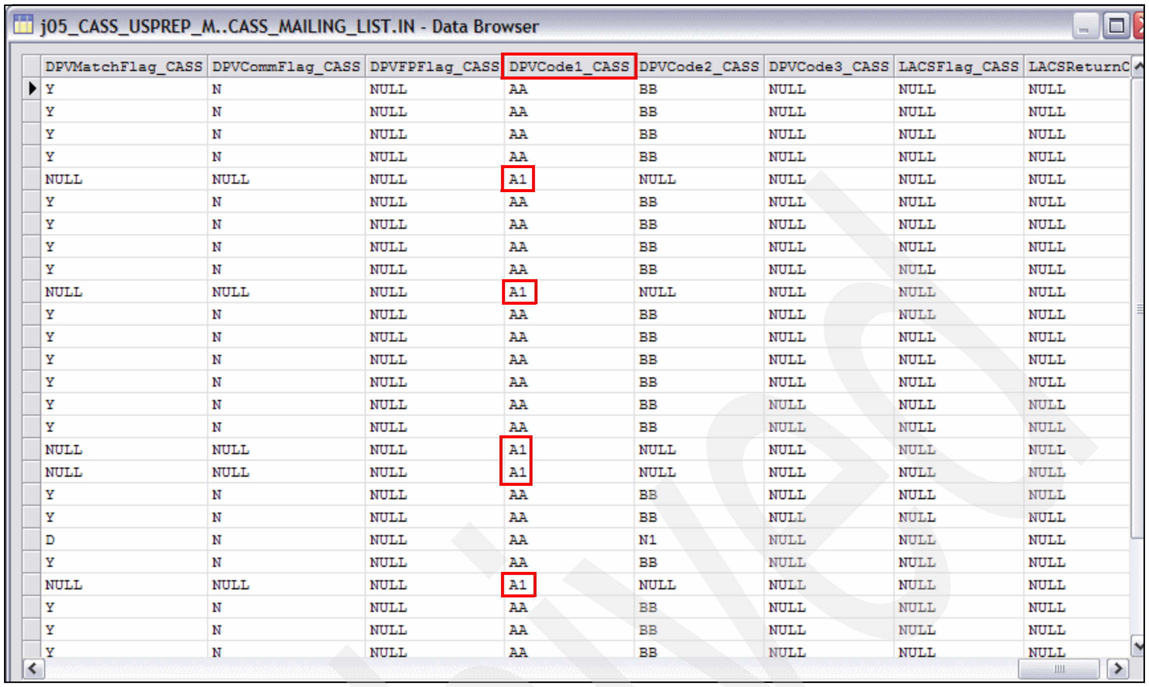
SOURCE	NAME	ADDRESS	CITY	STATE	ZIP	CELL_PHONE	HOME_PHONE	EMAI
A	Anders Olsson	2050 North First Street	San Jose	CA	95119	(800) 817-8231	(800) 817-8232	ande
D	Nagraj Alur	5528 Muir Dr	San Jose	CA	95119			
A	Lisa, B	911 Main St	Toronto	OH	43964		(740) 537-3607	blis
B	Alex Skov	3030 Orchard Pkwy	San Jose	CA	95119	(408) 953-6001	(408) 953-6000	
D	Lisa, C	1393 Fahy Dr Columbus	Columbus	OH	43223	(614) 405-2253	(614) 405-2252	clis
C	Torben Andersom	321 Curie Drivee	San Jose	CA	95119	(408) 782-7100	(408) 782-7100	
D	George, Carl A	4021 Gold Run	San Jose	CA	95136	(408) 224-5446		geor
A	Pattern, C	250 W 93rd St, Apt 10H	New York	NY	10025	(212) 933-0681		
A	Mathew, Ciby	1440 Koll Cir, Ste 107	San Jose	CA	95112	(408) 929-6651		math
B	Christina Anderson	618x Camino Verde Dr	San Jose	CA	95119	(408) 226-2321	(408) 226-2327	myma
B	Curtis Madison	1603 Bel Air Avenue	San Jose	CA	95119	(408) 900-6401	(408) 782-3700	
B	Laura, E	459 Clifton Ave	San Jose	CA	95128	(408) 995-5347		e_la
C	Mathew, Cib	1440 Koll Cir, Ste 107	San Josee	CAL	95112	(408) 929-6651		math
C	Russel, Dale	1743 Septembersong Ct	San Jose	CA	95131		(408) 923-5187	drus
A	Bruce H Anderson	6177 Purple Sage Ct	San Jose	CA	95119	(408) 365-1091		anym
D	Barry Rosen	5528 Muir Dr	San Jose	CA	95119			
B	Lisa, A R	20210 Oak St Weston	Weston	OH	43569		(419) 669-2025	lisa
D	Lisa, C	1393 Fahy Dr Columbus	Columbus	OH	43223	(614) 405-2253	(614) 405-2252	clis
D	George, Carl A	4021 Gold Run	San Jose	CA	95136	(408) 224-5446		geor
A	Anders Olsson	2050 N. First Street	San Jose	CA	95119	(800) 817-8231	(800) 817-8232	ande

Figure 1-449 Create J05_CASS_USPREP_M job 5/7



NameDomain_USPREP	AddressDomain_USPREP	AreaDomain_USPREP	Field1Pattern_USPREP	Field2Pattern_USPREP	Field3
ANDERS OLSSON	2050 NORTH FIRST STREET	SAN JOSE CA 95119	N++	A^D+T	R+FS^
NAGRAJ ALUR	5528 MUIR DR	SAN JOSE CA 95119	N++	A^+M	R+FS^
LISA , B	911 MAIN ST	TORONTO OH 43964	NF,I	A^+T	R+S^
ALEX SKOV	3030 ORCHARD PKWY	SAN JOSE CA 95119	NF+	A^TT	R+FS^
LISA , C	1393 FAHY DR COLUMBUS	COLUMBUS OH 43223	NF,I	A^+M+	R+S^
TORBEN ANDERSOM	321 CURIE DRIVEE	SAN JOSE CA 95119	N++	A^++	R+FS^
GEORGE , CARL A	4021 GOLD RUN	SAN JOSE CA 95136	NF,FI	A^+T	R+FS^
PATTERN , C	250 W 93RD ST , APT 10H	NEW YORK NY 10025	N+,I	A^D>T,U>	R++S^
MATHEW , CIBY	1440 KOLL CIR , STE 107	SAN JOSE CA 95112	NF,+	A^+T,U^	R+FS^
CHRISTINA ANDERSON	618X CAMINO VERDE DR	SAN JOSE CA 95119	NF+	A>++M	R+FS^
CURTIS MADISON	1603 BEL AIR AVENUE	SAN JOSE CA 95119	NF+	A^++T	R+FS^
LAURA , E	459 CLIFTON AVE	SAN JOSE CA 95128	NF,D	A^FT	R+FS^
MATHEW , CIB	1440 KOLL CIR , STE 107	SAN JOSEE CAL 95112	NF,+	A^+T,U^	R+++^
RUSSEL , DALE	1743 SEPTEMBERSONG CT	SAN JOSE CA 95131	N+,T	A^+M	R+FS^
BRUCE H ANDERSON	6177 PURPLE SAGE CT	SAN JOSE CA 95119	NFI+	A^++M	R+FS^
BARRY ROSEN	5528 MUIR DR	SAN JOSE CA 95119	NF+	A^+M	R+FS^
LISA , A R	20210 OAK ST WESTON	WESTON OH 43569	NF,II	A^+T+	R+S^
LISA , C	1393 FAHY DR COLUMBUS	COLUMBUS OH 43223	NF,I	A^+M+	R+S^
GEORGE , CARL A	4021 GOLD RUN	SAN JOSE CA 95136	NF,FI	A^+T	R+FS^
ANDERS OLSSON	2050 N . FIRST STREET	SAN JOSE CA 95119	N++	A^D.+T	R+FS^
CHRISTINA ANDERSON	6181 CAMINO VERDE DR	SAN JOSE CA 95119	NF+	A^++M	R+FS^
YESICA ANDERSON	6177 PURPLE SAGE CT	SAN JOSE CA 95119	N++	A^++M	R+FS^
GEORGE , A M	2 COTTAGE PL MOUNT KISCO	MOUNT KISCO N Y 10549	NF,II	A^+TT+	RT+DI^
ALEXANDRA ANDERSON	321 CURIE DR	SAN JOSE CA 95119	NF+	A^+M	R+FS^
PATTEN , C	250 W 93RD ST , APT 10H	NEW YORK NY 10025	N+,I	A^D>T,U>	R++S^
CAROL HANSSON	1361 CRESTWOOD DR .	SAN JOSE CA 95119	NF+	A^+M.	R+FS^
MATHEW , CIBY	1440 KOLL CIR , STE 107	SAN JOSE CA 95112	NF,+	A^+T,U^	R+FS^
RUSSEL , DAL	1743 SEPTEMBERSONG CT	SAN JOSE CA 95131	N+,+	A^+M	R+FS^
BRUCE H ANDERSON	6177 PURPLE SAGE CT	SAN JOSE CA 95119	NFI+	A^++M	R+FS^
LISA , A R	20210 OAK ST WESTON	VESTON OH 43569	NF,II	A^+T+	R+S^
CURTIS , M	1603 BEL AIR AVENUE	SAN JOSE CA 95119	NF,I	A^++T	R+FS^

Figure 1-450 Create J05_CASS_USPREP_M job 6/7


j05_CASS_USPREP_M..CASS_MAILING_LIST.IN - Data Browser

DPVMatchFlag_CASS	DPVCommFlag_CASS	DPVFPFlag_CASS	DPVCode1_CASS	DPVCode2_CASS	DPVCode3_CASS	LACSFlag_CASS	LACSReturnC
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
NULL	NULL	NULL	A1	NULL	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
NULL	NULL	NULL	A1	NULL	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
NULL	NULL	NULL	A1	NULL	NULL	NULL	NULL
NULL	NULL	NULL	A1	NULL	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
D	N	NULL	AA	N1	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
NULL	NULL	NULL	A1	NULL	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL

Figure 1-451 Create J05_CASS_USPREP_M job 7/7

J06_INVCC_CASS_M

In this step, we then run the Investigate stage with character concatenate on the (address related columns) results of the job "J05_CASS_USPREP_M" on page 374 step to determine investigate addresses that are not recognized by CASS (delivery point verification or DPV).

Note: Due to a bug with handling nulls, we introduced a Transformer stage to convert nulls to a space using column derivation.

We investigate using character concatenate (on CASS generated columns DPVMATCHFLAG_CASS and DPVCODE1_CASS) using a *C* mask, the output of CASS. A value of `A1` in the DPVCODE1_CASS field indicates a potential problem.

Figure 1-452 on page 381 shows the various stages that are used in this job, including the data set that was created in "J05_CASS_USPREP_M" on page 374, a Transformer stage for handling nulls, an Investigate stage, and an output Sequential File stage. We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object created earlier.

Because the configuration of this job is very similar to that described in "J06_INVCC_CASS" on page 228, we do not repeat it here. However, some of the configurations of interest are as follows:

- Figure 1-453 on page 381 shows the HANDLE NULL - Transformer Stage window with derivation substitution for all the columns with the following expression:

  ```
  If ISNULL(OUT.SOURCE) then " else OUT.SOURCE
  ```

  This derivative directs that if the column is null, then spaces are replaced by an empty string.

- Configure the INV_DPV_INFO Investigate stage by right-clicking the INV_DPV_INFO icon and selecting **Properties** in Figure 1-234 on page 232. Then, continue by clicking **Character Concatenate Investigate**. The columns propagated from the input data set are shown in the Available Data Columns list. Select the following columns:

  – DPVMatchFlag_CASS with C mask
  – DPVCode1_CASS with C mask
  – Delivery_AddressLine1_CASS with X mask
  – City_CASS with X mask
  – State_CASS with X mask
  – Zip5_CASS with X mask

  After adding these columns with these masks, click **Stage Properties** as shown in Figure 1-235 on page 232.

- After saving, compiling, and running this job (we do not show statistics), the output of the Investigate stage is shown in Figure 1-455 on page 382 through Figure 1-457 on page 383. It shows a count of six records with values `A1` in the DPVMatchFlag_CASS and DPVCode1_CASS character concatenated columns. As discussed in "J05_CASS_USPREP" on page 219, a value of `A1` in the DPVCode1_CASS indicates an address that did not match. Using a sample of six and rerunning this job shows the relevant records that need further investigation.

Now, proceed to "J07_STAN_CUSTOMER_Domain_Preprocessor" on page 234.

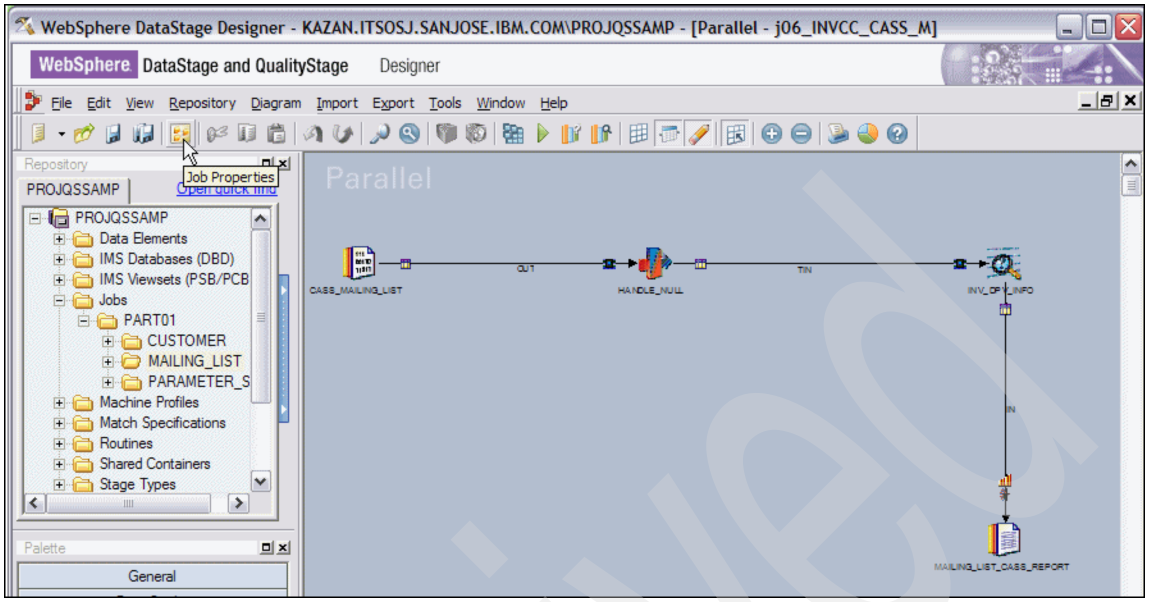
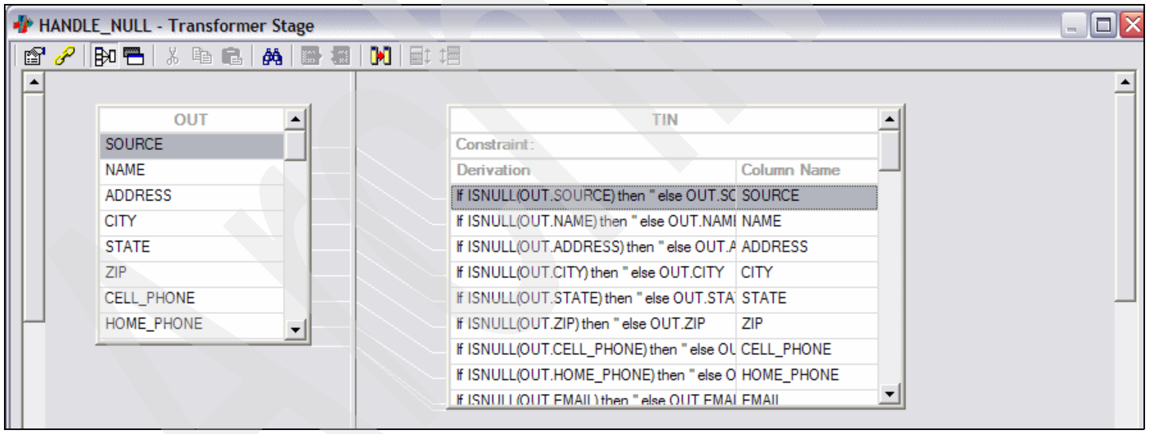


Figure 1-452 Create J06_INVCC_CASS_M job 1/6


HANDLE_NULL - Transformer Stage

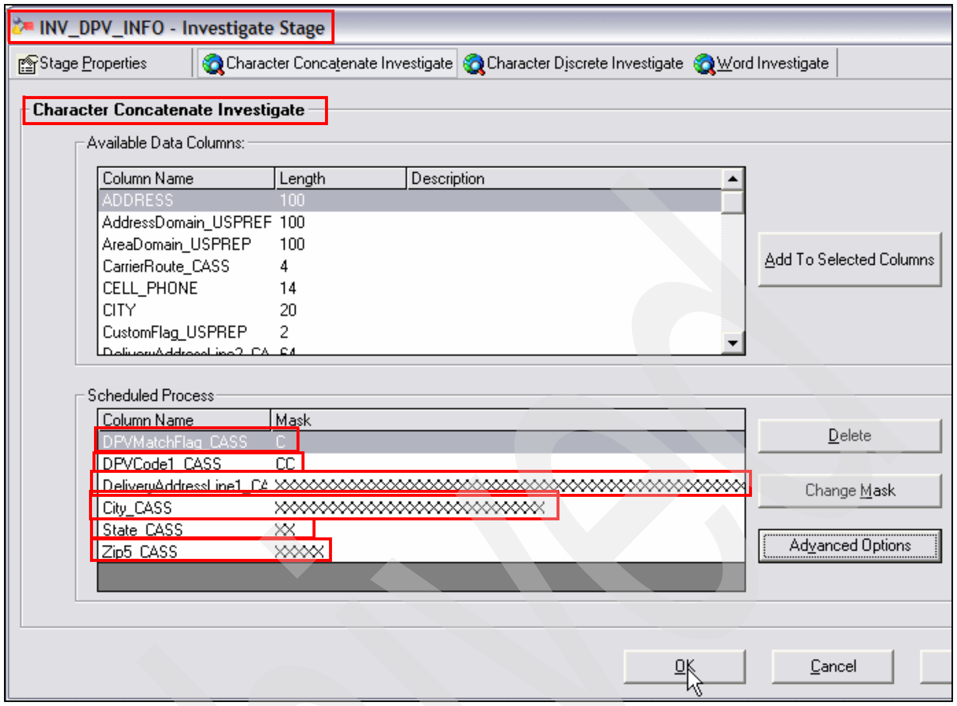
OUT
SOURCE
NAME
ADDRESS
CITY
STATE
ZIP
CELL_PHONE
HOME_PHONE

TIN

Constraint:

Derivation	Column Name
If ISNULL(OUT.SOURCE) then '' else OUT.SC	SOURCE
If ISNULL(OUT.NAME) then '' else OUT.NAMI	NAME
If ISNULL(OUT.ADDRESS) then '' else OUT.A	ADDRESS
If ISNULL(OUT.CITY) then '' else OUT.CITY	CITY
If ISNULL(OUT.STATE) then '' else OUT.STA	STATE
If ISNULL(OUT.ZIP) then '' else OUT.ZIP	ZIP
If ISNULL(OUT.CELL_PHONE) then '' else OU	CELL_PHONE
If ISNULL(OUT.HOME_PHONE) then '' else O	HOME_PHONE
If ISNULL(OUT.EMAIL) then '' else OUT.EMAI	EMAIL

Figure 1-453 Create J06_INVCC_CASS_M job 2/6


INV_DPV_INFO - Investigate Stage

Stage Properties
Character Concatenate Investigate
Character Discrete Investigate
Word Investigate

Character Concatenate Investigate

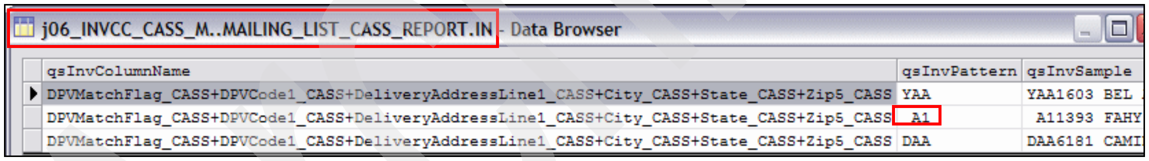
Available Data Columns:

Column Name	Length	Description
ADDRESS	100	
AddressDomain_USPREF	100	
AreaDomain_USPREP	100	
CarrierRoute_CASS	4	
CELL_PHONE	14	
CITY	20	
CustomFlag_USPREP	2	

Scheduled Process

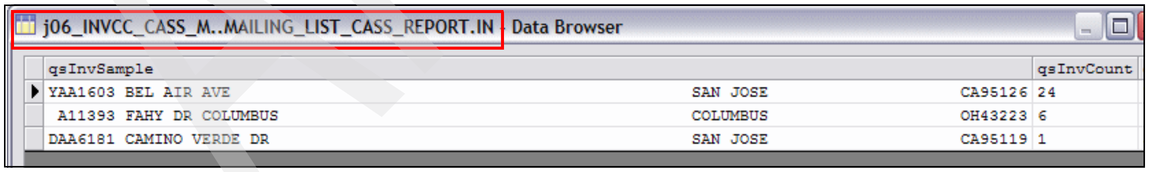
Column Name	Mask
DPVMatchFlag_CASS	C
DPVCode1_CASS	CC
DeliveryAddressLine1_CA	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
City_CASS	XXXXXXXXXXXXXXXXXXXX
State_CASS	XX
Zip5_CASS	XXXXX

Figure 1-454 Create J06_INVCC_CASS_M job 3/6


j06_INVCC_CASS_M..MAILING_LIST_CASS_REPORT.IN - Data Browser

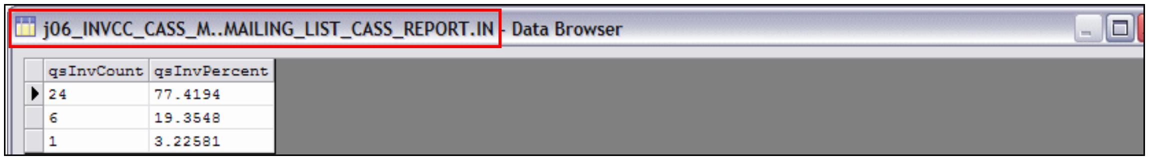
qsInvColumnName	qsInvPattern	qsInvSample
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_CASS+City_CASS+State_CASS+Zip5_CASS	YAA	YAA1603 BEL
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_CASS+City_CASS+State_CASS+Zip5_CASS	A1	A11393 FAHY
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_CASS+City_CASS+State_CASS+Zip5_CASS	DAA	DAA6181 CAMI

Figure 1-455 Create J06_INVCC_CASS_M job 4/6


j06_INVCC_CASS_M..MAILING_LIST_CASS_REPORT.IN - Data Browser

qsInvSample	qsInvCount
YAA1603 BEL AIR AVE SAN JOSE CA95126	24
A11393 FAHY DR COLUMBUS COLUMBUS OH43223	6
DAA6181 CAMINO VERDE DR SAN JOSE CA95119	1

Figure 1-456 Create J06_INVCC_CASS_M job 5/6


j06_INVCC_CASS_M..MAILING_LIST_CASS_REPORT.IN - Data Browser

qsInvCount	qsInvPercent
24	77.4194
6	19.3548
1	3.22581

Figure 1-457 Create J06_INVCC_CASS_M job 6/6

J07_STAN_MAILING_LIST_Domain_Preprocessor

In this step, we standardize the name and address contents of the output of job "J05_CASS_USPREP_M" on page 374 using the domain-preprocessor USPREP rule set. We also add a column to the output that only had the first three characters of the ZIP code using a Transformer stage. This new column (ZIP3) is used as a blocking variable in the next matching stage.

After CASS validates and corrects the address fields but not the name fields, the Standardize stage is run again with the domain-preprocessor rule set USPREP on the name fields and CASS corrected address fields into Name, Address, and Area domains.

Figure 1-458 on page 384 shows the various stages that are used in this job, including the data set that was created in "J05_CASS_USPREP_M" on page 374, a Standardize stage, a Transformer stage to add a column, and an output Data Set stage. We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object created earlier.

Because the configuration of this job is very similar to that described in "J07_STAN_CUSTOMER_Domain_Preprocessor" on page 234, we do not repeat it here. However, some of the configurations of interest are as follows:

- The Standardize Rule Process window in Figure 1-459 on page 385 through Figure 1-461 on page 385 show the domain preprocessor USPREP rule set with literals and selected columns as follows:

  ```
  ZQNAMEZQ NAME ZQPUTAZQ DeliveryAddressLine1_CASS
  DeliveryAddressLine2_CASS ZQPUTRZQ City_CASS State_CASS Zip5_CASS
  ```

- The ADD_ZIP3 - Transformer Stage window in Figure 1-462 on page 386 shows the addition of a ZIP3 column to the output that only contains the first three digits of the 5-digit ZIP5_CASS field.
- After saving, compiling, and running this job (Figure 1-463 on page 387), view the content of output data set as shown in Figure 1-464 on page 388 through Figure 1-466 on page 390.

  As discussed before, this report shows:

  – Columns NameDomain_USPREP (contains prefix, first name, last name, suffix tokens), AddressDomain_USPREP (contains apartment, street name and street type tokens), and AreaDomain_USPREP (contains state, ZIP code tokens) that were parsed from the input columns.

– InputPattern_USPREP and OutboundPattern_USPREP columns that contain the patterns that were generated after processing the name and address columns in the input file.

A visual analysis of the report shows some problems with the AddressDomain_USPREP column that has the name of the city in it as shown in Figure 1-465 on page 389. In the real-world, the volume of data would be too large to attempt a visual analysis of the report. Therefore, proceed to "J08_STAN_MAILING_LIST_Domain_Specific" on page 390.

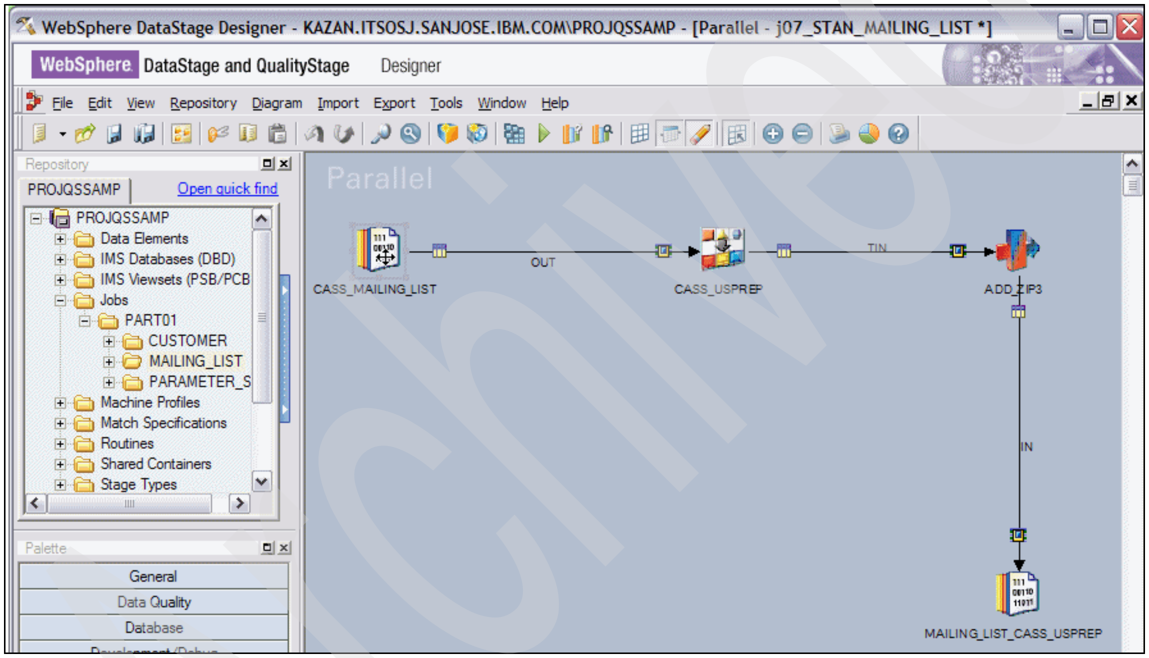
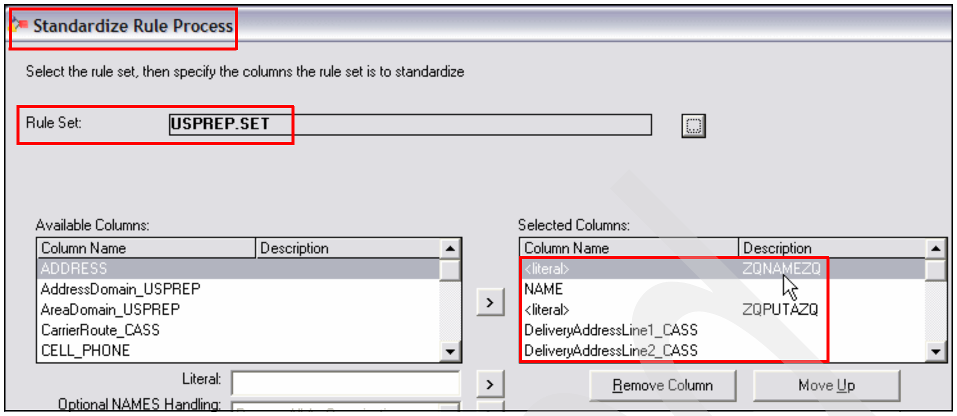


Figure 1-458 Create J07_STAN_MAILING_LIST_Domain_Preprocessor 1/9


Standardize Rule Process

Select the rule set, then specify the columns the rule set is to standardize

Rule Set: USPREP.SET

Available Columns:

Column Name	Description
ADDRESS	
AddressDomain_USPREP	
AreaDomain_USPREP	
CarrierRoute_CASS	
CELL_PHONE	

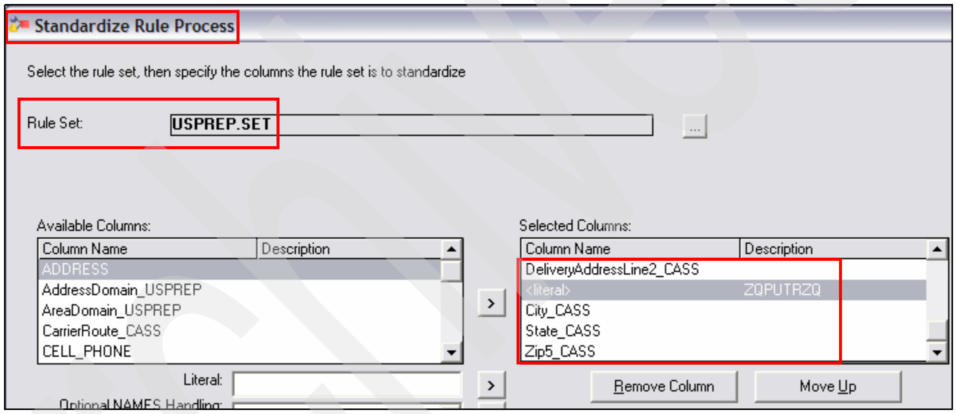
Selected Columns:

Column Name	Description
<literal>	ZQNAMEZQ
NAME	
<literal>	ZQPUTAZQ
DeliveryAddressLine1_CASS	
DeliveryAddressLine2_CASS	

Literal:

Optional NAMES Handling:

Figure 1-459 Create J07_STAN_MAILING_LIST_Domain_Preprocessor 2/9


Standardize Rule Process

Select the rule set, then specify the columns the rule set is to standardize

Rule Set: USPREP.SET

Available Columns:

Column Name	Description
ADDRESS	
AddressDomain_USPREP	
AreaDomain_USPREP	
CarrierRoute_CASS	
CELL_PHONE	

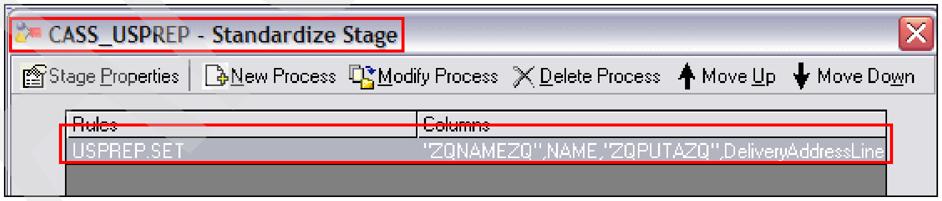
Selected Columns:

Column Name	Description
DeliveryAddressLine2_CASS	
<literal>	ZQPUTRZQ
City_CASS	
State_CASS	
Zip5_CASS	

Literal:

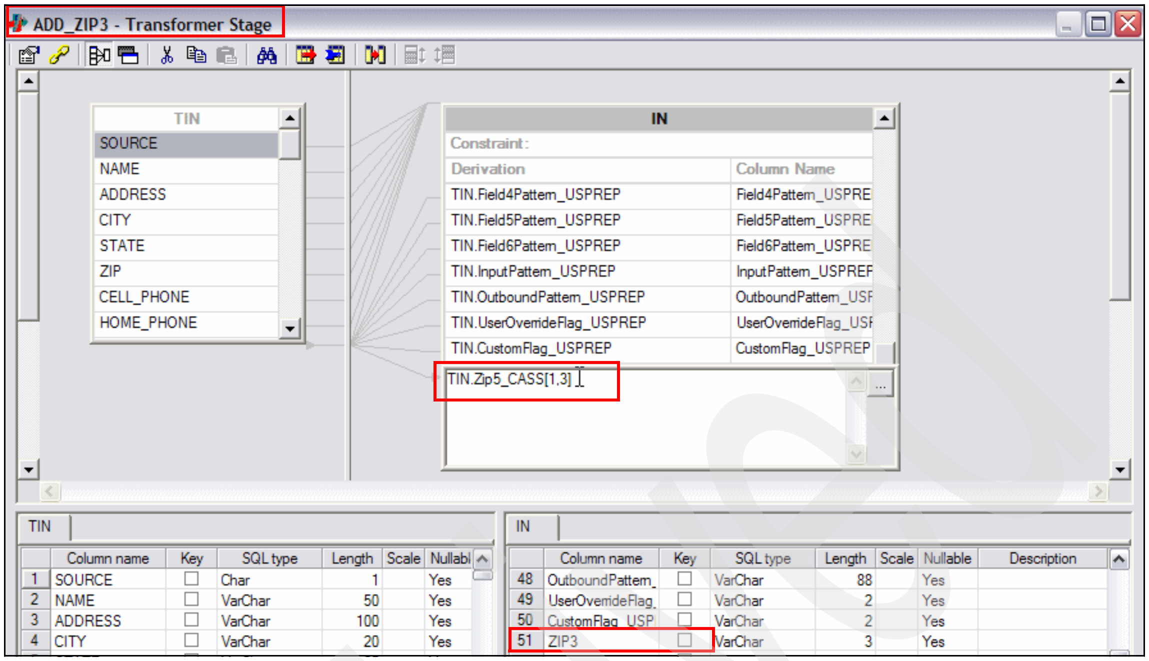
Optional NAMES Handling:

Figure 1-460 Create J07_STAN_MAILING_LIST_Domain_Preprocessor 3/9


CASS_USPREP - Standardize Stage

Rules	Columns
USPREP.SET	"ZQNAMEZQ",NAME,"ZQPUTAZQ",DeliveryAddressLine

Figure 1-461 Create J07_STAN_MAILING_LIST_Domain_Preprocessor 4/9


ADD_ZIP3 - Transformer Stage

TIN
SOURCE
NAME
ADDRESS
CITY
STATE
ZIP
CELL_PHONE
HOME_PHONE

IN

Constraint:

Derivation	Column Name
TIN.Field4Pattern_USPREP	Field4Pattern_USPRE
TIN.Field5Pattern_USPREP	Field5Pattern_USPRE
TIN.Field6Pattern_USPREP	Field6Pattern_USPRE
TIN.InputPattern_USPREP	InputPattern_USPREF
TIN.OutboundPattern_USPREP	OutboundPattern_USF
TIN.UserOverrideFlag_USPREP	UserOverrideFlag_USF
TIN.CustomFlag_USPREP	CustomFlag_USPREP

TIN.Zip5_CASS[1,3]

TIN

	Column name	Key	SQL type	Length	Scale	Nullabl
1	SOURCE	☐	Char	1		Yes
2	NAME	☐	VarChar	50		Yes
3	ADDRESS	☐	VarChar	100		Yes
4	CITY	☐	VarChar	20		Yes

IN

	Column name	Key	SQL type	Length	Scale	Nullable	Description
48	OutboundPattern_	☐	VarChar	88		Yes	
49	UserOverrideFlag_	☐	VarChar	2		Yes	
50	CustomFlag_USP	☐	VarChar	2		Yes	
51	ZIP3	☐	VarChar	3		Yes	

Figure 1-462   Create J07_STAN_MAILING_LIST_Domain_Preprocessor 5/9

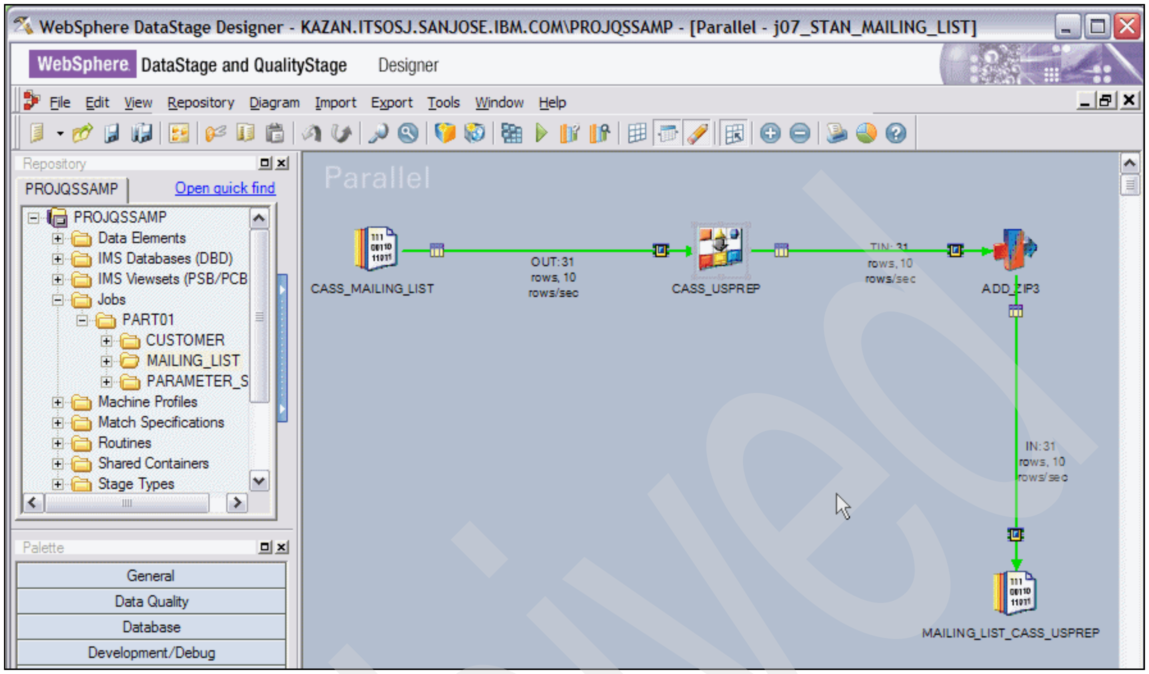
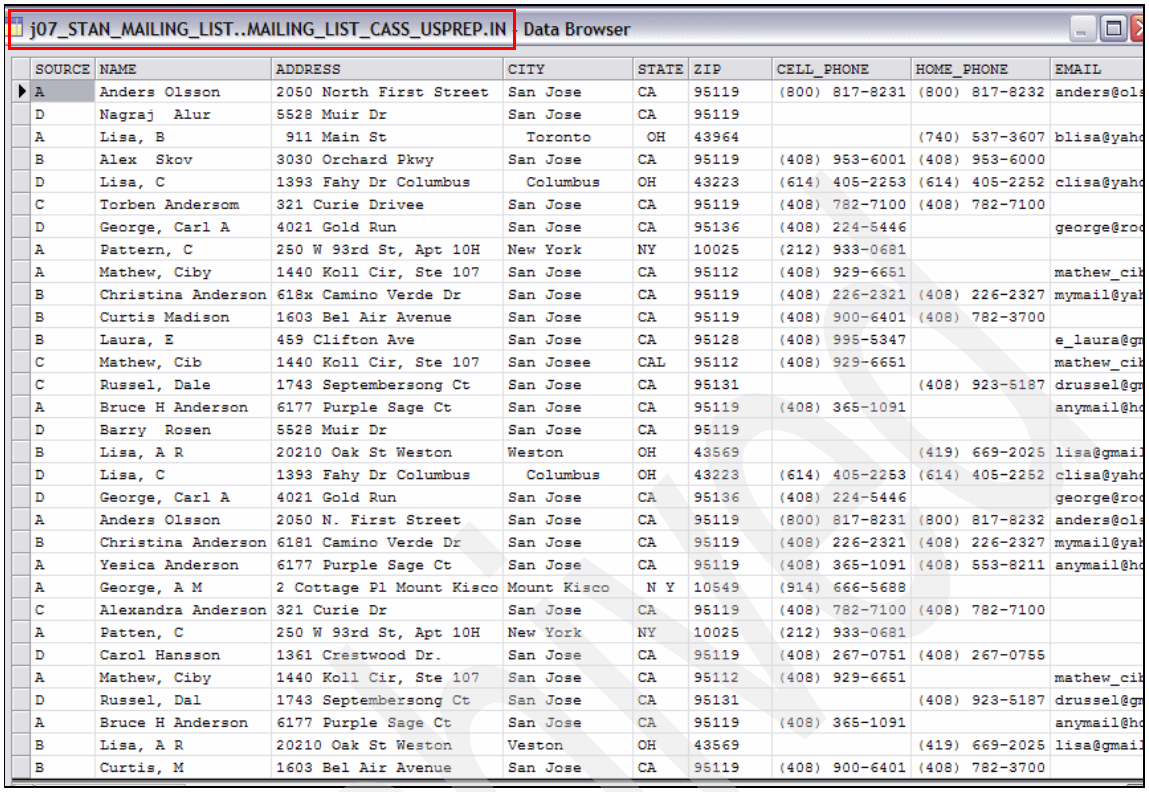


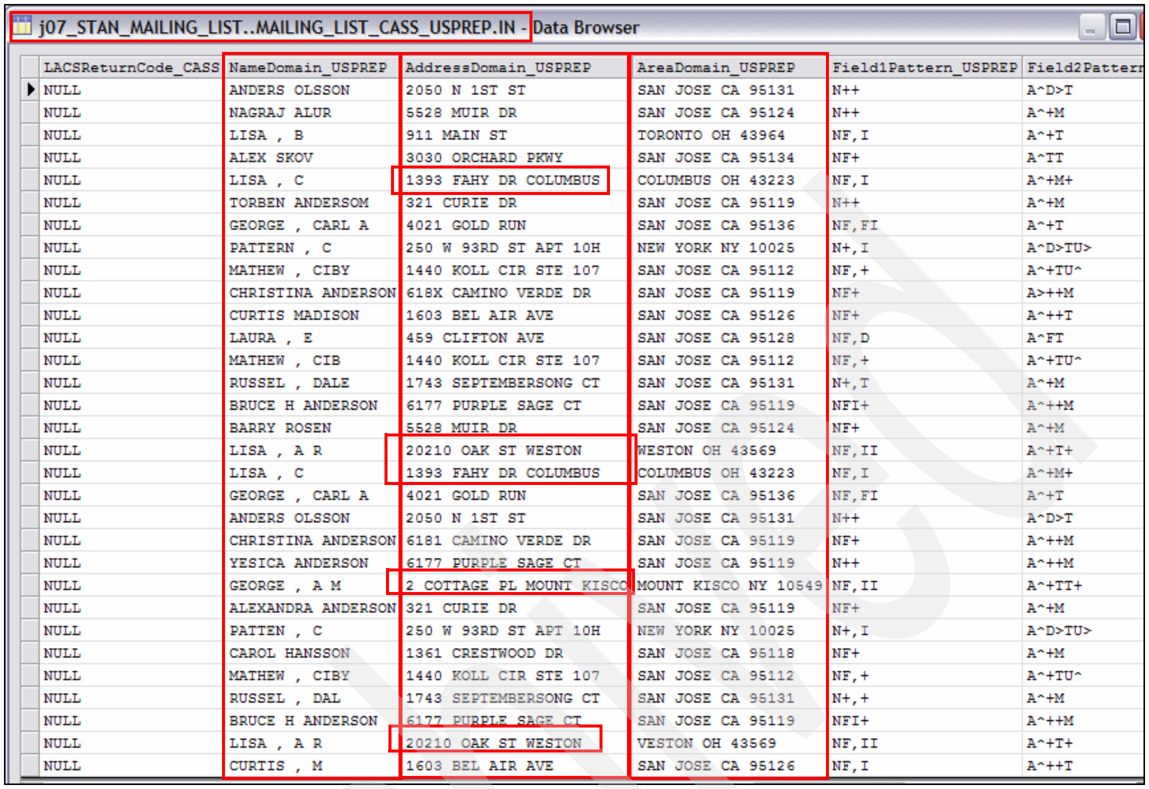
Figure 1-463 Create J07_STAN_MAILING_LIST_Domain_Preprocessor 6/9



j07_STAN_MAILING_LIST..MAILING_LIST_CASS_USPREP.IN - Data Browser

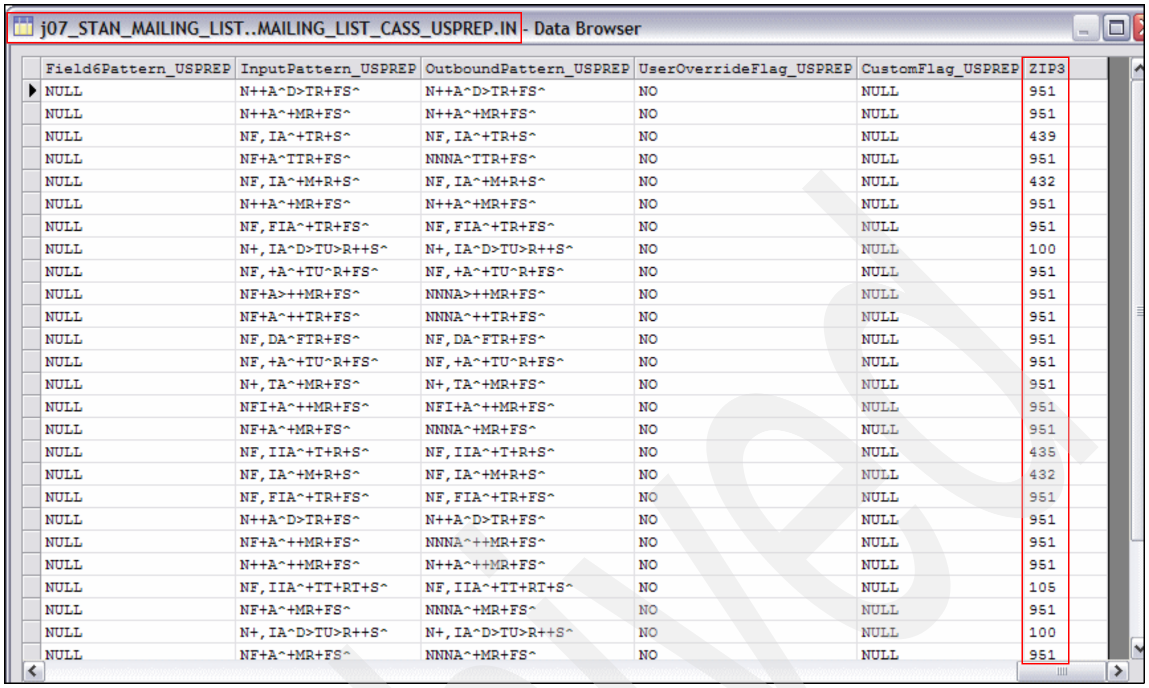
SOURCE	NAME	ADDRESS	CITY	STATE	ZIP	CELL_PHONE	HOME_PHONE	EMAIL
A	Anders Olsson	2050 North First Street	San Jose	CA	95119	(800) 817-8231	(800) 817-8232	anders@ols
D	Nagraj Alur	5528 Muir Dr	San Jose	CA	95119			
A	Lisa, B	911 Main St	Toronto	OH	43964		(740) 537-3607	blisa@yahc
B	Alex Skov	3030 Orchard Pkwy	San Jose	CA	95119	(408) 953-6001	(408) 953-6000	
D	Lisa, C	1393 Fahy Dr Columbus	Columbus	OH	43223	(614) 405-2253	(614) 405-2252	clisa@yahc
C	Torben Andersom	321 Curie Drivee	San Jose	CA	95119	(408) 782-7100	(408) 782-7100	
D	George, Carl A	4021 Gold Run	San Jose	CA	95136	(408) 224-5446		george@roc
A	Pattern, C	250 W 93rd St, Apt 10H	New York	NY	10025	(212) 933-0681		
A	Mathew, Ciby	1440 Koll Cir, Ste 107	San Jose	CA	95112	(408) 929-6651		mathew_cit
B	Christina Anderson	618x Camino Verde Dr	San Jose	CA	95119	(408) 226-2321	(408) 226-2327	mymail@yah
B	Curtis Madison	1603 Bel Air Avenue	San Jose	CA	95119	(408) 900-6401	(408) 782-3700	
B	Laura, E	459 Clifton Ave	San Jose	CA	95128	(408) 995-5347		e_laura@gm
C	Mathew, Cib	1440 Koll Cir, Ste 107	San Josee	CAL	95112	(408) 929-6651		mathew_cit
C	Russel, Dale	1743 Septembersong Ct	San Jose	CA	95131		(408) 923-5187	drussel@gm
A	Bruce H Anderson	6177 Purple Sage Ct	San Jose	CA	95119	(408) 365-1091		anymail@hc
D	Barry Rosen	5528 Muir Dr	San Jose	CA	95119			
B	Lisa, A R	20210 Oak St Weston	Weston	OH	43569		(419) 669-2025	lisa@gmail
D	Lisa, C	1393 Fahy Dr Columbus	Columbus	OH	43223	(614) 405-2253	(614) 405-2252	clisa@yahc
D	George, Carl A	4021 Gold Run	San Jose	CA	95136	(408) 224-5446		george@roc
A	Anders Olsson	2050 N. First Street	San Jose	CA	95119	(800) 817-8231	(800) 817-8232	anders@ols
B	Christina Anderson	6181 Camino Verde Dr	San Jose	CA	95119	(408) 226-2321	(408) 226-2327	mymail@yah
A	Yesica Anderson	6177 Purple Sage Ct	San Jose	CA	95119	(408) 365-1091	(408) 553-8211	anymail@hc
A	George, A M	2 Cottage Pl Mount Kisco	Mount Kisco	N Y	10549	(914) 666-5688		
C	Alexandra Anderson	321 Curie Dr	San Jose	CA	95119	(408) 782-7100	(408) 782-7100	
A	Patten, C	250 W 93rd St, Apt 10H	New York	NY	10025	(212) 933-0681		
D	Carol Hansson	1361 Crestwood Dr.	San Jose	CA	95119	(408) 267-0751	(408) 267-0755	
A	Mathew, Ciby	1440 Koll Cir, Ste 107	San Jose	CA	95112	(408) 929-6651		mathew_cit
D	Russel, Dal	1743 Septembersong Ct	San Jose	CA	95131		(408) 923-5187	drussel@gm
A	Bruce H Anderson	6177 Purple Sage Ct	San Jose	CA	95119	(408) 365-1091		anymail@hc
B	Lisa, A R	20210 Oak St Weston	Veston	OH	43569		(419) 669-2025	lisa@gmail
B	Curtis, M	1603 Bel Air Avenue	San Jose	CA	95119	(408) 900-6401	(408) 782-3700	

Figure 1-464 Create J07_STAN_MAILING_LIST_Domain_Preprocessor 7/9


j07_STAN_MAILING_LIST..MAILING_LIST_CASS_USPREP.IN - Data Browser

LACSReturnCode_CASS	NameDomain_USPREP	AddressDomain_USPREP	AreaDomain_USPREP	Field1Pattern_USPREP	Field2Pattern
NULL	ANDERS OLSSON	2050 N 1ST ST	SAN JOSE CA 95131	N++	A^D>T
NULL	NAGRAJ ALUR	5528 MUIR DR	SAN JOSE CA 95124	N++	A^+M
NULL	LISA , B	911 MAIN ST	TORONTO OH 43964	NF,I	A^+T
NULL	ALEX SKOV	3030 ORCHARD PKWY	SAN JOSE CA 95134	NF+	A^TT
NULL	LISA , C	1393 FAHY DR COLUMBUS	COLUMBUS OH 43223	NF,I	A^+M+
NULL	TORBEN ANDERSOM	321 CURIE DR	SAN JOSE CA 95119	N++	A^+M
NULL	GEORGE , CARL A	4021 GOLD RUN	SAN JOSE CA 95136	NF,FI	A^+T
NULL	PATTERN , C	250 W 93RD ST APT 10H	NEW YORK NY 10025	N+,I	A^D>TU>
NULL	MATHEW , CIBY	1440 KOLL CIR STE 107	SAN JOSE CA 95112	NF,+	A^+TU^
NULL	CHRISTINA ANDERSON	618X CAMINO VERDE DR	SAN JOSE CA 95119	NF+	A>++M
NULL	CURTIS MADISON	1603 BEL AIR AVE	SAN JOSE CA 95126	NF+	A^++T
NULL	LAURA , E	459 CLIFTON AVE	SAN JOSE CA 95128	NF,D	A^FT
NULL	MATHEW , CIB	1440 KOLL CIR STE 107	SAN JOSE CA 95112	NF,+	A^+TU^
NULL	RUSSEL , DALE	1743 SEPTEMBERSONG CT	SAN JOSE CA 95131	N+,T	A^+M
NULL	BRUCE H ANDERSON	6177 PURPLE SAGE CT	SAN JOSE CA 95119	NFI+	A^++M
NULL	BARRY ROSEN	5528 MUIR DR	SAN JOSE CA 95124	NF+	A^+M
NULL	LISA , A R	20210 OAK ST WESTON	WESTON OH 43569	NF,II	A^+T+
NULL	LISA , C	1393 FAHY DR COLUMBUS	COLUMBUS OH 43223	NF,I	A^+M+
NULL	GEORGE , CARL A	4021 GOLD RUN	SAN JOSE CA 95136	NF,FI	A^+T
NULL	ANDERS OLSSON	2050 N 1ST ST	SAN JOSE CA 95131	N++	A^D>T
NULL	CHRISTINA ANDERSON	6181 CAMINO VERDE DR	SAN JOSE CA 95119	NF+	A^++M
NULL	YESICA ANDERSON	6177 PURPLE SAGE CT	SAN JOSE CA 95119	N++	A^++M
NULL	GEORGE , A M	2 COTTAGE PL MOUNT KISCO	MOUNT KISCO NY 10549	NF,II	A^+TT+
NULL	ALEXANDRA ANDERSON	321 CURIE DR	SAN JOSE CA 95119	NF+	A^+M
NULL	PATTEN , C	250 W 93RD ST APT 10H	NEW YORK NY 10025	N+,I	A^D>TU>
NULL	CAROL HANSSON	1361 CRESTWOOD DR	SAN JOSE CA 95118	NF+	A^+M
NULL	MATHEW , CIBY	1440 KOLL CIR STE 107	SAN JOSE CA 95112	NF,+	A^+TU^
NULL	RUSSEL , DAL	1743 SEPTEMBERSONG CT	SAN JOSE CA 95131	N+,+	A^+M
NULL	BRUCE H ANDERSON	6177 PURPLE SAGE CT	SAN JOSE CA 95119	NFI+	A^++M
NULL	LISA , A R	20210 OAK ST WESTON	VESTON OH 43569	NF,II	A^+T+
NULL	CURTIS , M	1603 BEL AIR AVE	SAN JOSE CA 95126	NF,I	A^++T

Figure 1-465 Create J07_STAN_MAILING_LIST_Domain_Preprocessor 8/9


j07_STAN_MAILING_LIST..MAILING_LIST_CASS_USPREP.IN - Data Browser

Field6Pattern_USPREP	InputPattern_USPREP	OutboundPattern_USPREP	UserOverrideFlag_USPREP	CustomFlag_USPREP	ZIP3
NULL	N++A^D>TR+FS^	N++A^D>TR+FS^	NO	NULL	951
NULL	N++A^+MR+FS^	N++A^+MR+FS^	NO	NULL	951
NULL	NF,IA^+TR+S^	NF,IA^+TR+S^	NO	NULL	439
NULL	NF+A^TTR+FS^	NNNA^TTR+FS^	NO	NULL	951
NULL	NF,IA^+M+R+S^	NF,IA^+M+R+S^	NO	NULL	432
NULL	N++A^+MR+FS^	N++A^+MR+FS^	NO	NULL	951
NULL	NF,FIA^+TR+FS^	NF,FIA^+TR+FS^	NO	NULL	951
NULL	N+,IA^D>TU>R++S^	N+,IA^D>TU>R++S^	NO	NULL	100
NULL	NF,+A^+TU^R+FS^	NF,+A^+TU^R+FS^	NO	NULL	951
NULL	NF+A>++MR+FS^	NNNA>++MR+FS^	NO	NULL	951
NULL	NF+A^++TR+FS^	NNNA^++TR+FS^	NO	NULL	951
NULL	NF,DA^FTR+FS^	NF,DA^FTR+FS^	NO	NULL	951
NULL	NF,+A^+TU^R+FS^	NF,+A^+TU^R+FS^	NO	NULL	951
NULL	N+,TA^+MR+FS^	N+,TA^+MR+FS^	NO	NULL	951
NULL	NFI+A^++MR+FS^	NFI+A^++MR+FS^	NO	NULL	951
NULL	NF+A^+MR+FS^	NNNA^+MR+FS^	NO	NULL	951
NULL	NF,IIA^+T+R+S^	NF,IIA^+T+R+S^	NO	NULL	435
NULL	NF,IA^+M+R+S^	NF,IA^+M+R+S^	NO	NULL	432
NULL	NF,FIA^+TR+FS^	NF,FIA^+TR+FS^	NO	NULL	951
NULL	N++A^D>TR+FS^	N++A^D>TR+FS^	NO	NULL	951
NULL	NF+A^++MR+FS^	NNNA^++MR+FS^	NO	NULL	951
NULL	N++A^++MR+FS^	N++A^++MR+FS^	NO	NULL	951
NULL	NF,IIA^+TT+RT+S^	NF,IIA^+TT+RT+S^	NO	NULL	105
NULL	NF+A^+MR+FS^	NNNA^+MR+FS^	NO	NULL	951
NULL	N+,IA^D>TU>R++S^	N+,IA^D>TU>R++S^	NO	NULL	100
NULL	NF+A^+MR+FS^	NNNA^+MR+FS^	NO	NULL	951

Figure 1-466 Create J07_STAN_MAILING_LIST_Domain_Preprocessor 9/9

J08_STAN_MAILING_LIST_Domain_Specific

In this step, we standardize the name and address contents of the output of job "J07_STAN_MAILING_LIST_Domain_Preprocessor" on page 383 using the domain-specific USNAME (with column NameDomain_USPREP), USADDR (with column AddressDomain_USPREP) and USAREA (with column AreaDomain_USPREP) rule sets. Three separate processes were defined—one for each rule set.

> **Note:** Here again, we had to introduce a null handling Transformer stage to circumvent a bug.

Figure 1-467 on page 392 shows the various stages that are used in this job, including the data set that was created in "J07_STAN_CUSTOMER_Domain_Preprocessor" on page 234, a Standardize stage, a Transformer stage to handle nulls, and an output Data Set stage. We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object created earlier.

Because the configuration of this job is very similar to that described in “J08_STAN_CUSTOMER_Domain_Specific” on page 241, we do not repeat it here. However, some of the configurations of interest are as follows:

- Figure 1-468 on page 392 shows the Standardize Rule Process window with the USNAME rule set for the NameDomain_USPREP column. Figure 1-469 on page 393 shows the Standardize Rule Process window with the USADDR rule set for the AddressDomain_USPREP column. Figure 1-470 on page 393 shows the Standardize Rule Process window with the USAREA rule set for the AreaDomain_USPREP column. Figure 1-471 on page 393 shows the three rules that were used in this Standardize job.
- Figure 1-472 on page 394 shows the HANDLE_NULL Transformer Stage windows derivation substitution for all the columns with the following expression:

  ```
  If ISNULL(TIN.CountryCode_USAREA) then “ else TIN.CountryCode_USAREA
  ```

  This derivative directs that if the column is null, then spaces is replaced by an empty string.
- After saving, compiling, and running this job (Figure 1-473 on page 394), you view the content of the STAN_MAILING_LIST data set object as shown in Figure 1-474 on page 395 through Figure 1-478 on page 399.

  This report shows Standardize Stage added columns such as FirstName_USNAME and PrimaryName_USNAME, InputPattern_USNAME in Figure 1-475 on page 396; UnhandledPattern_USNAME, UnhandledData_USNAME and InputPattern_USNAME in Figure 1-476 on page 397; UnhandledPattern_USADDR, UnhandledData_USADDR and InputPattern_USADDR in Figure 1-477 on page 398, and UnhandledPattern_USAREA and UnhandledData_USAREA in Figure 1-478 on page 399.

The next step is to identify any unhandled patterns and classifications by performing an investigate as described in “J09_INVCC_STAN_MAILING_LIST” on page 399.

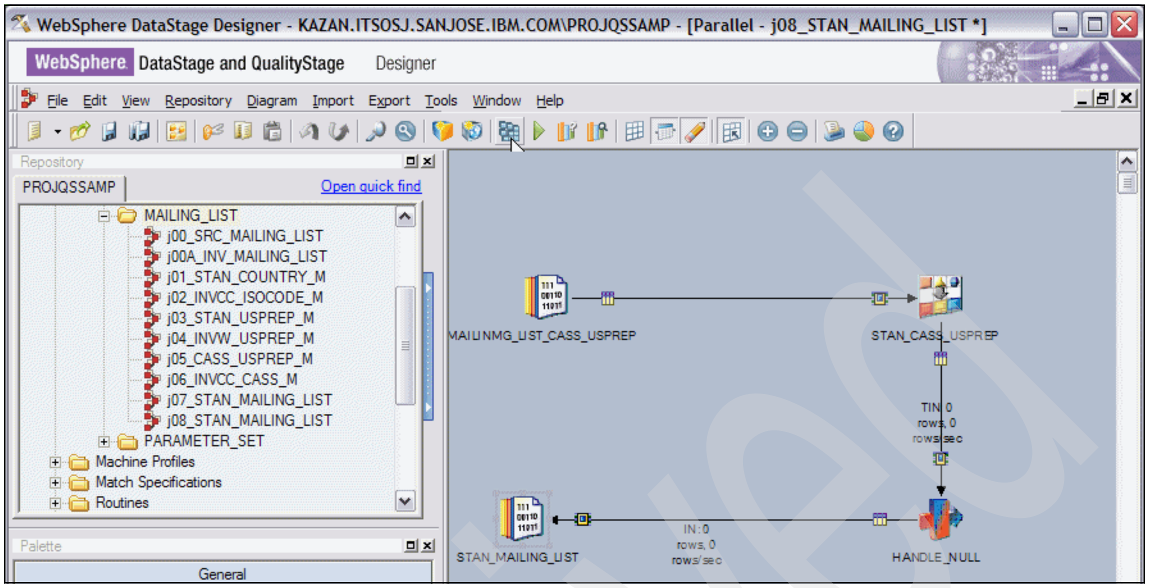
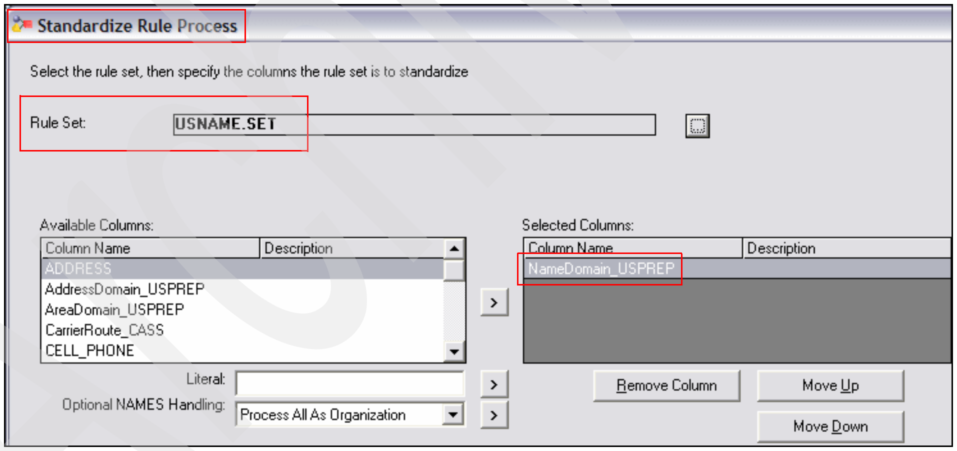


Figure 1-467   Create J08_STAN_MAILING_LIST_Domain_Specific 1/12



Select the rule set, then specify the columns the rule set is to standardize

Rule Set: USNAME.SET

Available Columns:

Column Name	Description
ADDRESS	
AddressDomain_USPREP	
AreaDomain_USPREP	
CarrierRoute_CASS	
CELL_PHONE	

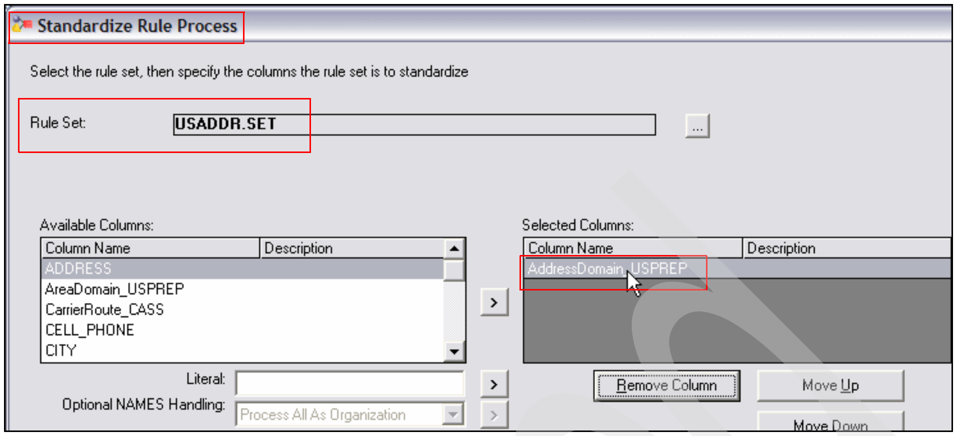
Selected Columns:

Column Name	Description
NameDomain_USPREP	

Literal:

Optional NAMES Handling: Process All As Organization

Figure 1-468   Create J08_STAN_MAILING_LIST_Domain_Specific 2/12


Standardize Rule Process

Select the rule set, then specify the columns the rule set is to standardize

Rule Set: USADDR.SET

Available Columns:

Column Name	Description
ADDRESS	
AreaDomain_USPREP	
CarrierRoute_CASS	
CELL_PHONE	
CITY	

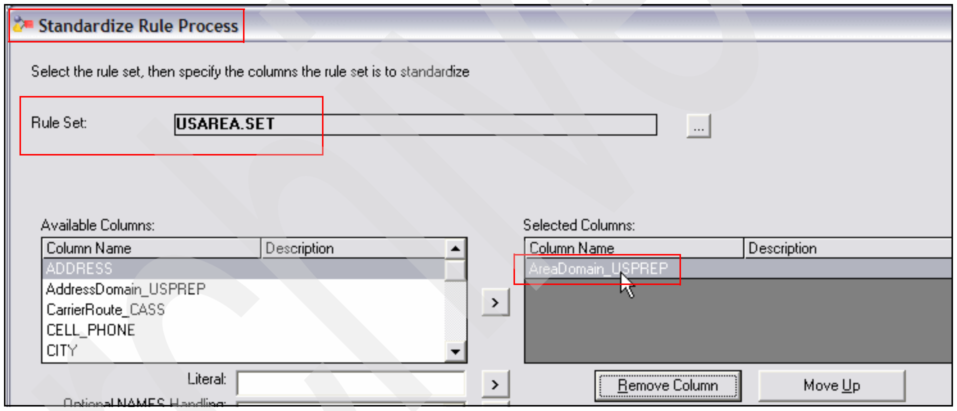
Selected Columns:

Column Name	Description
AddressDomain_USPREP	

Literal:

Optional NAMES Handling: Process All As Organization

Figure 1-469 Create J08_STAN_MAILING_LIST_Domain_Specific 3/12


Standardize Rule Process

Select the rule set, then specify the columns the rule set is to standardize

Rule Set: USAREA.SET

Available Columns:

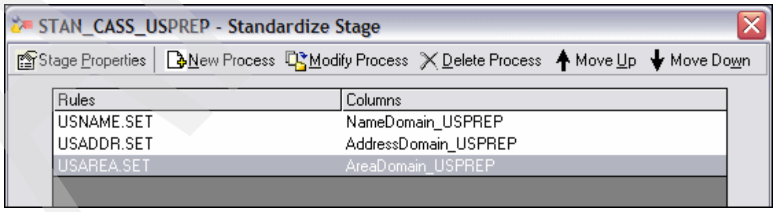
Column Name	Description
ADDRESS	
AddressDomain_USPREP	
CarrierRoute_CASS	
CELL_PHONE	
CITY	

Selected Columns:

Column Name	Description
AreaDomain_USPREP	

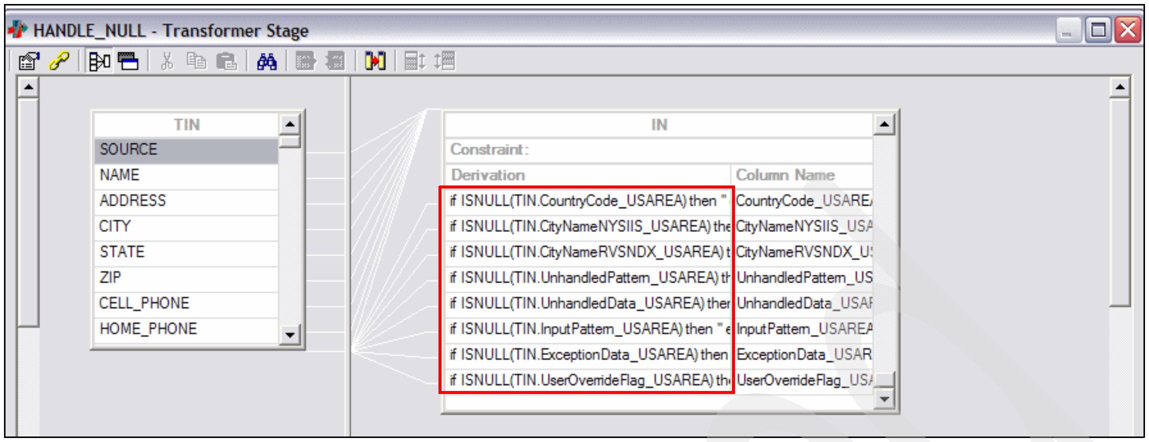
Literal:

Figure 1-470 Create J08_STAN_MAILING_LIST_Domain_Specific 4/12


STAN_CASS_USPREP - Standardize Stage

Rules	Columns
USNAME.SET	NameDomain_USPREP
USADDR.SET	AddressDomain_USPREP
USAREA.SET	AreaDomain_USPREP

Figure 1-471 Create J08_STAN_MAILING_LIST_Domain_Specific 5/12



Derivation	Column Name
if ISNULL(TIN.CountryCode_USAREA) then "	CountryCode_USARE
if ISNULL(TIN.CityNameNYSIIS_USAREA) the	CityNameNYSIIS_USA
if ISNULL(TIN.CityNameRVSNDX_USAREA) t	CityNameRVSNDX_U
if ISNULL(TIN.UnhandledPattern_USAREA) th	UnhandledPattern_US
if ISNULL(TIN.UnhandledData_USAREA) ther	UnhandledData_USAR
if ISNULL(TIN.InputPattern_USAREA) then " e	InputPattern_USAREA
if ISNULL(TIN.ExceptionData_USAREA) then	ExceptionData_USAR
if ISNULL(TIN.UserOverrideFlag_USAREA) th	UserOverrideFlag_USA

Figure 1-472   Create J08_STAN_MAILING_LIST_Domain_Specific 6/12

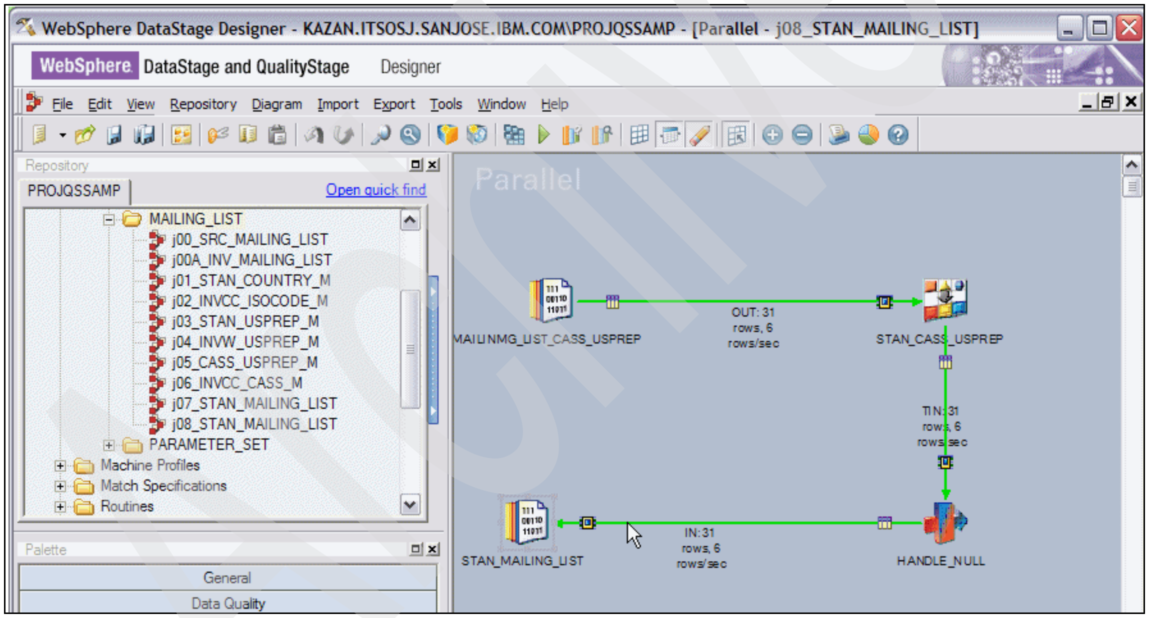
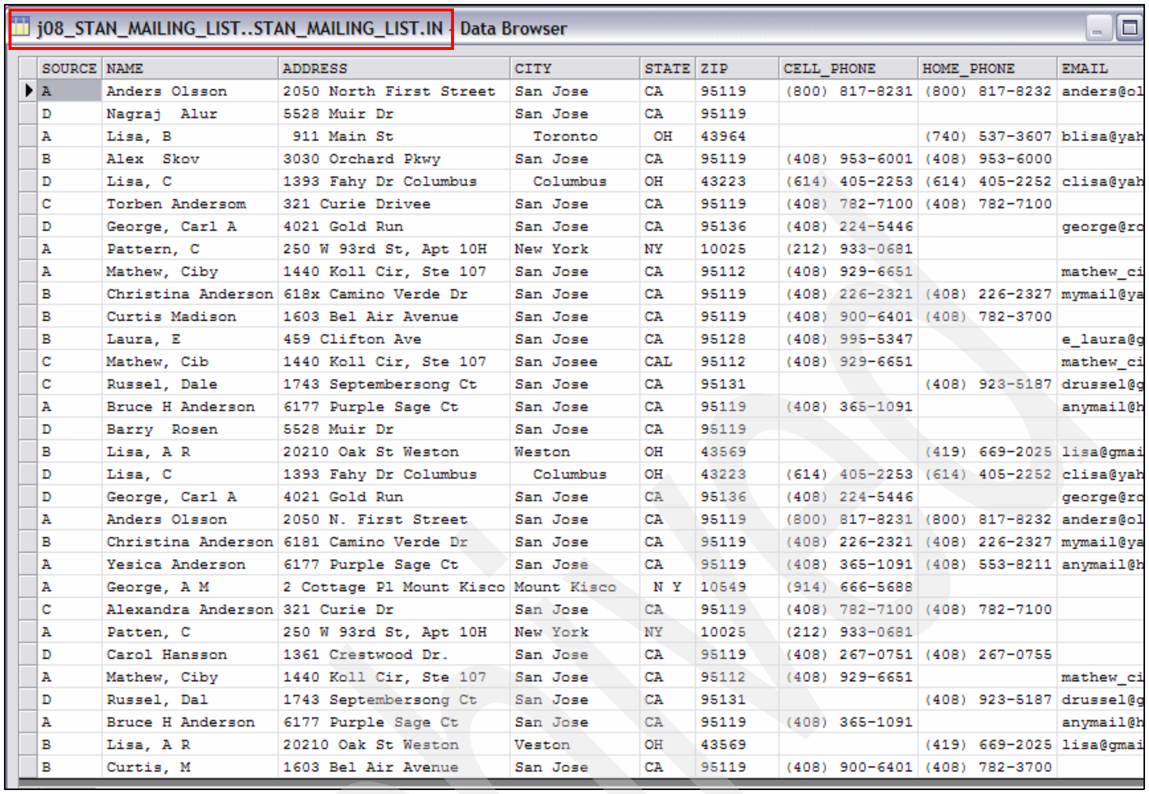
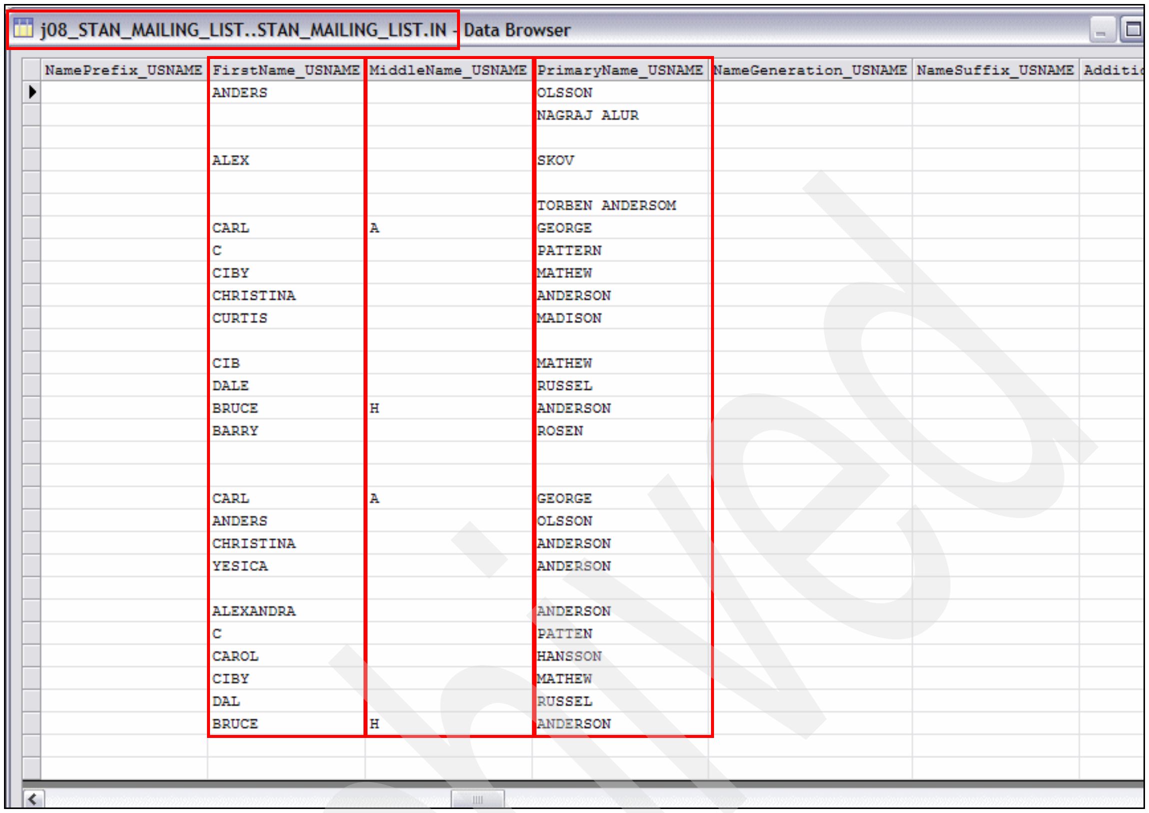


Figure 1-473   Create J08_STAN_MAILING_LIST_Domain_Specific 7/12


j08_STAN_MAILING_LIST..STAN_MAILING_LIST.IN - Data Browser

SOURCE	NAME	ADDRESS	CITY	STATE	ZIP	CELL_PHONE	HOME_PHONE	EMAIL
A	Anders Olsson	2050 North First Street	San Jose	CA	95119	(800) 817-8231	(800) 817-8232	anders@ol
D	Nagraj Alur	5528 Muir Dr	San Jose	CA	95119			
A	Lisa, B	911 Main St	Toronto	OH	43964		(740) 537-3607	blisa@yah
B	Alex Skov	3030 Orchard Pkwy	San Jose	CA	95119	(408) 953-6001	(408) 953-6000	
D	Lisa, C	1393 Fahy Dr Columbus	Columbus	OH	43223	(614) 405-2253	(614) 405-2252	clisa@yah
C	Torben Andersom	321 Curie Drivee	San Jose	CA	95119	(408) 782-7100	(408) 782-7100	
D	George, Carl A	4021 Gold Run	San Jose	CA	95136	(408) 224-5446		george@rc
A	Pattern, C	250 W 93rd St, Apt 10H	New York	NY	10025	(212) 933-0681		
A	Mathew, Ciby	1440 Koll Cir, Ste 107	San Jose	CA	95112	(408) 929-6651		mathew_ci
B	Christina Anderson	618x Camino Verde Dr	San Jose	CA	95119	(408) 226-2321	(408) 226-2327	mymail@ya
B	Curtis Madison	1603 Bel Air Avenue	San Jose	CA	95119	(408) 900-6401	(408) 782-3700	
B	Laura, E	459 Clifton Ave	San Jose	CA	95128	(408) 995-5347		e_laura@g
C	Mathew, Cib	1440 Koll Cir, Ste 107	San Josee	CAL	95112	(408) 929-6651		mathew_ci
C	Russel, Dale	1743 Septembersong Ct	San Jose	CA	95131		(408) 923-5187	drussel@g
A	Bruce H Anderson	6177 Purple Sage Ct	San Jose	CA	95119	(408) 365-1091		anymail@h
D	Barry Rosen	5528 Muir Dr	San Jose	CA	95119			
B	Lisa, A R	20210 Oak St Weston	Weston	OH	43569		(419) 669-2025	lisa@gmai
D	Lisa, C	1393 Fahy Dr Columbus	Columbus	OH	43223	(614) 405-2253	(614) 405-2252	clisa@yah
D	George, Carl A	4021 Gold Run	San Jose	CA	95136	(408) 224-5446		george@rc
A	Anders Olsson	2050 N. First Street	San Jose	CA	95119	(800) 817-8231	(800) 817-8232	anders@ol
B	Christina Anderson	6181 Camino Verde Dr	San Jose	CA	95119	(408) 226-2321	(408) 226-2327	mymail@ya
A	Yesica Anderson	6177 Purple Sage Ct	San Jose	CA	95119	(408) 365-1091	(408) 553-8211	anymail@h
A	George, A M	2 Cottage Pl Mount Kisco	Mount Kisco	N Y	10549	(914) 666-5688		
C	Alexandra Anderson	321 Curie Dr	San Jose	CA	95119	(408) 782-7100	(408) 782-7100	
A	Patten, C	250 W 93rd St, Apt 10H	New York	NY	10025	(212) 933-0681		
D	Carol Hansson	1361 Crestwood Dr.	San Jose	CA	95119	(408) 267-0751	(408) 267-0755	
A	Mathew, Ciby	1440 Koll Cir, Ste 107	San Jose	CA	95112	(408) 929-6651		mathew_ci
D	Russel, Dal	1743 Septembersong Ct	San Jose	CA	95131		(408) 923-5187	drussel@g
A	Bruce H Anderson	6177 Purple Sage Ct	San Jose	CA	95119	(408) 365-1091		anymail@h
B	Lisa, A R	20210 Oak St Weston	Veston	OH	43569		(419) 669-2025	lisa@gmai
B	Curtis, M	1603 Bel Air Avenue	San Jose	CA	95119	(408) 900-6401	(408) 782-3700	

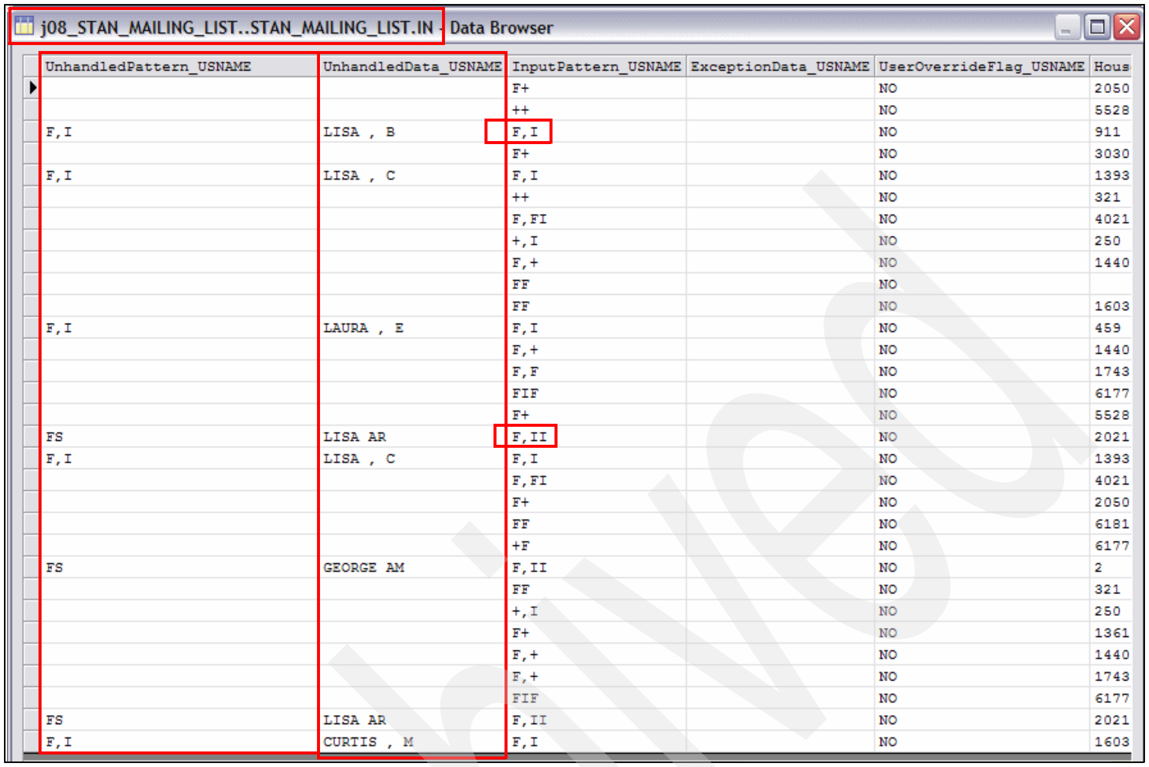
Figure 1-474 Create J08_STAN_MAILING_LIST_Domain_Specific 8/12



j08_STAN_MAILING_LIST..STAN_MAILING_LIST.IN - Data Browser

NamePrefix_USNAME	FirstName_USNAME	MiddleName_USNAME	PrimaryName_USNAME	NameGeneration_USNAME	NameSuffix_USNAME	Additi
	ANDERS		OLSSON			
			NAGRAJ ALUR			
	ALEX		SKOV			
			TORBEN ANDERSOM			
	CARL	A	GEORGE			
	C		PATTERN			
	CIBY		MATHEW			
	CHRISTINA		ANDERSON			
	CURTIS		MADISON			
	CIB		MATHEW			
	DALE		RUSSEL			
	BRUCE	H	ANDERSON			
	BARRY		ROSEN			
	CARL	A	GEORGE			
	ANDERS		OLSSON			
	CHRISTINA		ANDERSON			
	YESICA		ANDERSON			
	ALEXANDRA		ANDERSON			
	C		PATTEN			
	CAROL		HANSSON			
	CIBY		MATHEW			
	DAL		RUSSEL			
	BRUCE	H	ANDERSON			

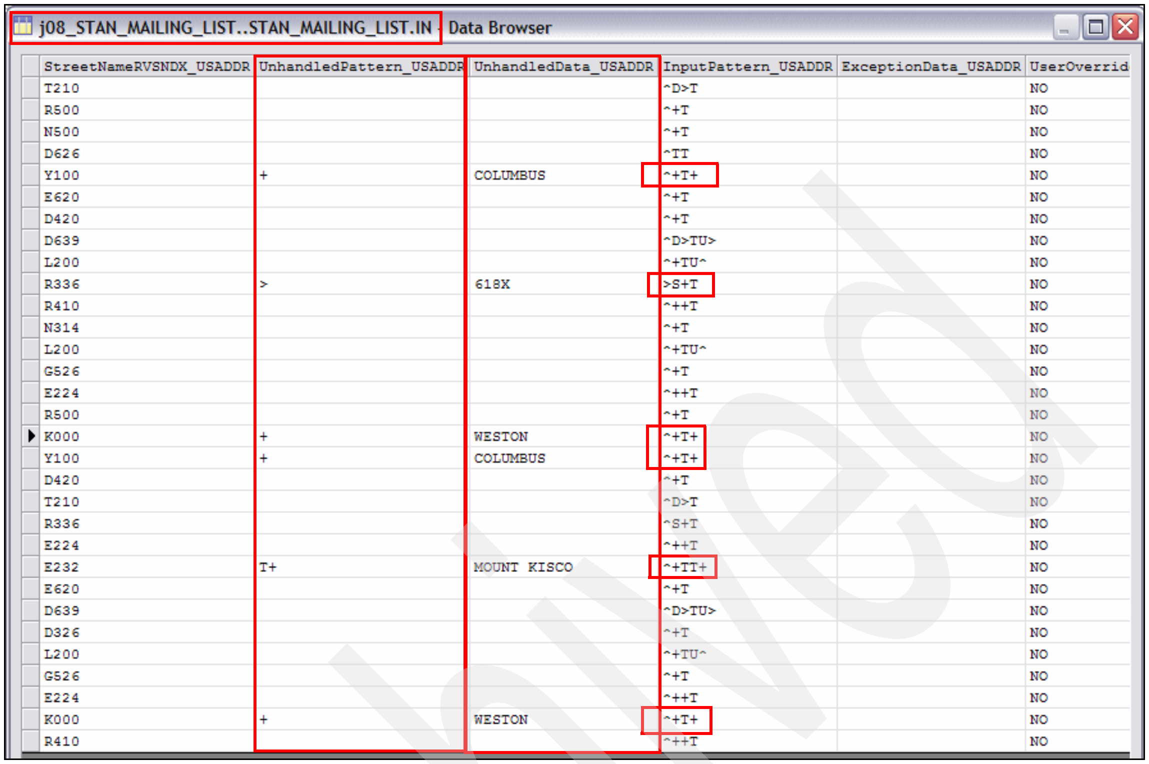
Figure 1-475 Create J08_STAN_MAILING_LIST_Domain_Specific 9/12



j08_STAN_MAILING_LIST..STAN_MAILING_LIST.IN - Data Browser

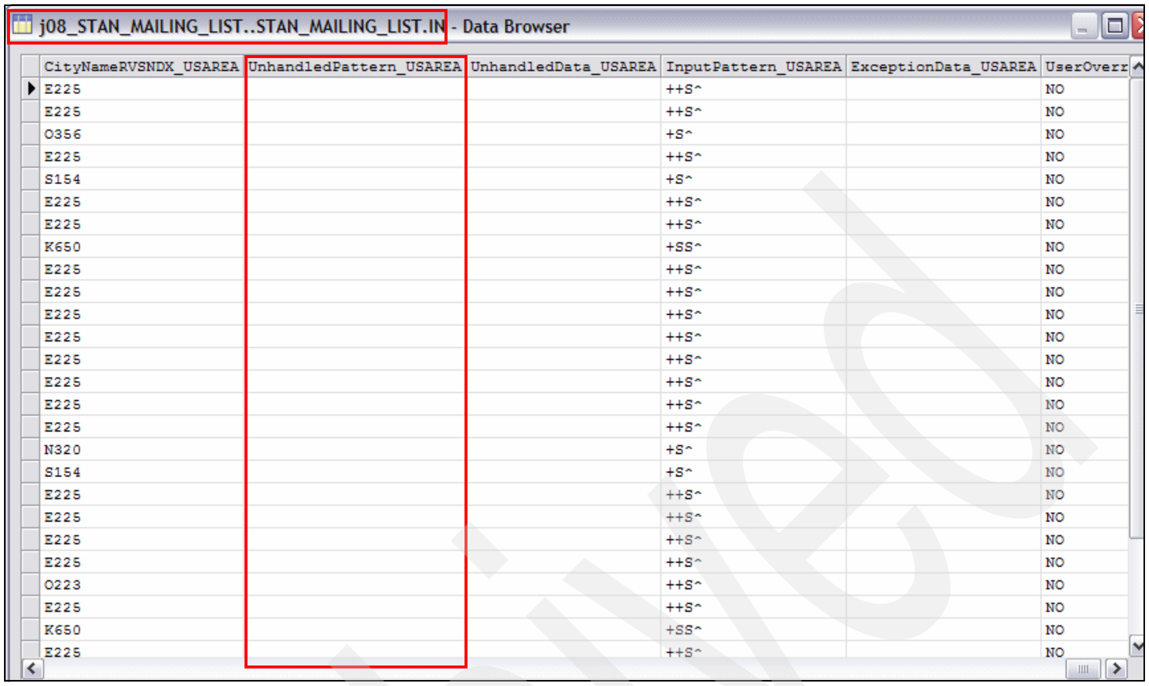
UnhandledPattern_USNAME	UnhandledData_USNAME	InputPattern_USNAME	ExceptionData_USNAME	UserOverrideFlag_USNAME	Hous
		F+		NO	2050
		++		NO	5528
F,I	LISA , B	F,I		NO	911
		F+		NO	3030
F,I	LISA , C	F,I		NO	1393
		++		NO	321
		F,FI		NO	4021
		+,I		NO	250
		F,+		NO	1440
		FF		NO	
		FF		NO	1603
F,I	LAURA , E	F,I		NO	459
		F,+		NO	1440
		F,F		NO	1743
		FIF		NO	6177
		F+		NO	5528
FS	LISA AR	F,II		NO	2021
F,I	LISA , C	F,I		NO	1393
		F,FI		NO	4021
		F+		NO	2050
		FF		NO	6181
		+F		NO	6177
FS	GEORGE AM	F,II		NO	2
		FF		NO	321
		+,I		NO	250
		F+		NO	1361
		F,+		NO	1440
		F,+		NO	1743
		FIF		NO	6177
FS	LISA AR	F,II		NO	2021
F,I	CURTIS , M	F,I		NO	1603

Figure 1-476 Create J08_STAN_MAILING_LIST_Domain_Specific 10/12


j08_STAN_MAILING_LIST..STAN_MAILING_LIST.IN - Data Browser

StreetNameRVSNDX_USADDR	UnhandledPattern_USADDR	UnhandledData_USADDR	InputPattern_USADDR	ExceptionData_USADDR	UserOverrid
T210			^D>T		NO
R500			^+T		NO
N500			^+T		NO
D626			^TT		NO
Y100	+	COLUMBUS	^+T+		NO
E620			^+T		NO
D420			^+T		NO
D639			^D>TU>		NO
L200			^+TU^		NO
R336	>	618X	>S+T		NO
R410			^++T		NO
N314			^+T		NO
L200			^+TU^		NO
G526			^+T		NO
E224			^++T		NO
R500			^+T		NO
K000	+	WESTON	^+T+		NO
Y100	+	COLUMBUS	^+T+		NO
D420			^+T		NO
T210			^D>T		NO
R336			^S+T		NO
E224			^++T		NO
E232	T+	MOUNT KISCO	^+TT+		NO
E620			^+T		NO
D639			^D>TU>		NO
D326			^+T		NO
L200			^+TU^		NO
G526			^+T		NO
E224			^++T		NO
K000	+	WESTON	^+T+		NO
R410			^++T		NO

Figure 1-477 Create J08_STAN_MAILING_LIST_Domain_Specific 11/12


j08_STAN_MAILING_LIST..STAN_MAILING_LIST.IN - Data Browser

CityNameRVSNDX_USAREA	UnhandledPattern_USAREA	UnhandledData_USAREA	InputPattern_USAREA	ExceptionData_USAREA	UserOverr
E225			++S^		NO
E225			++S^		NO
O356			+S^		NO
E225			++S^		NO
S154			+S^		NO
E225			++S^		NO
E225			++S^		NO
K650			+SS^		NO
E225			++S^		NO
E225			++S^		NO
E225			++S^		NO
E225			++S^		NO
E225			++S^		NO
E225			++S^		NO
E225			++S^		NO
E225			++S^		NO
N320			+S^		NO
S154			+S^		NO
E225			++S^		NO
E225			++S^		NO
E225			++S^		NO
E225			++S^		NO
O223			++S^		NO
E225			++S^		NO
K650			+SS^		NO
E225			++S^		NO

Figure 1-478 Create J08_STAN_MAILING_LIST_Domain_Specific 12/12

J09_INVCC_STAN_MAILING_LIST

In this step, we investigate unhandled patterns in "J08_STAN_MAILING_LIST_Domain_Specific" on page 390. We run the Investigate stage with character concatenate using the *C* mask on the unhandled pattern columns of "J08_STAN_MAILING_LIST_Domain_Specific" on page 390. The columns that are investigated correspond to the name, address, and area domains.

Because a single Investigate stage can only have a single rule set associated with it, we split the output data set of the "J08_STAN_MAILING_LIST_Domain_Specific" on page 390 job (using a Copy stage) and process it by three independent Investigate stages, with each stage using a particular domain-specific rule set. The column frequency report is generated in each Investigate stage.

Figure 1-479 on page 401 shows the various stages that are used in this job, including the data set that was created in "J08_STAN_MAILING_LIST_Domain_Specific" on page 390, a Copy stage, three Investigate stages that each use a different domain-specific rule set, and two data set stages (one each for the token report and pattern report) for each

Investigate stage. We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object created earlier.

Because the configuration of this job is very similar to that described in "J09_INVCC_STAN_CUSTOMER" on page 253, we do not repeat it here. However, some of the configurations of interest are as follows:

- Figure 1-480 on page 401 shows the INV01 - Investigate Stage window with the USNAME rule set and the columns selected for Character Concatenate Investigate as follows:
  - UnhandledPattern_USNAME with *C* mask
  - InputPattern_USNAME with *C* mask
  - UnhandledData_USNAME with *X* mask
  - NameDomain_USPREP with *X* mask

  Click **Advanced Options** in the INV01 Investigate Stage window to set the number of samples that display. (We do not show this process here. The default is 1.)
- Repeat the process for the INV02 Investigate stage with the USADDR rule set (Figure 1-481 on page 402) and INV03 Investigate stage with the USAREA rule set (Figure 1-482 on page 402) with the appropriate columns as shown.
- After saving, compiling, and running this job (Figure 1-483 on page 403), the contents of the outputs of the Investigate stages that are written to the sequential file are shown in Figure 1-484 on page 404 through Figure 1-490 on page 408.
  - The output of the INV01 Investigate stage in Figure 1-484 on page 404 and Figure 1-486 on page 406 shows a total of two types of unhandled patterns in the name domain—five occurrences of the `F,I` pattern (corresponding to `F,I` in the input pattern) and three occurrences of the `FS` pattern (corresponding to `F,II` in the input pattern).
  - The output of the INV02 Investigate stage in Figure 1-487 on page 407 and Figure 1-489 on page 408 show a total of three types of unhandled patterns for the address domain—four occurrences of the + pattern (corresponding to `^+T+` in the input pattern), one occurrence of the `>` pattern (corresponding to `^S+T` in the input pattern), and one occurrence of the `T+` pattern (corresponding to `^+TT+` in the input pattern).
  - The output of the INV03 Investigate stage in Figure 1-490 on page 408 shows that there were no unhandled patterns in the area domain.

You need to manage the unhandled patterns in the address domain as described in the step "J09_Z_Override_And_After" on page 408.

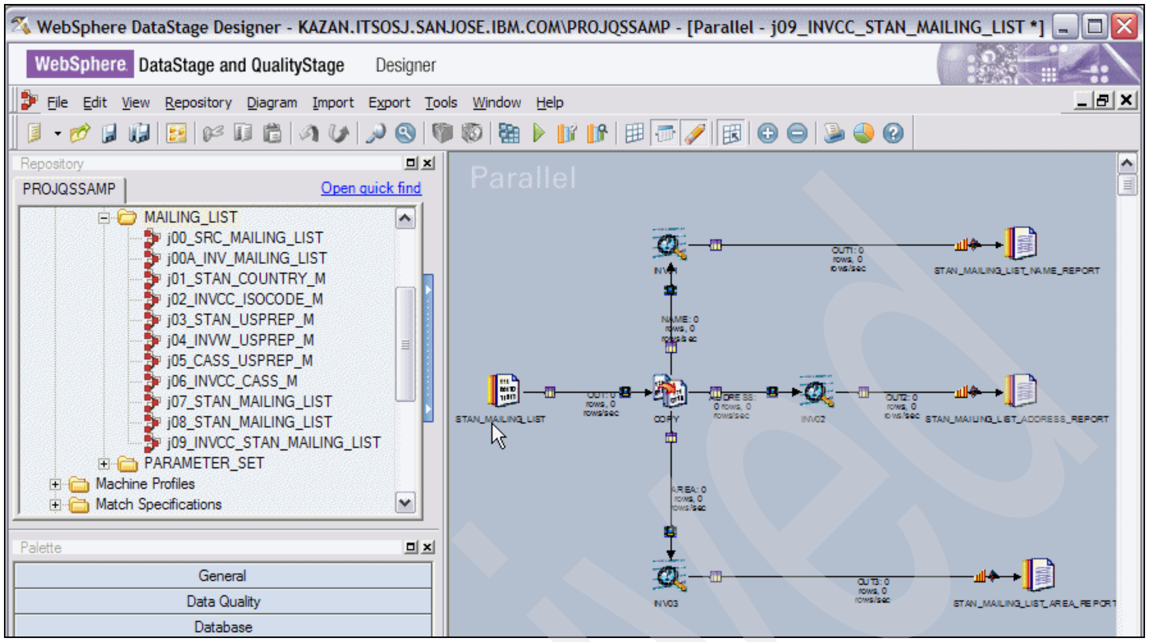
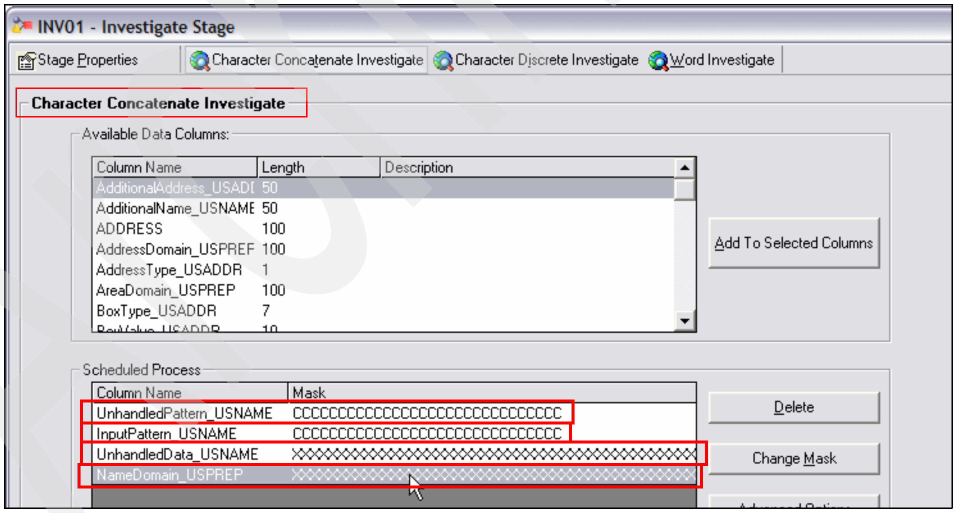


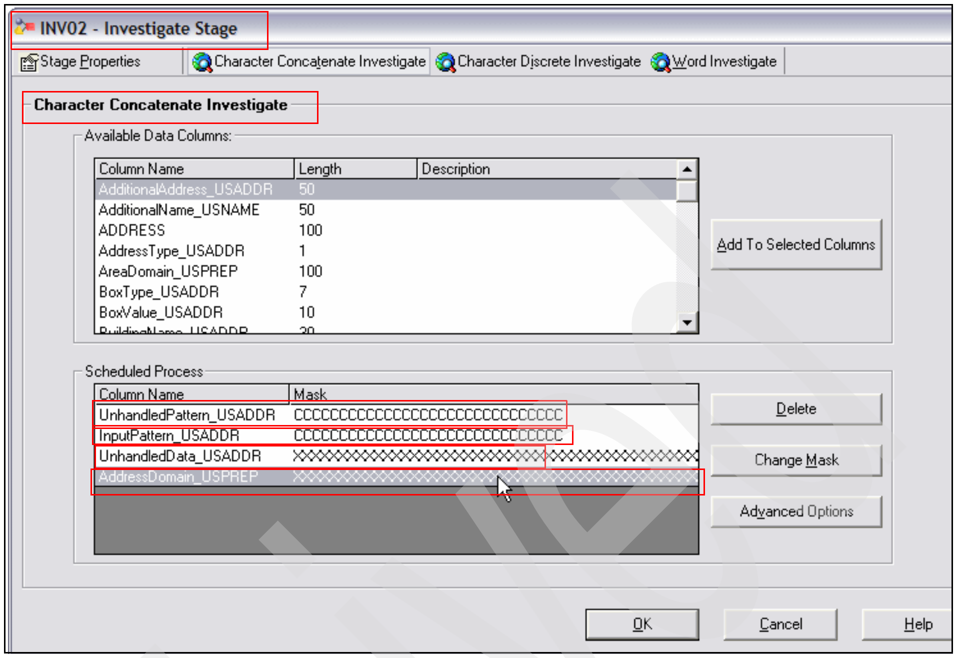
Figure 1-479 Create J09_INVCC_STAN_MAILING_LIST job 1/12


INV01 - Investigate Stage
Stage Properties
Character Concatenate Investigate
Character Discrete Investigate
Word Investigate
Character Concatenate Investigate
Available Data Columns:
Add To Selected Columns
Scheduled Process
Delete
Change Mask

Column Name	Length	Description
AdditionalAddress_USADI	50	
AdditionalName_USNAME	50	
ADDRESS	100	
AddressDomain_USPREP	100	
AddressType_USADDR	1	
AreaDomain_USPREP	100	
BoxType_USADDR	7	

Column Name	Mask
UnhandledPattern_USNAME	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
InputPattern_USNAME	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
UnhandledData_USNAME	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
NameDomain_USPREP	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Figure 1-480 Create J09_INVCC_STAN_MAILING_LIST job 2/12


INV02 - Investigate Stage

Stage Properties | Character Concatenate Investigate | Character Discrete Investigate | Word Investigate

Character Concatenate Investigate

Available Data Columns:

Column Name	Length	Description
AdditionalAddress_USADDR	50	
AdditionalName_USNAME	50	
ADDRESS	100	
AddressType_USADDR	1	
AreaDomain_USPREP	100	
BoxType_USADDR	7	
BoxValue_USADDR	10	
BuildingName_USADDR	30	

Add To Selected Columns

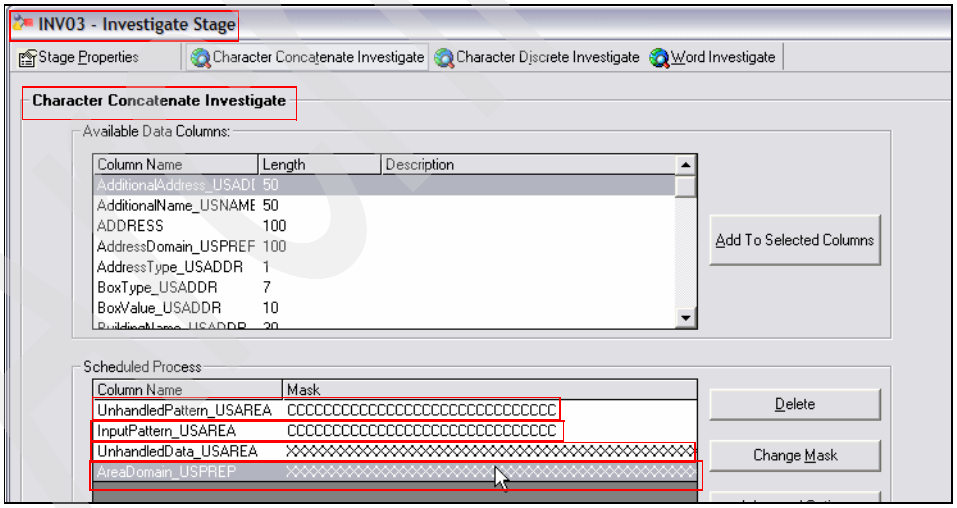
Scheduled Process

Column Name	Mask
UnhandledPattern_USADDR	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
InputPattern_USADDR	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
UnhandledData_USADDR	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
AddressDomain_USPREP	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Delete | Change Mask | Advanced Options

OK | Cancel | Help

Figure 1-481 Create J09_INVCC_STAN_MAILING_LIST job 3/12


INV03 - Investigate Stage

Stage Properties | Character Concatenate Investigate | Character Discrete Investigate | Word Investigate

Character Concatenate Investigate

Available Data Columns:

Column Name	Length	Description
AdditionalAddress_USADI	50	
AdditionalName_USNAME	50	
ADDRESS	100	
AddressDomain_USPREF	100	
AddressType_USADDR	1	
BoxType_USADDR	7	
BoxValue_USADDR	10	
BuildingName_USADDR	30	

Add To Selected Columns

Scheduled Process

Column Name	Mask
UnhandledPattern_USAREA	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
InputPattern_USAREA	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
UnhandledData_USAREA	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
AreaDomain_USPREP	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Delete | Change Mask

Figure 1-482 Create J09_INVCC_STAN_MAILING_LIST job 4/12

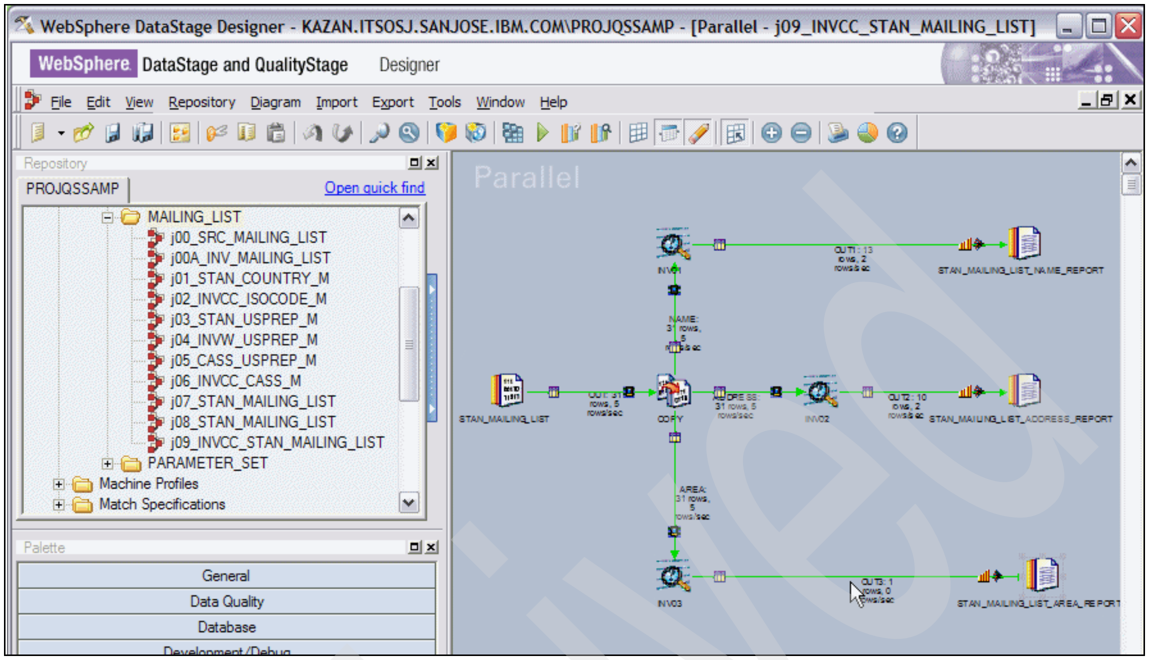
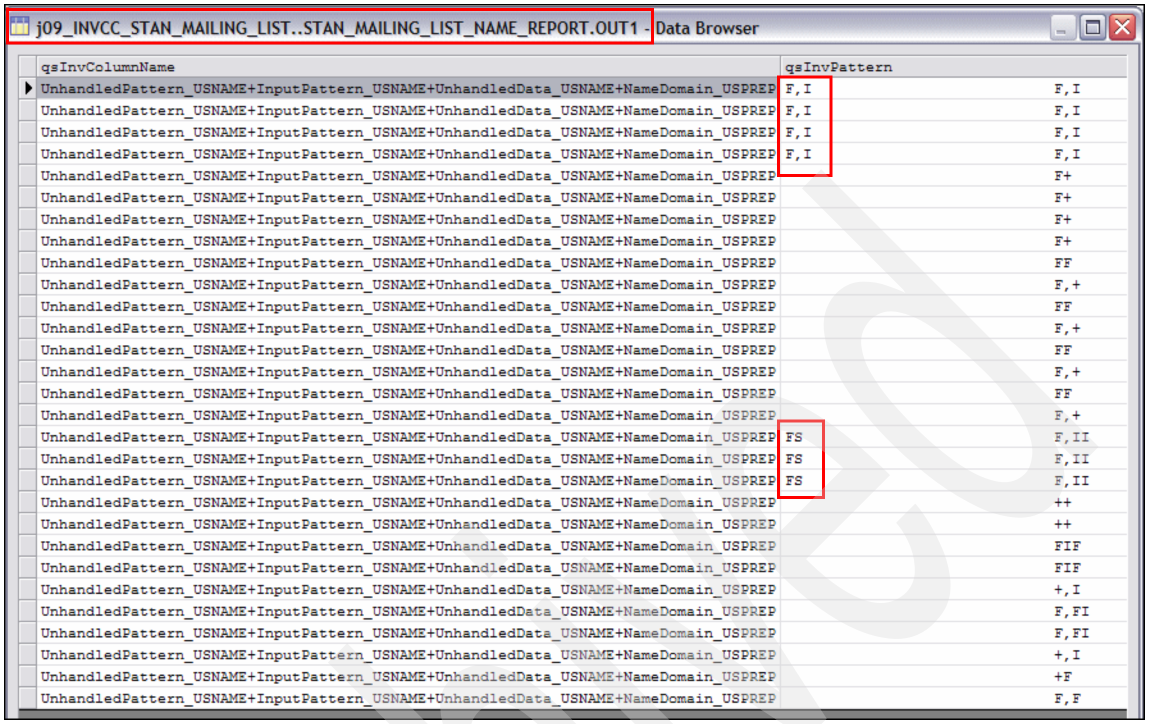


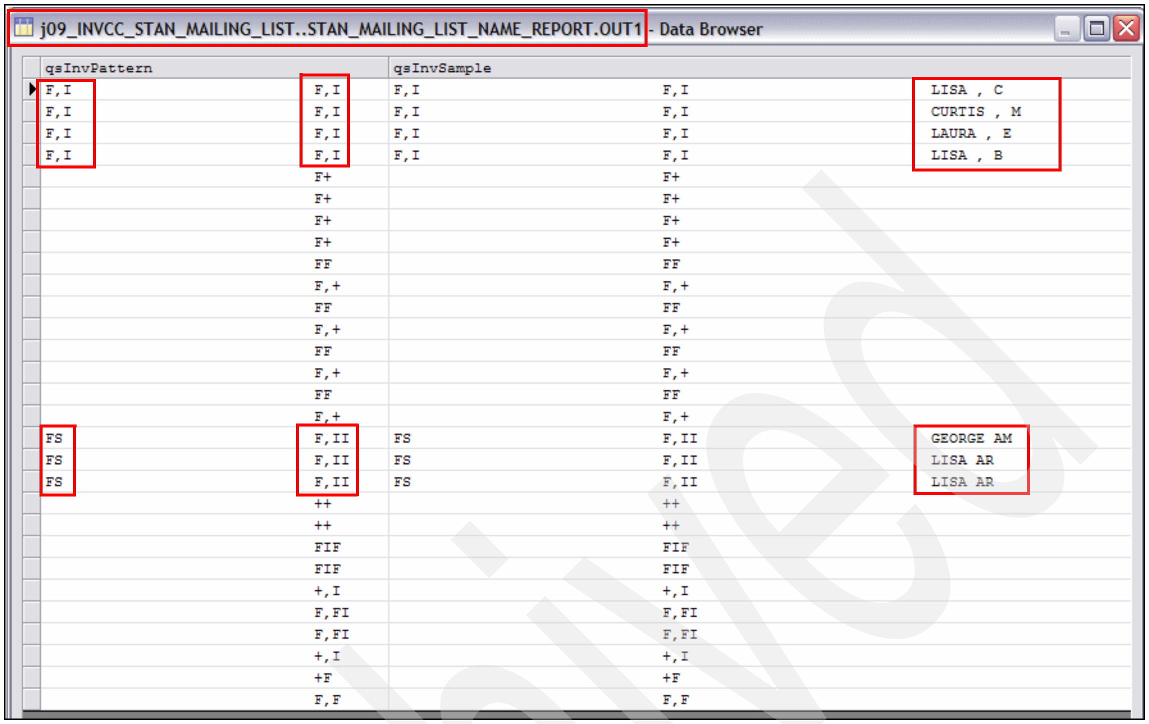
Figure 1-483 Create J09_INVCC_STAN_MAILING_LIST job 5/12



j09_INVCC_STAN_MAILING_LIST..STAN_MAILING_LIST_NAME_REPORT.OUT1 - Data Browser

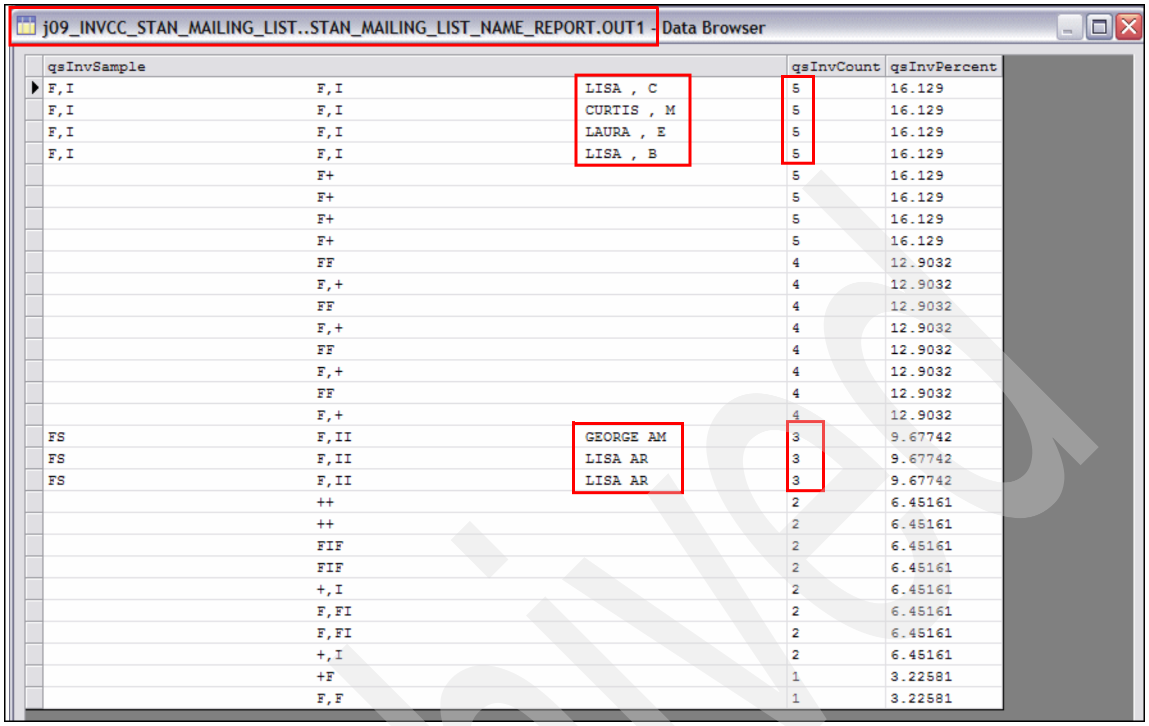
qsInvColumnName	qsInvPattern	
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP	F,I	F,I
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP	F,I	F,I
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP	F,I	F,I
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP	F,I	F,I
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP		F+
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP		F+
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP		F+
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP		F+
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP		FF
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP		F,+
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP		FF
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP		F,+
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP		FF
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP		F,+
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP		FF
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP		F,+
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP	FS	F,II
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP	FS	F,II
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP	FS	F,II
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP		++
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP		++
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP		FIF
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP		FIF
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP		+,I
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP		F,FI
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP		F,FI
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP		+,I
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP		+F
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+NameDomain_USPREP		F,F

Figure 1-484 Create J09_INVCC_STAN_MAILING_LIST job 6/12


j09_INVCC_STAN_MAILING_LIST..STAN_MAILING_LIST_NAME_REPORT.OUT1 - Data Browser

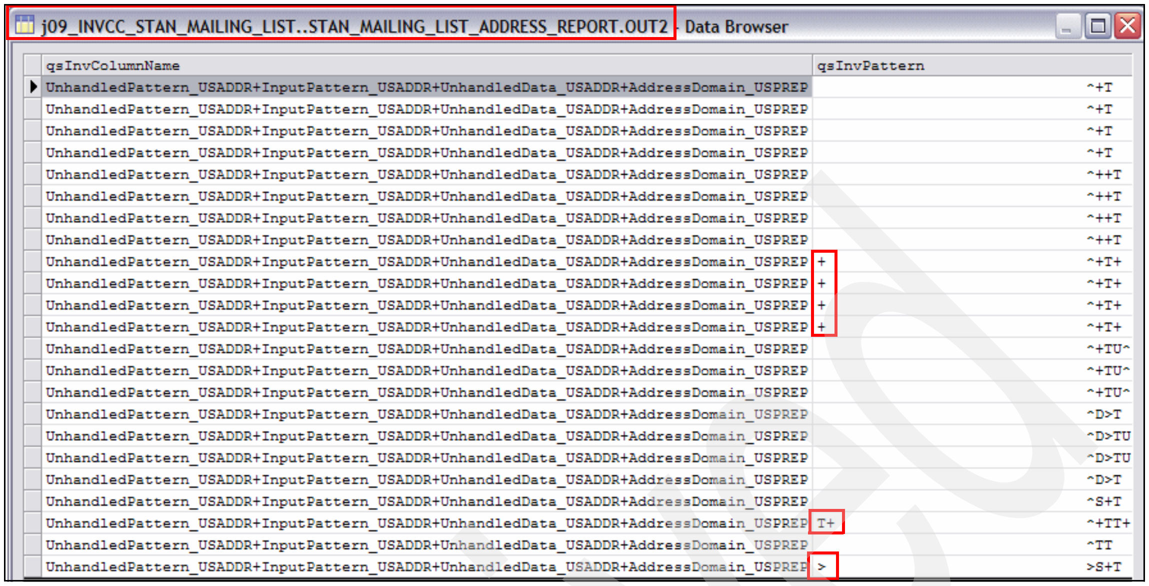
qsInvPattern		qsInvSample		
F,I	F,I	F,I	F,I	LISA , C
F,I	F,I	F,I	F,I	CURTIS , M
F,I	F,I	F,I	F,I	LAURA , E
F,I	F,I	F,I	F,I	LISA , B
	F+		F+	
	F+		F+	
	F+		F+	
	F+		F+	
	FF		FF	
	F,+		F,+	
	FF		FF	
	F,+		F,+	
	FF		FF	
	F,+		F,+	
	FF		FF	
	F,+		F,+	
FS	F,II	FS	F,II	GEORGE AM
FS	F,II	FS	F,II	LISA AR
FS	F,II	FS	F,II	LISA AR
	++		++	
	++		++	
	FIF		FIF	
	FIF		FIF	
	+,I		+,I	
	F,FI		F,FI	
	F,FI		F,FI	
	+,I		+,I	
	+F		+F	
	F,F		F,F	

Figure 1-485 Create J09_INVCC_STAN_MAILING_LIST job 7/12


j09_INVCC_STAN_MAILING_LIST..STAN_MAILING_LIST_NAME_REPORT.OUT1 - Data Browser

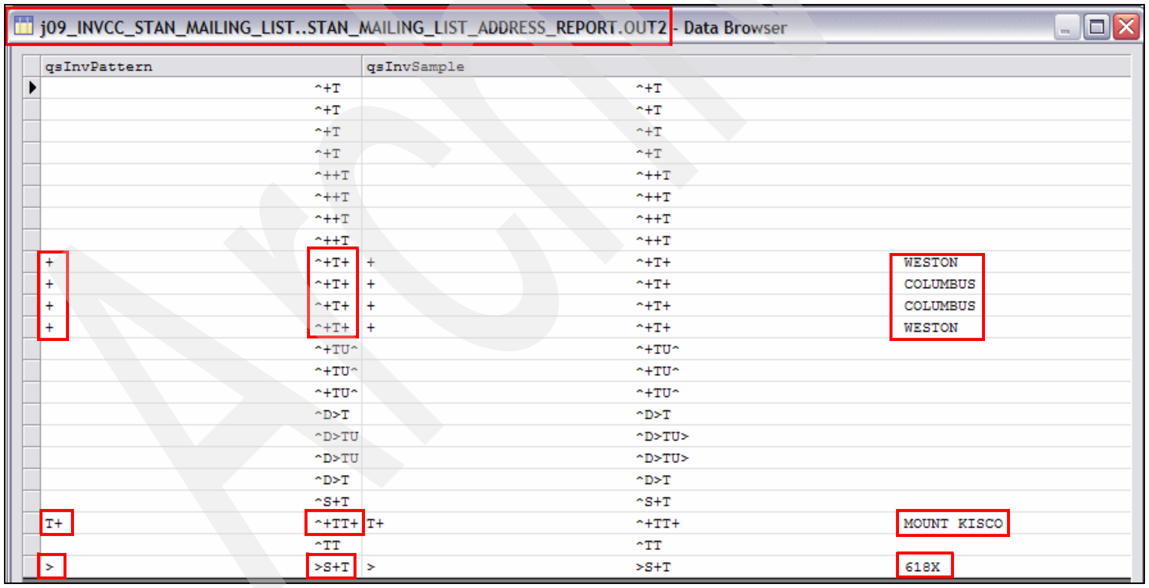
qsInvSample			qsInvCount	qsInvPercent
F,I	F,I	LISA , C	5	16.129
F,I	F,I	CURTIS , M	5	16.129
F,I	F,I	LAURA , E	5	16.129
F,I	F,I	LISA , B	5	16.129
	F+		5	16.129
	F+		5	16.129
	F+		5	16.129
	F+		5	16.129
	FF		4	12.9032
	F,+		4	12.9032
	FF		4	12.9032
	F,+		4	12.9032
	FF		4	12.9032
	F,+		4	12.9032
	FF		4	12.9032
	F,+		4	12.9032
FS	F,II	GEORGE AM	3	9.67742
FS	F,II	LISA AR	3	9.67742
FS	F,II	LISA AR	3	9.67742
	++		2	6.45161
	++		2	6.45161
	FIF		2	6.45161
	FIF		2	6.45161
	+,I		2	6.45161
	F,FI		2	6.45161
	F,FI		2	6.45161
	+,I		2	6.45161
	+F		1	3.22581
	F,F		1	3.22581

Figure 1-486 Create J09_INVCC_STAN_MAILING_LIST job 8/12


j09_INVCC_STAN_MAILING_LIST..STAN_MAILING_LIST_ADDRESS_REPORT.OUT2 - Data Browser

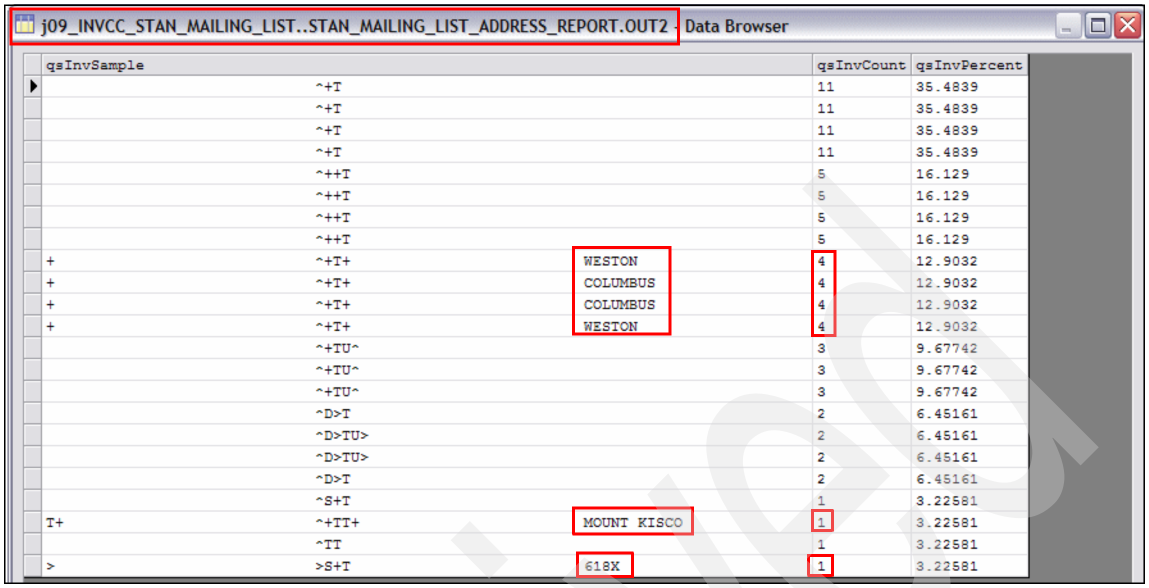
qsInvColumnName	qsInvPattern
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	^+T
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	^+T
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	^+T
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	^+T
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	^++T
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	^++T
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	^++T
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	^++T
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	+ ^+T+
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	+ ^+T+
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	+ ^+T+
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	+ ^+T+
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	^+TU^
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	^+TU^
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	^+TU^
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	^D>T
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	^D>TU
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	^D>TU
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	^D>T
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	^S+T
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	T+ ^+TT+
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	^TT
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+AddressDomain_USPREP	> >S+T

Figure 1-487 Create J09_INVCC_STAN_MAILING_LIST job 9/12


j09_INVCC_STAN_MAILING_LIST..STAN_MAILING_LIST_ADDRESS_REPORT.OUT2 - Data Browser

qsInvPattern	qsInvSample
^+T	^+T
^+T	^+T
^+T	^+T
^+T	^+T
^++T	^++T
^++T	^++T
^++T	^++T
^++T	^++T
+ ^+T+	+ ^+T+ WESTON
+ ^+T+	+ ^+T+ COLUMBUS
+ ^+T+	+ ^+T+ COLUMBUS
+ ^+T+	+ ^+T+ WESTON
^+TU^	^+TU^
^+TU^	^+TU^
^+TU^	^+TU^
^D>T	^D>T
^D>TU	^D>TU>
^D>TU	^D>TU>
^D>T	^D>T
^S+T	^S+T
T+ ^+TT+	T+ ^+TT+ MOUNT KISCO
^TT	^TT
> >S+T	> >S+T 618X

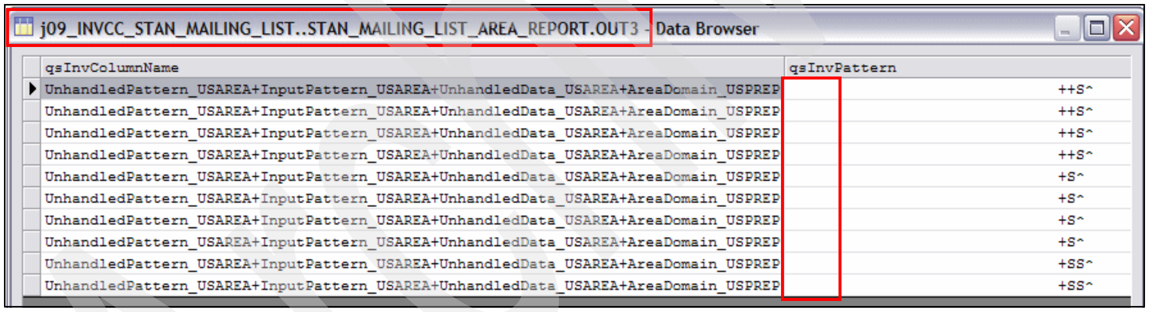
Figure 1-488 Create J09_INVCC_STAN_MAILING_LIST job 10/12



j09_INVCC_STAN_MAILING_LIST..STAN_MAILING_LIST_ADDRESS_REPORT.OUT2 - Data Browser

qsInvSample	qsInvCount	qsInvPercent
^+T	11	35.4839
^+T	11	35.4839
^+T	11	35.4839
^+T	11	35.4839
^++T	5	16.129
^++T	5	16.129
^++T	5	16.129
^++T	5	16.129
+ ^+T+ WESTON	4	12.9032
+ ^+T+ COLUMBUS	4	12.9032
+ ^+T+ COLUMBUS	4	12.9032
+ ^+T+ WESTON	4	12.9032
^+TU^	3	9.67742
^+TU^	3	9.67742
^+TU^	3	9.67742
^D>T	2	6.45161
^D>TU>	2	6.45161
^D>TU>	2	6.45161
^D>T	2	6.45161
^S+T	1	3.22581
T+ ^+TT+ MOUNT KISCO	1	3.22581
^TT	1	3.22581
> >S+T 618X	1	3.22581

Figure 1-489 Create J09_INVCC_STAN_MAILING_LIST job 11/12



j09_INVCC_STAN_MAILING_LIST..STAN_MAILING_LIST_AREA_REPORT.OUT3 - Data Browser

qsInvColumnName	qsInvPattern
UnhandledPattern_USAREA+InputPattern_USAREA+UnhandledData_USAREA+AreaDomain_USPREP	++S^
UnhandledPattern_USAREA+InputPattern_USAREA+UnhandledData_USAREA+AreaDomain_USPREP	++S^
UnhandledPattern_USAREA+InputPattern_USAREA+UnhandledData_USAREA+AreaDomain_USPREP	++S^
UnhandledPattern_USAREA+InputPattern_USAREA+UnhandledData_USAREA+AreaDomain_USPREP	++S^
UnhandledPattern_USAREA+InputPattern_USAREA+UnhandledData_USAREA+AreaDomain_USPREP	+S^
UnhandledPattern_USAREA+InputPattern_USAREA+UnhandledData_USAREA+AreaDomain_USPREP	+S^
UnhandledPattern_USAREA+InputPattern_USAREA+UnhandledData_USAREA+AreaDomain_USPREP	+S^
UnhandledPattern_USAREA+InputPattern_USAREA+UnhandledData_USAREA+AreaDomain_USPREP	+S^
UnhandledPattern_USAREA+InputPattern_USAREA+UnhandledData_USAREA+AreaDomain_USPREP	+SS^
UnhandledPattern_USAREA+InputPattern_USAREA+UnhandledData_USAREA+AreaDomain_USPREP	+SS^

Figure 1-490 Create J09_INVCC_STAN_MAILING_LIST job 12/12

J09_Z_Override_And_After

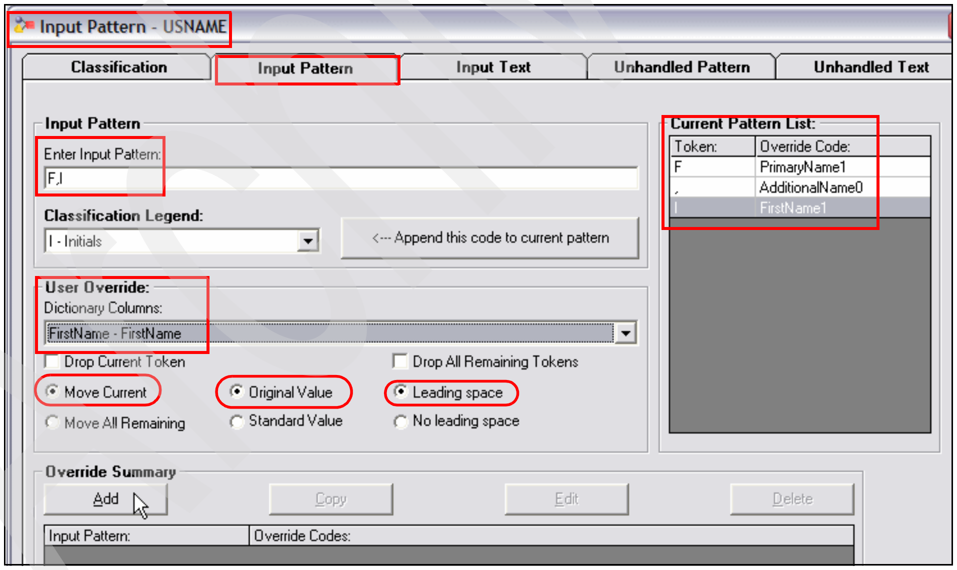
Because we find unhandled patterns (such as `F,I`) in the name domain, we create pattern overrides in the domain-specific USNAME rule set. We also find unhandled patterns in the address domain and create overrides in the domain-specific USADDR rule set.

Because the configuration of overrides is described in "J09_Z_Override_And_After" on page 262, we do not repeat it here except for the following:

- Figure 1-491 that shows the input pattern `F,I` overridden for the USNAME rule set. It shows the Override Code for the `F` token as PrimaryName, the "`,`" token as AdditionalName, and the `I` token as FirstName.
- Figure 1-492 on page 410 that shows the Override Summary input patterns `F,I` and `F,II` with the appropriate Override Code for each token for the USNAME rule set.
- Figure 1-493 on page 410 that shows the Override Summary input patterns ++ (from the overrides done in "J09_Z_Override_And_After" on page 262 in Figure 1-290 on page 264), `>S+T`, `^+T+`, and `^+TT+` with the appropriate Override Code for each token for the USADDR rule set.

We then re-run the "J08_STAN_MAILING_LIST_Domain_Specific" on page 390 step and confirm that there are no unhandled patterns. We do not show this process here.

Now, proceed to "J10_MATCHFREQ_STAN_MAILING_LIST" on page 411.


Input Pattern - USNAME

Classification | Input Pattern | Input Text | Unhandled Pattern | Unhandled Text

Input Pattern
Enter Input Pattern: F,I
Classification Legend: I - Initials
<-- Append this code to current pattern

Current Pattern List:

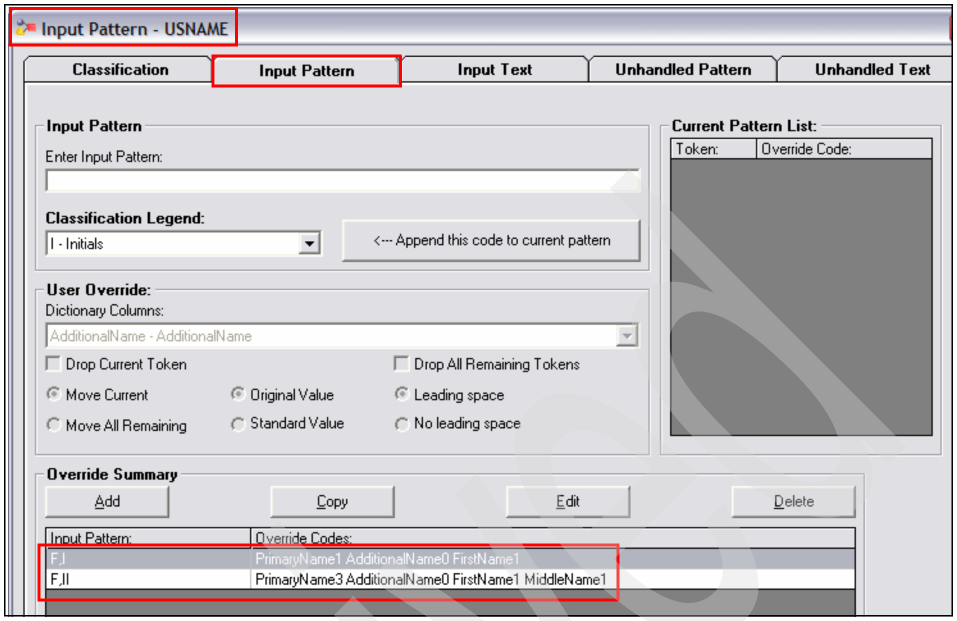
Token:	Override Code:
F	PrimaryName1
,	AdditionalName0
I	FirstName1

User Override:
Dictionary Columns: FirstName - FirstName
Drop Current Token
Drop All Remaining Tokens
[✓ Move Current] Move All Remaining
[✓ Original Value] Standard Value
[✓ Leading space] No leading space

Override Summary
Add | Copy | Edit | Delete

Input Pattern:	Override Codes:

Figure 1-491 Create J09_Z_After_Override 1/3


Input Pattern - USNAME

Input Pattern
Enter Input Pattern:

Classification Legend: I - Initials

User Override:
Dictionary Columns: AdditionalName - AdditionalName

Drop Current Token [ ] Drop All Remaining Tokens [ ]
[✓ Move Current] Move All Remaining
[✓ Original Value] Standard Value
[✓ Leading space] No leading space

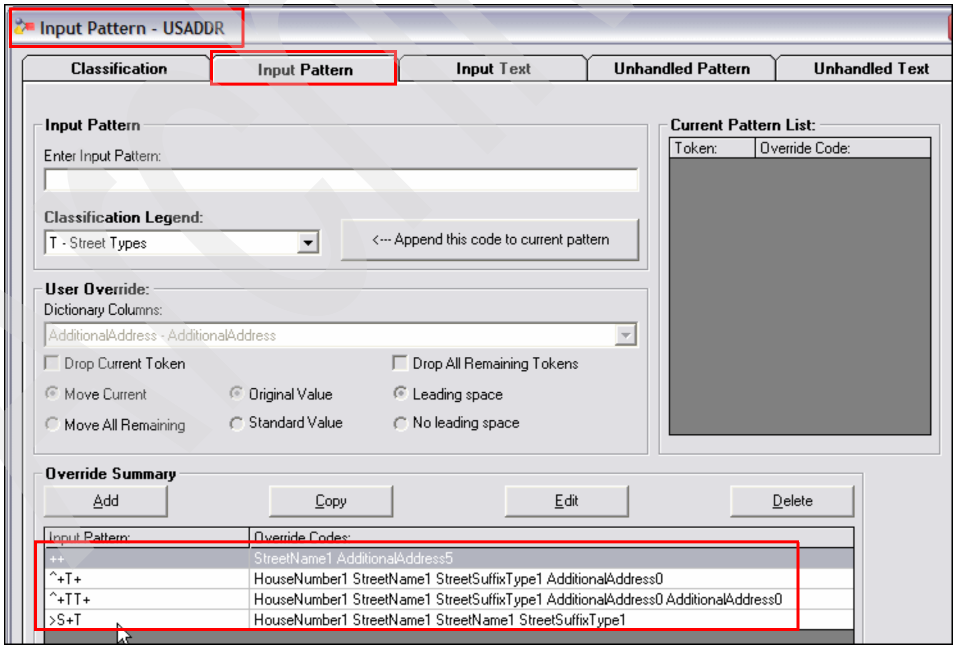
Current Pattern List:

Token:	Override Code:

Override Summary

Input Pattern:	Override Codes:
F,I	PrimaryName1 AdditionalName0 FirstName1
F,II	PrimaryName3 AdditionalName0 FirstName1 MiddleName1

Figure 1-492 Create J09_Z_After_Override 2/3


Input Pattern - USADDR

Input Pattern
Enter Input Pattern:

Classification Legend: T - Street Types

User Override:
Dictionary Columns: AdditionalAddress - AdditionalAddress

Drop Current Token [ ] Drop All Remaining Tokens [ ]
[✓ Move Current] Move All Remaining
[✓ Original Value] Standard Value
[✓ Leading space] No leading space

Current Pattern List:

Token:	Override Code:

Override Summary

Input Pattern:	Override Codes:
++	StreetName1 AdditionalAddress5
^+T+	HouseNumber1 StreetName1 StreetSuffixType1 AdditionalAddress0
^+TT+	HouseNumber1 StreetName1 StreetSuffixType1 AdditionalAddress0 AdditionalAddress0
>S+T	HouseNumber1 StreetName1 StreetName1 StreetSuffixType1

Figure 1-493 Create J09_Z_After_Override 3/3

J10_MATCHFREQ_STAN_MAILING_LIST

After the unhandled patterns are handled, we generate frequency distribution on *all* the columns in the mailing list file using the Match Frequency stage. The idea is to generate match frequency for all the columns so that it can be used with any match specification.

Figure 1-494 shows the various stages that are used in this job, including the data set that was created in “J08_STAN_MAILING_LIST_Domain_Specific” on page 390, a Match Frequency stage, and a Sequential File stage. We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object created earlier.

Because the configuration of this job is very similar to that described in “J10_MATCHFREQ_STAN_CUSTOMER” on page 266, we do not repeat it here. However, some of the configurations of interest are as follows:

- Figure 1-495 on page 412 shows the MATCH_FREQUENCY - Match Frequency Stage window with Match Specification of NONE.
- After saving, compiling, and running this job (Figure 1-496 on page 412), the contents of the output of this stage are listed in Figure 1-497 on page 413. We do not document the interpretation of the format and content of this file.

Now, proceed to “J10_REFERENCE_MatchSpec_MAILING_LIST” on page 413.

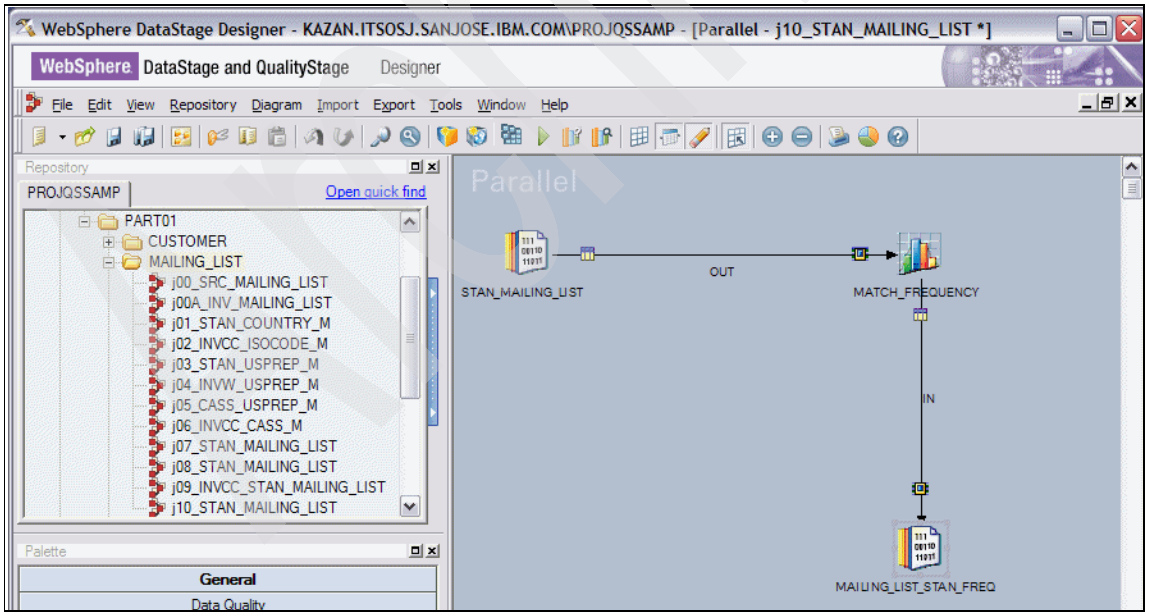
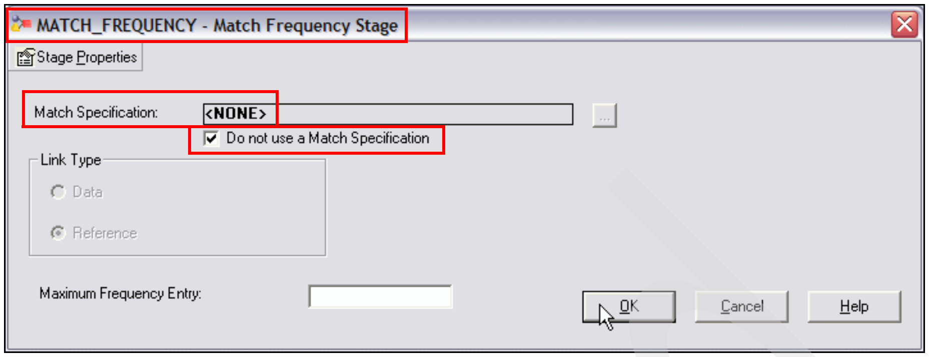


Figure 1-494 Create J10_MATCHFREQ_STAN_MAILING_LIST job 1/4


MATCH_FREQUENCY - Match Frequency Stage

Stage Properties

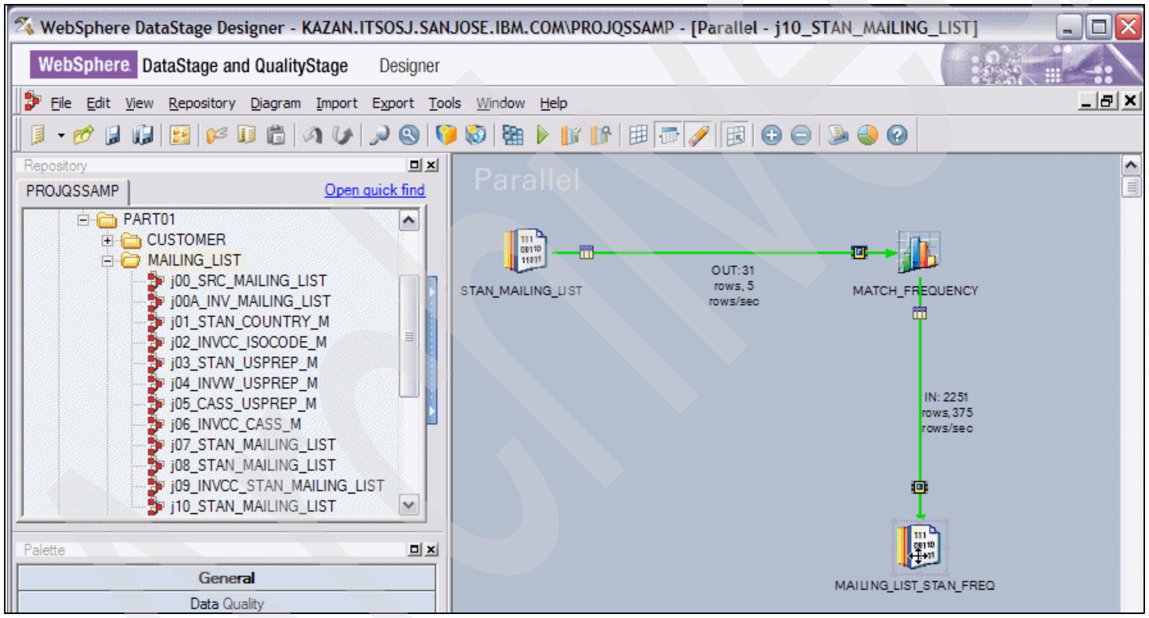
Match Specification: <NONE>

[✓] Do not use a Match Specification

Link Type: Data [✓ Reference]

Maximum Frequency Entry:

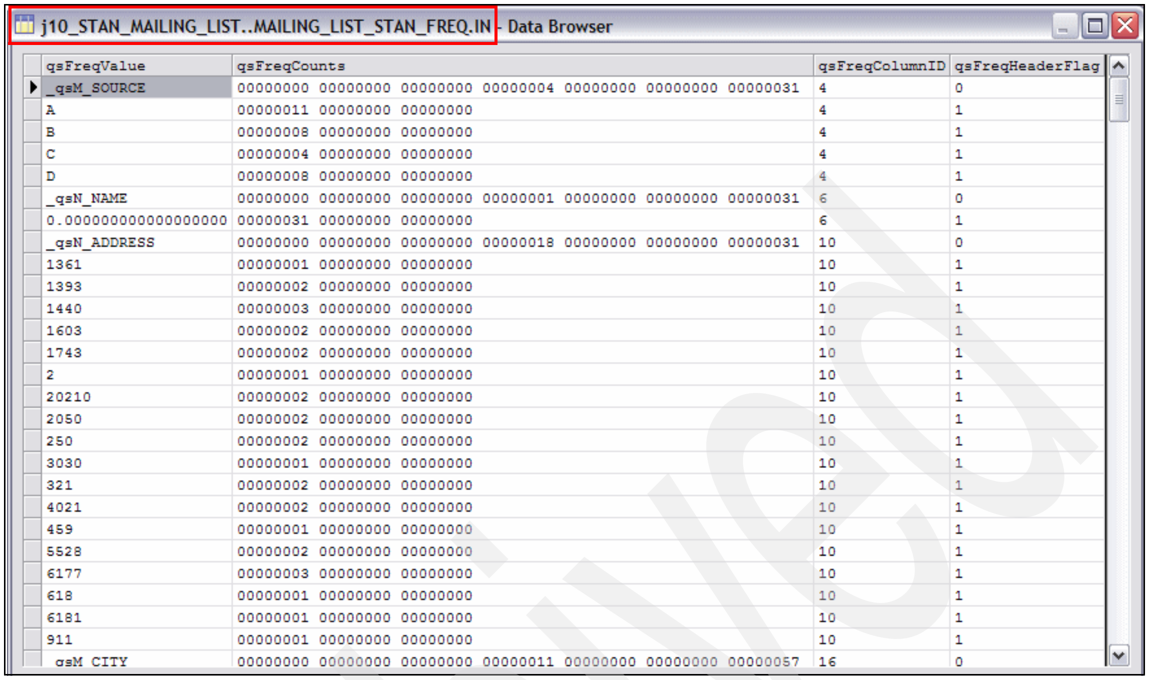
Figure 1-495 Create J10_MATCHFREQ_STAN_MAILING_LIST job 2/4


WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j10_STAN_MAILING_LIST]

WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Repository
PROJQSSAMP
Open quick find

- PART01
  - CUSTOMER
  - MAILING_LIST
    - j00_SRC_MAILING_LIST
    - j00A_INV_MAILING_LIST
    - j01_STAN_COUNTRY_M
    - j02_INVCC_ISOCODE_M
    - j03_STAN_USPREP_M
    - j04_INVW_USPREP_M
    - j05_CASS_USPREP_M
    - j06_INVCC_CASS_M
    - j07_STAN_MAILING_LIST
    - j08_STAN_MAILING_LIST
    - j09_INVCC_STAN_MAILING_LIST
    - j10_STAN_MAILING_LIST

Figure 1-496 Create J10_MATCHFREQ_STAN_MAILING_LIST job 3/4



j10_STAN_MAILING_LIST..MAILING_LIST_STAN_FREQ.IN - Data Browser

qsFreqValue	qsFreqCounts	qsFreqColumnID	qsFreqHeaderFlag
_qsM_SOURCE	00000000 00000000 00000000 00000004 00000000 00000000 00000031	4	0
A	00000011 00000000 00000000	4	1
B	00000008 00000000 00000000	4	1
C	00000004 00000000 00000000	4	1
D	00000008 00000000 00000000	4	1
_qsN_NAME	00000000 00000000 00000000 00000001 00000000 00000000 00000031	6	0
0.000000000000000000	00000031 00000000 00000000	6	1
_qsN_ADDRESS	00000000 00000000 00000000 00000018 00000000 00000000 00000031	10	0
1361	00000001 00000000 00000000	10	1
1393	00000002 00000000 00000000	10	1
1440	00000003 00000000 00000000	10	1
1603	00000002 00000000 00000000	10	1
1743	00000002 00000000 00000000	10	1
2	00000001 00000000 00000000	10	1
20210	00000002 00000000 00000000	10	1
2050	00000002 00000000 00000000	10	1
250	00000002 00000000 00000000	10	1
3030	00000001 00000000 00000000	10	1
321	00000002 00000000 00000000	10	1
4021	00000002 00000000 00000000	10	1
459	00000001 00000000 00000000	10	1
5528	00000002 00000000 00000000	10	1
6177	00000003 00000000 00000000	10	1
618	00000001 00000000 00000000	10	1
6181	00000001 00000000 00000000	10	1
911	00000001 00000000 00000000	10	1
qsM CITY	00000000 00000000 00000000 00000011 00000000 00000000 00000057	16	0

Figure 1-497 Create J10_MATCHFREQ_STAN_MAILING_LIST job 4/4

J10_REFERENCE_MatchSpec_MAILING_LIST

In this next step, we generate a match specification for a Reference match stage using as input the match frequency data that was created in the "J10_MATCHFREQ_STAN_MAILING_LIST" on page 411 job. The specification includes a single-pass based on the three digits ZIP code and the street name.

Because the creation of a match specification is described in "J15_Undup_MatchSpec_CUSTOMER" on page 318, we do not repeat it here. However, in this case it is for a Reference Match Type and there are some differences as follows:

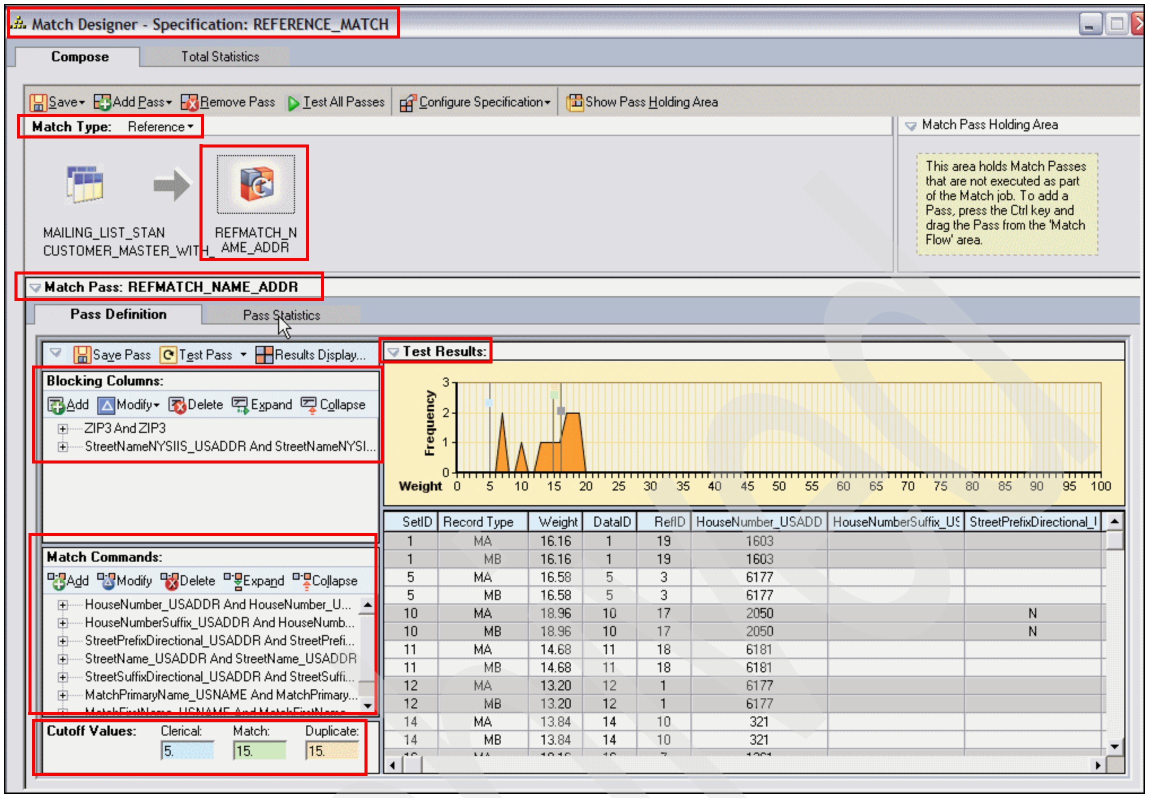
- Two data sets are specified as input for the Match Type is Reference, and four data sets are specified in the set up of the test environment.
- Figure 1-498 on page 415 shows the Blocking Columns (ZIP3 and StreetNameNYSIIS_USADDR), Match Commands, and Cutoff Values chosen for the REFMATCH_NAME_ADDR pass in the REFERENCE_MATCH specification. In the Test Results pane, there are a number of matched pairs as follows:
  - MA and MB in the Record Type column indicate that they are A and B records of a matched pair.

  - CA and CB in the Record Type column indicate that they constitute a clerical review (borderline match) pair—none in Figure 1-498 on page 415.
  - DA and DB in the Record Type indicate a duplicate A or B record. These follow matched pairs or clerical cases—none in Figure 1-498 on page 415.
  - UA and UB in the Record Type indicate unmatched records on the A file or the B file—none in Figure 1-498 on page 415.

  An asterisk (*) displaying after a record type designation (such as MA) indicates that the match (or duplicate) was an exact match. This is indicated on reports and extracts as a flag value of *X*.

- Figure 1-499 on page 416 shows the statistics of this pass in tabular and graphical form. Here again, the OVERFLOW blocks statistic shows a zero value, which indicates a no buffer overflow condition.

The test of the match specification with the full volume of data appears to deliver results that are accurate. This specification is then used in a Reference match stage as described in the step “J11_REFMATCH” on page 416.



Match Type: Reference

This area holds Match Passes that are not executed as part of the Match job. To add a Pass, press the Ctrl key and drag the Pass from the 'Match Flow' area.

Match Pass: REFMATCH_NAME_ADDR

Blocking Columns:

- ZIP3 And ZIP3
- StreetNameNYSIIS_USADDR And StreetNameNYSI...

Match Commands:

- HouseNumber_USADDR And HouseNumber_U...
- HouseNumberSuffix_USADDR And HouseNumb...
- StreetPrefixDirectional_USADDR And StreetPrefi...
- StreetName_USADDR And StreetName_USADDR
- StreetSuffixDirectional_USADDR And StreetSuffi...
- MatchPrimaryName_USNAME And MatchPrimary...

Cutoff Values: Clerical: 5. Match: 15. Duplicate: 15.

SetID	Record Type	Weight	DataID	RefID	HouseNumber_USADD	HouseNumberSuffix_US	StreetPrefixDirectional_I
1	MA	16.16	1	19	1603		
1	MB	16.16	1	19	1603		
5	MA	16.58	5	3	6177		
5	MB	16.58	5	3	6177		
10	MA	18.96	10	17	2050		N
10	MB	18.96	10	17	2050		N
11	MA	14.68	11	18	6181		
11	MB	14.68	11	18	6181		
12	MA	13.20	12	1	6177		
12	MB	13.20	12	1	6177		
14	MA	13.84	14	10	321		
14	MB	13.84	14	10	321		

Figure 1-498 Create J10_REFERENCE_MatchSpec_MAILING_LIST 1/2

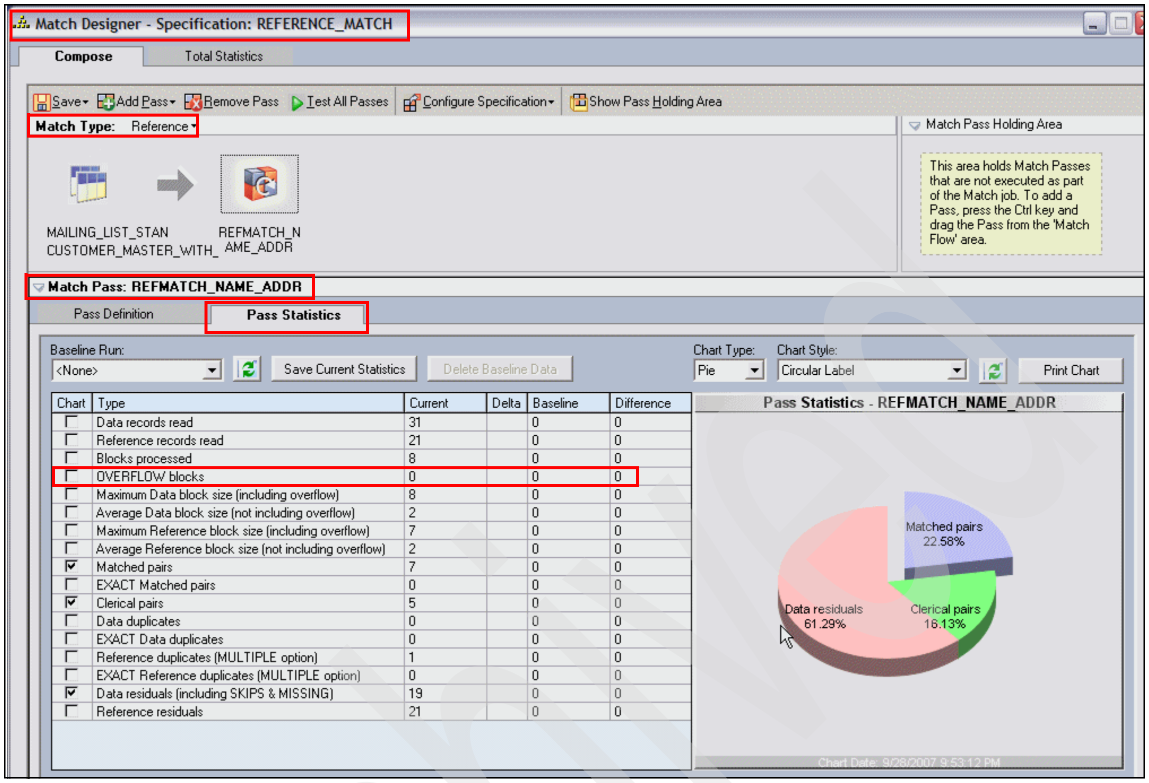


Chart	Type	Current	Delta	Baseline	Difference
☐	Data records read	31		0	0
☐	Reference records read	21		0	0
☐	Blocks processed	8		0	0
☐	OVERFLOW blocks	0		0	0
☐	Maximum Data block size (including overflow)	8		0	0
☐	Average Data block size (not including overflow)	2		0	0
☐	Maximum Reference block size (including overflow)	7		0	0
☐	Average Reference block size (not including overflow)	2		0	0
☑	Matched pairs	7		0	0
☐	EXACT Matched pairs	0		0	0
☑	Clerical pairs	5		0	0
☐	Data duplicates	0		0	0
☐	EXACT Data duplicates	0		0	0
☐	Reference duplicates (MULTIPLE option)	1		0	0
☐	EXACT Reference duplicates (MULTIPLE option)	0		0	0
☑	Data residuals (including SKIPS & MISSING)	19		0	0
☐	Reference residuals	21		0	0

Figure 1-499 Create J10_REFERENCE_MatchSpec_MAILING_LIST 2/2

J11_REFMATCH

In this step, we determine if there were any matches between the records in the credit card customer file and mailing list file by running the Reference stage, which requires standardized data (from "J08_STAN_MAILING_LIST_Domain_Specific" on page 390) and reference data (from "J17_CUSTOMER_MASTER_WITH_HOUSEHOLD" on page 327) as source data, a reference match specification (from "J10_REFERENCE_MatchSpec_MAILING_LIST" on page 413), and frequency information for both sources (from "J18_MATCHFREQ_CUSTOMER_WITH_HOUSEHOLD" on page 332 and "J10_MATCHFREQ_STAN_MAILING_LIST" on page 411). The output of the Reference Match stage includes master records, clerical review records, duplicates, and residuals. You can use this output as input to the Survive stage.

Figure 1-500 on page 419 through Figure 1-524 on page 430 describe the main steps using Designer Client to build and execute the Reference Match job.

The main steps are as follows:

1. Figure 1-500 on page 419 shows the Reference Match stage that creates six data sets that contain:
   - Matched records
   - Clerical review records
   - Mailing list duplicates
   - Credit card customer duplicates
   - Mailing list residuals
   - Credit card residuals

   We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object created earlier.
2. Configure the four input data sets shown in Figure 1-500 on page 419. This process is similar to earlier configurations, thus we do not repeat it here.
3. Next, configure the REFERENCE_MATCH stage as follows:
   a. Double-click the **REFERENCE_MATCH** stage icon and click the ... Match Specification button. We do not show this process here.
   b. From the Repository window, double-click the Match Specifications folder and select REFERENCE_MATCH as the Reference Match specification that was created in “J10_REFERENCE_MatchSpec_MAILING_LIST” on page 413. We do not show this process here.
   c. Check all the Match Outputs fields in the REFEREN CE_MATCH - Reference Match Stage shown in Figure 1-501 on page 419:
      - Match sends matched (master) records for both inputs.
      - Clerical separates those records that require clerical review for both inputs.
      - Data Duplicates include duplicate records that in the data input.
      - Reference Duplicates include duplicate records in the reference input.
      - Data Residuals includes records that are not matched in the data input.
      - Reference Residuals includes records that are not matched in the reference input.

      Click the **Many-to-one** Match Type, and click **Stage Properties** to configure link ordering.
   d. The REFERENCE_MATCH - Reference Match window in Figure 1-502 on page 420 shows the input and output links in this stage. Click the **Output** tab.
   e. For the MATCHED link (Output name field), click **Mapping** as shown in Figure 1-503 on page 420. Copy all the columns from Columns pane to the MATCHED pane.

Repeat the process for the CLERICAL, MLDUPLICATE, CDUPLICATE, MLRESIDUAL, and CRESIDUAL links (Output name field). We do not repeat this process here.

4. Next, configure the six output data set objects. We do not repeat this process here.
5. After saving, compiling, and running this job j11_REFMATCH (Figure 1-509 on page 424), view the content of the six data set objects as shown in Figure 1-510 on page 424 through Figure 1-524 on page 430. The input credit card customer master input had 21 records, while the mailing list input had 31 records. The output of the reference match stage is as follows:
   – Matched records found are 8 as shown in Figure 1-510 on page 424 and Figure 1-511 on page 425.
   – Clerical review records are 2 as shown in Figure 1-512 on page 425 and Figure 1-513 on page 425.
   – No duplicates are found relating to the mailing list.
   – Credit card customer duplicate records found are 3 as shown in Figure 1-514 on page 425 and Figure 1-515 on page 425.
   – Mailing list residuals has 21 records as shown in Figure 1-516 on page 426 and Figure 1-517 on page 426.
   – Credit card customer residuals has 13 records as shown in Figure 1-518 on page 427 and Figure 1-524 on page 430. This output is shown using the Data Set Management functionality.
6. Save the data set metadata to Table Definitions. We do not repeat this process here.

The next step is to create a single report that contains each clerical review record followed by the master record to which it relates so that the clerical reviewer is able to view record details next to each other. We described this step in “J12_CLERICAL_REPORT_MAILING_LIST” on page 430.

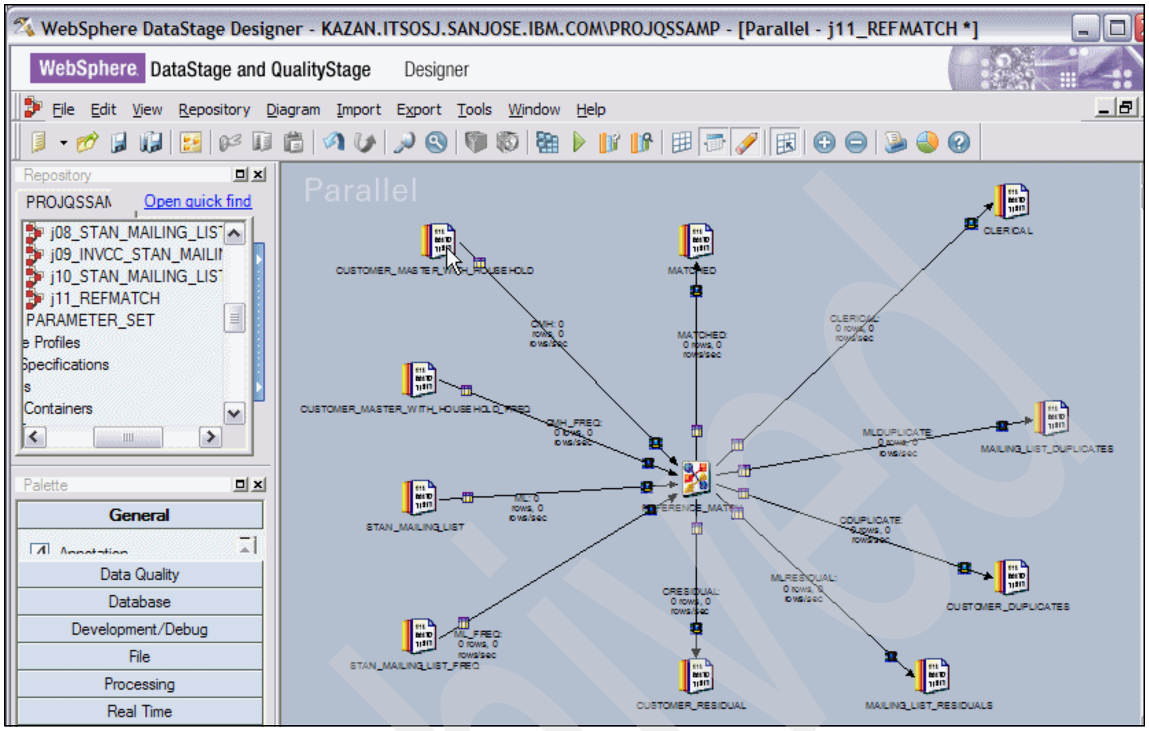
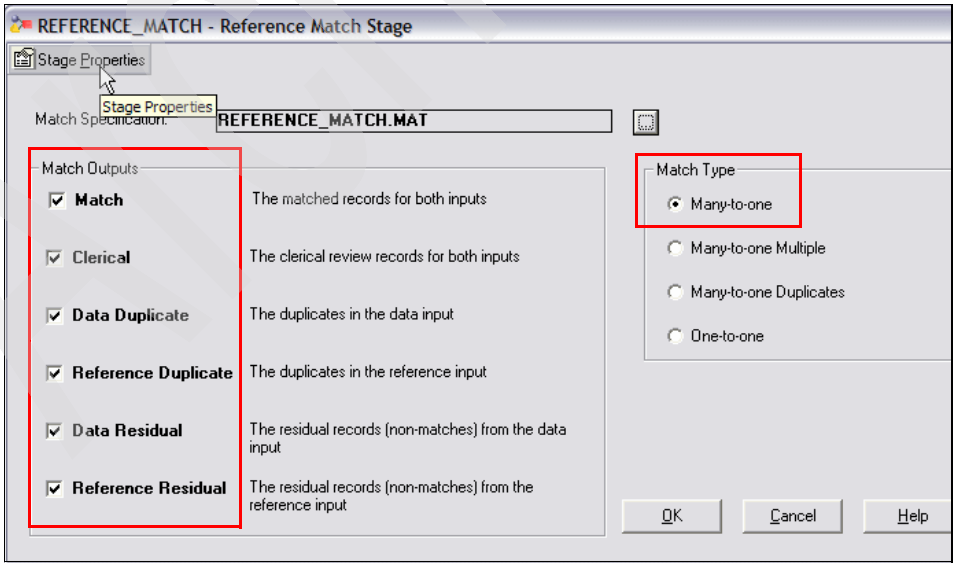

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j11_REFMATCH *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Repository
PROJQSSAN
Open quick find
j08_STAN_MAILING_LIS
j09_INVCC_STAN_MAILI
j10_STAN_MAILING_LIS
j11_REFMATCH
PARAMETER_SET
Profiles
Specifications
Containers
Palette
General
Data Quality
Database
Development/Debug
File
Processing
Real Time
Parallel
CUSTOMER_MASTER_WITH_HOUSEHOLD
CUSTOMER_MASTER_WITH_HOUSEHOLD_FREQ
STAN_MAILING_LIST
STAN_MAILING_LIST_FREQ
MATCHED
CLERICAL
MAILING_LIST_DUPLICATES
CUSTOMER_DUPLICATES
CUSTOMER_RESIDUAL
MAILING_LIST_RESIDUALS
REFERENCE_MATCH
CMH: 0 rows, 0 rows/sec
CMH_FREQ: 0 rows, 0 rows/sec
ML: 0 rows, 0 rows/sec
ML_FREQ: 0 rows, 0 rows/sec
MATCHED: 0 rows, 0 rows/sec
CLERICAL: 0 rows, 0 rows/sec
MLDUPLICATE: 0 rows, 0 rows/sec
CDUPLICATE: 0 rows, 0 rows/sec
CRESIDUAL: 0 rows, 0 rows/sec
MLRESIDUAL: 0 rows, 0 rows/sec

Figure 1-500 Create J11_REFMATCH job 1/25



REFERENCE_MATCH - Reference Match Stage

Stage Properties

Match Specification: REFERENCE_MATCH.MAT

Match Outputs

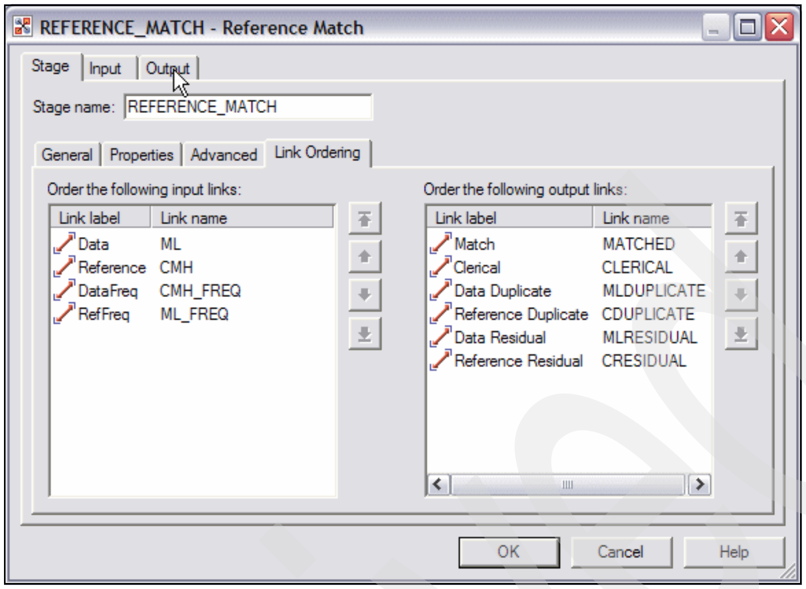
☑ Match	The matched records for both inputs
☑ Clerical	The clerical review records for both inputs
☑ Data Duplicate	The duplicates in the data input
☑ Reference Duplicate	The duplicates in the reference input
☑ Data Residual	The residual records (non-matches) from the data input
☑ Reference Residual	The residual records (non-matches) from the reference input

Match Type

[✓ Many-to-one] Many-to-one Multiple Many-to-one Duplicates One-to-one

OK Cancel Help

Figure 1-501 Create J11_REFMATCH job 2/25


REFERENCE_MATCH - Reference Match

Stage name: REFERENCE_MATCH

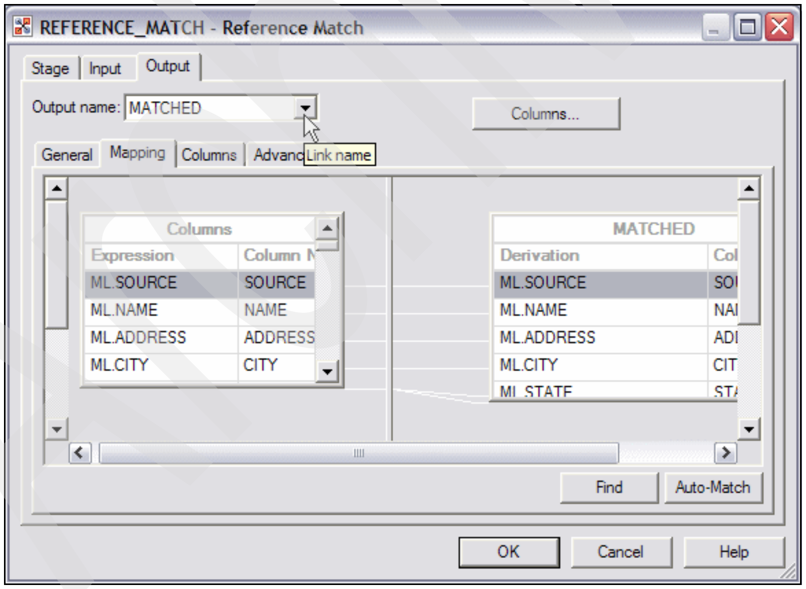
Order the following input links:

Link label	Link name
Data	ML
Reference	CMH
DataFreq	CMH_FREQ
RefFreq	ML_FREQ

Order the following output links:

Link label	Link name
Match	MATCHED
Clerical	CLERICAL
Data Duplicate	MLDUPLICATE
Reference Duplicate	CDUPLICATE
Data Residual	MLRESIDUAL
Reference Residual	CRESIDUAL

Figure 1-502 Create J11_REFMATCH job 3/25

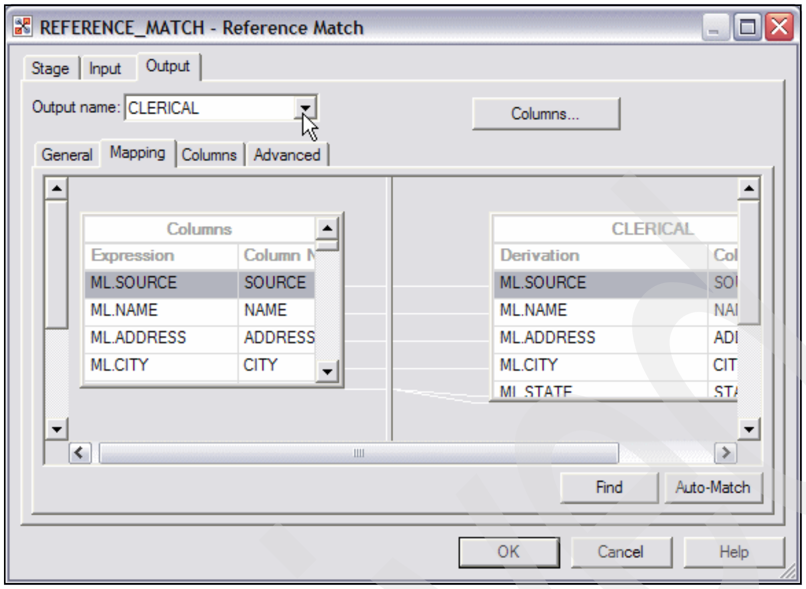

REFERENCE_MATCH - Reference Match

Output name: MATCHED

Expression	Column N
ML.SOURCE	SOURCE
ML.NAME	NAME
ML.ADDRESS	ADDRESS
ML.CITY	CITY

Derivation	Col
ML.SOURCE	SO
ML.NAME	NA
ML.ADDRESS	AD
ML.CITY	CIT
ML.STATE	ST

Figure 1-503 Create J11_REFMATCH job 4/25



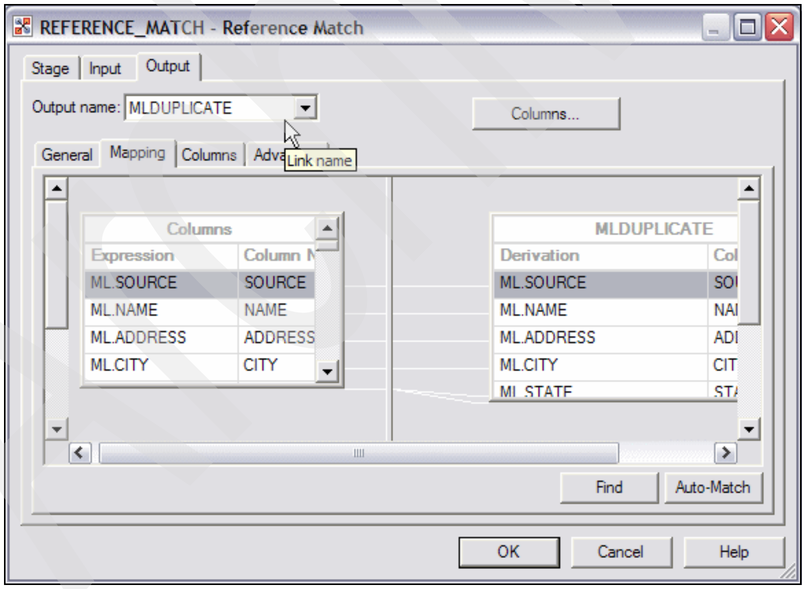
REFERENCE_MATCH - Reference Match

Output name: CLERICAL

Columns	
Expression	Column N
ML.SOURCE	SOURCE
ML.NAME	NAME
ML.ADDRESS	ADDRESS
ML.CITY	CITY

CLERICAL	
Derivation	Col
ML.SOURCE	SO
ML.NAME	NA
ML.ADDRESS	AD
ML.CITY	CIT
ML.STATE	ST

Figure 1-504   Create J11_REFMATCH job 5/25



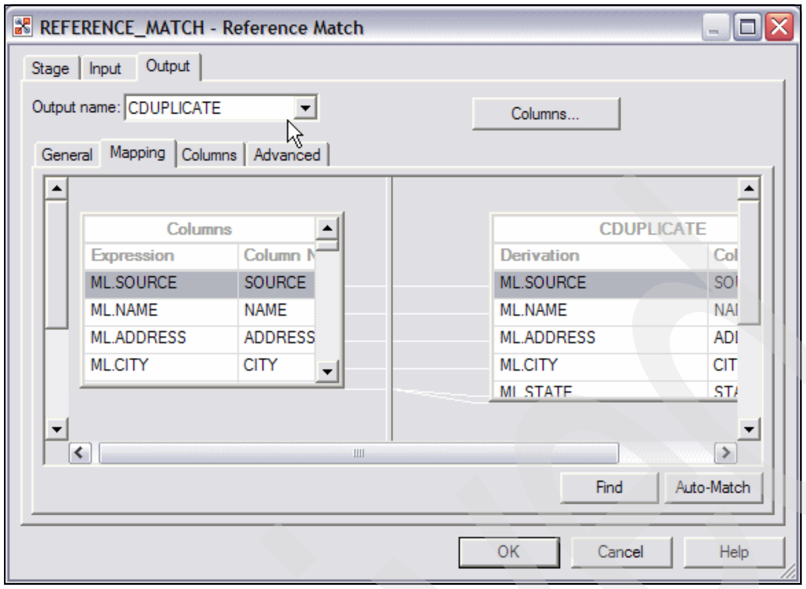
REFERENCE_MATCH - Reference Match

Output name: MLDUPLICATE

Columns	
Expression	Column N
ML.SOURCE	SOURCE
ML.NAME	NAME
ML.ADDRESS	ADDRESS
ML.CITY	CITY

MLDUPLICATE	
Derivation	Col
ML.SOURCE	SO
ML.NAME	NA
ML.ADDRESS	AD
ML.CITY	CIT
ML.STATE	ST

Figure 1-505   Create J11_REFMATCH job 6/25

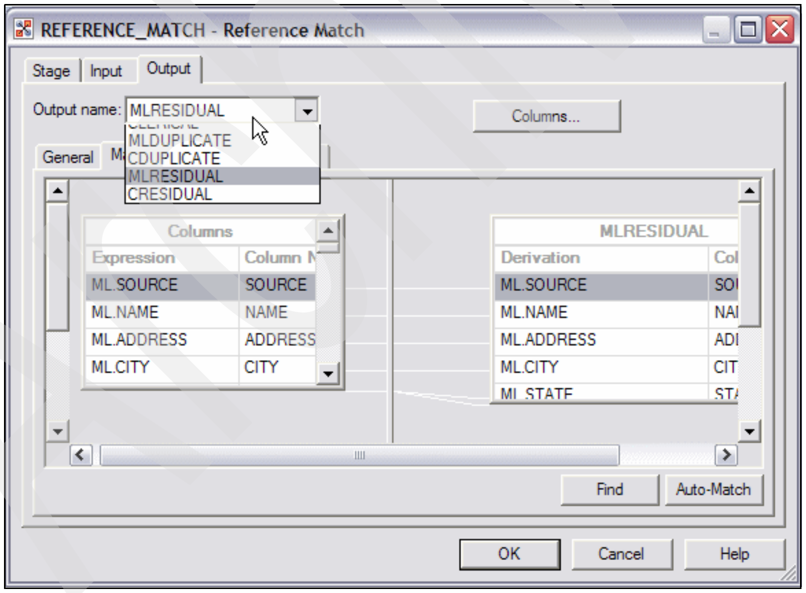


Output name: CDUPLICATE

Columns	
Expression	Column N
ML.SOURCE	SOURCE
ML.NAME	NAME
ML.ADDRESS	ADDRESS
ML.CITY	CITY

CDUPLICATE	
Derivation	Col
ML.SOURCE	SO
ML.NAME	NA
ML.ADDRESS	AD
ML.CITY	CIT
ML.STATE	ST

Figure 1-506 Create J11_REFMATCH job 7/25



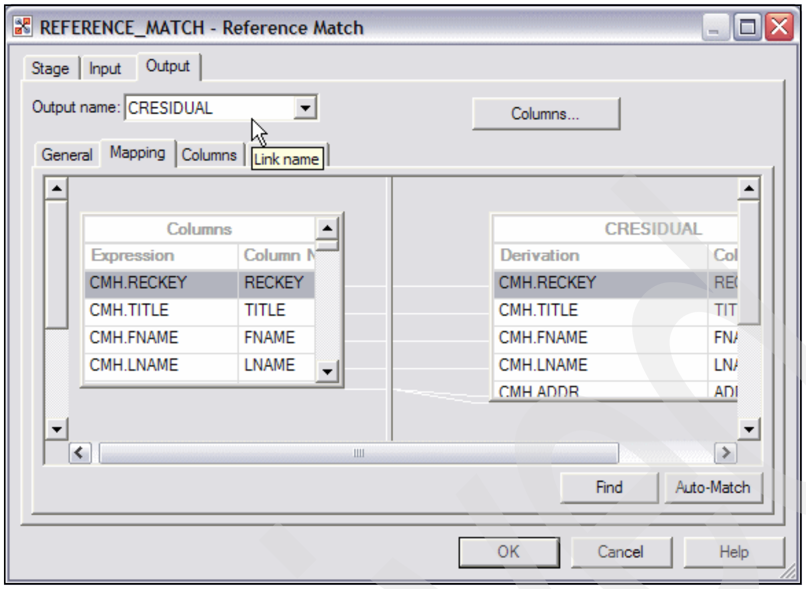
Output name: MLRESIDUAL

MLDUPLICATE
CDUPLICATE
MLRESIDUAL
CRESIDUAL

Columns	
Expression	Column N
ML.SOURCE	SOURCE
ML.NAME	NAME
ML.ADDRESS	ADDRESS
ML.CITY	CITY

MLRESIDUAL	
Derivation	Col
ML.SOURCE	SO
ML.NAME	NA
ML.ADDRESS	AD
ML.CITY	CIT
ML.STATE	ST

Figure 1-507 Create J11_REFMATCH job 8/25



Output name: CRESIDUAL

Columns	
Expression	Column N
CMH.RECKEY	RECKEY
CMH.TITLE	TITLE
CMH.FNAME	FNAME
CMH.LNAME	LNAME

CRESIDUAL	
Derivation	Col
CMH.RECKEY	REC
CMH.TITLE	TIT
CMH.FNAME	FNA
CMH.LNAME	LNA
CMH.ADDR	ADI

Figure 1-508   Create J11_REFMATCH job 9/25

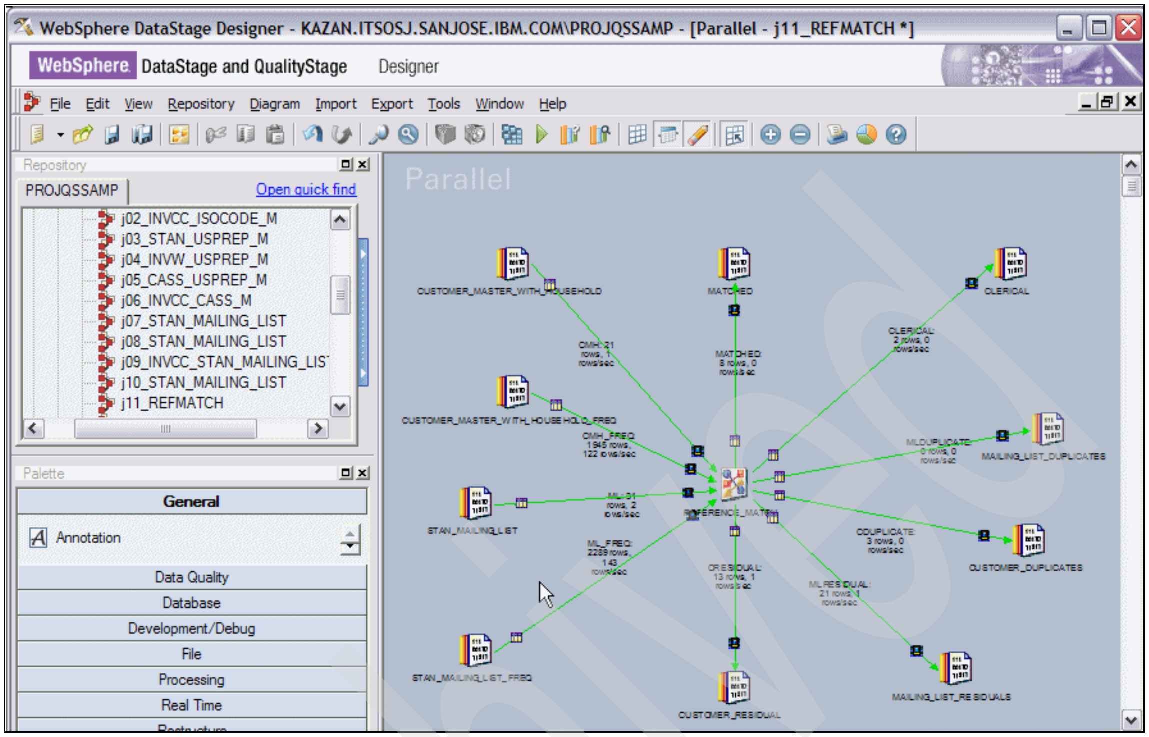
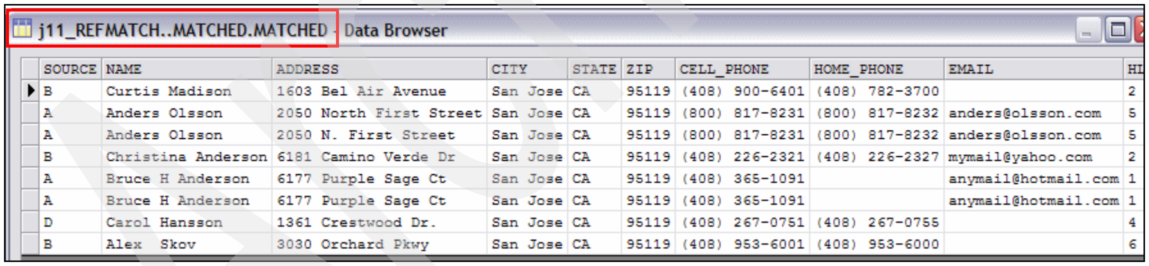
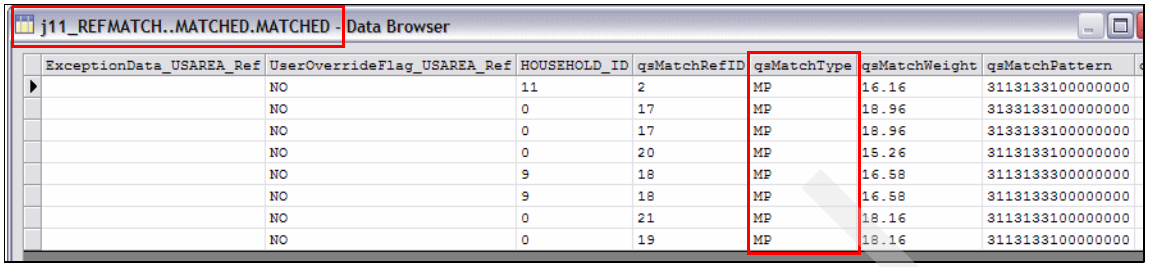

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j11_REFMATCH *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Repository
PROJQSSAMP
Open quick find
j02_INVCC_ISOCODE_M
j03_STAN_USPREP_M
j04_INVW_USPREP_M
j05_CASS_USPREP_M
j06_INVCC_CASS_M
j07_STAN_MAILING_LIST
j08_STAN_MAILING_LIST
j09_INVCC_STAN_MAILING_LIS
j10_STAN_MAILING_LIST
j11_REFMATCH
Palette
General
Annotation
Data Quality
Database
Development/Debug
File
Processing
Real Time
Parallel
CUSTOMER_MASTER_WITH_HOUSEHOLD
MATCHED
CLERICAL
CMH: 21 rows, 1 rows/sec
MATCHED: 8 rows, 0 rows/sec
CLERICAL: 2 rows, 0 rows/sec
CUSTOMER_MASTER_WITH_HOUSEHOLD_FREQ
CMH_FREQ: 1945 rows, 122 rows/sec
MLDUPLICATE: 0 rows, 0 rows/sec
MAILING_LIST_DUPLICATES
STAN_MAILING_LIST
ML: 31 rows, 2 rows/sec
REFERENCE_MATCH
ML_FREQ: 2289 rows, 143 rows/sec
CRESIDUAL: 13 rows, 1 rows/sec
MLRESIDUAL: 21 rows, 1 rows/sec
CDUPLICATE: 3 rows, 0 rows/sec
CUSTOMER_DUPLICATES
STAN_MAILING_LIST_FREQ
CUSTOMER_RESIDUAL
MAILING_LIST_RESIDUALS

Figure 1-509 Create J11_REFMATCH job 10/25


j11_REFMATCH..MATCHED.MATCHED - Data Browser

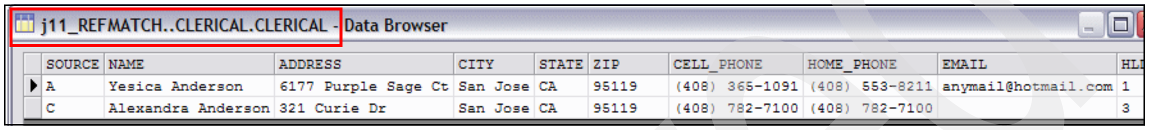
SOURCE	NAME	ADDRESS	CITY	STATE	ZIP	CELL_PHONE	HOME_PHONE	EMAIL	HI
B	Curtis Madison	1603 Bel Air Avenue	San Jose	CA	95119	(408) 900-6401	(408) 782-3700		2
A	Anders Olsson	2050 North First Street	San Jose	CA	95119	(800) 817-8231	(800) 817-8232	anders@olsson.com	5
A	Anders Olsson	2050 N. First Street	San Jose	CA	95119	(800) 817-8231	(800) 817-8232	anders@olsson.com	5
B	Christina Anderson	6181 Camino Verde Dr	San Jose	CA	95119	(408) 226-2321	(408) 226-2327	mymail@yahoo.com	2
A	Bruce H Anderson	6177 Purple Sage Ct	San Jose	CA	95119	(408) 365-1091		anymail@hotmail.com	1
A	Bruce H Anderson	6177 Purple Sage Ct	San Jose	CA	95119	(408) 365-1091		anymail@hotmail.com	1
D	Carol Hansson	1361 Crestwood Dr.	San Jose	CA	95119	(408) 267-0751	(408) 267-0755		4
B	Alex Skov	3030 Orchard Pkwy	San Jose	CA	95119	(408) 953-6001	(408) 953-6000		6

Figure 1-510 Create J11_REFMATCH job 11/25


j11_REFMATCH..MATCHED.MATCHED - Data Browser

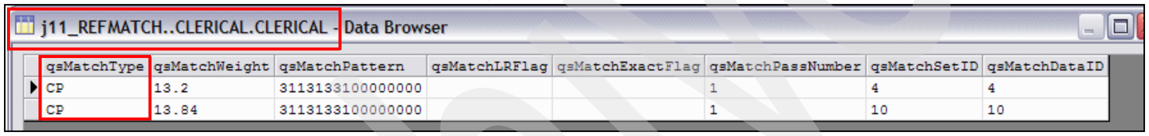
ExceptionData_USAREA_Ref	UserOverrideFlag_USAREA_Ref	HOUSEHOLD_ID	qsMatchRefID	qsMatchType	qsMatchWeight	qsMatchPattern
	NO	11	2	MP	16.16	3113133100000000
	NO	0	17	MP	18.96	3133133100000000
	NO	0	17	MP	18.96	3133133100000000
	NO	0	20	MP	15.26	3113133100000000
	NO	9	18	MP	16.58	3113133300000000
	NO	9	18	MP	16.58	3113133300000000
	NO	0	21	MP	18.16	3113133100000000
	NO	0	19	MP	18.16	3113133100000000

Figure 1-511   Create J11_REFMATCH job 12/25


j11_REFMATCH..CLERICAL.CLERICAL - Data Browser

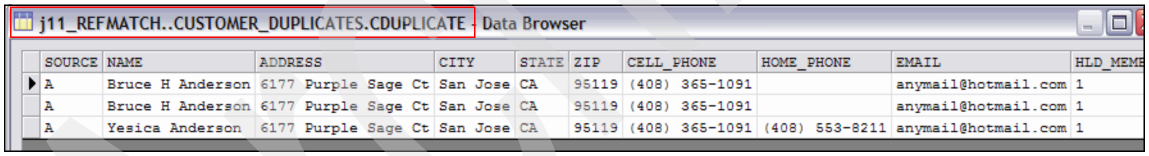
SOURCE	NAME	ADDRESS	CITY	STATE	ZIP	CELL_PHONE	HOME_PHONE	EMAIL	HLI
A	Yesica Anderson	6177 Purple Sage Ct	San Jose	CA	95119	(408) 365-1091	(408) 553-8211	anymail@hotmail.com	1
C	Alexandra Anderson	321 Curie Dr	San Jose	CA	95119	(408) 782-7100	(408) 782-7100		3

Figure 1-512   Create J11_REFMATCH job 13/25


j11_REFMATCH..CLERICAL.CLERICAL - Data Browser

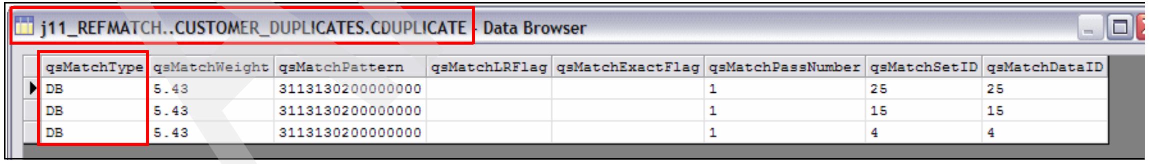
qsMatchType	qsMatchWeight	qsMatchPattern	qsMatchLRFlag	qsMatchExactFlag	qsMatchPassNumber	qsMatchSetID	qsMatchDataID
CP	13.2	3113133100000000			1	4	4
CP	13.84	3113133100000000			1	10	10

Figure 1-513   Create J11_REFMATCH job 14/25


j11_REFMATCH..CUSTOMER_DUPLICATES.CDUPLICATE - Data Browser

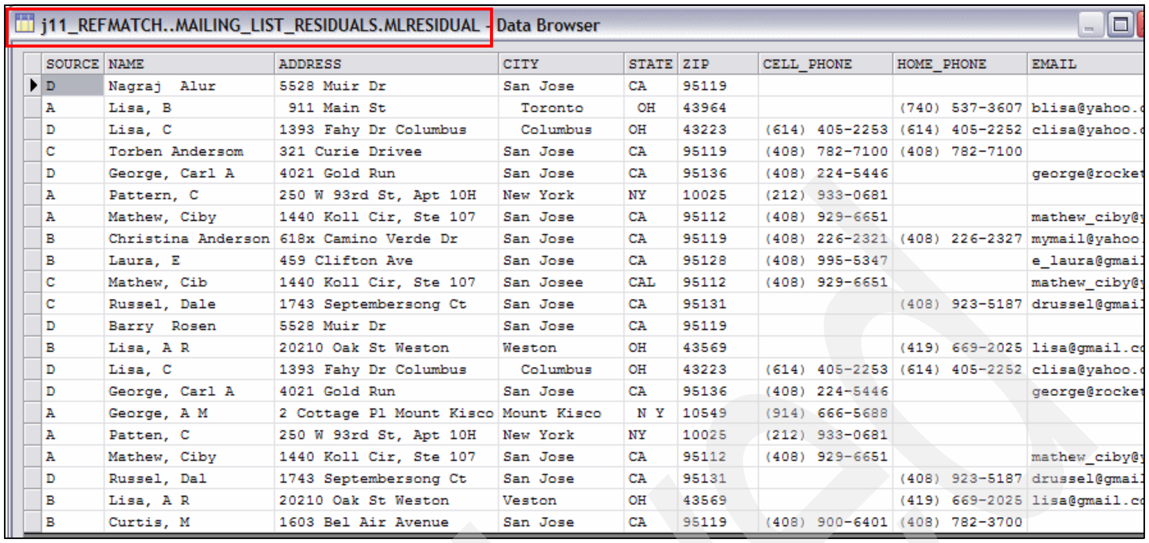
SOURCE	NAME	ADDRESS	CITY	STATE	ZIP	CELL_PHONE	HOME_PHONE	EMAIL	HLD_MEM
A	Bruce H Anderson	6177 Purple Sage Ct	San Jose	CA	95119	(408) 365-1091		anymail@hotmail.com	1
A	Bruce H Anderson	6177 Purple Sage Ct	San Jose	CA	95119	(408) 365-1091		anymail@hotmail.com	1
A	Yesica Anderson	6177 Purple Sage Ct	San Jose	CA	95119	(408) 365-1091	(408) 553-8211	anymail@hotmail.com	1

Figure 1-514   Create J11_REFMATCH job 15/25


j11_REFMATCH..CUSTOMER_DUPLICATES.CDUPLICATE - Data Browser

qsMatchType	qsMatchWeight	qsMatchPattern	qsMatchLRFlag	qsMatchExactFlag	qsMatchPassNumber	qsMatchSetID	qsMatchDataID
DB	5.43	3113130200000000			1	25	25
DB	5.43	3113130200000000			1	15	15
DB	5.43	3113130200000000			1	4	4

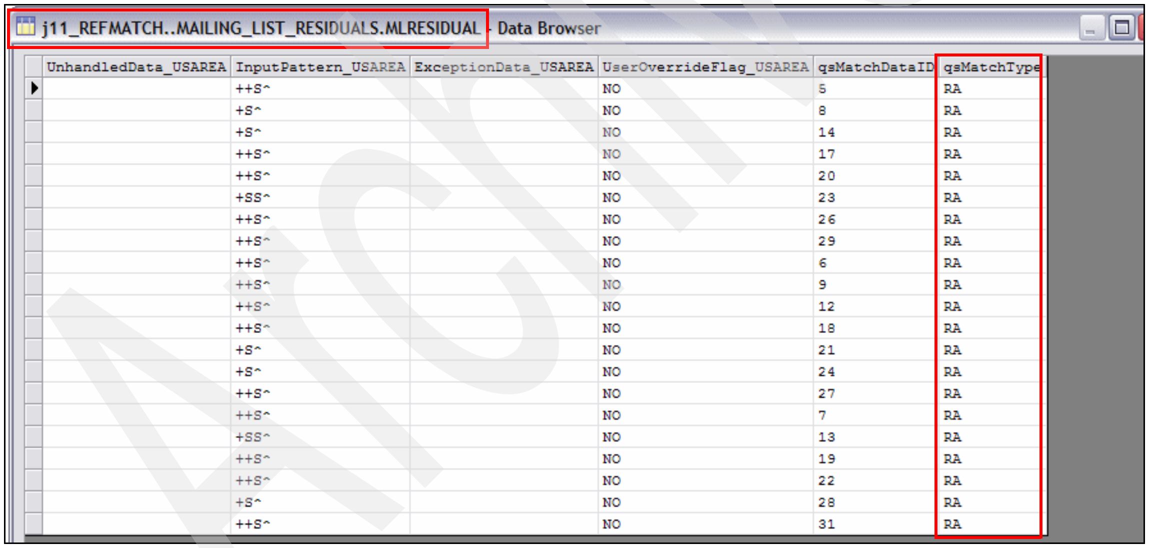
Figure 1-515   Create J11_REFMATCH job 16/25



j11_REFMATCH..MAILING_LIST_RESIDUALS.MLRESIDUAL - Data Browser

SOURCE	NAME	ADDRESS	CITY	STATE	ZIP	CELL_PHONE	HOME_PHONE	EMAIL
D	Nagraj Alur	5528 Muir Dr	San Jose	CA	95119			
A	Lisa, B	911 Main St	Toronto	OH	43964		(740) 537-3607	blisa@yahoo.c
D	Lisa, C	1393 Fahy Dr Columbus	Columbus	OH	43223	(614) 405-2253	(614) 405-2252	clisa@yahoo.c
C	Torben Andersom	321 Curie Drivee	San Jose	CA	95119	(408) 782-7100	(408) 782-7100	
D	George, Carl A	4021 Gold Run	San Jose	CA	95136	(408) 224-5446		george@rocket
A	Pattern, C	250 W 93rd St, Apt 10H	New York	NY	10025	(212) 933-0681		
A	Mathew, Ciby	1440 Koll Cir, Ste 107	San Jose	CA	95112	(408) 929-6651		mathew_ciby@y
B	Christina Anderson	618x Camino Verde Dr	San Jose	CA	95119	(408) 226-2321	(408) 226-2327	mymail@yahoo.
B	Laura, E	459 Clifton Ave	San Jose	CA	95128	(408) 995-5347		e_laura@gmail
C	Mathew, Cib	1440 Koll Cir, Ste 107	San Josee	CAL	95112	(408) 929-6651		mathew_ciby@y
C	Russel, Dale	1743 Septembersong Ct	San Jose	CA	95131		(408) 923-5187	drussel@gmail
D	Barry Rosen	5528 Muir Dr	San Jose	CA	95119			
B	Lisa, A R	20210 Oak St Weston	Weston	OH	43569		(419) 669-2025	lisa@gmail.co
D	Lisa, C	1393 Fahy Dr Columbus	Columbus	OH	43223	(614) 405-2253	(614) 405-2252	clisa@yahoo.c
D	George, Carl A	4021 Gold Run	San Jose	CA	95136	(408) 224-5446		george@rocket
A	George, A M	2 Cottage Pl Mount Kisco	Mount Kisco	N Y	10549	(914) 666-5688		
A	Patten, C	250 W 93rd St, Apt 10H	New York	NY	10025	(212) 933-0681		
A	Mathew, Ciby	1440 Koll Cir, Ste 107	San Jose	CA	95112	(408) 929-6651		mathew_ciby@y
D	Russel, Dal	1743 Septembersong Ct	San Jose	CA	95131		(408) 923-5187	drussel@gmail
B	Lisa, A R	20210 Oak St Weston	Veston	OH	43569		(419) 669-2025	lisa@gmail.co
B	Curtis, M	1603 Bel Air Avenue	San Jose	CA	95119	(408) 900-6401	(408) 782-3700	

Figure 1-516   Create J11_REFMATCH job 17/25



j11_REFMATCH..MAILING_LIST_RESIDUALS.MLRESIDUAL - Data Browser

UnhandledData_USAREA	InputPattern_USAREA	ExceptionData_USAREA	UserOverrideFlag_USAREA	qsMatchDataID	qsMatchType
	++S^		NO	5	RA
	+S^		NO	8	RA
	+S^		NO	14	RA
	++S^		NO	17	RA
	++S^		NO	20	RA
	+SS^		NO	23	RA
	++S^		NO	26	RA
	++S^		NO	29	RA
	++S^		NO	6	RA
	++S^		NO	9	RA
	++S^		NO	12	RA
	++S^		NO	18	RA
	+S^		NO	21	RA
	+S^		NO	24	RA
	++S^		NO	27	RA
	++S^		NO	7	RA
	+SS^		NO	13	RA
	++S^		NO	19	RA
	++S^		NO	22	RA
	+S^		NO	28	RA
	++S^		NO	31	RA

Figure 1-517   Create J11_REFMATCH job 18/25

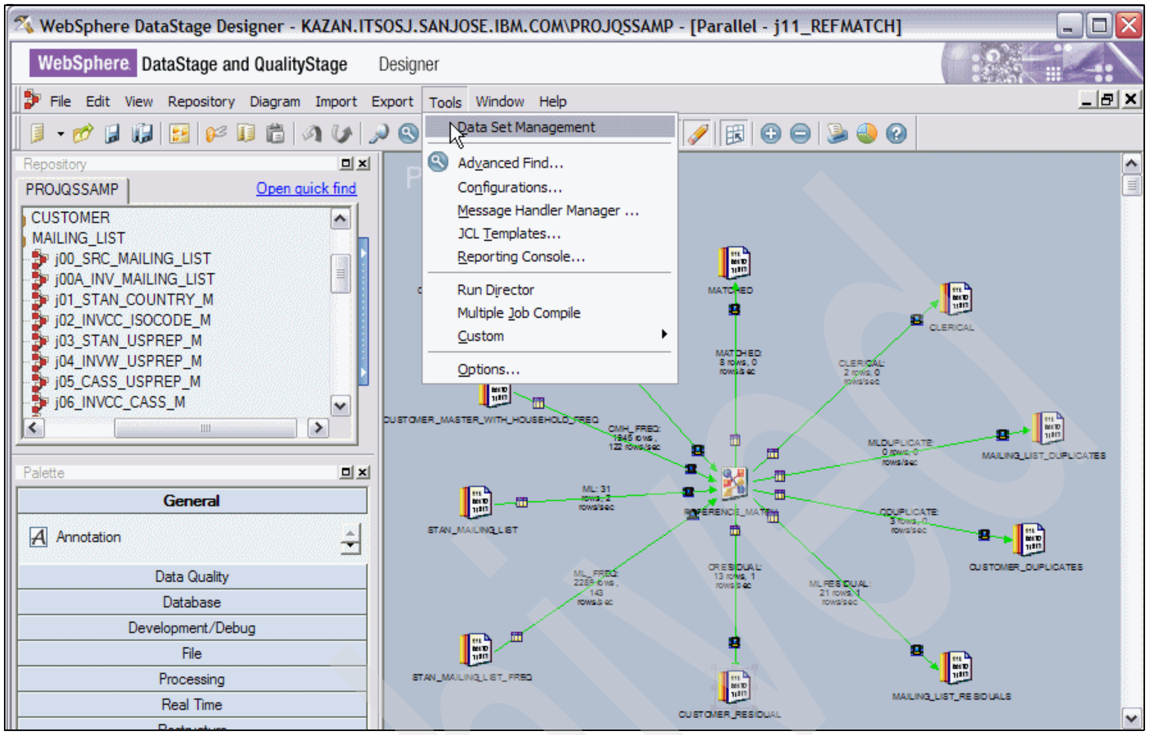
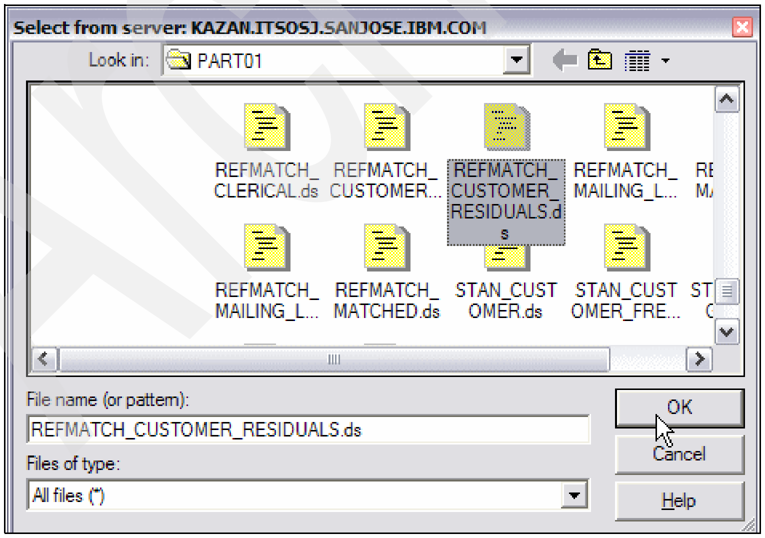


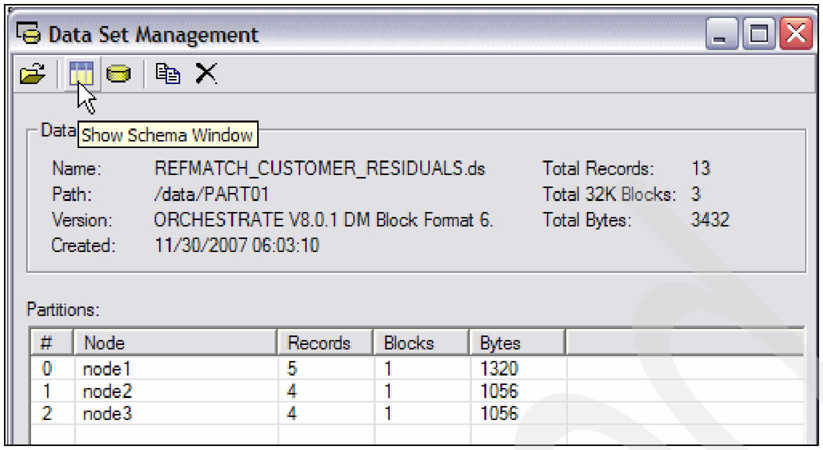
Figure 1-518 Create J11_REFMATCH job 19/25



File name (or pattern): REFMATCH_CUSTOMER_RESIDUALS.ds

Files of type: All files (*)

Figure 1-519 Create J11_REFMATCH job 20/25



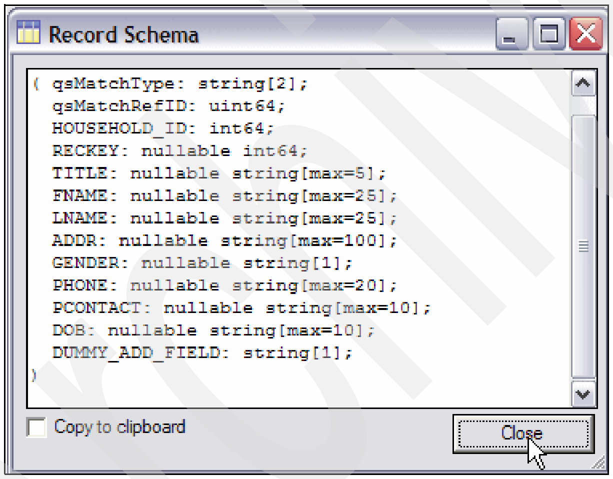
Name: REFMATCH_CUSTOMER_RESIDUALS.ds
Path: /data/PART01
Version: ORCHESTRATE V8.0.1 DM Block Format 6.
Created: 11/30/2007 06:03:10

Total Records: 13
Total 32K Blocks: 3
Total Bytes: 3432

Partitions:

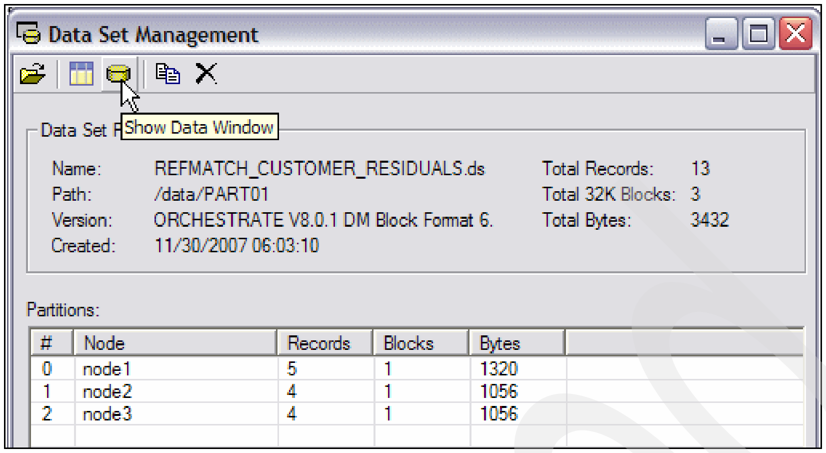
#	Node	Records	Blocks	Bytes
0	node1	5	1	1320
1	node2	4	1	1056
2	node3	4	1	1056

Figure 1-520 Create J11_REFMATCH job 21/25



```
( qsMatchType: string[2];
  qsMatchRefID: uint64;
  HOUSEHOLD_ID: int64;
  RECKEY: nullable int64;
  TITLE: nullable string[max=5];
  FNAME: nullable string[max=25];
  LNAME: nullable string[max=25];
  ADDR: nullable string[max=100];
  GENDER: nullable string[1];
  PHONE: nullable string[max=20];
  PCONTACT: nullable string[max=10];
  DOB: nullable string[max=10];
  DUMMY_ADD_FIELD: string[1];
)
```

Figure 1-521 Create J11_REFMATCH job 22/25



Data Set Management

Show Data Window

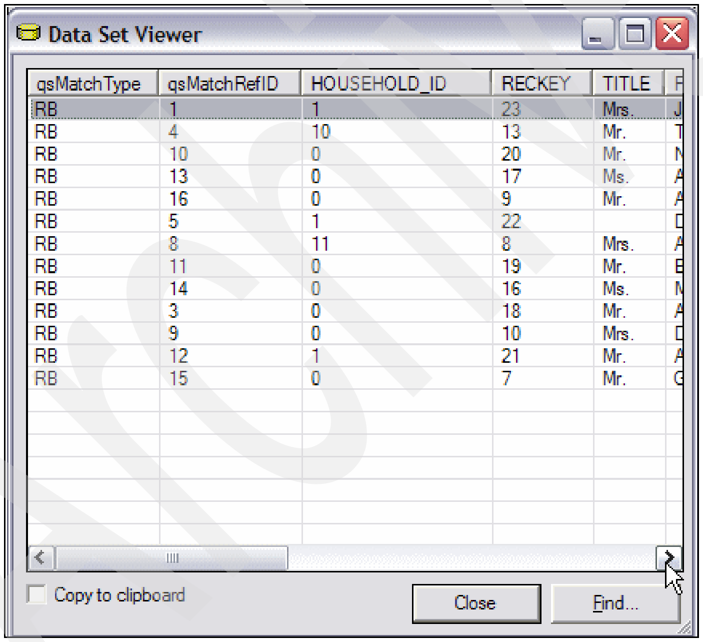
Data Set

Name:	REFMATCH_CUSTOMER_RESIDUALS.ds	Total Records:	13
Path:	/data/PART01	Total 32K Blocks:	3
Version:	ORCHESTRATE V8.0.1 DM Block Format 6.	Total Bytes:	3432
Created:	11/30/2007 06:03:10		

Partitions:

#	Node	Records	Blocks	Bytes
0	node1	5	1	1320
1	node2	4	1	1056
2	node3	4	1	1056

Figure 1-522 Create J11_REFMATCH job 23/25



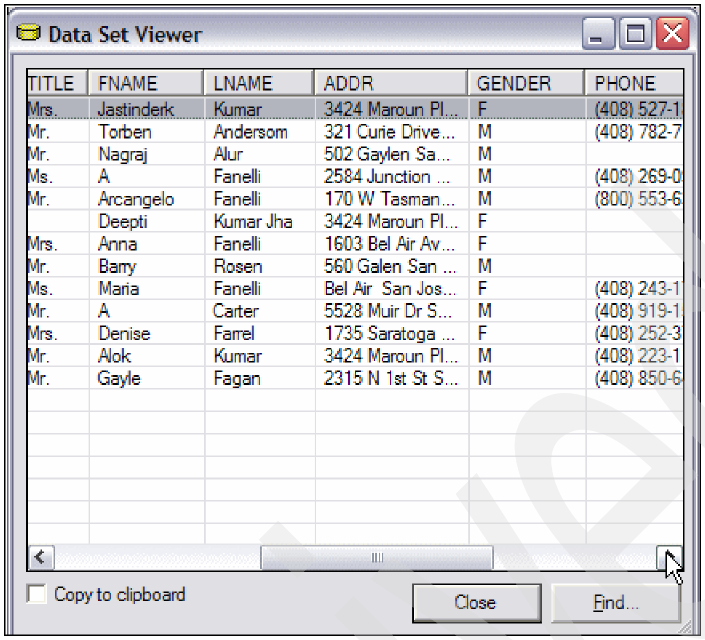
Data Set Viewer

qsMatchType	qsMatchRefID	HOUSEHOLD_ID	RECKEY	TITLE	F
RB	1	1	23	Mrs.	J
RB	4	10	13	Mr.	T
RB	10	0	20	Mr.	N
RB	13	0	17	Ms.	A
RB	16	0	9	Mr.	A
RB	5	1	22		D
RB	8	11	8	Mrs.	A
RB	11	0	19	Mr.	E
RB	14	0	16	Ms.	M
RB	3	0	18	Mr.	A
RB	9	0	10	Mrs.	D
RB	12	1	21	Mr.	A
RB	15	0	7	Mr.	G

Copy to clipboard

Close Find...

Figure 1-523 Create J11_REFMATCH job 24/25



TITLE	FNAME	LNAME	ADDR	GENDER	PHONE
Mrs.	Jastinderk	Kumar	3424 Maroun Pl...	F	(408) 527-1
Mr.	Torben	Andersom	321 Curie Drive...	M	(408) 782-7
Mr.	Nagraj	Alur	502 Gaylen Sa...	M	
Ms.	A	Fanelli	2584 Junction ...	M	(408) 269-0
Mr.	Arcangelo	Fanelli	170 W Tasman...	M	(800) 553-6
	Deepti	Kumar Jha	3424 Maroun Pl...	F	
Mrs.	Anna	Fanelli	1603 Bel Air Av...	F	
Mr.	Barry	Rosen	560 Galen San ...	M	
Ms.	Maria	Fanelli	Bel Air San Jos...	F	(408) 243-1
Mr.	A	Carter	5528 Muir Dr S...	M	(408) 919-1
Mrs.	Denise	Farrel	1735 Saratoga ...	F	(408) 252-3
Mr.	Alok	Kumar	3424 Maroun Pl...	M	(408) 223-1
Mr.	Gayle	Fagan	2315 N 1st St S...	M	(408) 850-6

Figure 1-524 Create J11_REFMATCH job 25/25

J12_CLERICAL_REPORT_MAILING_LIST

In this step, we create a single report of the matched mailing list persons and the records in the clerical review records to enable a manual review of potential match records.

The format of the CLERICAL data set that was created in "J11_REFMATCH" on page 416 has the two records for clerical review that are concatenated together, which makes reviewing them difficult. In this step, we use a Transformer stage to split the single concatenated record into two records that we then merge using a Funnel stage so that they can be viewed more conveniently for manual review of potential match records.

Figure 1-525 on page 432 through Figure 1-535 on page 437 describe the main steps using Designer Client to perform this task.

The main steps are as follows:

1. Figure 1-525 on page 432 shows the various stages that are used in this job, including the data set that was created in "J11_REFMATCH" on page 416, a Transformer stage, and a Funnel stage. One data set is created that contains the merged clerical review records organized for convenient manual review. We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object that was created earlier.

2. Configure the input CLERICAL data set. This is similar to earlier configurations, and we do not repeat the process here.
3. Configure the SPLIT_REC Transformer stage to split each record in the input data set into two records that correspond to:
   - The record in the mailing list file
   - The credit card customer

   Figure 1-526 on page 433 shows the SPLIT_REC - Transformer Stage window that maps the mailing list portion of the input record to the MATCHED_MAILING_OUT output link with an added column with the literal `Mailing`. Figure 1-527 on page 434 shows the SPLIT_REC - Transformer Stage window that maps the credit card customer portion of the input record to the MATCHED_CUSTOMER_OUT output link with an added column with the literal "CUSTOMER". The two literals distinguish the source of the two records to be reviewed.
4. Configure the Funnel stage to merge the two inputs from the SPLIT_REC Transformer stage output links. Because the configuration of the Funnel stage is described earlier, we do not repeat it here. Figure 1-528 on page 434 through Figure 1-531 on page 436 show the configuration of this stage with the input and output links.
5. Configure the output data set object CLERICAL_REVIEW_REPORT to store the results of the merge. This is similar to earlier configurations, and we do not repeat this process here.
6. After saving, compiling, and running this job j12_CLERICAL_REPORT_MAILING_LIST (Figure 1-532 on page 436), you view the content of the output data set object CLERICAL_REVIEW_REPORT as shown in Figure 1-533 on page 437 through Figure 1-535 on page 437.

   This report shows four clerical review records (qsMatchType has value `CP`)—two pairs of mailing and credit card customer records. This method of organizing the records makes it more convenient for manual review. The report shows that the two pairs are actual duplicates.
7. Save the data set metadata to Table Definitions. We do not repeat the process here.

After clerical review is complete, we enhance the information in the credit card customer with information in the matched mailing list record as described in the step "J13_ENHANCE_CUSTOMER" on page 437.

Note: We first created a complete set of matched records by merging the matched records created in "J11_REFMATCH" on page 416 with the clerical reviewed matches that were found to be matching. We do not show this process here.

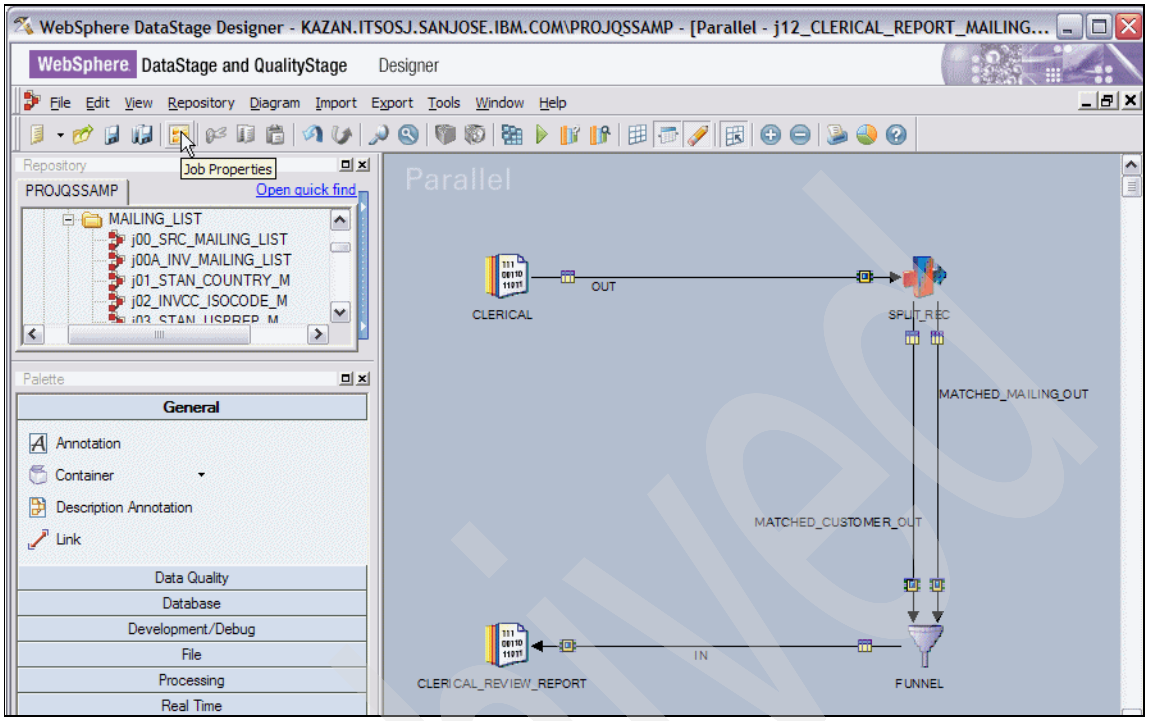
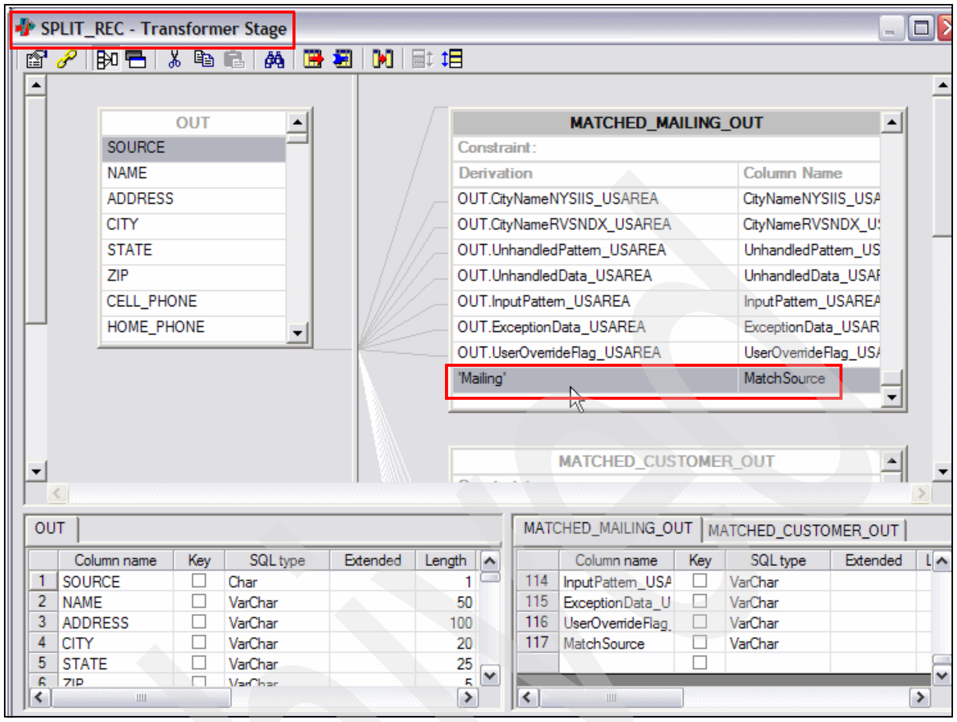


Figure 1-525 Create J12_CLERICAL_REPORT_MAILING_LIST 1/11


SPLIT_REC - Transformer Stage

OUT
SOURCE
NAME
ADDRESS
CITY
STATE
ZIP
CELL_PHONE
HOME_PHONE

MATCHED_MAILING_OUT

Constraint:

Derivation	Column Name
OUT.CityNameNYSIIS_USAREA	CityNameNYSIIS_USA
OUT.CityNameRVSNDX_USAREA	CityNameRVSNDX_US
OUT.UnhandledPattern_USAREA	UnhandledPattern_US
OUT.UnhandledData_USAREA	UnhandledData_USAR
OUT.InputPattern_USAREA	InputPattern_USAREA
OUT.ExceptionData_USAREA	ExceptionData_USAR
OUT.UserOverrideFlag_USAREA	UserOverrideFlag_USA
'Mailing'	MatchSource

MATCHED_CUSTOMER_OUT

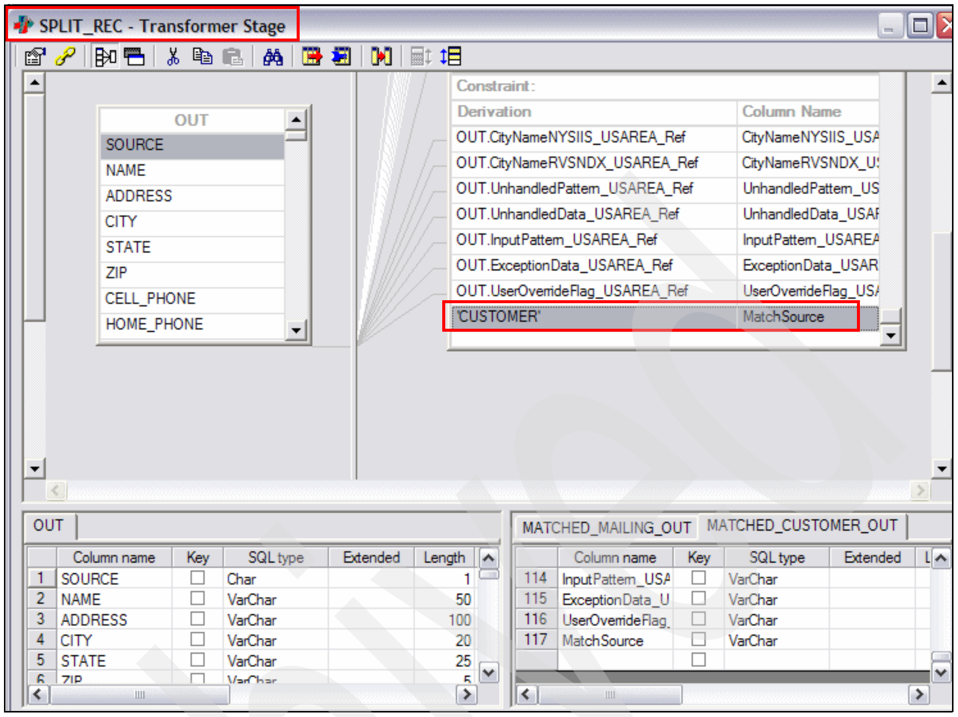
OUT

	Column name	Key	SQL type	Extended	Length
1	SOURCE	☐	Char		1
2	NAME	☐	VarChar		50
3	ADDRESS	☐	VarChar		100
4	CITY	☐	VarChar		20
5	STATE	☐	VarChar		25
6	ZIP	☐	VarChar		5

MATCHED_MAILING_OUT | MATCHED_CUSTOMER_OUT

	Column name	Key	SQL type	Extended
114	InputPattern_USA	☐	VarChar	
115	ExceptionData_U	☐	VarChar	
116	UserOverrideFlag_	☐	VarChar	
117	MatchSource	☐	VarChar	
		☐		

Figure 1-526   Create J12_CLERICAL_REPORT_MAILING_LIST 2/11


SPLIT_REC - Transformer Stage

OUT
- SOURCE
- NAME
- ADDRESS
- CITY
- STATE
- ZIP
- CELL_PHONE
- HOME_PHONE

Constraint:

Derivation	Column Name
OUT.CityNameNYSIIS_USAREA_Ref	CityNameNYSIIS_USA
OUT.CityNameRVSNDX_USAREA_Ref	CityNameRVSNDX_U:
OUT.UnhandledPattern_USAREA_Ref	UnhandledPattern_US
OUT.UnhandledData_USAREA_Ref	UnhandledData_USAI
OUT.InputPattern_USAREA_Ref	InputPattern_USAREA
OUT.ExceptionData_USAREA_Ref	ExceptionData_USAR
OUT.UserOverrideFlag_USAREA_Ref	UserOverrideFlag_US/
'CUSTOMER'	MatchSource

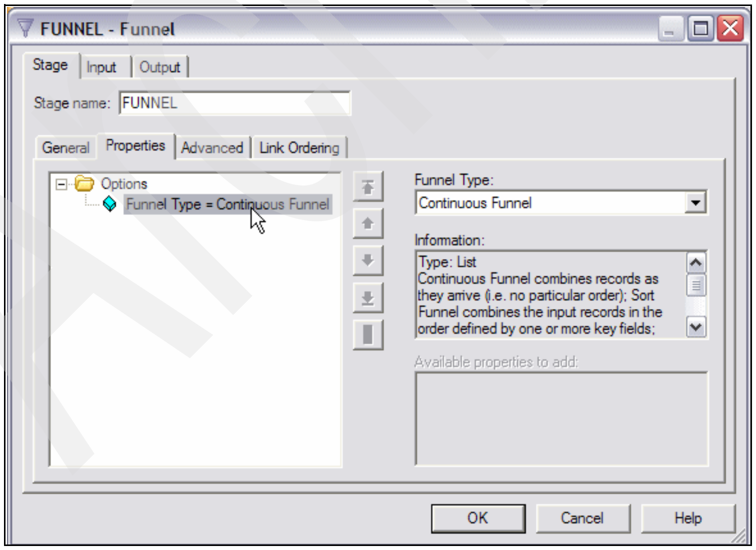
OUT

	Column name	Key	SQL type	Extended	Length
1	SOURCE	☐	Char		1
2	NAME	☐	VarChar		50
3	ADDRESS	☐	VarChar		100
4	CITY	☐	VarChar		20
5	STATE	☐	VarChar		25
6	ZIP	☐	VarChar		5

MATCHED_MAILING_OUT | MATCHED_CUSTOMER_OUT

	Column name	Key	SQL type	Extended
114	InputPattern_USA	☐	VarChar	
115	ExceptionData_U	☐	VarChar	
116	UserOverrideFlag_	☐	VarChar	
117	MatchSource	☐	VarChar	

Figure 1-527 Create J12_CLERICAL_REPORT_MAILING_LIST 3/11


FUNNEL - Funnel

Stage | Input | Output

Stage name: FUNNEL

General | Properties | Advanced | Link Ordering

- Options
  - Funnel Type = Continuous Funnel

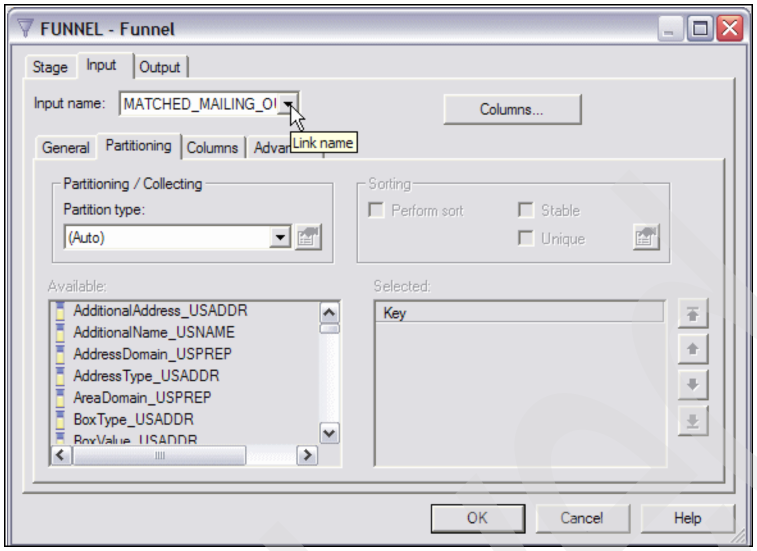
Funnel Type: Continuous Funnel

Information:
Type: List
Continuous Funnel combines records as they arrive (i.e. no particular order); Sort Funnel combines the input records in the order defined by one or more key fields;

Available properties to add:

OK | Cancel | Help

Figure 1-528 Create J12_CLERICAL_REPORT_MAILING_LIST 4/11


FUNNEL - Funnel

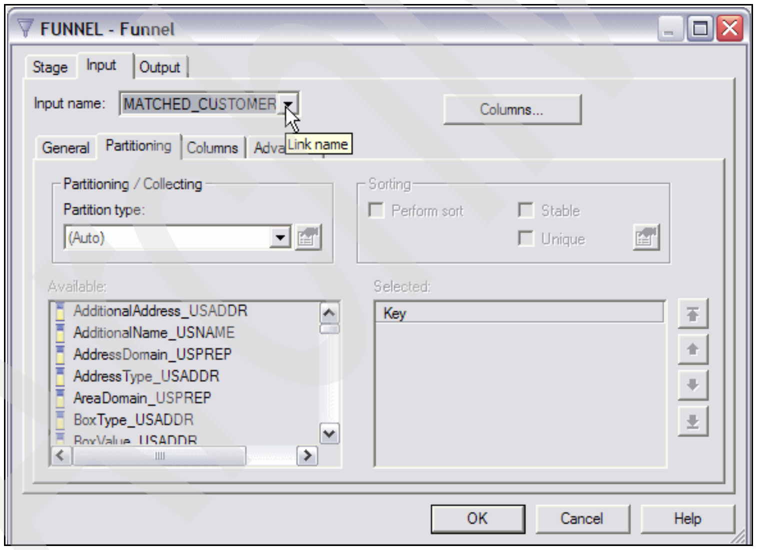
Input name: MATCHED_MAILING_OI

Partition type: (Auto)

Available:
AdditionalAddress_USADDR
AdditionalName_USNAME
AddressDomain_USPREP
AddressType_USADDR
AreaDomain_USPREP
BoxType_USADDR
BoxValue_USADDR

Selected:
Key

Figure 1-529 Create J12_CLERICAL_REPORT_MAILING_LIST 5/11


FUNNEL - Funnel

Input name: MATCHED_CUSTOMER

Partition type: (Auto)

Available:
AdditionalAddress_USADDR
AdditionalName_USNAME
AddressDomain_USPREP
AddressType_USADDR
AreaDomain_USPREP
BoxType_USADDR
BoxValue_USADDR

Selected:
Key

Figure 1-530 Create J12_CLERICAL_REPORT_MAILING_LIST 6/11

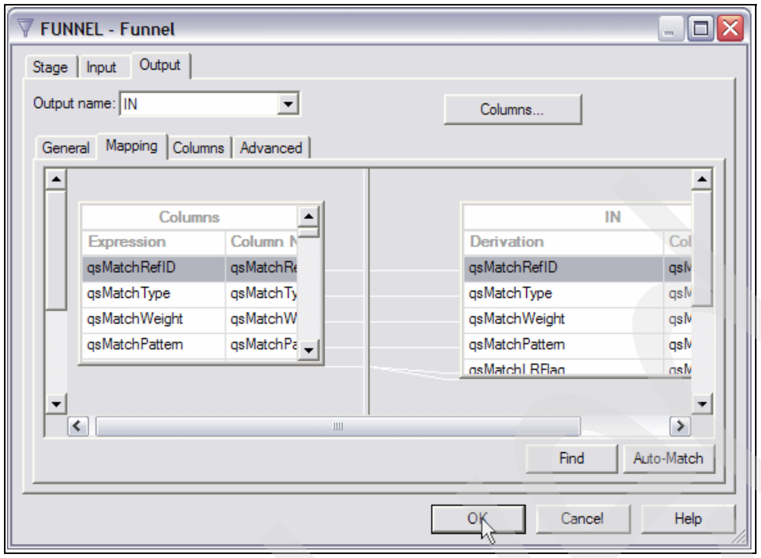


Figure 1-531 Create J12_CLERICAL_REPORT_MAILING_LIST 7/11

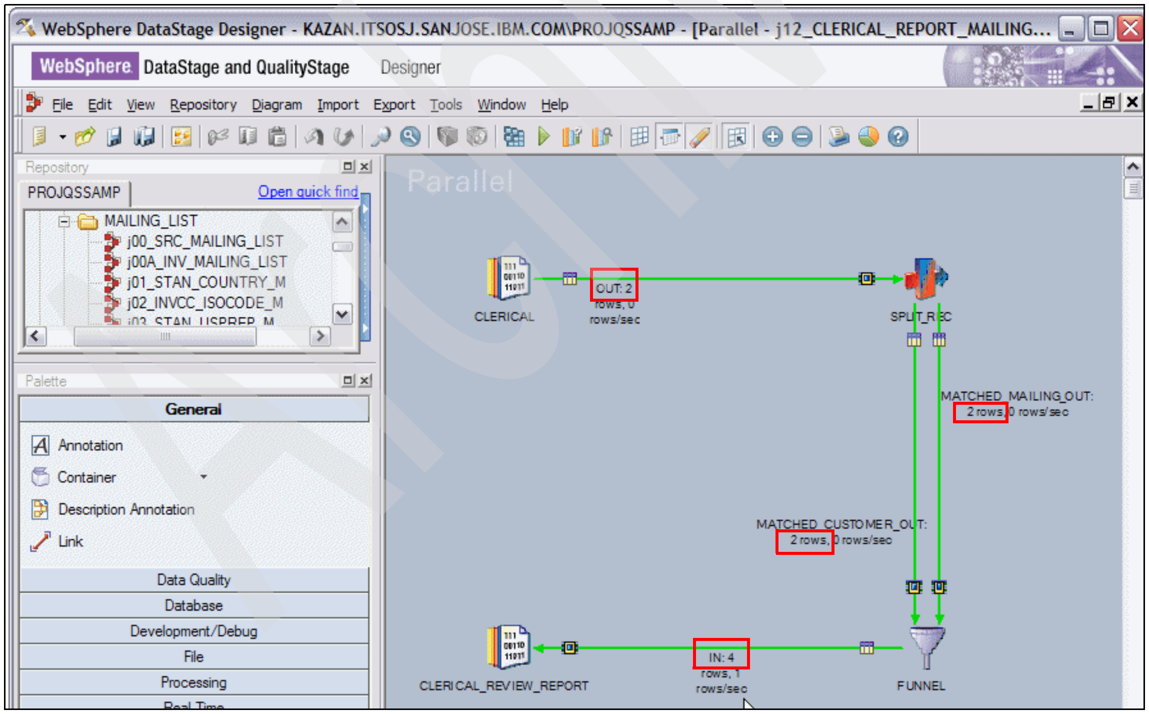
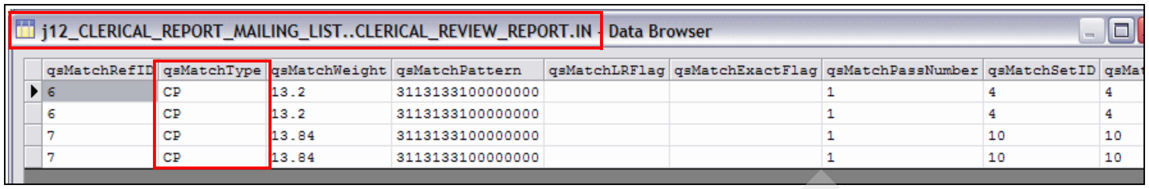
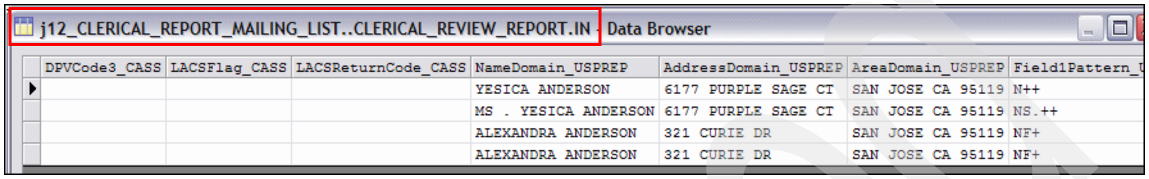


Figure 1-532 Create J12_CLERICAL_REPORT_MAILING_LIST 8/11


j12_CLERICAL_REPORT_MAILING_LIST..CLERICAL_REVIEW_REPORT.IN - Data Browser

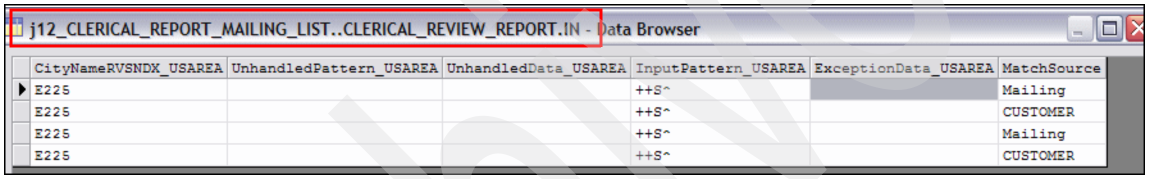
qsMatchRefID	qsMatchType	qsMatchWeight	qsMatchPattern	qsMatchLRFlag	qsMatchExactFlag	qsMatchPassNumber	qsMatchSetID	qsMat
6	CP	13.2	3113133100000000			1	4	4
6	CP	13.2	3113133100000000			1	4	4
7	CP	13.84	3113133100000000			1	10	10
7	CP	13.84	3113133100000000			1	10	10

Figure 1-533 Create J12_CLERICAL_REPORT_MAILING_LIST 9/11


j12_CLERICAL_REPORT_MAILING_LIST..CLERICAL_REVIEW_REPORT.IN - Data Browser

DPVCode3_CASS	LACSFlag_CASS	LACSReturnCode_CASS	NameDomain_USPREP	AddressDomain_USPREP	AreaDomain_USPREP	Field1Pattern_U
			YESICA ANDERSON	6177 PURPLE SAGE CT	SAN JOSE CA 95119	N++
			MS . YESICA ANDERSON	6177 PURPLE SAGE CT	SAN JOSE CA 95119	NS.++
			ALEXANDRA ANDERSON	321 CURIE DR	SAN JOSE CA 95119	NF+
			ALEXANDRA ANDERSON	321 CURIE DR	SAN JOSE CA 95119	NF+

Figure 1-534 Create J12_CLERICAL_REPORT_MAILING_LIST 10/11


j12_CLERICAL_REPORT_MAILING_LIST..CLERICAL_REVIEW_REPORT.IN - Data Browser

CityNameRVSNDX_USAREA	UnhandledPattern_USAREA	UnhandledData_USAREA	InputPattern_USAREA	ExceptionData_USAREA	MatchSource
E225			++S^		Mailing
E225			++S^		CUSTOMER
E225			++S^		Mailing
E225			++S^		CUSTOMER

Figure 1-535 Create J12_CLERICAL_REPORT_MAILING_LIST 11/11

1.10.7 Enhance credit card customers

A manual review of the CLERICAL_REVIEW_REPORT finds duplicates that need to be considered as part of the file that contains matched records by merging them using a Funnel stage. We do not show this process here.

In this step, we enhance the information in the credit card customer file with middle name information in the mailing list portion of the matched record if it had a longer middle name. We use a Transformer stage to enhance the middle name in the credit card customer file.

J13_ENHANCE_CUSTOMER

Figure 1-536 on page 439 through Figure 1-542 on page 442 describe the main steps using Designer Client to perform this task.

The main steps are as follows:

1. Figure 1-536 on page 439 shows the ENHANCE_CUSTOMER Transformer stage used in this job, including the MATCHED data set that was created in

"J11_REFMATCH" on page 416. One data set is created that contains the enhanced credit card customer information. We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object created earlier.

2. Configure the input MATCHED data set. This process is similar to earlier configurations, and we do not repeat this process here.
3. Configure the ENHANCE_CUSTOMER Transformer stage to update the middle name information in the credit card customer portion with that of the middle name in the mailing list portion if appropriate. Figure 1-537 on page 440 shows the ENHANCE_CUSTOMER - Transformer Stage window where the Derivation for the MiddleName_USNAME_REF column (corresponding to the middle name in the credit card customer portion) has logic to perform the update as follows:

   ```
   if (Len(OUT.MiddleName_USNAME) > Len (OUT.MiddleName_USNAME_Ref))
   then OUT.MiddleName_USNAME else OUT.MiddleName_USNAME_Ref
   ```

4. Configure the output data set object ENHANCED_CUSTOMER to store the enhanced credit card customer information. This process is similar to earlier configurations, and we do not repeat this process here.
5. After saving, compiling, and running this job j13_ENHANCE_CUSTOMER (Figure 1-538 on page 441), view the content of the output data set object ENHANCED_CUSTOMER as shown in Figure 1-538 on page 441 through Figure 1-542 on page 442.

   This report shows eight records—the same as the number of input records to the ENHANCE_CUSTOMER Transformer stage.

   Note: To determine the records that were enhanced, you need to compare the input and output data sets. We do not show this process here.

6. Save the data set metadata to Table Definitions. We do not repeat this process here.

After you enhance credit card customer information, you generate match frequency information for the residual mailing list persons (those not matched with a person in the credit card customer file) as described in the step "J14_MAILING_LIST_RESIDUAL_FREQ" on page 442.

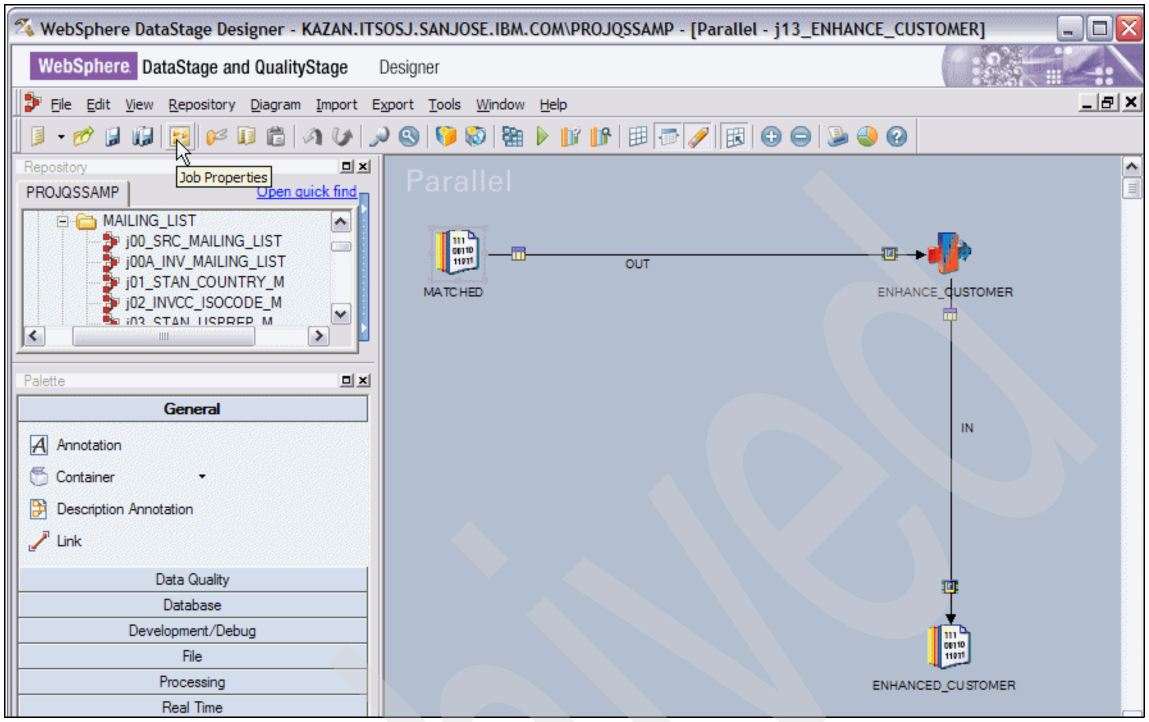
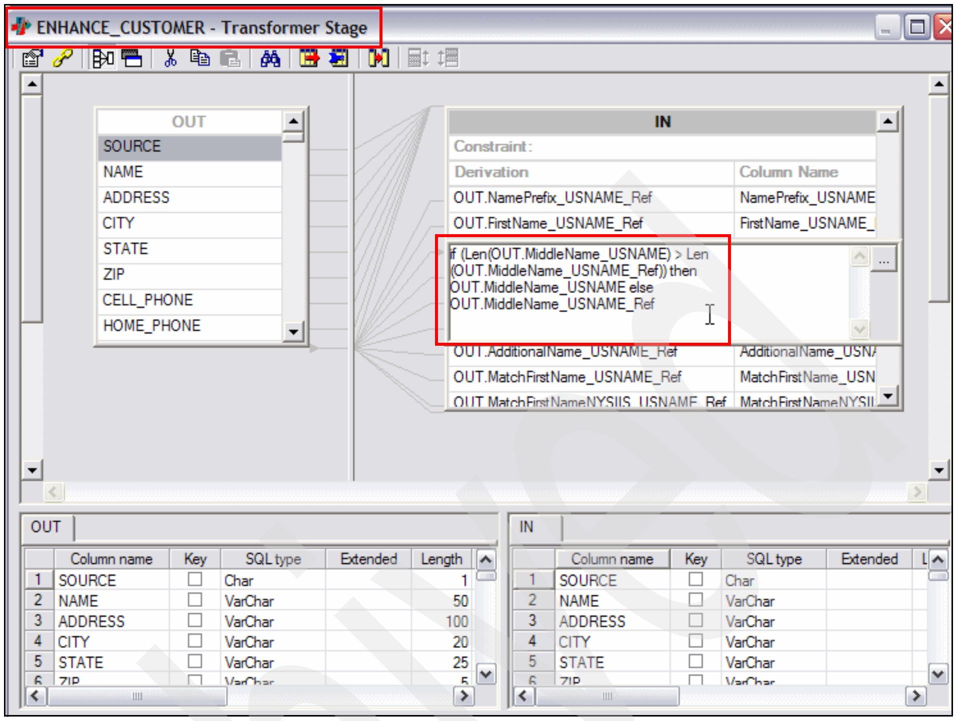


Figure 1-536 Create J13_ENHANCE_CUSTOMER 1/7


ENHANCE_CUSTOMER - Transformer Stage

OUT
SOURCE
NAME
ADDRESS
CITY
STATE
ZIP
CELL_PHONE
HOME_PHONE

IN

Constraint:

Derivation	Column Name
OUT.NamePrefix_USNAME_Ref	NamePrefix_USNAME
OUT.FirstName_USNAME_Ref	FirstName_USNAME_
if (Len(OUT.MiddleName_USNAME) > Len (OUT.MiddleName_USNAME_Ref)) then OUT.MiddleName_USNAME else OUT.MiddleName_USNAME_Ref	
OUT.AdditionalName_USNAME_Ref	AdditionalName_USNA
OUT.MatchFirstName_USNAME_Ref	MatchFirstName_USN
OUT.MatchFirstNameNYSIIS_USNAME_Ref	MatchFirstNameNYSII

OUT

	Column name	Key	SQL type	Extended	Length
1	SOURCE	☐	Char		1
2	NAME	☐	VarChar		50
3	ADDRESS	☐	VarChar		100
4	CITY	☐	VarChar		20
5	STATE	☐	VarChar		25
6	ZIP	☐	VarChar		5

IN

	Column name	Key	SQL type	Extended
1	SOURCE	☐	Char	
2	NAME	☐	VarChar	
3	ADDRESS	☐	VarChar	
4	CITY	☐	VarChar	
5	STATE	☐	VarChar	
6	ZIP	☐	VarChar	

Figure 1-537 Create J13_ENHANCE_CUSTOMER 2/7

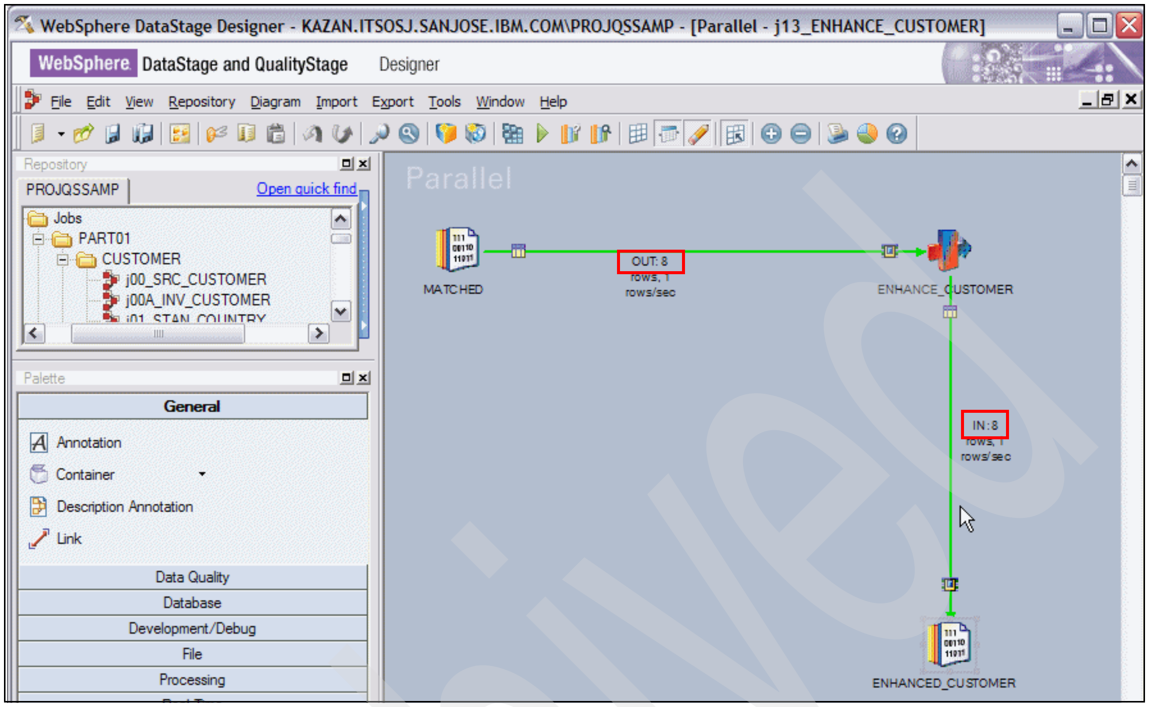
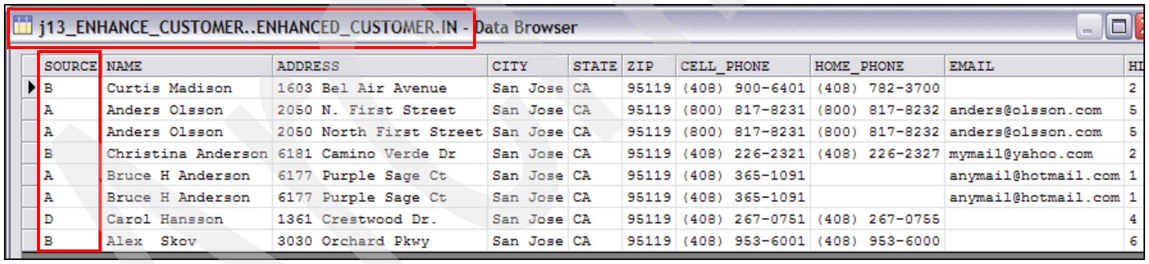
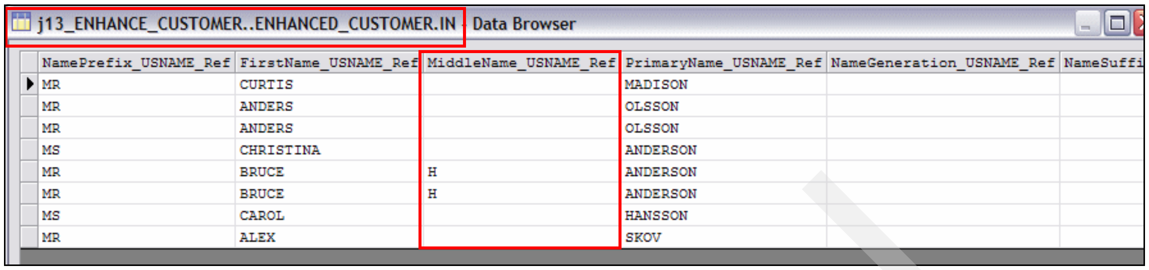


Figure 1-538 Create J13_ENHANCE_CUSTOMER 3/7


j13_ENHANCE_CUSTOMER..ENHANCED_CUSTOMER.IN - Data Browser

SOURCE	NAME	ADDRESS	CITY	STATE	ZIP	CELL_PHONE	HOME_PHONE	EMAIL	HI
B	Curtis Madison	1603 Bel Air Avenue	San Jose	CA	95119	(408) 900-6401	(408) 782-3700		2
A	Anders Olsson	2050 N. First Street	San Jose	CA	95119	(800) 817-8231	(800) 817-8232	anders@olsson.com	5
A	Anders Olsson	2050 North First Street	San Jose	CA	95119	(800) 817-8231	(800) 817-8232	anders@olsson.com	5
B	Christina Anderson	6181 Camino Verde Dr	San Jose	CA	95119	(408) 226-2321	(408) 226-2327	mymail@yahoo.com	2
A	Bruce H Anderson	6177 Purple Sage Ct	San Jose	CA	95119	(408) 365-1091		anymail@hotmail.com	1
A	Bruce H Anderson	6177 Purple Sage Ct	San Jose	CA	95119	(408) 365-1091		anymail@hotmail.com	1
D	Carol Hansson	1361 Crestwood Dr.	San Jose	CA	95119	(408) 267-0751	(408) 267-0755		4
B	Alex Skov	3030 Orchard Pkwy	San Jose	CA	95119	(408) 953-6001	(408) 953-6000		6

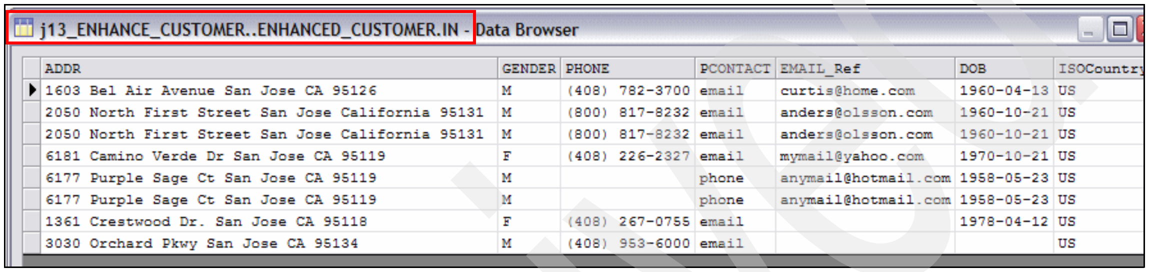
Figure 1-539 Create J13_ENHANCE_CUSTOMER 4/7



j13_ENHANCE_CUSTOMER..ENHANCED_CUSTOMER.IN - Data Browser

NamePrefix_USNAME_Ref	FirstName_USNAME_Ref	MiddleName_USNAME_Ref	PrimaryName_USNAME_Ref	NameGeneration_USNAME_Ref	NameSuffi
MR	CURTIS		MADISON		
MR	ANDERS		OLSSON		
MR	ANDERS		OLSSON		
MS	CHRISTINA		ANDERSON		
MR	BRUCE	H	ANDERSON		
MR	BRUCE	H	ANDERSON		
MS	CAROL		HANSSON		
MR	ALEX		SKOV		

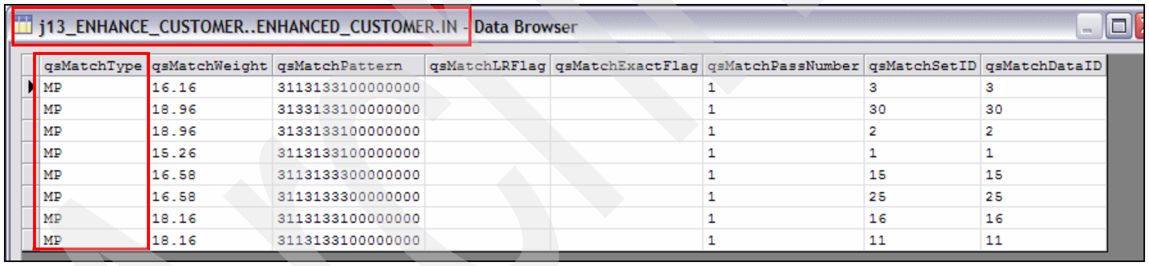
Figure 1-540 Create J13_ENHANCE_CUSTOMER 5/7



j13_ENHANCE_CUSTOMER..ENHANCED_CUSTOMER.IN - Data Browser

ADDR	GENDER	PHONE	PCONTACT	EMAIL_Ref	DOB	ISOCountry
1603 Bel Air Avenue San Jose CA 95126	M	(408) 782-3700	email	curtis@home.com	1960-04-13	US
2050 North First Street San Jose California 95131	M	(800) 817-8232	email	anders@olsson.com	1960-10-21	US
2050 North First Street San Jose California 95131	M	(800) 817-8232	email	anders@olsson.com	1960-10-21	US
6181 Camino Verde Dr San Jose CA 95119	F	(408) 226-2327	email	mymail@yahoo.com	1970-10-21	US
6177 Purple Sage Ct San Jose CA 95119	M		phone	anymail@hotmail.com	1958-05-23	US
6177 Purple Sage Ct San Jose CA 95119	M		phone	anymail@hotmail.com	1958-05-23	US
1361 Crestwood Dr. San Jose CA 95118	F	(408) 267-0755	email		1978-04-12	US
3030 Orchard Pkwy San Jose CA 95134	M	(408) 953-6000	email			US

Figure 1-541 Create J13_ENHANCE_CUSTOMER 6/7



j13_ENHANCE_CUSTOMER..ENHANCED_CUSTOMER.IN - Data Browser

qsMatchType	qsMatchWeight	qsMatchPattern	qsMatchLRFlag	qsMatchExactFlag	qsMatchPassNumber	qsMatchSetID	qsMatchDataID
MP	16.16	3113133100000000			1	3	3
MP	18.96	3133133100000000			1	30	30
MP	18.96	3133133100000000			1	2	2
MP	15.26	3113133100000000			1	1	1
MP	16.58	3113133300000000			1	15	15
MP	16.58	3113133300000000			1	25	25
MP	18.16	3113133100000000			1	16	16
MP	18.16	3113133100000000			1	11	11

Figure 1-542 Create J13_ENHANCE_CUSTOMER 7/7

J14_MAILING_LIST_RESIDUAL_FREQ

In this step, we generate frequency distribution on *all* the columns in the residual mailing list file that was created in the "J11_REFMATCH" on page 416 step using the Match Frequency stage. Here again, the idea is to generate match frequency for all the columns so that it can be used with any match specification.

Note: If the clerical review records that are generated in "J11_REFMATCH" on page 416 are deemed to be residuals, then you need to merge them with the residuals. We do not show this process here.

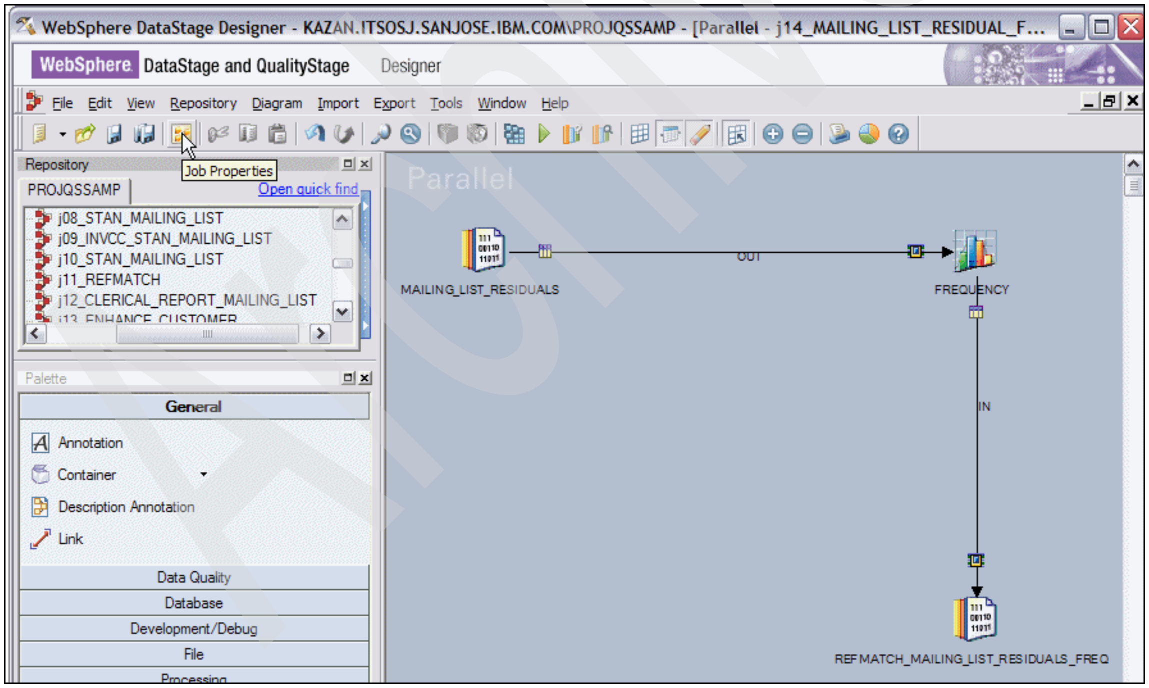
Figure 1-543 through Figure 1-546 on page 445 describe the main steps using Designer Client to perform this task.

Figure 1-543 on page 443 shows the various stages that are used in this job, including the data set that was created in “J11_REFMATCH” on page 416, a Match Frequency stage, and an output Data Set stage. We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object created earlier.

Because the configuration of this job is very similar to that described in “J10_MATCHFREQ_STAN_CUSTOMER” on page 266, we do not repeat it here. However, some of the configurations of interest are as follows:

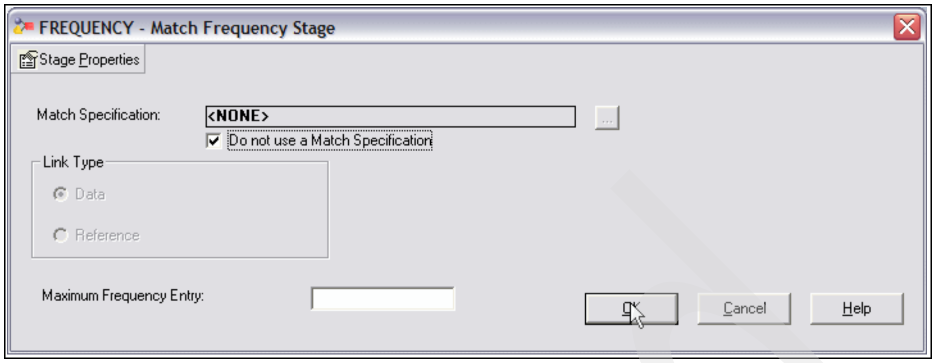
- Figure 1-544 on page 444 shows the FREQUENCY - Match Frequency Stage window with Match Specification of NONE.
- After saving, compiling, and running this job (Figure 1-545 on page 444), the contents of the output of this stage are listed in Figure 1-546 on page 445. The interpretation of the format and content of this file is not documented.

Now, proceed to “J14_UNDUP_DEP_MATCHSPEC_MAILING” on page 445.



WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j14_MAILING_LIST_RESIDUAL_F...
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Job Properties
Repository
PROJQSSAMP
Open quick find
j08_STAN_MAILING_LIST
j09_INVCC_STAN_MAILING_LIST
j10_STAN_MAILING_LIST
j11_REFMATCH
j12_CLERICAL_REPORT_MAILING_LIST
j13_ENHANCE_CUSTOMER
Palette
General
Annotation
Container
Description Annotation
Link
Data Quality
Database
Development/Debug
File
Processing
Parallel
MAILING_LIST_RESIDUALS
OUT
FREQUENCY
IN
REFMATCH_MAILING_LIST_RESIDUALS_FREQ

Figure 1-543 Create J14_MAILING_LIST_RESIDUAL_FREQ 1/4


FREQUENCY - Match Frequency Stage

Stage Properties

Match Specification: <NONE>

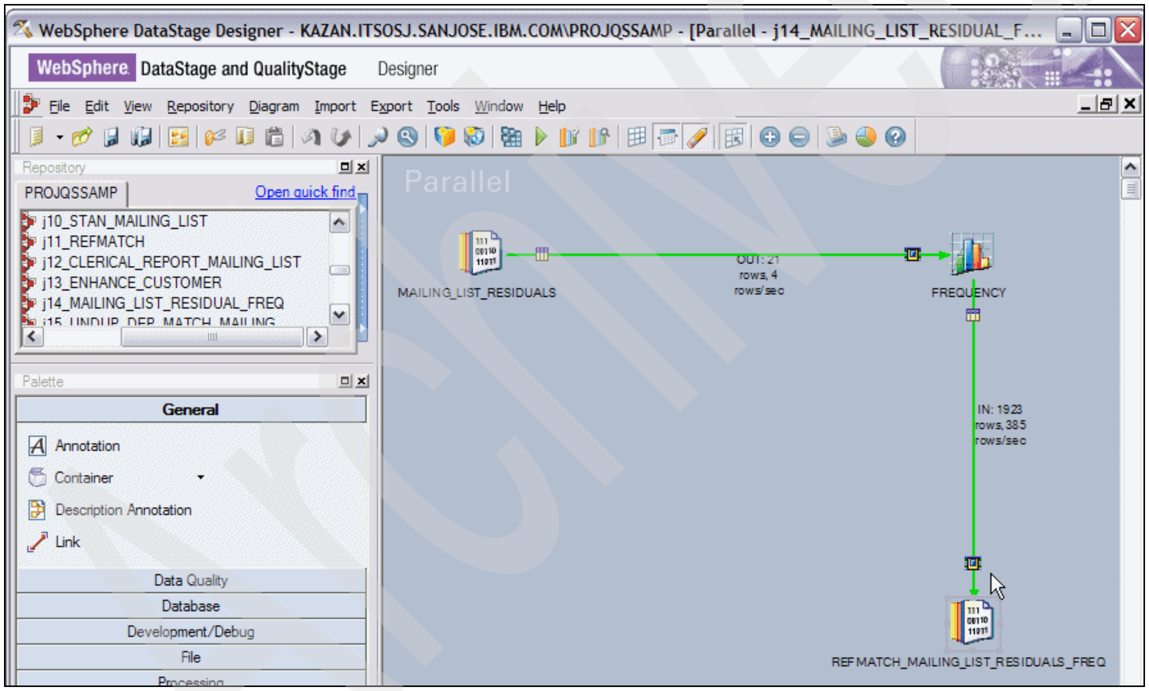
[✓] Do not use a Match Specification

Link Type

[✓ Data] Reference

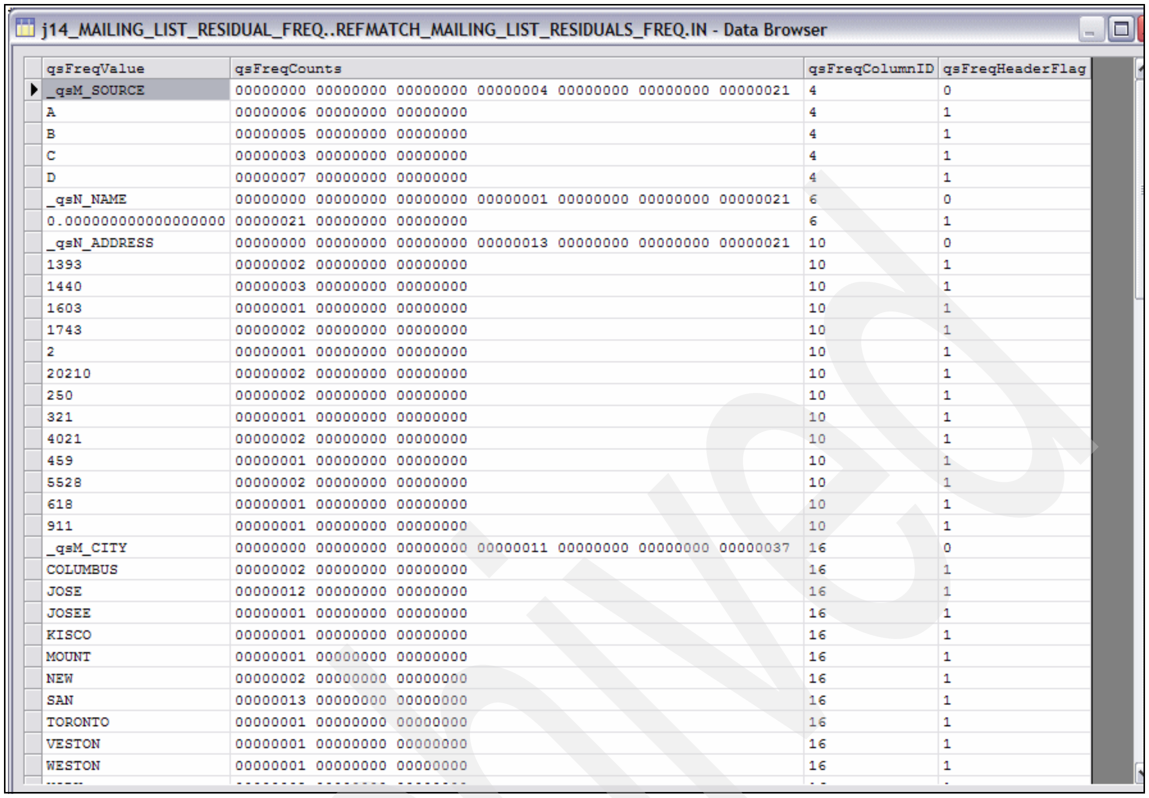
Maximum Frequency Entry:

Figure 1-544   Create J14_MAILING_LIST_RESIDUAL_FREQ 2/4


WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j14_MAILING_LIST_RESIDUAL_F...

WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Repository
PROJQSSAMP
Open quick find
j10_STAN_MAILING_LIST
j11_REFMATCH
j12_CLERICAL_REPORT_MAILING_LIST
j13_ENHANCE_CUSTOMER
j14_MAILING_LIST_RESIDUAL_FREQ
j15_UNDUP_DEP_MATCH_MAILING
Palette
General
Annotation
Container
Description Annotation
Link
Data Quality
Database
Development/Debug
File
Processing
Parallel
MAILING_LIST_RESIDUALS
OUT: 21 rows, 4 rows/sec
FREQUENCY
IN: 1923 rows, 385 rows/sec
REFMATCH_MAILING_LIST_RESIDUALS_FREQ

Figure 1-545   Create J14_MAILING_LIST_RESIDUAL_FREQ 3/4


j14_MAILING_LIST_RESIDUAL_FREQ..REFMATCH_MAILING_LIST_RESIDUALS_FREQ.IN - Data Browser

qsFreqValue	qsFreqCounts	qsFreqColumnID	qsFreqHeaderFlag
_qsM_SOURCE	00000000 00000000 00000000 00000004 00000000 00000000 00000021	4	0
A	00000006 00000000 00000000	4	1
B	00000005 00000000 00000000	4	1
C	00000003 00000000 00000000	4	1
D	00000007 00000000 00000000	4	1
_qsN_NAME	00000000 00000000 00000000 00000001 00000000 00000000 00000021	6	0
0.000000000000000000	00000021 00000000 00000000	6	1
_qsN_ADDRESS	00000000 00000000 00000000 00000013 00000000 00000000 00000021	10	0
1393	00000002 00000000 00000000	10	1
1440	00000003 00000000 00000000	10	1
1603	00000001 00000000 00000000	10	1
1743	00000002 00000000 00000000	10	1
2	00000001 00000000 00000000	10	1
20210	00000002 00000000 00000000	10	1
250	00000002 00000000 00000000	10	1
321	00000001 00000000 00000000	10	1
4021	00000002 00000000 00000000	10	1
459	00000001 00000000 00000000	10	1
5528	00000002 00000000 00000000	10	1
618	00000001 00000000 00000000	10	1
911	00000001 00000000 00000000	10	1
_qsM_CITY	00000000 00000000 00000000 00000011 00000000 00000000 00000037	16	0
COLUMBUS	00000002 00000000 00000000	16	1
JOSE	00000012 00000000 00000000	16	1
JOSEE	00000001 00000000 00000000	16	1
KISCO	00000001 00000000 00000000	16	1
MOUNT	00000001 00000000 00000000	16	1
NEW	00000002 00000000 00000000	16	1
SAN	00000013 00000000 00000000	16	1
TORONTO	00000001 00000000 00000000	16	1
VESTON	00000001 00000000 00000000	16	1
WESTON	00000001 00000000 00000000	16	1

Figure 1-546 Create J14_MAILING_LIST_RESIDUAL_FREQ 4/4

J14_UNDUP_DEP_MATCHSPEC_MAILING

In this step, we generate a match specification for an Unduplicate match stage using as input the match frequency data that was created in the job "J14_MAILING_LIST_RESIDUAL_FREQ" on page 442. The specification included two passes:

- The first pass blocks on a phonetic encoding (NYSIIS) of the primary (last) name, the address, phonetic encoding (NYSIIS) of the street name, and the ZIP code.
- The second pass blocks on the three digit ZIP code.

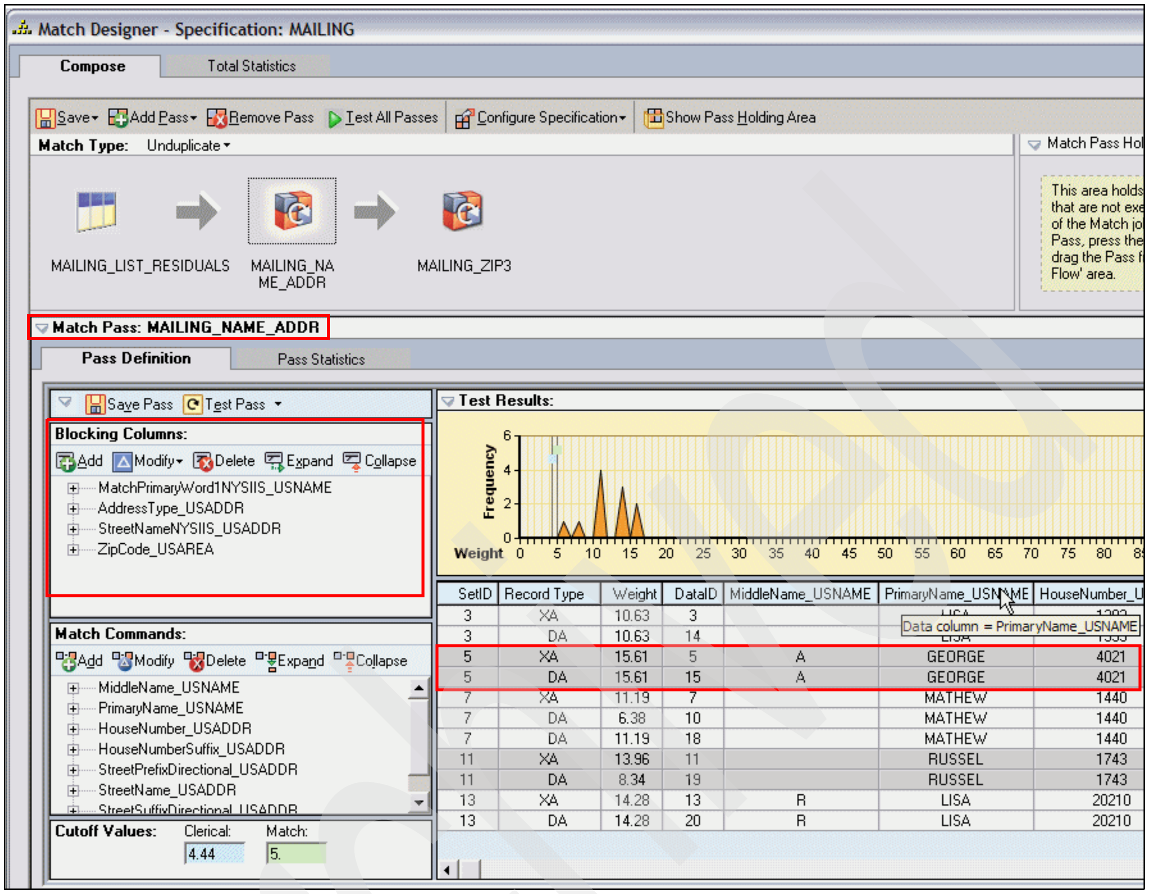
Because the creation of this match specification is very similar to that described in "J10_Undup_MatchSpec_STAN_CUSTOMER" on page 269, we do not repeat the steps here.

A summary of the passes is follows:

- MAILING_NAME_ADDR pass (Figure 1-547 on page 447) with the blocking columns, match commands, cutoff values, and test results. The results show multiple master records (Record Type “XA”) and duplicates (Record Type `DA`).
- MAILING_ZIP3 pass (Figure 1-548 on page 448) with the blocking columns, match commands, cutoff values, and test results. The results show a single master record (Record Type `XA`) and duplicate (Record Type `DA`).
- Total Statistics of the two passes in Figure 1-549 on page 449. It shows the total number of Pseudo matches (5+1 for a total of 6).

  Of particular interest is the statistic OVERFLOW blocks—a non-zero value indicates the need to increase the block size or define more restrictive blocking columns.

The test of the MAILING match specification with representative sample data appears to deliver results that are accurate. This specification can then be used in an Unduplicate match stage as described in the step “J15_UNDUP_DEP_MATCH_MAILING” on page 449.



Match Type: Unduplicate

This area holds that are not exe of the Match jo Pass, press the drag the Pass f Flow' area.

Match Pass: MAILING_NAME_ADDR

Blocking Columns:

- MatchPrimaryWord1NYSIIS_USNAME
- AddressType_USADDR
- StreetNameNYSIIS_USADDR
- ZipCode_USAREA

Match Commands:

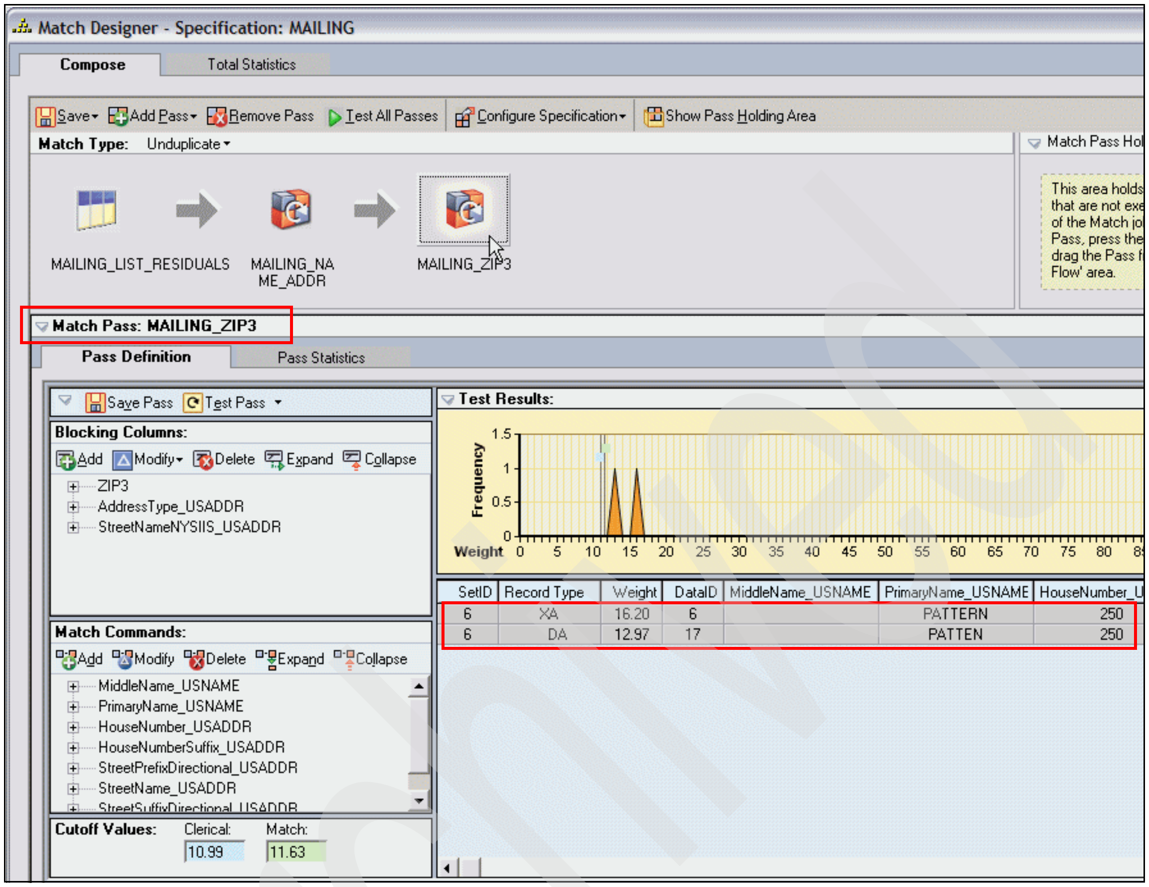
- MiddleName_USNAME
- PrimaryName_USNAME
- HouseNumber_USADDR
- HouseNumberSuffix_USADDR
- StreetPrefixDirectional_USADDR
- StreetName_USADDR
- StreetSuffixDirectional_USADDR

Cutoff Values: Clerical: 4.44 Match: 5.

SetID	Record Type	Weight	DataID	MiddleName_USNAME	PrimaryName_USNAME	HouseNumber_U
3	XA	10.63	3		LISA	[illegible]
3	DA	10.63	14		LISA	[illegible]
5	XA	15.61	5	A	GEORGE	4021
5	DA	15.61	15	A	GEORGE	4021
7	XA	11.19	7		MATHEW	1440
7	DA	6.38	10		MATHEW	1440
7	DA	11.19	18		MATHEW	1440
11	XA	13.96	11		RUSSEL	1743
11	DA	8.34	19		RUSSEL	1743
13	XA	14.28	13	R	LISA	20210
13	DA	14.28	20	R	LISA	20210

Data column = PrimaryName_USNAME

Figure 1-547 Create J14_UNDUP_DEP_MATCHSPEC_MAILING 1/3



This area holds
that are not exe
of the Match jo
Pass, press the
drag the Pass f
Flow' area.

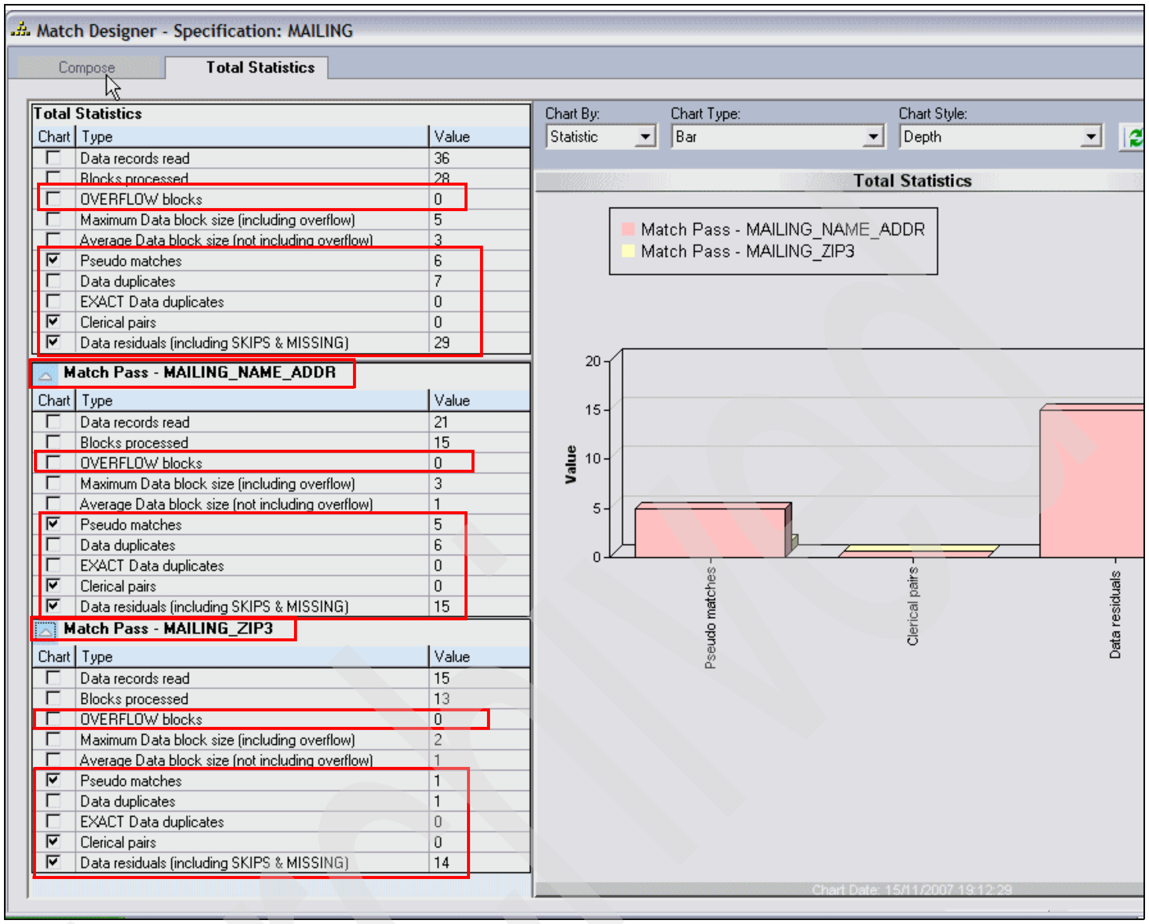
Blocking Columns:
- ZIP3
- AddressType_USADDR
- StreetNameNYSIIS_USADDR

Match Commands:
- MiddleName_USNAME
- PrimaryName_USNAME
- HouseNumber_USADDR
- HouseNumberSuffix_USADDR
- StreetPrefixDirectional_USADDR
- StreetName_USADDR
- StreetSuffixDirectional_USADDR

Cutoff Values: Clerical: 10.99 Match: 11.63

SetID	Record Type	Weight	DataID	MiddleName_USNAME	PrimaryName_USNAME	HouseNumber_U
6	XA	16.20	6		PATTERN	250
6	DA	12.97	17		PATTEN	250

Figure 1-548 Create J14_UNDUP_DEP_MATCHSPEC_MAILING 2/3



Total Statistics

Chart	Type	Value
☐	Data records read	36
☐	Blocks processed	28
☐	OVERFLOW blocks	0
☐	Maximum Data block size (including overflow)	5
☐	Average Data block size (not including overflow)	3
☑	Pseudo matches	6
☐	Data duplicates	7
☐	EXACT Data duplicates	0
☑	Clerical pairs	0
☑	Data residuals (including SKIPS & MISSING)	29

Match Pass - MAILING_NAME_ADDR

Chart	Type	Value
☐	Data records read	21
☐	Blocks processed	15
☐	OVERFLOW blocks	0
☐	Maximum Data block size (including overflow)	3
☐	Average Data block size (not including overflow)	1
☑	Pseudo matches	5
☐	Data duplicates	6
☐	EXACT Data duplicates	0
☑	Clerical pairs	0
☑	Data residuals (including SKIPS & MISSING)	15

Match Pass - MAILING_ZIP3

Chart	Type	Value
☐	Data records read	15
☐	Blocks processed	13
☐	OVERFLOW blocks	0
☐	Maximum Data block size (including overflow)	2
☐	Average Data block size (not including overflow)	1
☑	Pseudo matches	1
☐	Data duplicates	1
☐	EXACT Data duplicates	0
☑	Clerical pairs	0
☑	Data residuals (including SKIPS & MISSING)	14

Figure 1-549 Create J14_UNDUP_DEP_MATCHSPEC_MAILING 3/3

J15_UNDUP_DEP_MATCH_MAILING

Next, we determine whether there are duplicates in the mailing list persons file using the Unduplicate stage with the match specification and match frequency information that was created in steps "J14_UNDUP_DEP_MATCHSPEC_MAILING" on page 445 and "J14_MAILING_LIST_RESIDUAL_FREQ" on page 442 respectively. As before, the output is matched records (merge of master and duplicates using a Funnel stage), records for clerical review, and residuals (records that do not match).

Figure 1-550 on page 451 through Figure 1-558 on page 454 describe the main steps using Designer Client to perform this task.

Figure 1-550 on page 451 shows the various stages that are used in this job, including the data set that was created in “J11_REFMATCH” on page 416, the match frequency data set that was created in “J10_MATCHFREQ_STAN_CUSTOMER” on page 266, an Unduplicate stage, and a Funnel stage to merge master and duplicate records. Three data sets are created:

- One data set contains the merged master and duplicates by the Funnel stage.
- The other two data sets contain the clerical and residual records as the output of the Unduplicate stage.

We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object created earlier.

Because the creation of this Unduplicate stage Dependent Match Type is very similar to that described in “J11_UNDUP_DEP_MATCH_CUSTOMER” on page 282, we do not repeat the steps that are involved here. However, some of the configurations of interest are as follows:

- Figure 1-551 on page 451 shows the UNDUP_DEP - Unduplicate Match Stage window with all the Match Outputs checked, the Match Specification of MAILING, and the Match Type of Dependent.
- After saving, compiling, and running this job j15_UNDUP_DEP_MATCH_MAILING (Figure 1-552 on page 452), view the content of the three data set objects as shown in Figure 1-553 on page 452 through Figure 1-558 on page 454.

  Out of a total of 21 records in the input to this process:

  – Thirteen (13) records are masters and duplicates. There are not in sorted order of qsMatchSetId.

  – Zero (0) records are for clerical review.

  – Remaining eight (8) records are residuals.

Because there are no records for clerical review, you next create a clean set of master and duplicate records and survive the best information into the master record using the SURVIVE stage as described in the step “J16_SURVIVE_MAILING” on page 454.

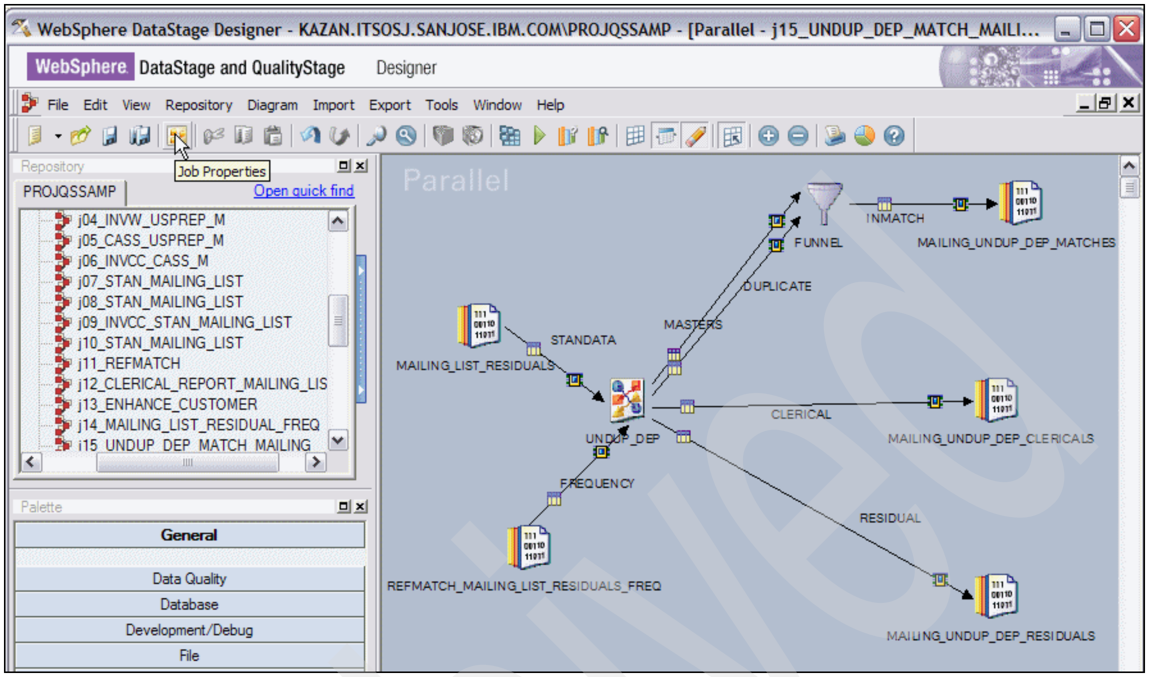
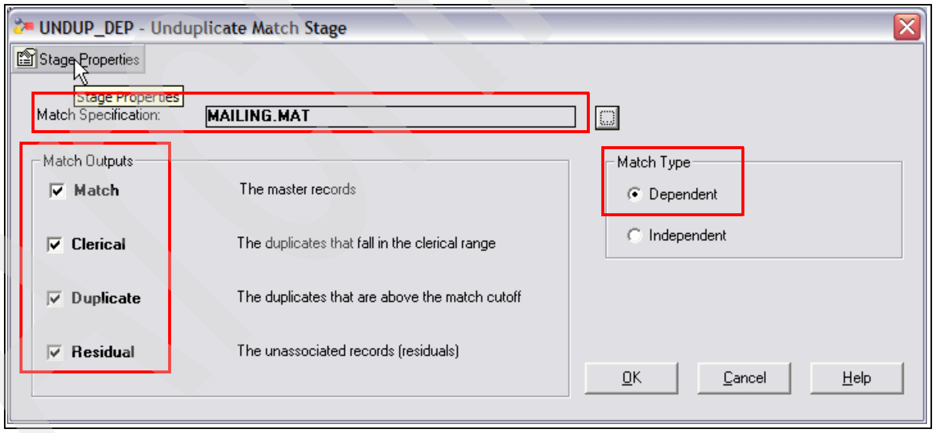

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j15_UNDUP_DEP_MATCH_MAILI...
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Job Properties
Repository
PROJQSSAMP
Open quick find
j04_INVW_USPREP_M
j05_CASS_USPREP_M
j06_INVCC_CASS_M
j07_STAN_MAILING_LIST
j08_STAN_MAILING_LIST
j09_INVCC_STAN_MAILING_LIST
j10_STAN_MAILING_LIST
j11_REFMATCH
j12_CLERICAL_REPORT_MAILING_LIS
j13_ENHANCE_CUSTOMER
j14_MAILING_LIST_RESIDUAL_FREQ
j15_UNDUP_DEP_MATCH_MAILING
Palette
General
Data Quality
Database
Development/Debug
File
Parallel
FUNNEL
INMATCH
MAILING_UNDUP_DEP_MATCHES
DUPLICATE
MASTERS
STANDATA
MAILING_LIST_RESIDUALS
UNDUP_DEP
CLERICAL
MAILING_UNDUP_DEP_CLERICALS
FREQUENCY
RESIDUAL
REFMATCH_MAILING_LIST_RESIDUALS_FREQ
MAILING_UNDUP_DEP_RESIDUALS

Figure 1-550 Create J15_UNDUP_DEP_MATCH_MAILING 1/9



Match Specification: MAILING.MAT

Match Outputs

☑ Match	The master records
☑ Clerical	The duplicates that fall in the clerical range
☑ Duplicate	The duplicates that are above the match cutoff
☑ Residual	The unassociated records (residuals)

Match Type: [✓ Dependent] Independent

Figure 1-551 Create J15_UNDUP_DEP_MATCH_MAILING 2/9

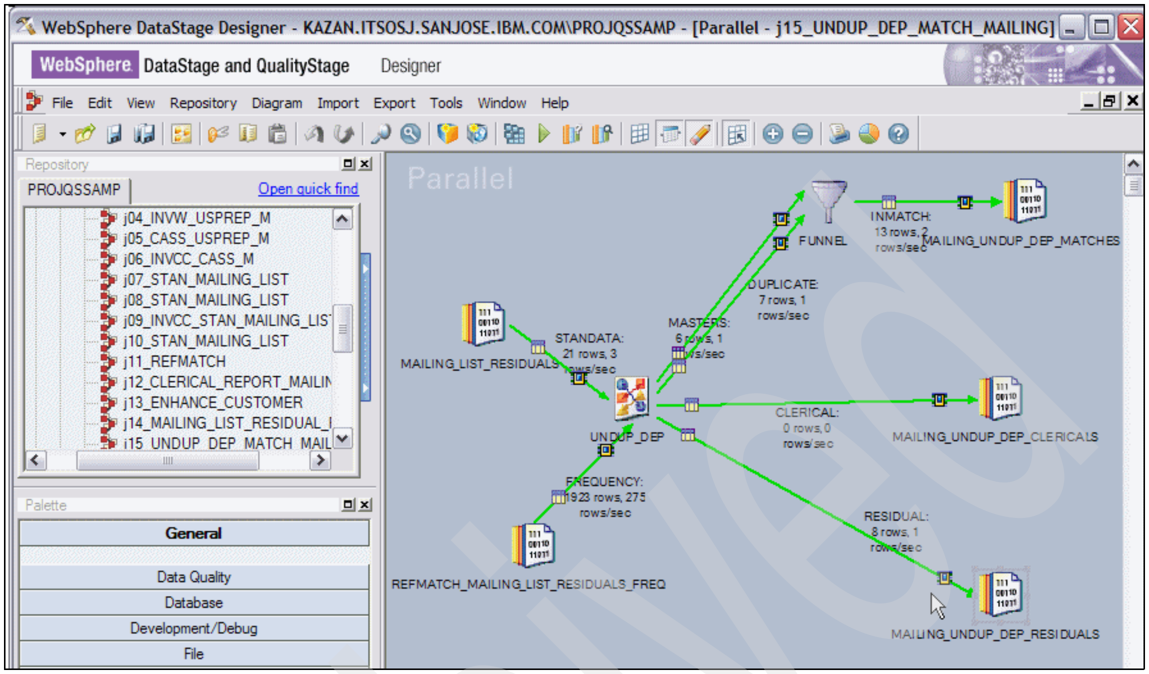
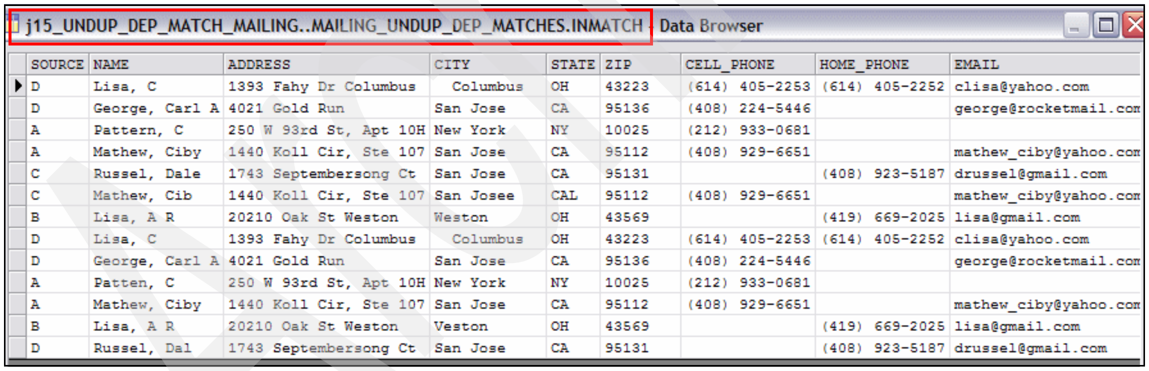
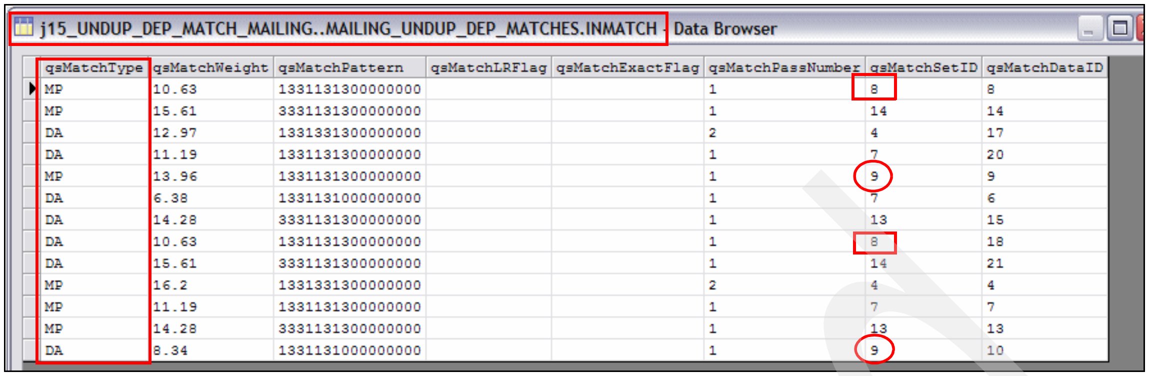

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j15_UNDUP_DEP_MATCH_MAILING]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Repository
PROJQSSAMP
Open quick find
j04_INVW_USPREP_M
j05_CASS_USPREP_M
j06_INVCC_CASS_M
j07_STAN_MAILING_LIST
j08_STAN_MAILING_LIST
j09_INVCC_STAN_MAILING_LIST
j10_STAN_MAILING_LIST
j11_REFMATCH
j12_CLERICAL_REPORT_MAILIN
j13_ENHANCE_CUSTOMER
j14_MAILING_LIST_RESIDUAL_I
j15_UNDUP_DEP_MATCH_MAIL
Palette
General
Data Quality
Database
Development/Debug
File
Parallel
MAILING_LIST_RESIDUALS
STANDATA: 21 rows, 3 rows/sec
UNDUP_DEP
FREQUENCY: 1923 rows, 275 rows/sec
REFMATCH_MAILING_LIST_RESIDUALS_FREQ
MASTERS: 6 rows, 1 rows/sec
DUPLICATE: 7 rows, 1 rows/sec
FUNNEL
INMATCH: 13 rows, 2 rows/sec
MAILING_UNDUP_DEP_MATCHES
CLERICAL: 0 rows, 0 rows/sec
MAILING_UNDUP_DEP_CLERICALS
RESIDUAL: 8 rows, 1 rows/sec
MAILING_UNDUP_DEP_RESIDUALS

Figure 1-552 Create J15_UNDUP_DEP_MATCH_MAILING 3/9


j15_UNDUP_DEP_MATCH_MAILING..MAILING_UNDUP_DEP_MATCHES.INMATCH - Data Browser

SOURCE	NAME	ADDRESS	CITY	STATE	ZIP	CELL_PHONE	HOME_PHONE	EMAIL
D	Lisa, C	1393 Fahy Dr Columbus	Columbus	OH	43223	(614) 405-2253	(614) 405-2252	clisa@yahoo.com
D	George, Carl A	4021 Gold Run	San Jose	CA	95136	(408) 224-5446		george@rocketmail.com
A	Pattern, C	250 W 93rd St, Apt 10H	New York	NY	10025	(212) 933-0681		
A	Mathew, Ciby	1440 Koll Cir, Ste 107	San Jose	CA	95112	(408) 929-6651		mathew_ciby@yahoo.com
C	Russel, Dale	1743 Septembersong Ct	San Jose	CA	95131		(408) 923-5187	drussel@gmail.com
C	Mathew, Cib	1440 Koll Cir, Ste 107	San Josee	CAL	95112	(408) 929-6651		mathew_ciby@yahoo.com
B	Lisa, A R	20210 Oak St Weston	Weston	OH	43569		(419) 669-2025	lisa@gmail.com
D	Lisa, C	1393 Fahy Dr Columbus	Columbus	OH	43223	(614) 405-2253	(614) 405-2252	clisa@yahoo.com
D	George, Carl A	4021 Gold Run	San Jose	CA	95136	(408) 224-5446		george@rocketmail.com
A	Patten, C	250 W 93rd St, Apt 10H	New York	NY	10025	(212) 933-0681		
A	Mathew, Ciby	1440 Koll Cir, Ste 107	San Jose	CA	95112	(408) 929-6651		mathew_ciby@yahoo.com
B	Lisa, A R	20210 Oak St Weston	Veston	OH	43569		(419) 669-2025	lisa@gmail.com
D	Russel, Dal	1743 Septembersong Ct	San Jose	CA	95131		(408) 923-5187	drussel@gmail.com

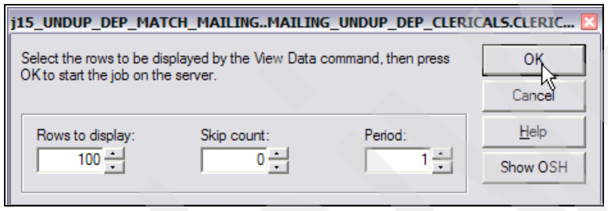
Figure 1-553 Create J15_UNDUP_DEP_MATCH_MAILING 4/9



j15_UNDUP_DEP_MATCH_MAILING..MAILING_UNDUP_DEP_MATCHES.INMATCH - Data Browser

qsMatchType	qsMatchWeight	qsMatchPattern	qsMatchLRFlag	qsMatchExactFlag	qsMatchPassNumber	qsMatchSetID	qsMatchDataID
MP	10.63	1331131300000000			1	8	8
MP	15.61	3331131300000000			1	14	14
DA	12.97	1331331300000000			2	4	17
DA	11.19	1331131300000000			1	7	20
MP	13.96	1331131300000000			1	9	9
DA	6.38	1331131000000000			1	7	6
DA	14.28	3331131300000000			1	13	15
DA	10.63	1331131300000000			1	8	18
DA	15.61	3331131300000000			1	14	21
MP	16.2	1331331300000000			2	4	4
MP	11.19	1331131300000000			1	7	7
MP	14.28	3331131300000000			1	13	13
DA	8.34	1331131000000000			1	9	10

Figure 1-554   Create J15_UNDUP_DEP_MATCH_MAILING 5/9

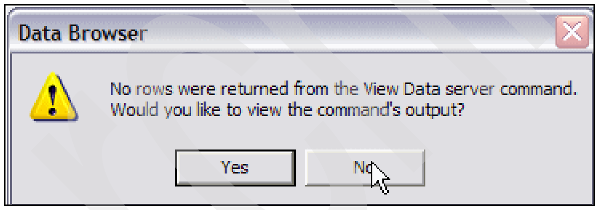


j15_UNDUP_DEP_MATCH_MAILING..MAILING_UNDUP_DEP_CLERICALS.CLERIC...

Select the rows to be displayed by the View Data command, then press OK to start the job on the server.

Rows to display: 100
Skip count: 0
Period: 1

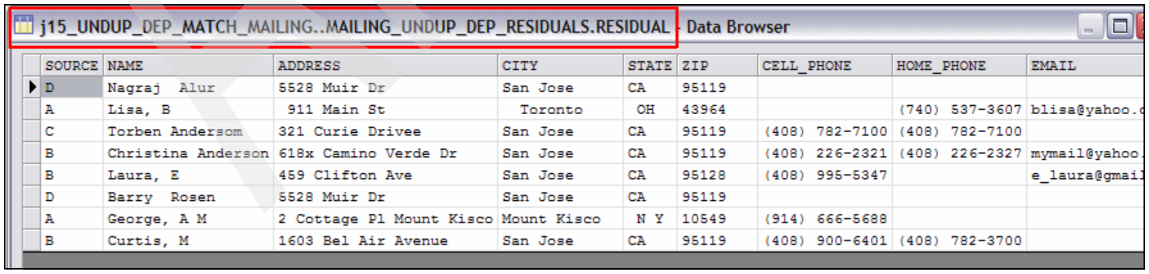
Figure 1-555   Create J15_UNDUP_DEP_MATCH_MAILING 6/9



Data Browser

No rows were returned from the View Data server command. Would you like to view the command's output?

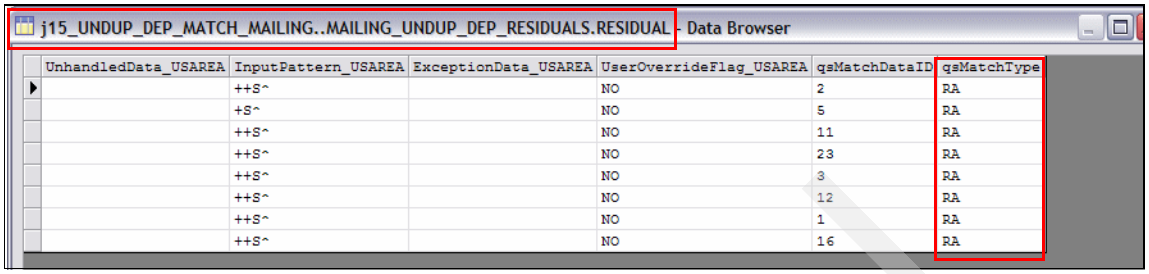
Figure 1-556   Create J15_UNDUP_DEP_MATCH_MAILING 7/9



j15_UNDUP_DEP_MATCH_MAILING..MAILING_UNDUP_DEP_RESIDUALS.RESIDUAL - Data Browser

SOURCE	NAME	ADDRESS	CITY	STATE	ZIP	CELL_PHONE	HOME_PHONE	EMAIL
D	Nagraj Alur	5528 Muir Dr	San Jose	CA	95119			
A	Lisa, B	911 Main St	Toronto	OH	43964		(740) 537-3607	blisa@yahoo.c
C	Torben Andersom	321 Curie Drivee	San Jose	CA	95119	(408) 782-7100	(408) 782-7100	
B	Christina Anderson	618x Camino Verde Dr	San Jose	CA	95119	(408) 226-2321	(408) 226-2327	mymail@yahoo.
B	Laura, E	459 Clifton Ave	San Jose	CA	95128	(408) 995-5347		e_laura@gmail
D	Barry Rosen	5528 Muir Dr	San Jose	CA	95119			
A	George, A M	2 Cottage Pl Mount Kisco	Mount Kisco	N Y	10549	(914) 666-5688		
B	Curtis, M	1603 Bel Air Avenue	San Jose	CA	95119	(408) 900-6401	(408) 782-3700	

Figure 1-557   Create J15_UNDUP_DEP_MATCH_MAILING 8/9



j15_UNDUP_DEP_MATCH_MAILING..MAILING_UNDUP_DEP_RESIDUALS.RESIDUAL - Data Browser

UnhandledData_USAREA	InputPattern_USAREA	ExceptionData_USAREA	UserOverrideFlag_USAREA	qsMatchDataID	qsMatchType
	++S^		NO	2	RA
	+S^		NO	5	RA
	++S^		NO	11	RA
	++S^		NO	23	RA
	++S^		NO	3	RA
	++S^		NO	12	RA
	++S^		NO	1	RA
	++S^		NO	16	RA

Figure 1-558 Create J15_UNDUP_DEP_MATCH_MAILING 9/9

J16_SURVIVE_MAILING

Because no duplicates are found for clerical review, in this step we survive the best information from the set of duplicates in the matched records that are created in the “J15_UNDUP_DEP_MATCH_MAILING” on page 449 step using the SURVIVE stage.

Figure 1-559 on page 455 through Figure 1-564 on page 457 describe the main steps using Designer Client to perform this task.

Figure 1-559 on page 455 shows the various stages that are used in this job, including the input data set MAILING_UNDUP_DEP_MATCHES that was created in “J15_UNDUP_DEP_MATCH_MAILING” on page 449 containing matched and duplicate records, a Survive stage, and an output data set containing the survived records. We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object created earlier.

Because the creation of this survive job is very similar to that described in “J13_SURVIVE_CUSTOMER” on page 295, we do not repeat the steps that are involved here. However, some of the configurations of interest are as follows:

- Figure 1-560 on page 456 and Figure 1-561 on page 456 show the SURVIVE - Survive Stage windows with the survive rules involving the first name, middle name, and last name for the target master record (qsMatchType column has a value `MP`). It specifies the following rules:
  - Choose the survivor for the MatchFirstName_USNAME to be the most frequent value of multiple occurrences.
  - Choose the survivor for the MiddleName_USNAME to be the longest value of multiple occurrences.
  - Choose the survivor for the PrimaryName_USNAME to be the longest value of multiple occurrences.
- After saving, compiling, and running this job j16_SURVIVE_MAILING (Figure 1-562 on page 457), you view the content of the output data set object

MAILING_SURVIVE_MP_DA as shown in Figure 1-563 on page 457 through Figure 1-564 on page 457.

This report shows the six master records with the survived information from the duplicates.

Now, proceed to “J17_MAILING_MASTER” on page 458.

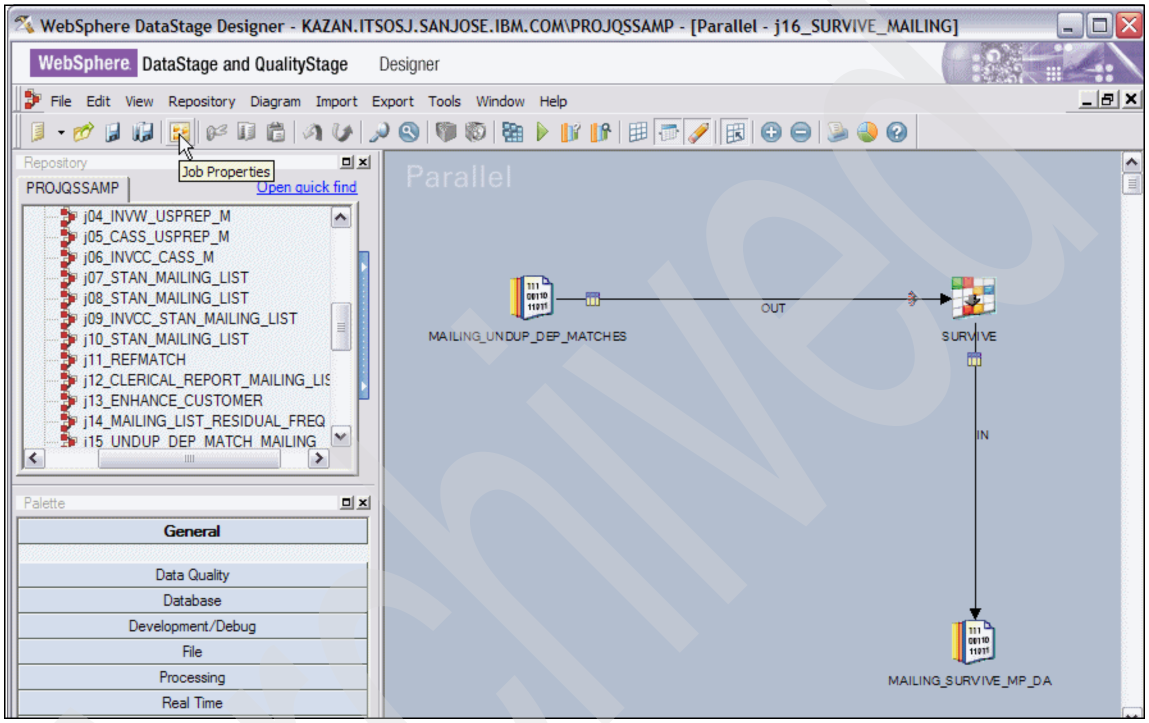
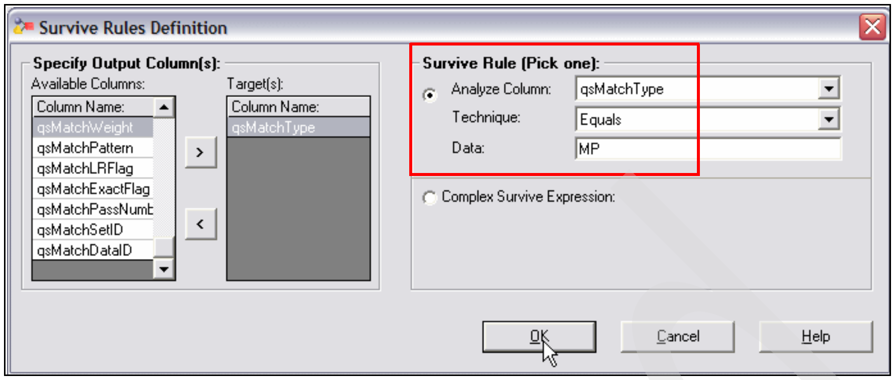

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j16_SURVIVE_MAILING]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Job Properties
Repository
PROJQSSAMP
Open quick find
j04_INVW_USPREP_M
j05_CASS_USPREP_M
j06_INVCC_CASS_M
j07_STAN_MAILING_LIST
j08_STAN_MAILING_LIST
j09_INVCC_STAN_MAILING_LIST
j10_STAN_MAILING_LIST
j11_REFMATCH
j12_CLERICAL_REPORT_MAILING_LIS
j13_ENHANCE_CUSTOMER
j14_MAILING_LIST_RESIDUAL_FREQ
j15_UNDUP_DEP_MATCH_MAILING
Palette
General
Data Quality
Database
Development/Debug
File
Processing
Real Time
Parallel
MAILING_UNDUP_DEP_MATCHES
OUT
SURVIVE
IN
MAILING_SURVIVE_MP_DA

Figure 1-559 Create J16_SURVIVE_MAILING 1/6


Survive Rules Definition

Specify Output Column(s):

Available Columns:

Column Name:
qsMatchWeight
qsMatchPattern
qsMatchLRFlag
qsMatchExactFlag
qsMatchPassNumb
qsMatchSetID
qsMatchDataID

Target(s):

Column Name:
qsMatchType

Survive Rule (Pick one):

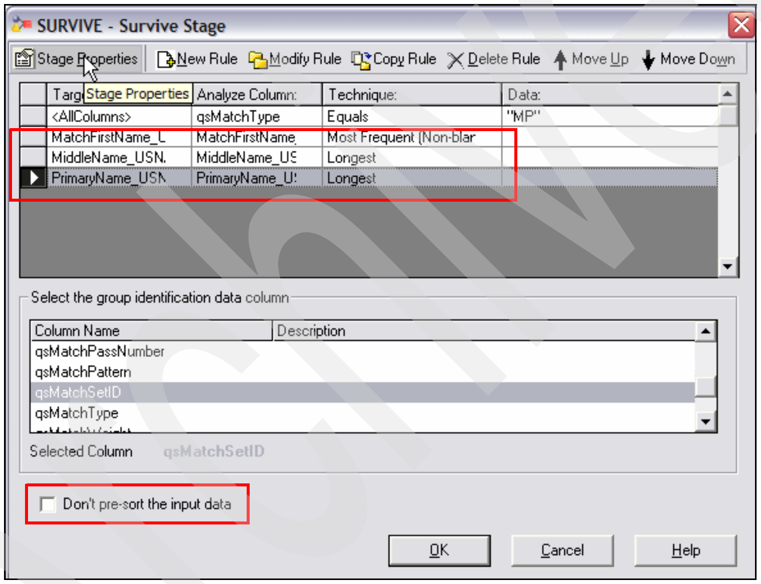
(●) Analyze Column: qsMatchType

Technique: Equals

Data: MP

( ) Complex Survive Expression:

Figure 1-560 Create J16_SURVIVE_MAILING 2/6


SURVIVE - Survive Stage

	Targ Stage Properties	Analyze Column:	Technique:	Data:
	<AllColumns>	qsMatchType	Equals	"MP"
	MatchFirstName_L	MatchFirstName	Most Frequent (Non-blar	
	MiddleName_USN.	MiddleName_US	Longest	
▶	PrimaryName_USN	PrimaryName_U:	Longest	

Select the group identification data column

Column Name	Description
qsMatchPassNumber	
qsMatchPattern	
qsMatchSetID	
qsMatchType	

Selected Column qsMatchSetID

☐ Don't pre-sort the input data

Figure 1-561 Create J16_SURVIVE_MAILING 3/6

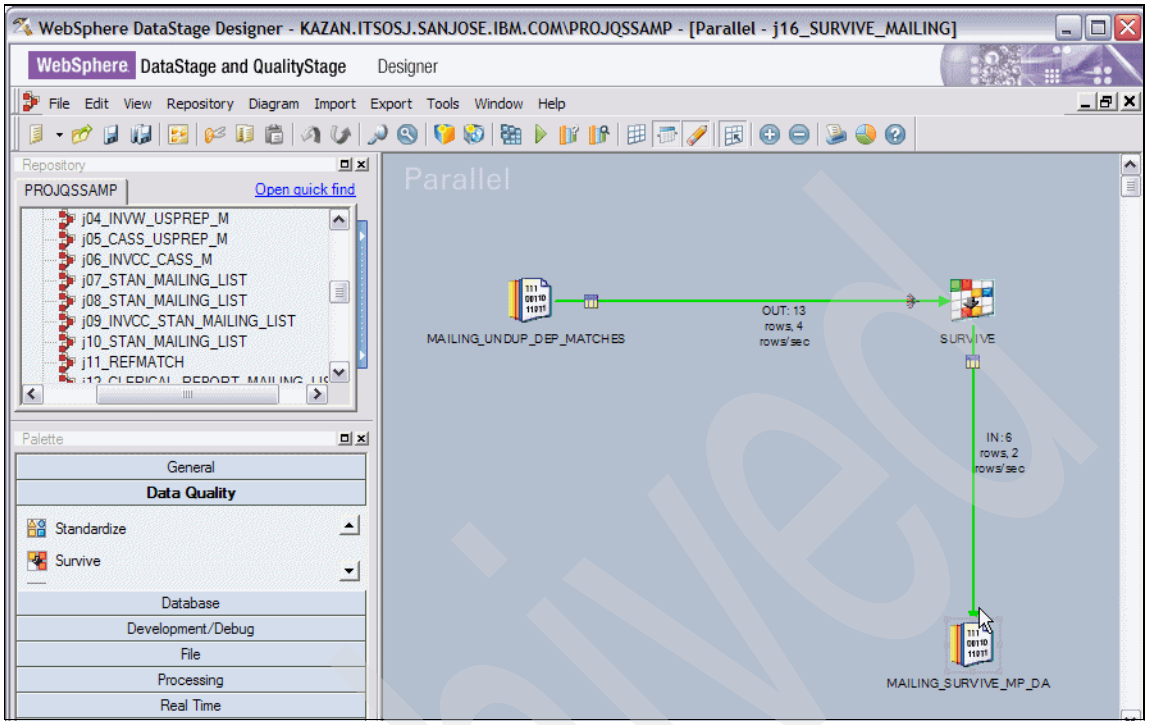
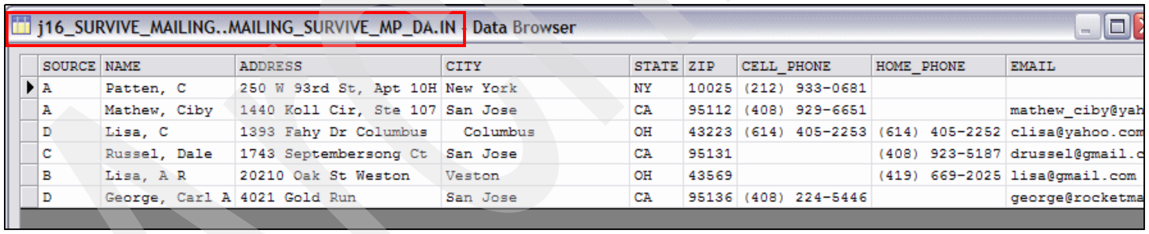
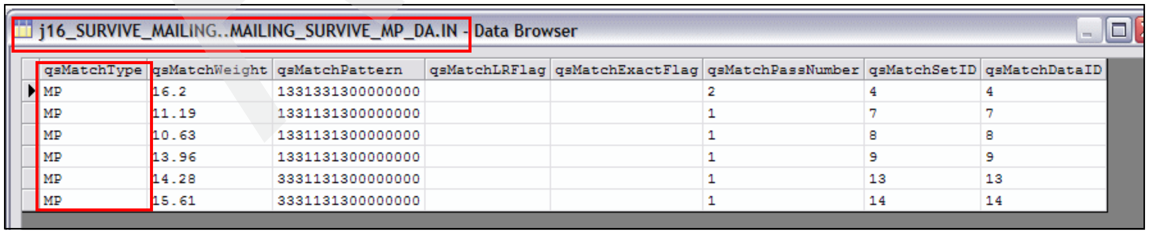


Figure 1-562 Create J16_SURVIVE_MAILING 4/6


j16_SURVIVE_MAILING..MAILING_SURVIVE_MP_DA.IN - Data Browser

SOURCE	NAME	ADDRESS	CITY	STATE	ZIP	CELL_PHONE	HOME_PHONE	EMAIL
A	Patten, C	250 W 93rd St, Apt 10H	New York	NY	10025	(212) 933-0681		
A	Mathew, Ciby	1440 Koll Cir, Ste 107	San Jose	CA	95112	(408) 929-6651		mathew_ciby@yah
D	Lisa, C	1393 Fahy Dr Columbus	Columbus	OH	43223	(614) 405-2253	(614) 405-2252	clisa@yahoo.com
C	Russel, Dale	1743 Septembersong Ct	San Jose	CA	95131		(408) 923-5187	drussel@gmail.c
B	Lisa, A R	20210 Oak St Weston	Veston	OH	43569		(419) 669-2025	lisa@gmail.com
D	George, Carl A	4021 Gold Run	San Jose	CA	95136	(408) 224-5446		george@rocketma

Figure 1-563 Create J16_SURVIVE_MAILING 5/6


j16_SURVIVE_MAILING..MAILING_SURVIVE_MP_DA.IN - Data Browser

qsMatchType	qsMatchWeight	qsMatchPattern	qsMatchLRFlag	qsMatchExactFlag	qsMatchPassNumber	qsMatchSetID	qsMatchDataID
MP	16.2	1331331300000000			2	4	4
MP	11.19	1331131300000000			1	7	7
MP	10.63	1331131300000000			1	8	8
MP	13.96	1331131300000000			1	9	9
MP	14.28	3331131300000000			1	13	13
MP	15.61	3331131300000000			1	14	14

Figure 1-564 Create J16_SURVIVE_MAILING 6/6

J17_MAILING_MASTER

In this step, we create a clean master of mailing list persons by merging the master that was created in the “J16_SURVIVE_MAILING” on page 454 step and residual records that were created in the “J15_UNDUP_DEP_MATCH_MAILING” on page 449 step into a data set using the FUNNEL stage.

Figure 1-565 on page 459 through Figure 1-569 on page 461 describe the main steps using Designer Client to perform this task.

Figure 1-565 on page 459 shows the various stages that are used in this job, including the input data sets MAILING_SURVIVE_MP_DA (created in “J16_SURVIVE_MAILING” on page 454 containing survived records) and MAILING_UNDUP_DEP_RESIDUALS created in the “J15_UNDUP_DEP_MATCH_MAILING” on page 449, a FUNNEL stage, and an output data set containing the merged records from the input. We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object created earlier.

Because the creation of this survive job is very similar to that described in “J14_CUSTOMER_MASTER” on page 303, we do not repeat the steps that are involved here. However, some of the configurations of interest are as follows:

- Figure 1-566 on page 459 and Figure 1-567 on page 460 show the FUNNEL - Funnel windows with the Continuous Funnel and the mapping of all columns from the Columns pane to the IN pane.
- After saving, compiling, and running this job j17_MAILING_MASTER (Figure 1-568 on page 460), view the content of the output data set object MAILING_MASTER_SF as shown in Figure 1-569 on page 461.

  This report shows the cleansed mailing list file (14 records) that has no duplicates.

Now, proceed to “J18_FREQ_MAILING_MASTER” on page 461.

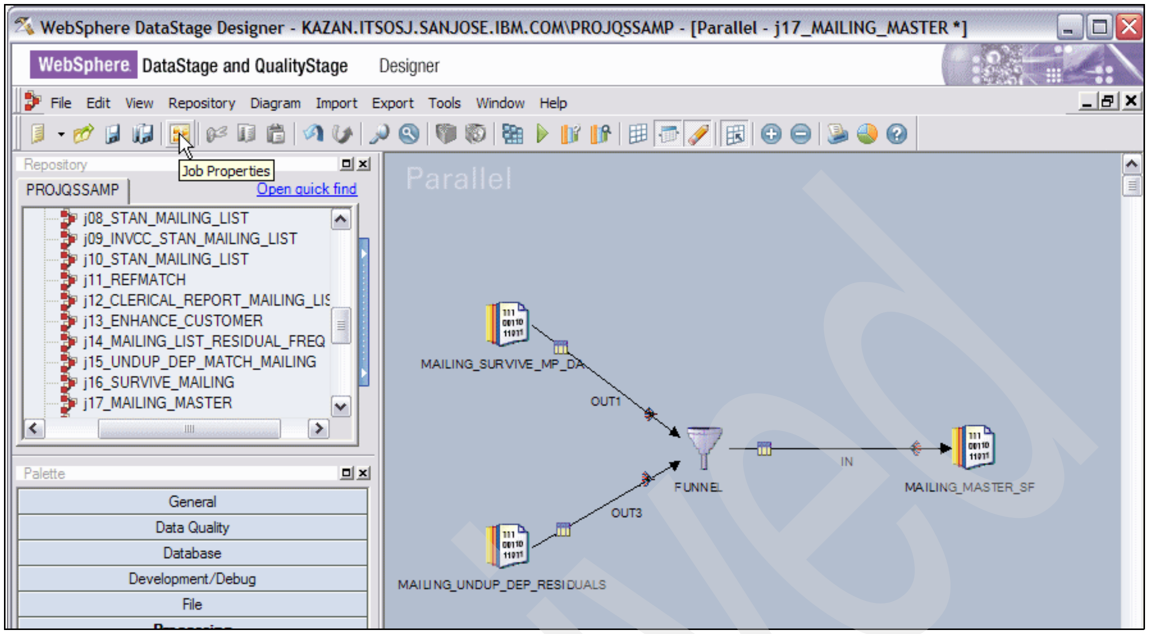
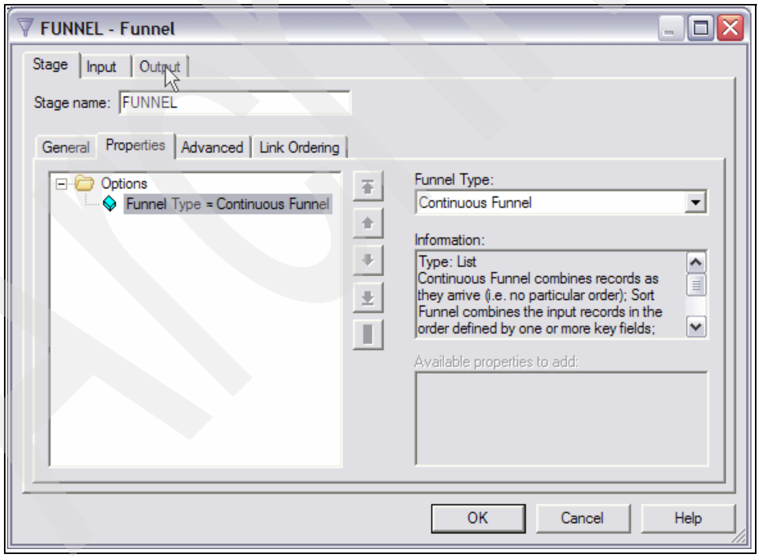

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j17_MAILING_MASTER *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Job Properties
Repository
PROJQSSAMP
Open quick find
j08_STAN_MAILING_LIST
j09_INVCC_STAN_MAILING_LIST
j10_STAN_MAILING_LIST
j11_REFMATCH
j12_CLERICAL_REPORT_MAILING_LIS
j13_ENHANCE_CUSTOMER
j14_MAILING_LIST_RESIDUAL_FREQ
j15_UNDUP_DEP_MATCH_MAILING
j16_SURVIVE_MAILING
j17_MAILING_MASTER
Palette
General
Data Quality
Database
Development/Debug
File
Parallel
MAILING_SURVIVE_MP_DA
OUT1
FUNNEL
IN
MAILING_MASTER_SF
OUT3
MAILING_UNDUP_DEP_RESIDUALS

Figure 1-565 Create J17_MAILING_MASTER 1/5



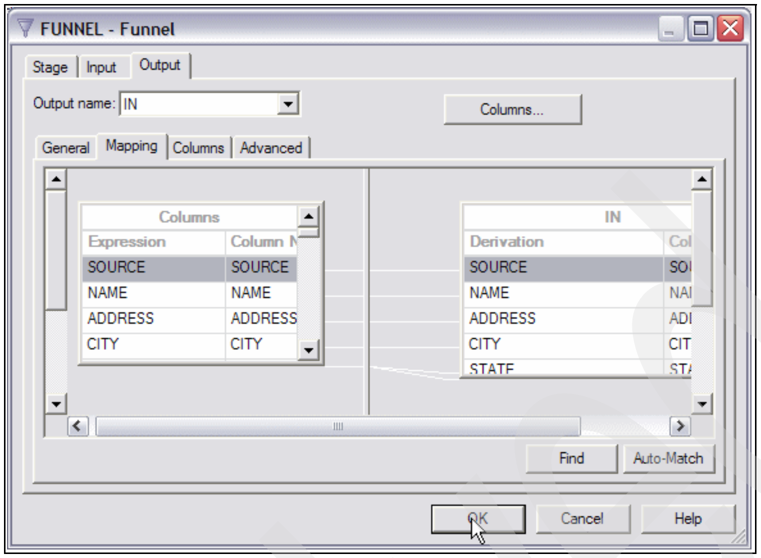
Stage name: FUNNEL

Funnel Type = Continuous Funnel

Funnel Type: Continuous Funnel

Information:
Type: List
Continuous Funnel combines records as they arrive (i.e. no particular order); Sort Funnel combines the input records in the order defined by one or more key fields;

Figure 1-566 Create J17_MAILING_MASTER 2/5



Columns	
Expression	Column N
SOURCE	SOURCE
NAME	NAME
ADDRESS	ADDRESS
CITY	CITY

IN	
Derivation	Col
SOURCE	SOI
NAME	NAI
ADDRESS	ADI
CITY	CIT
STATE	STA

Figure 1-567   Create J17_MAILING_MASTER 3/5

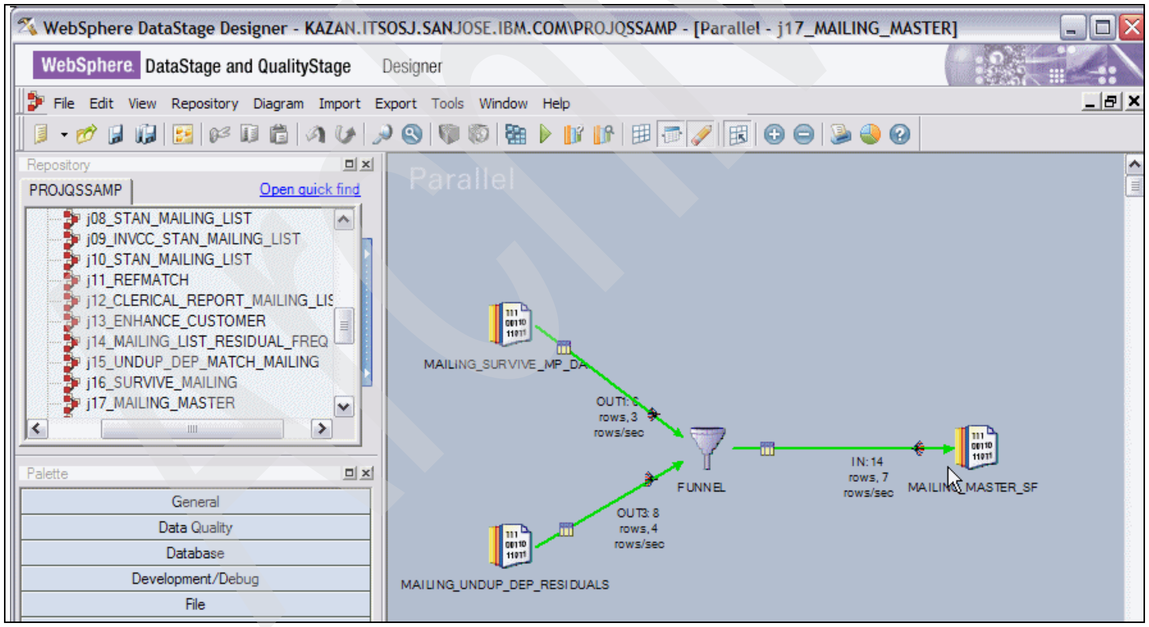
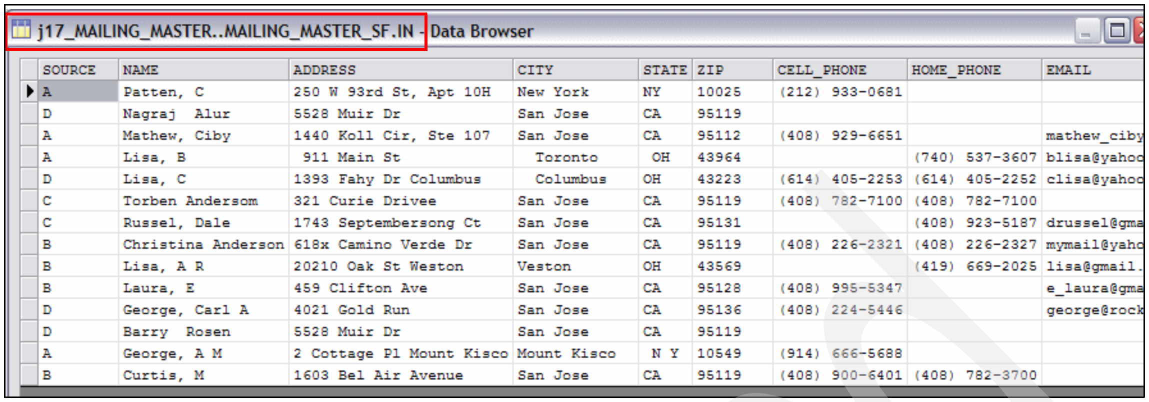

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j17_MAILING_MASTER]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Repository
PROJQSSAMP
Open quick find
j08_STAN_MAILING_LIST
j09_INVCC_STAN_MAILING_LIST
j10_STAN_MAILING_LIST
j11_REFMATCH
j12_CLERICAL_REPORT_MAILING_LIS
j13_ENHANCE_CUSTOMER
j14_MAILING_LIST_RESIDUAL_FREQ
j15_UNDUP_DEP_MATCH_MAILING
j16_SURVIVE_MAILING
j17_MAILING_MASTER
Palette
General
Data Quality
Database
Development/Debug
File
Parallel
MAILING_SURVIVE_MP_DA
OUT1: 6 rows, 3 rows/sec
FUNNEL
IN: 14 rows, 7 rows/sec
MAILING_MASTER_SF
OUT3: 8 rows, 4 rows/sec
MAILING_UNDUP_DEP_RESIDUALS

Figure 1-568   Create J17_MAILING_MASTER 4/5


j17_MAILING_MASTER..MAILING_MASTER_SF.IN - Data Browser

SOURCE	NAME	ADDRESS	CITY	STATE	ZIP	CELL_PHONE	HOME_PHONE	EMAIL
A	Patten, C	250 W 93rd St, Apt 10H	New York	NY	10025	(212) 933-0681		
D	Nagraj Alur	5528 Muir Dr	San Jose	CA	95119			
A	Mathew, Ciby	1440 Koll Cir, Ste 107	San Jose	CA	95112	(408) 929-6651		mathew_ciby
A	Lisa, B	911 Main St	Toronto	OH	43964		(740) 537-3607	blisa@yahoo
D	Lisa, C	1393 Fahy Dr Columbus	Columbus	OH	43223	(614) 405-2253	(614) 405-2252	clisa@yahoo
C	Torben Andersom	321 Curie Drivee	San Jose	CA	95119	(408) 782-7100	(408) 782-7100	
C	Russel, Dale	1743 Septembersong Ct	San Jose	CA	95131		(408) 923-5187	drussel@gma
B	Christina Anderson	618x Camino Verde Dr	San Jose	CA	95119	(408) 226-2321	(408) 226-2327	mymail@yaho
B	Lisa, A R	20210 Oak St Weston	Veston	OH	43569		(419) 669-2025	lisa@gmail.
B	Laura, E	459 Clifton Ave	San Jose	CA	95128	(408) 995-5347		e_laura@gma
D	George, Carl A	4021 Gold Run	San Jose	CA	95136	(408) 224-5446		george@rock
D	Barry Rosen	5528 Muir Dr	San Jose	CA	95119			
A	George, A M	2 Cottage Pl Mount Kisco	Mount Kisco	N Y	10549	(914) 666-5688		
B	Curtis, M	1603 Bel Air Avenue	San Jose	CA	95119	(408) 900-6401	(408) 782-3700	

Figure 1-569 Create J17_MAILING_MASTER 5/5

J18_FREQ_MAILING_MASTER

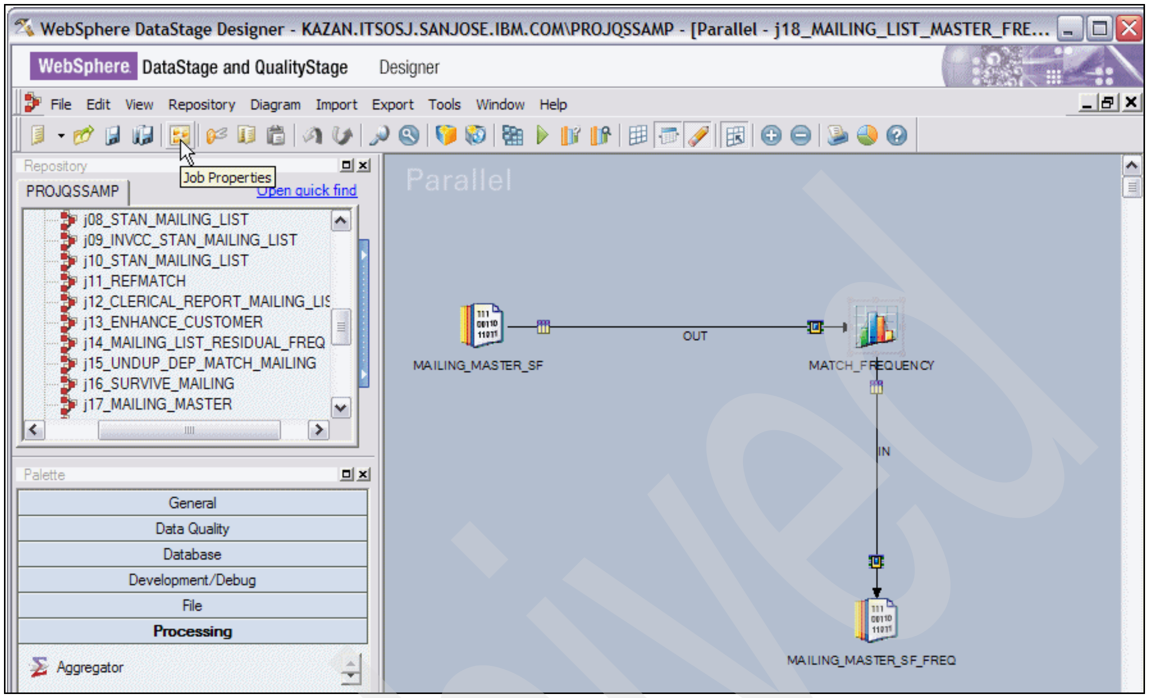
To generate household information for the master mailing list persons that were created in the "J17_MAILING_MASTER" on page 458 step, we generate a frequency distribution on *all* the columns in the master mailing list file using the Match Frequency stage. Here again, the idea is to generate match frequency for all the columns so that it can be used with any match specification.

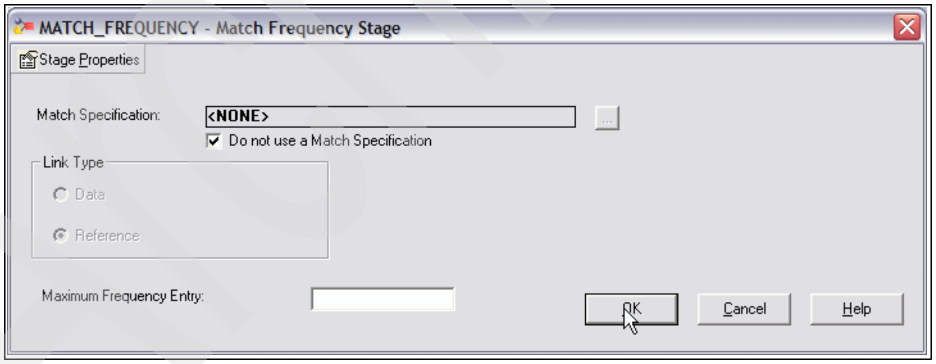
Figure 1-570 on page 462 shows the various stages that are used in this job, including the data set that was created in "J17_MAILING_MASTER" on page 458, a Match Frequency stage, and a Data Set stage. We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object created earlier.

Because the configuration of this job is very similar to that described in "J10_MATCHFREQ_STAN_CUSTOMER" on page 266, we do not repeat the process here. However, some of the configurations of interest are as follows:

- Figure 1-571 on page 462 shows the MATCH_FREQUENCY - Match Frequency Stage window with Match Specification of NONE.
- After saving, compiling, and running this job (Figure 1-572 on page 463), the contents of the output of this stage are listed in Figure 1-573 on page 464. We do not document the interpretation of the format and content of this file.

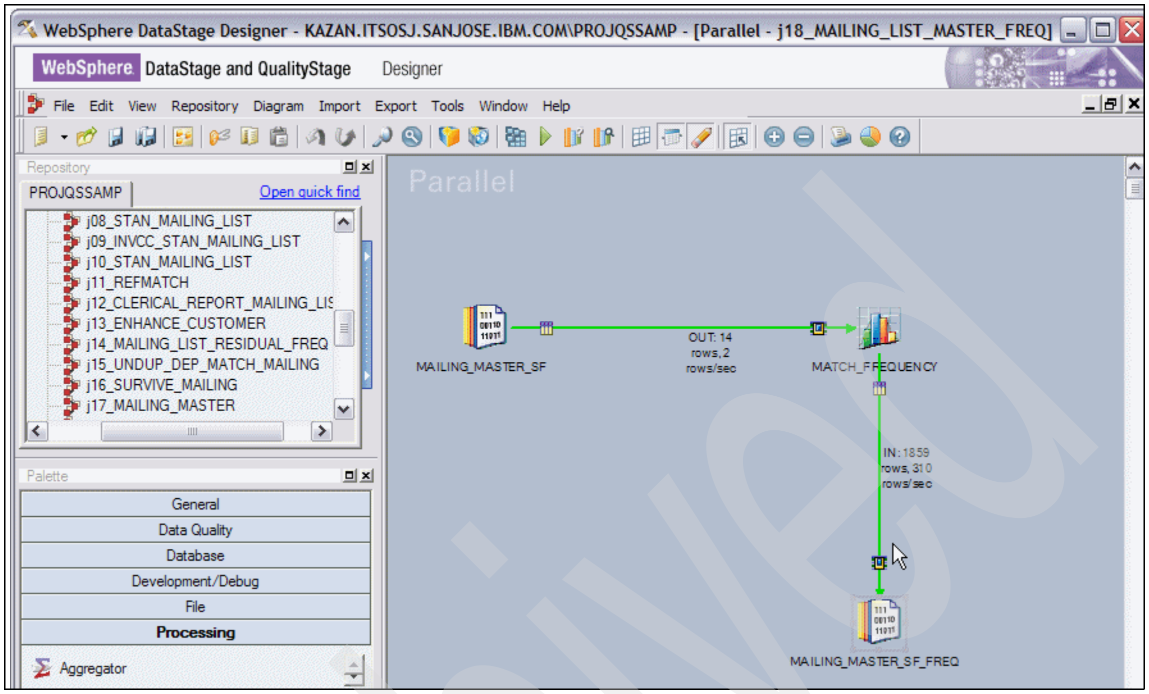
Now, proceed to "J18_UNDUP_IND_MATCHSPEC_MAILING" on page 464.


WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j18_MAILING_LIST_MASTER_FRE...
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Job Properties
Repository
PROJQSSAMP
Open quick find
j08_STAN_MAILING_LIST
j09_INVCC_STAN_MAILING_LIST
j10_STAN_MAILING_LIST
j11_REFMATCH
j12_CLERICAL_REPORT_MAILING_LIS
j13_ENHANCE_CUSTOMER
j14_MAILING_LIST_RESIDUAL_FREQ
j15_UNDUP_DEP_MATCH_MAILING
j16_SURVIVE_MAILING
j17_MAILING_MASTER
Palette
General
Data Quality
Database
Development/Debug
File
Processing
Aggregator
Parallel
MAILING_MASTER_SF
OUT
MATCH_FREQUENCY
IN
MAILING_MASTER_SF_FREQ
Figure 1-570 Create J18_FREQ_MAILING_MASTER 1/4



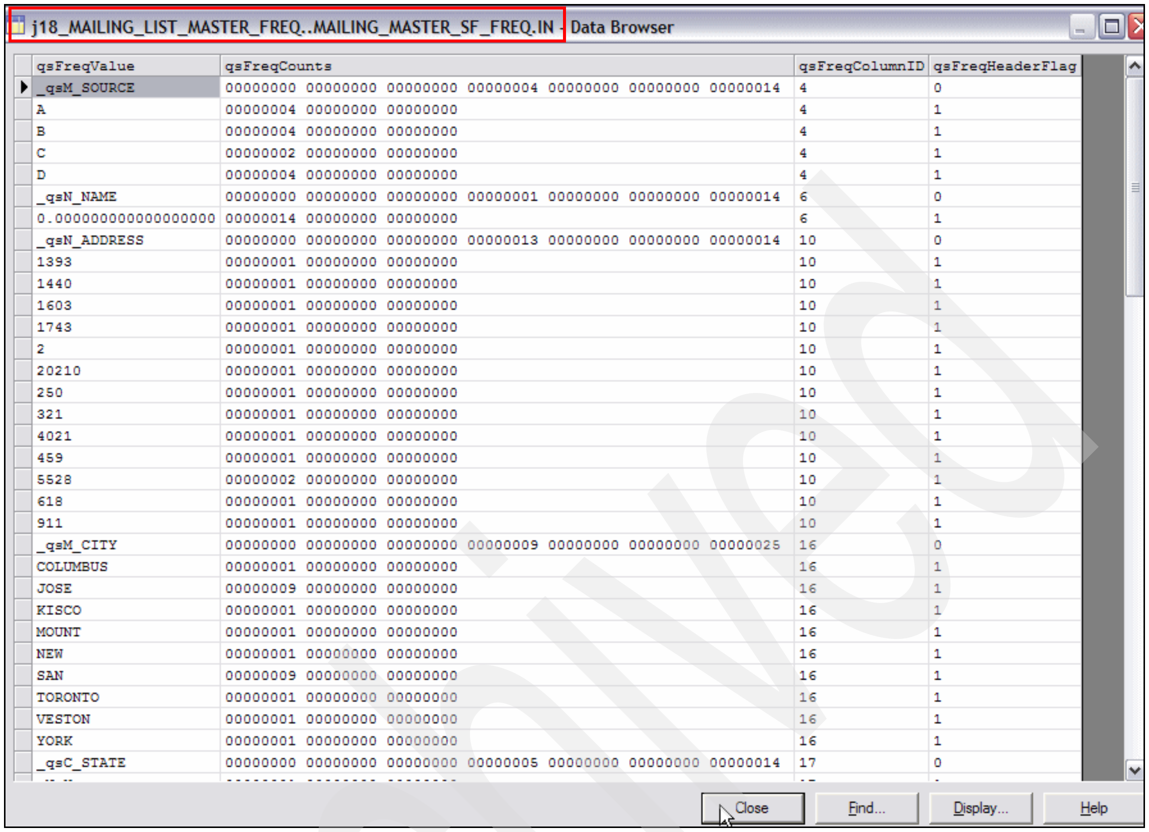
Match Specification: <NONE>
[✓] Do not use a Match Specification
Data [✓ Reference]
Maximum Frequency Entry:

Figure 1-571 Create J18_FREQ_MAILING_MASTER 2/4



WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j18_MAILING_LIST_MASTER_FREQ]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Repository
PROJQSSAMP
Open quick find
j08_STAN_MAILING_LIST
j09_INVCC_STAN_MAILING_LIST
j10_STAN_MAILING_LIST
j11_REFMATCH
j12_CLERICAL_REPORT_MAILING_LIS
j13_ENHANCE_CUSTOMER
j14_MAILING_LIST_RESIDUAL_FREQ
j15_UNDUP_DEP_MATCH_MAILING
j16_SURVIVE_MAILING
j17_MAILING_MASTER
Palette
General
Data Quality
Database
Development/Debug
File
Processing
Aggregator
Parallel
MAILING_MASTER_SF
OUT: 14 rows, 2 rows/sec
MATCH_FREQUENCY
IN: 1859 rows, 310 rows/sec
MAILING_MASTER_SF_FREQ

Figure 1-572 Create J18_FREQ_MAILING_MASTER 3/4



j18_MAILING_LIST_MASTER_FREQ..MAILING_MASTER_SF_FREQ.IN - Data Browser

qsFreqValue	qsFreqCounts	qsFreqColumnID	qsFreqHeaderFlag
_qsM_SOURCE	00000000 00000000 00000000 00000004 00000000 00000000 00000014	4	0
A	00000004 00000000 00000000	4	1
B	00000004 00000000 00000000	4	1
C	00000002 00000000 00000000	4	1
D	00000004 00000000 00000000	4	1
_qsN_NAME	00000000 00000000 00000000 00000001 00000000 00000000 00000014	6	0
0.000000000000000000	00000014 00000000 00000000	6	1
_qsN_ADDRESS	00000000 00000000 00000000 00000013 00000000 00000000 00000014	10	0
1393	00000001 00000000 00000000	10	1
1440	00000001 00000000 00000000	10	1
1603	00000001 00000000 00000000	10	1
1743	00000001 00000000 00000000	10	1
2	00000001 00000000 00000000	10	1
20210	00000001 00000000 00000000	10	1
250	00000001 00000000 00000000	10	1
321	00000001 00000000 00000000	10	1
4021	00000001 00000000 00000000	10	1
459	00000001 00000000 00000000	10	1
5528	00000002 00000000 00000000	10	1
618	00000001 00000000 00000000	10	1
911	00000001 00000000 00000000	10	1
_qsM_CITY	00000000 00000000 00000000 00000009 00000000 00000000 00000025	16	0
COLUMBUS	00000001 00000000 00000000	16	1
JOSE	00000009 00000000 00000000	16	1
KISCO	00000001 00000000 00000000	16	1
MOUNT	00000001 00000000 00000000	16	1
NEW	00000001 00000000 00000000	16	1
SAN	00000009 00000000 00000000	16	1
TORONTO	00000001 00000000 00000000	16	1
VESTON	00000001 00000000 00000000	16	1
YORK	00000001 00000000 00000000	16	1
_qsC_STATE	00000000 00000000 00000000 00000005 00000000 00000000 00000014	17	0

Figure 1-573 Create J18_FREQ_MAILING_MASTER 4/4

J18_UNDUP_IND_MATCHSPEC_MAILING

In this step, we generate a match specification for an Unduplicate match stage using as input the match frequency data that was created in the "J18_FREQ_MAILING_MASTER" on page 461 job. The specification included two passes:

- The first pass blocks on the primary (last) name and a phonetic encoding (NYSIIS) of the street name.
- The second pass blocks on the five digit ZIP code and a phonetic encoding (NYSIIS) of the street name.

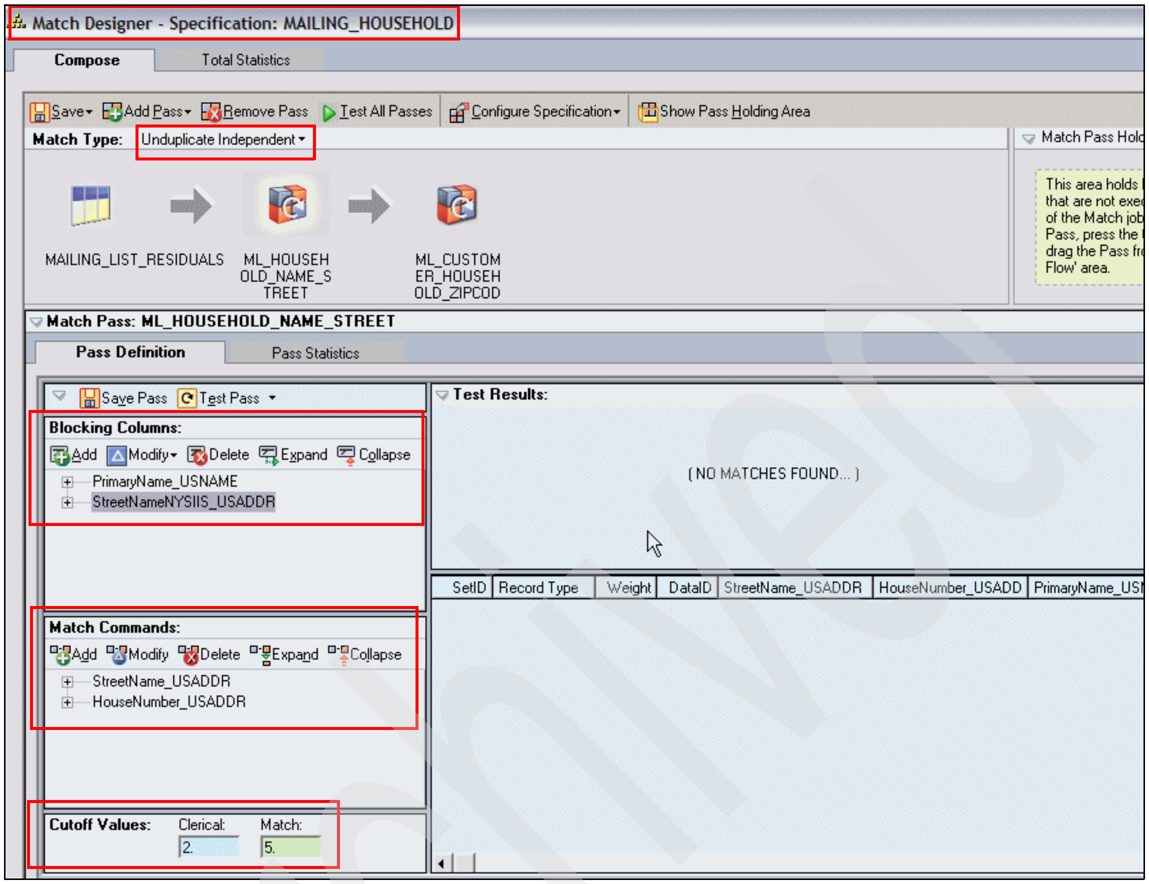
Because the creation of a match specification is similar to that described in "J10_Undup_MatchSpec_STAN_CUSTOMER" on page 269, we do not repeat the process here.

However, a summary of the passes is as follows:

- ML_HOUSEHOLD_NAME_STREET pass (Figure 1-574 on page 466) with the blocking columns, match commands, cutoff values, and test results. The Match Type was Unduplicate Independent because we need to group all the matching records for the purpose of identifying household information. No matching records are detected in this pass.
- ML_CUSTOMER_HOUSEHOLD_ZIPCODE pass (Figure 1-575 on page 467) with the blocking columns, match commands, cutoff values, and test results. The results show a single master record (Record Type `XA`) and duplicate (Record Type `DA`).
- Total Statistics of the two passes in Figure 1-576 on page 468. It shows the total number of Pseudo matches (0+1 for a total of 1).

  Of particular interest is the statistic OVERFLOW blocks—a non-zero value indicates the need to increase the block size or define more restrictive blocking columns.

The test of the MAILING_HOUSEHOLD match specification with representative sample data appears to deliver results that are accurate. You can then use this specification in an Unduplicate Independent match stage as described in the step “J19_UNDUP_IND_MATCH_MAILING” on page 468.


Match Designer - Specification: MAILING_HOUSEHOLD

Match Type: Unduplicate Independent

MAILING_LIST_RESIDUALS → ML_HOUSEHOLD_NAME_STREET → ML_CUSTOMER_HOUSEHOLD_ZIPCOD

This area holds that are not exe of the Match job Pass, press the drag the Pass fr Flow' area.

Match Pass: ML_HOUSEHOLD_NAME_STREET

Blocking Columns:
- PrimaryName_USNAME
- StreetNameNYSIIS_USADDR

Match Commands:
- StreetName_USADDR
- HouseNumber_USADDR

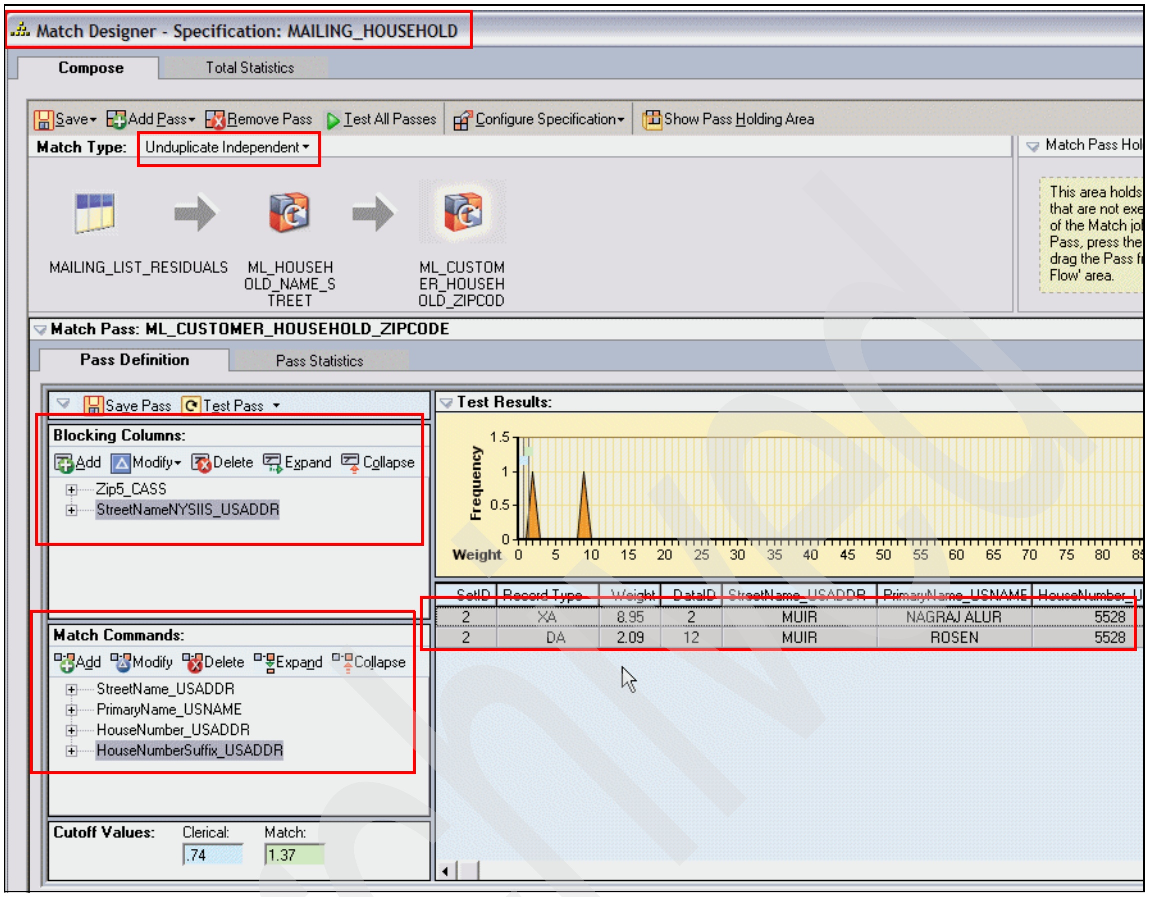
Cutoff Values: Clerical: 2. Match: 5.

Test Results:

(NO MATCHES FOUND...)

SetID	Record Type	Weight	DataID	StreetName_USADDR	HouseNumber_USADD	PrimaryName_USI

Figure 1-574 Create J18_UNDUP_IND_MATCHSPEC_MAILING 1/3



Match Type: Unduplicate Independent

This area holds that are not exe of the Match jo Pass, press the drag the Pass f Flow' area.

Match Pass: ML_CUSTOMER_HOUSEHOLD_ZIPCODE

Blocking Columns:

- Zip5_CASS
- StreetNameNYSIIS_USADDR

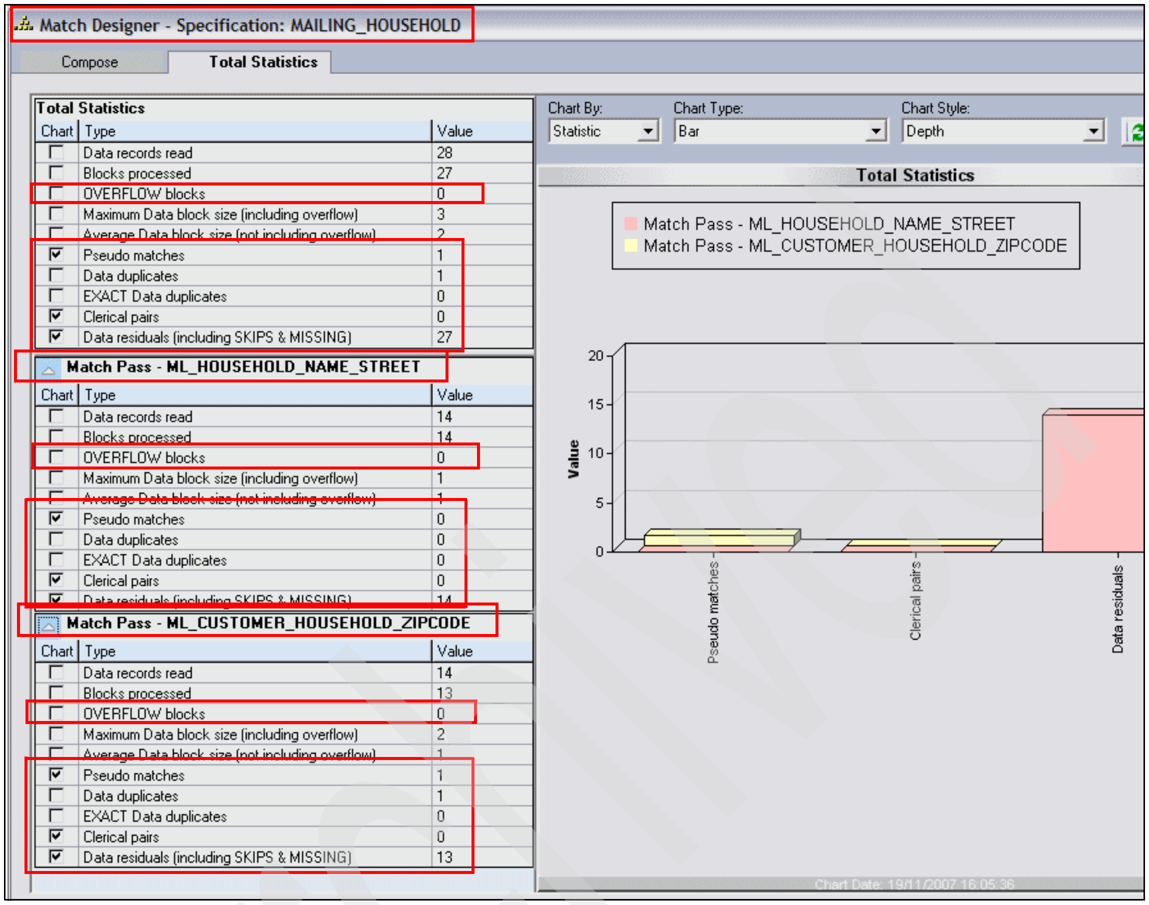
Match Commands:

- StreetName_USADDR
- PrimaryName_USNAME
- HouseNumber_USADDR
- HouseNumberSuffix_USADDR

Cutoff Values: Clerical: .74 Match: 1.37

SetID	Record Type	Weight	DataID	StreetName_USADDR	PrimaryName_USNAME	HouseNumber_U
2	XA	8.95	2	MUIR	NAGRAJ ALUR	5528
2	DA	2.09	12	MUIR	ROSEN	5528

Figure 1-575 Create J18_UNDUP_IND_MATCHSPEC_MAILING 2/3



Match Designer - Specification: MAILING_HOUSEHOLD
Compose
Total Statistics
Chart By:
Statistic
Chart Type:
Bar
Chart Style:
Depth
Total Statistics
Match Pass - ML_HOUSEHOLD_NAME_STREET
Match Pass - ML_CUSTOMER_HOUSEHOLD_ZIPCODE
Value
20
15
10
5
0
Pseudo matches
Clerical pairs
Data residuals
Chart Date: 19/11/2007 16:05:36

Total Statistics

Chart	Type	Value
	Data records read	28
	Blocks processed	27
	OVERFLOW blocks	0
	Maximum Data block size (including overflow)	3
	Average Data block size (not including overflow)	2
✓	Pseudo matches	1
	Data duplicates	1
	EXACT Data duplicates	0
✓	Clerical pairs	0
✓	Data residuals (including SKIPS & MISSING)	27

Match Pass - ML_HOUSEHOLD_NAME_STREET

Chart	Type	Value
	Data records read	14
	Blocks processed	14
	OVERFLOW blocks	0
	Maximum Data block size (including overflow)	1
	Average Data block size (not including overflow)	1
✓	Pseudo matches	0
	Data duplicates	0
	EXACT Data duplicates	0
✓	Clerical pairs	0
✓	Data residuals (including SKIPS & MISSING)	14

Match Pass - ML_CUSTOMER_HOUSEHOLD_ZIPCODE

Chart	Type	Value
	Data records read	14
	Blocks processed	13
	OVERFLOW blocks	0
	Maximum Data block size (including overflow)	2
	Average Data block size (not including overflow)	1
✓	Pseudo matches	1
	Data duplicates	1
	EXACT Data duplicates	0
✓	Clerical pairs	0
✓	Data residuals (including SKIPS & MISSING)	13

Figure 1-576 Create J18_UNDUP_IND_MATCHSPEC_MAILING 3/3

J19_UNDUP_IND_MATCH_MAILING

In this step, we determine whether there were duplicates in the mailing list persons file using the Unduplicate stage with the match specification and match frequency information that was created in steps "J18_UNDUP_IND_MATCHSPEC_MAILING" on page 464 and "J18_FREQ_MAILING_MASTER" on page 461 respectively. As before, the output is matched records (merge of master and duplicates using a Funnel stage), records for clerical review, and residuals (records that do not match).

Because the creation of this Unduplicate stage is very similar to that described in "J11_UNDUP_DEP_MATCH_CUSTOMER" on page 282, we do not repeat the steps that are involved here. However, the key differences are as follows:

- Figure 1-577 on page 470 shows the various stages that are used in this job, including the data set that was created in "J17_MAILING_MASTER" on page 458, the match frequency data set that was created in "J18_UNDUP_IND_MATCHSPEC_MAILING" on page 464, an Unduplicate stage, and a Funnel stage to merge master and duplicate records. Three data sets are created:
  - One data set contains the merged master and duplicates by the Funnel stage.
  - The other data sets contain the clerical and residual records as the output of the Unduplicate stage.

  We modify the names of the stages as shown, and this job uses the same QSPARAMETERSET object created earlier.
- In the UNDUP_IND - Unduplicate Match Stage window shown in Figure 1-578 on page 470, select the Match Specification MAILING_HOUSEHOLD. Then, select the Match Type Independent in order to find groups of all records (for household information) across multiple match passes as described in Table 1-11 on page 106.

7. After saving, compiling, and running this job j19_UNDUP_IND_MATCH_MAILING (Figure 1-579 on page 471—shows zero rows being written to the MAILING_UNDUP_IND_CLERICALS), view the content of the two data set objects as shown in Figure 1-580 on page 471 through Figure 1-583 on page 472.

   Out of a total of 14 records in the input to this process:
   - Two (2) records are masters and duplicates
   - Zero (0) records are for clerical review
   - Remaining twelve (12) records are residuals.

The next step is to create a single mailing list file with household information as described in "J20_MAILING_MASTER_WITH_HOUSEHOLD" on page 473.

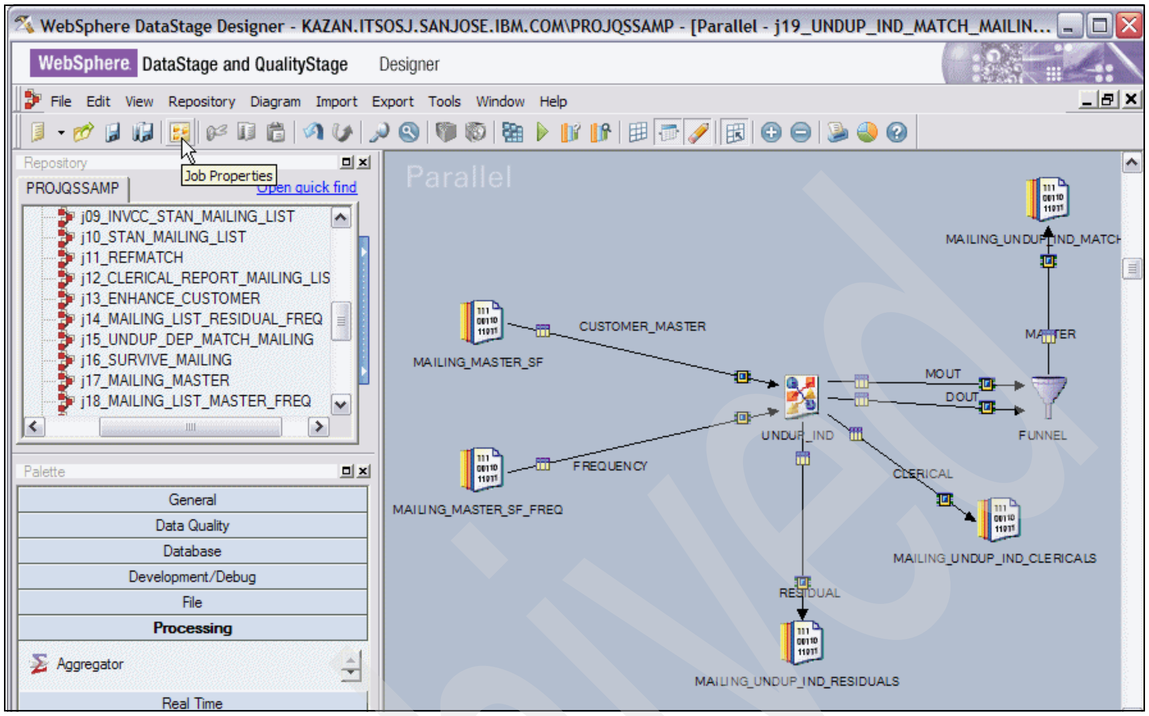
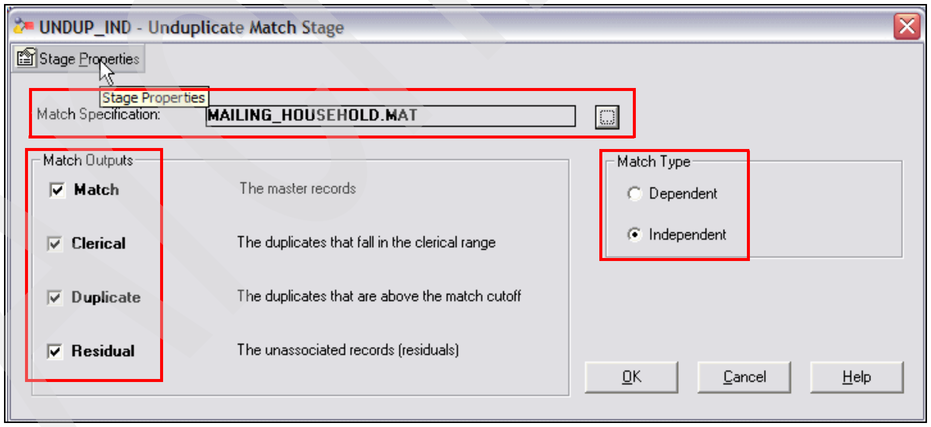

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j19_UNDUP_IND_MATCH_MAILIN...
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Job Properties
Repository
PROJQSSAMP
Open quick find
j09_INVCC_STAN_MAILING_LIST
j10_STAN_MAILING_LIST
j11_REFMATCH
j12_CLERICAL_REPORT_MAILING_LIS
j13_ENHANCE_CUSTOMER
j14_MAILING_LIST_RESIDUAL_FREQ
j15_UNDUP_DEP_MATCH_MAILING
j16_SURVIVE_MAILING
j17_MAILING_MASTER
j18_MAILING_LIST_MASTER_FREQ
Palette
General
Data Quality
Database
Development/Debug
File
Processing
Aggregator
Real Time
Parallel
MAILING_UNDUP_IND_MATCH
MASTER
CUSTOMER_MASTER
MAILING_MASTER_SF
MOUT
DOUT
UNDUP_IND
FUNNEL
FREQUENCY
MAILING_MASTER_SF_FREQ
CLERICAL
MAILING_UNDUP_IND_CLERICALS
RESIDUAL
MAILING_UNDUP_IND_RESIDUALS

Figure 1-577 Create J19_UNDUP_IND_MATCH_MAILING 1/7


UNDUP_IND - Unduplicate Match Stage

Stage Properties

Match Specification: MAILING_HOUSEHOLD.MAT

Match Outputs

- [✓] **Match** The master records
- [✓] **Clerical** The duplicates that fall in the clerical range
- [✓] **Duplicate** The duplicates that are above the match cutoff
- [✓] **Residual** The unassociated records (residuals)

Match Type

- ○ Dependent
- ● Independent

Figure 1-578 Create J19_UNDUP_IND_MATCH_MAILING 2/7

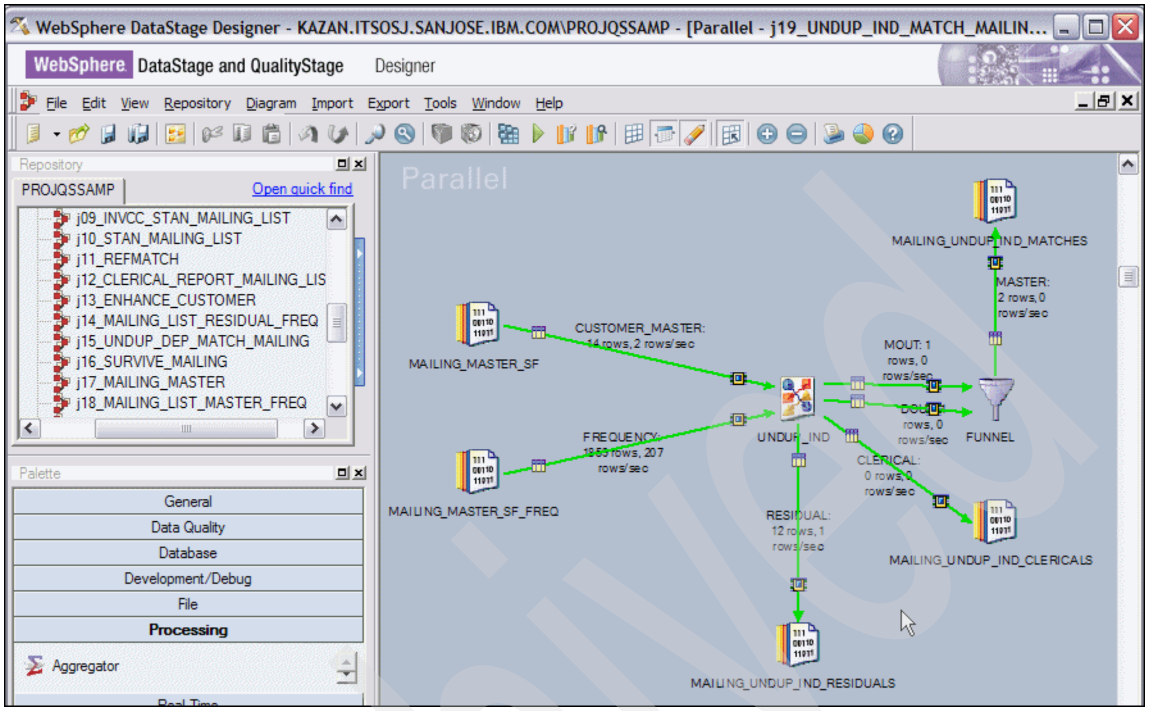
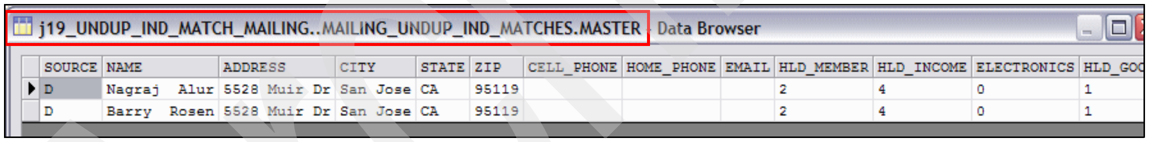

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j19_UNDUP_IND_MATCH_MAILIN...
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Repository
PROJQSSAMP
Open quick find
j09_INVCC_STAN_MAILING_LIST
j10_STAN_MAILING_LIST
j11_REFMATCH
j12_CLERICAL_REPORT_MAILING_LIS
j13_ENHANCE_CUSTOMER
j14_MAILING_LIST_RESIDUAL_FREQ
j15_UNDUP_DEP_MATCH_MAILING
j16_SURVIVE_MAILING
j17_MAILING_MASTER
j18_MAILING_LIST_MASTER_FREQ
Palette
General
Data Quality
Database
Development/Debug
File
Processing
Aggregator
Real Time
Parallel
MAILING_MASTER_SF
CUSTOMER_MASTER: 14 rows, 2 rows/sec
MAILING_MASTER_SF_FREQ
FREQUENCY: 1853 rows, 207 rows/sec
UNDUP_IND
MOUT: 1 rows, 0 rows/sec
DOUT: rows, 0 rows/sec
FUNNEL
MASTER: 2 rows, 0 rows/sec
MAILING_UNDUP_IND_MATCHES
CLERICAL: 0 rows, 0 rows/sec
MAILING_UNDUP_IND_CLERICALS
RESIDUAL: 12 rows, 1 rows/sec
MAILING_UNDUP_IND_RESIDUALS

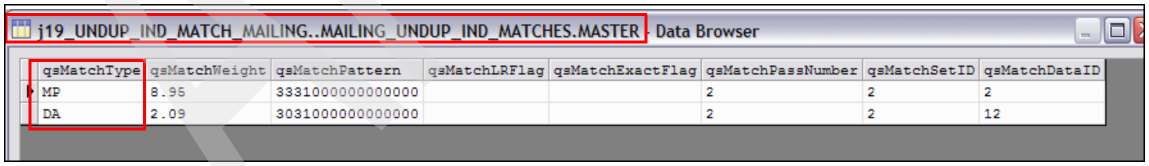
Figure 1-579 Create J19_UNDUP_IND_MATCH_MAILING 3/7



j19_UNDUP_IND_MATCH_MAILING..MAILING_UNDUP_IND_MATCHES.MASTER - Data Browser

SOURCE	NAME	ADDRESS	CITY	STATE	ZIP	CELL_PHONE	HOME_PHONE	EMAIL	HLD_MEMBER	HLD_INCOME	ELECTRONICS	HLD_GOO
D	Nagraj Alur	5528 Muir Dr	San Jose	CA	95119				2	4	0	1
D	Barry Rosen	5528 Muir Dr	San Jose	CA	95119				2	4	0	1

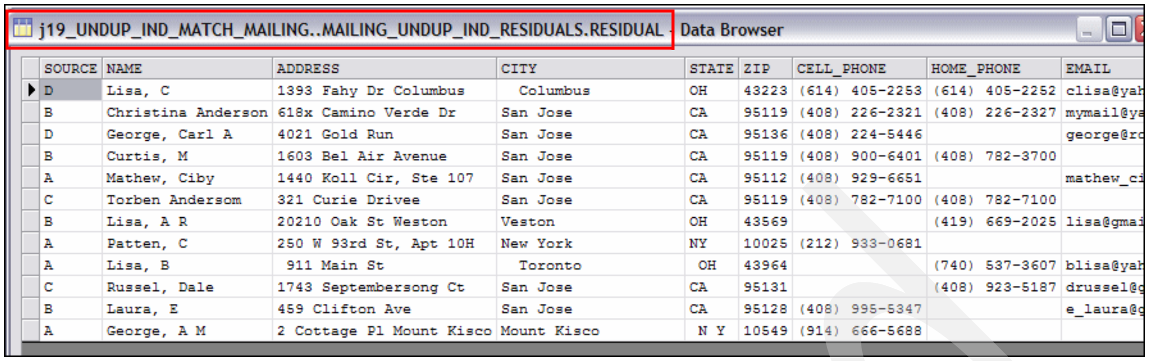
Figure 1-580 Create J19_UNDUP_IND_MATCH_MAILING 4/7



j19_UNDUP_IND_MATCH_MAILING..MAILING_UNDUP_IND_MATCHES.MASTER - Data Browser

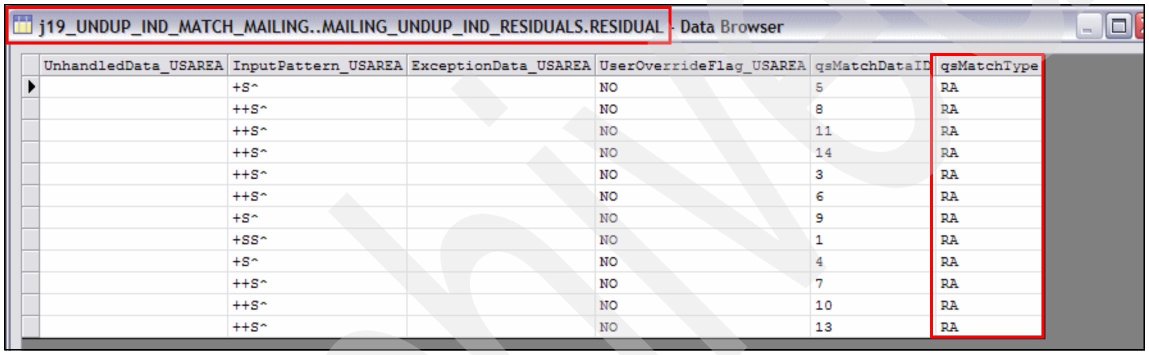
qsMatchType	qsMatchWeight	qsMatchPattern	qsMatchLRFlag	qsMatchExactFlag	qsMatchPassNumber	qsMatchSetID	qsMatchDataID
MP	8.95	3331000000000000			2	2	2
DA	2.09	3031000000000000			2	2	12

Figure 1-581 Create J19_UNDUP_IND_MATCH_MAILING 5/7


j19_UNDUP_IND_MATCH_MAILING..MAILING_UNDUP_IND_RESIDUALS.RESIDUAL - Data Browser

SOURCE	NAME	ADDRESS	CITY	STATE	ZIP	CELL_PHONE	HOME_PHONE	EMAIL
D	Lisa, C	1393 Fahy Dr Columbus	Columbus	OH	43223	(614) 405-2253	(614) 405-2252	clisa@yah
B	Christina Anderson	618x Camino Verde Dr	San Jose	CA	95119	(408) 226-2321	(408) 226-2327	mymail@ya
D	George, Carl A	4021 Gold Run	San Jose	CA	95136	(408) 224-5446		george@ro
B	Curtis, M	1603 Bel Air Avenue	San Jose	CA	95119	(408) 900-6401	(408) 782-3700	
A	Mathew, Ciby	1440 Koll Cir, Ste 107	San Jose	CA	95112	(408) 929-6651		mathew_ci
C	Torben Andersom	321 Curie Drivee	San Jose	CA	95119	(408) 782-7100	(408) 782-7100	
B	Lisa, A R	20210 Oak St Weston	Veston	OH	43569		(419) 669-2025	lisa@gmai
A	Patten, C	250 W 93rd St, Apt 10H	New York	NY	10025	(212) 933-0681		
A	Lisa, B	911 Main St	Toronto	OH	43964		(740) 537-3607	blisa@yah
C	Russel, Dale	1743 Septembersong Ct	San Jose	CA	95131		(408) 923-5187	drussel@g
B	Laura, E	459 Clifton Ave	San Jose	CA	95128	(408) 995-5347		e_laura@g
A	George, A M	2 Cottage Pl Mount Kisco	Mount Kisco	N Y	10549	(914) 666-5688		

Figure 1-582 Create J19_UNDUP_IND_MATCH_MAILING 6/7


j19_UNDUP_IND_MATCH_MAILING..MAILING_UNDUP_IND_RESIDUALS.RESIDUAL - Data Browser

UnhandledData_USAREA	InputPattern_USAREA	ExceptionData_USAREA	UserOverrideFlag_USAREA	qsMatchDataID	qsMatchType
	+S^		NO	5	RA
	++S^		NO	8	RA
	++S^		NO	11	RA
	++S^		NO	14	RA
	++S^		NO	3	RA
	++S^		NO	6	RA
	+S^		NO	9	RA
	+SS^		NO	1	RA
	+S^		NO	4	RA
	++S^		NO	7	RA
	++S^		NO	10	RA
	++S^		NO	13	RA

Figure 1-583 Create J19_UNDUP_IND_MATCH_MAILING 7/7

1.10.8 Generate mailing master with household for promotion mailing

In this step, we generate a clean master of mailing list persons by merging the master and residual records that are generated in the "J19_UNDUP_IND_MATCH_MAILING" on page 468 step into a data set using the FUNNEL stage. We add a household ID (using a Transformer stage) to these records in the output with a value (qsMatchSetID) when a record belongs to a household and a zero when it does not belong to a household.

J20_MAILING_MASTER_WITH_HOUSEHOLD

Because this step is very similar to that described in "J17_CUSTOMER_MASTER_WITH_HOUSEHOLD" on page 327, we do not repeat those steps here. However, some of the key points of interest are as follows:

- Figure 1-584 on page 474 shows the various stages that are used in this job, including the two input data sets (MAILING_UNDUP_IND_MATCHES created in "J19_UNDUP_IND_MATCH_MAILING" on page 468 and MAILING_UNDUP_IND_RESIDUALS created in "J19_UNDUP_IND_MATCH_MAILING" on page 468), two Transformer stages (each adds a household identifier), one FUNNEL stage, and an output Data Set stage. We modified the names of the stages as shown, and this job uses the same QSPARAMETERSET object created earlier.
- The ADD_HOUSEHOLD_ID - Transformer Stage shows the addition of a column HOUSEHOLD_ID (Figure 1-585 on page 475) which has the value stored in the column qsMatchSetID in the Derivation Substitution column.
- The ADD_BLANK_HOUSEHOLD - Transformer Stage (Figure 1-586 on page 476) shows the addition of a column HOUSEHOLD_ID, which has the value zero in the Derivation Substitution column.
- After saving, compiling, and running this job J20_MAILING_MASTER_WITH_HOUSEHOLD (Figure 1-587 on page 477), view the content of the MAILING_MASTER_WITH_HOUSEHOLD data set object as shown in Figure 1-588 on page 477 and Figure 1-589 on page 478.

  This report shows the cleansed mailing list file with household information added. This file can then be used for promotional mailing

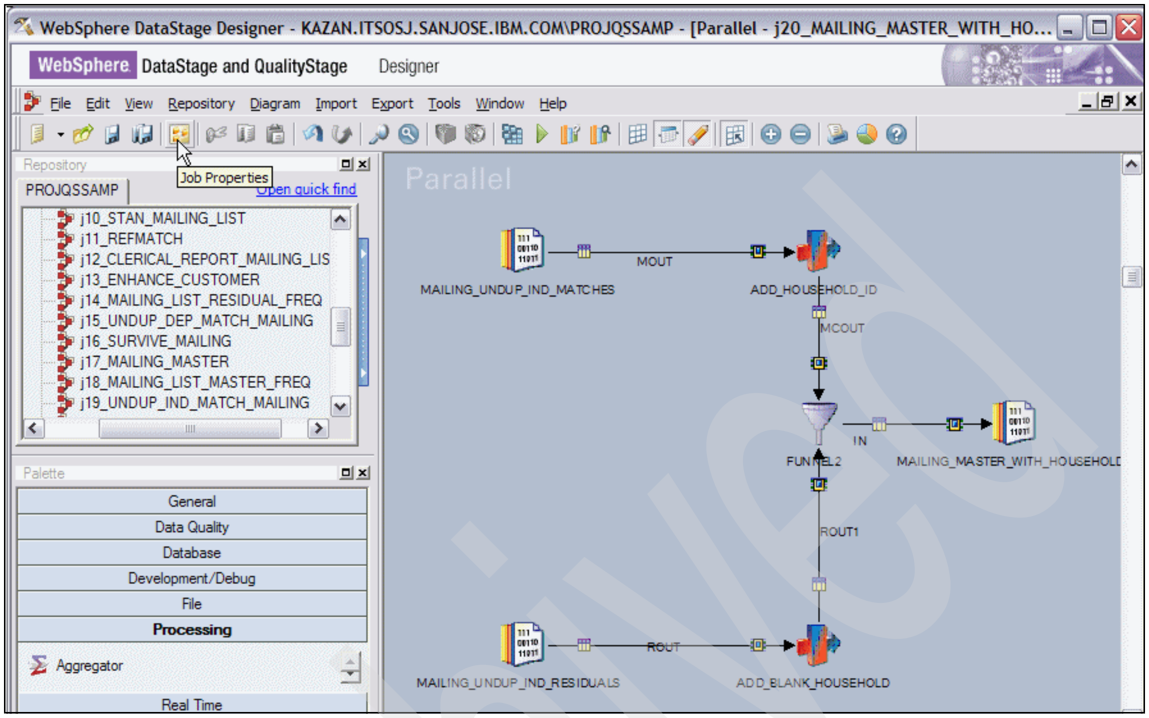
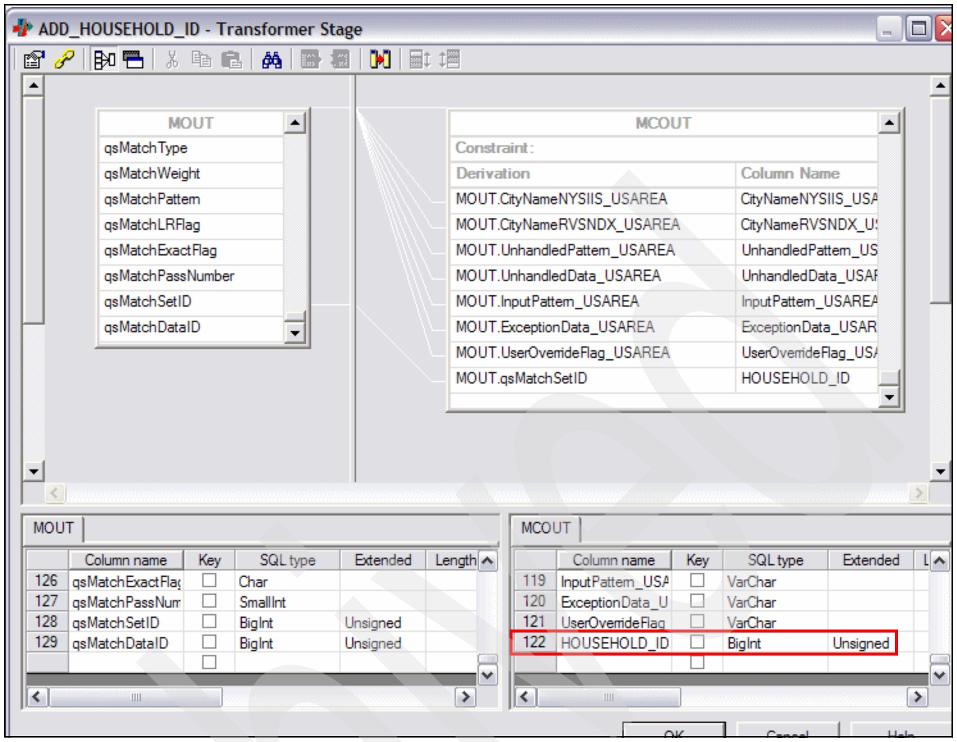

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j20_MAILING_MASTER_WITH_HO...
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Job Properties
Repository
PROJQSSAMP
Open quick find
j10_STAN_MAILING_LIST
j11_REFMATCH
j12_CLERICAL_REPORT_MAILING_LIS
j13_ENHANCE_CUSTOMER
j14_MAILING_LIST_RESIDUAL_FREQ
j15_UNDUP_DEP_MATCH_MAILING
j16_SURVIVE_MAILING
j17_MAILING_MASTER
j18_MAILING_LIST_MASTER_FREQ
j19_UNDUP_IND_MATCH_MAILING
Palette
General
Data Quality
Database
Development/Debug
File
Processing
Aggregator
Real Time
Parallel
MAILING_UNDUP_IND_MATCHES
MOUT
ADD_HOUSEHOLD_ID
MCOUT
FUNNEL2
IN
MAILING_MASTER_WITH_HOUSEHOLD
ROUT1
MAILING_UNDUP_IND_RESIDUALS
ROUT
ADD_BLANK_HOUSEHOLD

Figure 1-584 Create J20_MAILING_MASTER_WITH_HOUSEHOLD 1/6


ADD_HOUSEHOLD_ID - Transformer Stage

MOUT

qsMatchType
qsMatchWeight
qsMatchPattern
qsMatchLRFlag
qsMatchExactFlag
qsMatchPassNumber
qsMatchSetID
qsMatchDataID

MCOUT

Constraint:

Derivation	Column Name
MOUT.CityNameNYSIIS_USAREA	CityNameNYSIIS_USA
MOUT.CityNameRVSNDX_USAREA	CityNameRVSNDX_U
MOUT.UnhandledPattern_USAREA	UnhandledPattern_US
MOUT.UnhandledData_USAREA	UnhandledData_USAR
MOUT.InputPattern_USAREA	InputPattern_USAREA
MOUT.ExceptionData_USAREA	ExceptionData_USAR
MOUT.UserOverrideFlag_USAREA	UserOverrideFlag_US
MOUT.qsMatchSetID	HOUSEHOLD_ID

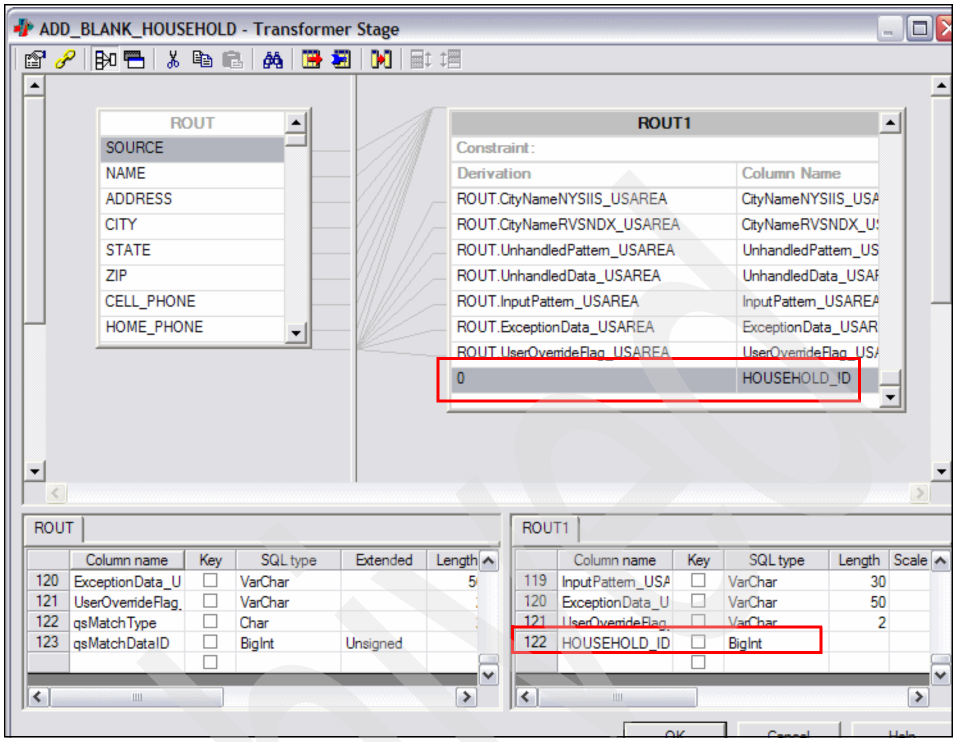
MOUT

	Column name	Key	SQL type	Extended	Length
126	qsMatchExactFlag	☐	Char		
127	qsMatchPassNum	☐	SmallInt		
128	qsMatchSetID	☐	BigInt	Unsigned	
129	qsMatchDataID	☐	BigInt	Unsigned	

MCOUT

	Column name	Key	SQL type	Extended
119	InputPattern_USA	☐	VarChar	
120	ExceptionData_U	☐	VarChar	
121	UserOverrideFlag	☐	VarChar	
122	HOUSEHOLD_ID	☐	BigInt	Unsigned

Figure 1-585 Create J20_MAILING_MASTER_WITH_HOUSEHOLD 2/6


ADD_BLANK_HOUSEHOLD - Transformer Stage

ROUT
SOURCE
NAME
ADDRESS
CITY
STATE
ZIP
CELL_PHONE
HOME_PHONE

ROUT1

Constraint:

Derivation	Column Name
ROUT.CityNameNYSIIS_USAREA	CityNameNYSIIS_USA
ROUT.CityNameRVSNDX_USAREA	CityNameRVSNDX_U
ROUT.UnhandledPattern_USAREA	UnhandledPattern_US
ROUT.UnhandledData_USAREA	UnhandledData_USAI
ROUT.InputPattern_USAREA	InputPattern_USAREA
ROUT.ExceptionData_USAREA	ExceptionData_USAR
ROUT.UserOverrideFlag_USAREA	UserOverrideFlag_USA
0	HOUSEHOLD_ID

ROUT

	Column name	Key	SQL type	Extended	Length
120	ExceptionData_U	☐	VarChar		5
121	UserOverrideFlag_	☐	VarChar		
122	qsMatchType	☐	Char		
123	qsMatchDataID	☐	BigInt	Unsigned	
		☐			

ROUT1

	Column name	Key	SQL type	Length	Scale
119	InputPattern_USA	☐	VarChar	30	
120	ExceptionData_U	☐	VarChar	50	
121	UserOverrideFlag_	☐	VarChar	2	
122	HOUSEHOLD_ID	☐	BigInt		
		☐			

Figure 1-586 Create J20_MAILING_MASTER_WITH_HOUSEHOLD 3/6

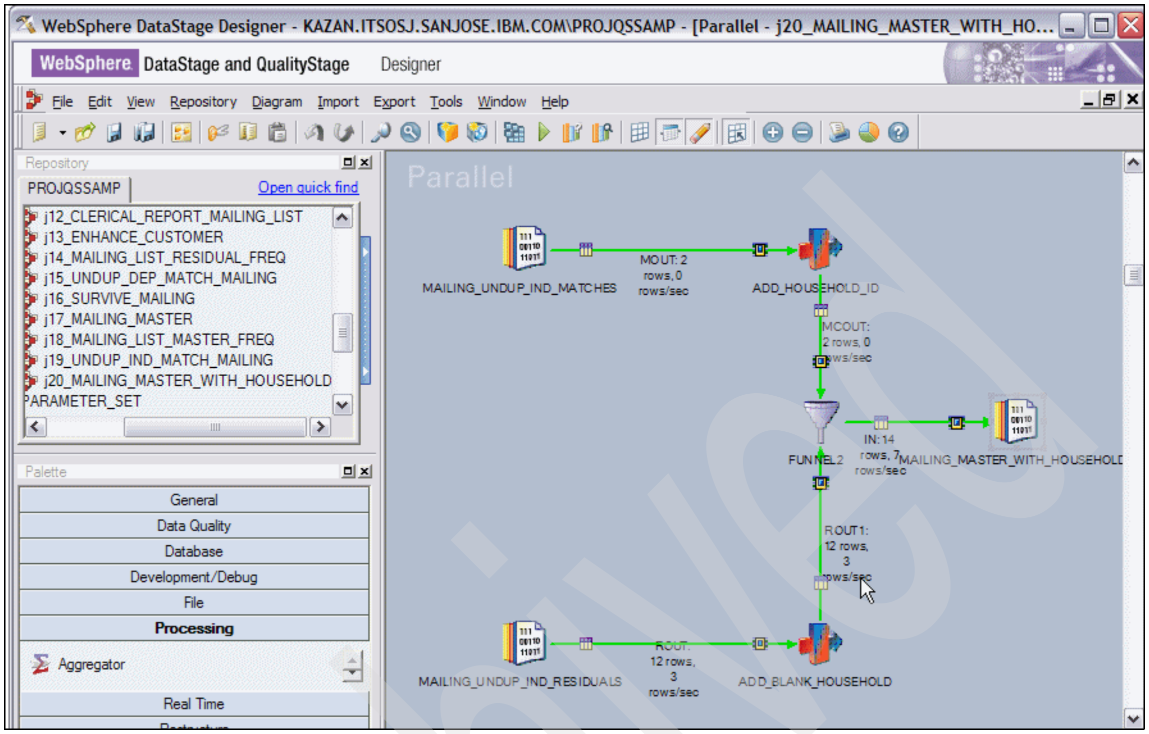
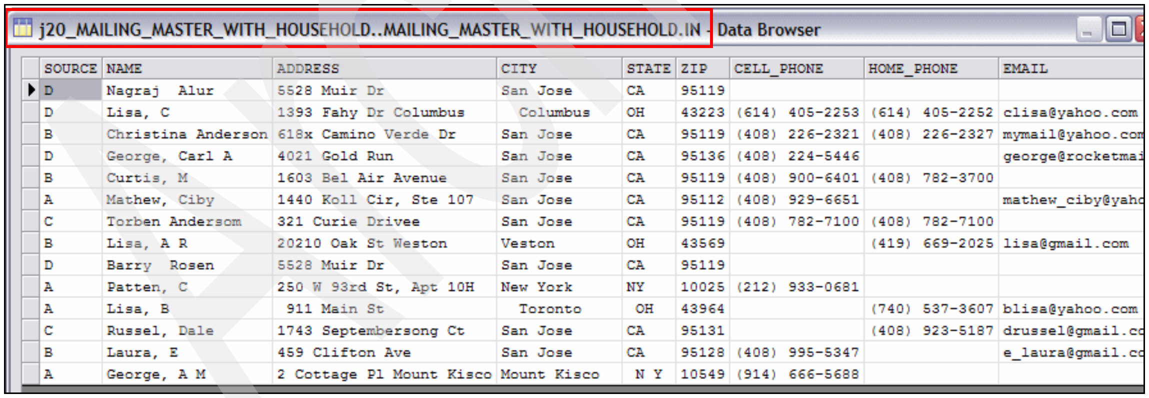
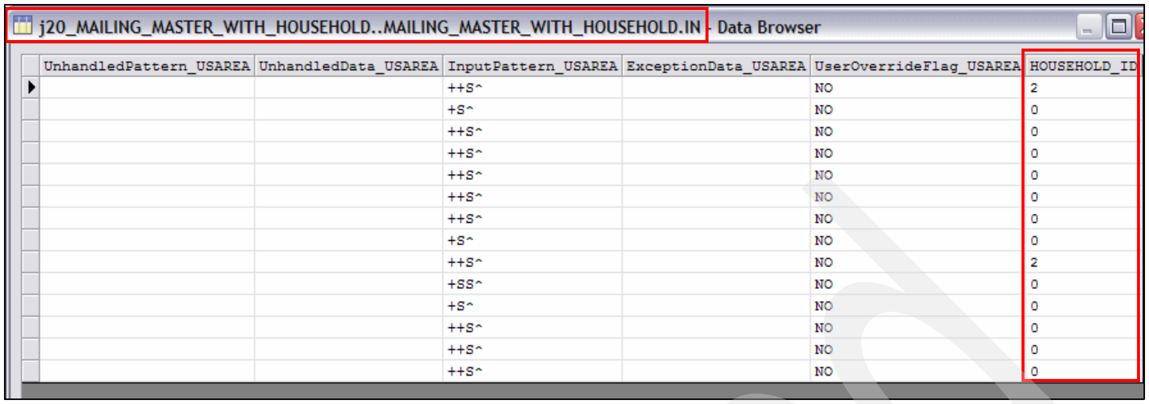

WebSphere DataStage Designer - KAZAN.ITSOSJ.SANJOSE.IBM.COM\PROJQSSAMP - [Parallel - j20_MAILING_MASTER_WITH_HO...
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Repository
PROJQSSAMP
Open quick find
j12_CLERICAL_REPORT_MAILING_LIST
j13_ENHANCE_CUSTOMER
j14_MAILING_LIST_RESIDUAL_FREQ
j15_UNDUP_DEP_MATCH_MAILING
j16_SURVIVE_MAILING
j17_MAILING_MASTER
j18_MAILING_LIST_MASTER_FREQ
j19_UNDUP_IND_MATCH_MAILING
j20_MAILING_MASTER_WITH_HOUSEHOLD
PARAMETER_SET
Palette
General
Data Quality
Database
Development/Debug
File
Processing
Aggregator
Real Time
Parallel
MAILING_UNDUP_IND_MATCHES
MOUT: 2 rows, 0 rows/sec
ADD_HOUSEHOLD_ID
MCOUT: 2 rows, 0 rows/sec
FUNNEL2
IN: 14 rows, 7 rows/sec
MAILING_MASTER_WITH_HOUSEHOLD
ROUT1: 12 rows, 3 rows/sec
MAILING_UNDUP_IND_RESIDUALS
ROUT: 12 rows, 3 rows/sec
ADD_BLANK_HOUSEHOLD

Figure 1-587 Create J20_MAILING_MASTER_WITH_HOUSEHOLD 4/6


j20_MAILING_MASTER_WITH_HOUSEHOLD..MAILING_MASTER_WITH_HOUSEHOLD.IN - Data Browser

SOURCE	NAME	ADDRESS	CITY	STATE	ZIP	CELL_PHONE	HOME_PHONE	EMAIL
D	Nagraj Alur	5528 Muir Dr	San Jose	CA	95119			
D	Lisa, C	1393 Fahy Dr Columbus	Columbus	OH	43223	(614) 405-2253	(614) 405-2252	clisa@yahoo.com
B	Christina Anderson	618x Camino Verde Dr	San Jose	CA	95119	(408) 226-2321	(408) 226-2327	mymail@yahoo.com
D	George, Carl A	4021 Gold Run	San Jose	CA	95136	(408) 224-5446		george@rocketmai
B	Curtis, M	1603 Bel Air Avenue	San Jose	CA	95119	(408) 900-6401	(408) 782-3700	
A	Mathew, Ciby	1440 Koll Cir, Ste 107	San Jose	CA	95112	(408) 929-6651		mathew_ciby@yahc
C	Torben Andersom	321 Curie Drivee	San Jose	CA	95119	(408) 782-7100	(408) 782-7100	
B	Lisa, A R	20210 Oak St Weston	Veston	OH	43569		(419) 669-2025	lisa@gmail.com
D	Barry Rosen	5528 Muir Dr	San Jose	CA	95119			
A	Patten, C	250 W 93rd St, Apt 10H	New York	NY	10025	(212) 933-0681		
A	Lisa, B	911 Main St	Toronto	OH	43964		(740) 537-3607	blisa@yahoo.com
C	Russel, Dale	1743 Septembersong Ct	San Jose	CA	95131		(408) 923-5187	drussel@gmail.cc
B	Laura, E	459 Clifton Ave	San Jose	CA	95128	(408) 995-5347		e_laura@gmail.cc
A	George, A M	2 Cottage Pl Mount Kisco	Mount Kisco	N Y	10549	(914) 666-5688		

Figure 1-588 Create J20_MAILING_MASTER_WITH_HOUSEHOLD 5/6



j20_MAILING_MASTER_WITH_HOUSEHOLD..MAILING_MASTER_WITH_HOUSEHOLD.IN - Data Browser

UnhandledPattern_USAREA	UnhandledData_USAREA	InputPattern_USAREA	ExceptionData_USAREA	UserOverrideFlag_USAREA	HOUSEHOLD_ID
		++S^		NO	2
		+S^		NO	0
		++S^		NO	0
		++S^		NO	0
		++S^		NO	0
		++S^		NO	0
		++S^		NO	0
		+S^		NO	0
		++S^		NO	2
		+SS^		NO	0
		+S^		NO	0
		++S^		NO	0
		++S^		NO	0
		++S^		NO	0

Figure 1-589 Create J20_MAILING_MASTER_WITH_HOUSEHOLD 6/6

2

Financial services business scenario

Note: The scenario that we present in this chapter is based upon two fictitious financial institutions. In addition, the data that we present in our model is fictional also. Any resemblance to real institutions or data is totally coincidental.

In this chapter we describe a step-by-step approach to implementing IBM WebSphere QualityStage on a Red Hat Advanced Server Enterprise 4.0 platform using a typical financial services business scenario.

The topics covered include:

- ► Business requirement
- ► Environment configuration
- ► General approach
- ► Scope of this book
- ► Data integration scenario
- ► Post migration scenario

2.1 Introduction

North American Bank provides core banking services such as savings, checking, and auto and home loans in North America and has significant market share in the eastern and midwest regions of the U.S. It additionally provides credit card and auto insurance services to its customer base. The primary IT platform of this bank is z/OS with DB2 and VSAM data sources.

Northern California Bank is a regional bank that provides core banking services such as savings, checking, and auto and home loans in the western region of the U.S. and has significant market share in that region. It also provides brokerage services to its customer base. The primary IT platform of this bank is AIX with DB2 UDB as the data source.

Seeking a synergistic relationship, the two banks entered into a merger with the intent of becoming a national bank by growing their individual customer bases, and upselling and cross selling each others products to their individual customers.

The two banks expect there to be some overlap of customers or groups of customers (such as members of a single family), who when identified could be granted special status. However, the two customer bases are mostly expected to be non-overlapped.

2.2 Business requirement

With the objective of streamlining the combined IT operations of the two banks and taking advantage of the synergy of the products that are offered by the individual banks in the most expeditious manner, management made a business decision to implement the following strategy:

- Migrate the core services such as saving, checking, and auto and home loans, from the North American Bank system implementations to the those implemented by the Northern California Bank (generally considered to be superior in function and architecture).

Important: The general guideline when mismatches are found in the migration scenario, is to have the target system have the overriding authority and not be modified except in absolutely unavoidable situations. This guideline means ignoring data elements in the source that have no correspondence in the target, to truncate source data element content when the target data element's precision is lesser than that of the source, and to accept coarser granularity data content values supported by the target where codes are involved. Judicious decisions involving minimal modifications to the target system should be the norm when mismatches are encountered.

- Build a customer relationship management (CRM) system to take advantage quickly of the individual brokerage, credit card, and auto insurance businesses of each bank to upsell and cross-sell these products throughout the customer base.

  The CRM design is an "off-the-shelf" design that is customized to suit the particular requirements of the merged bank. It integrates information from the following sources with no changes in the interim to any of the existing systems:

  – Core business services from both the Northern California Bank and the North American Bank
  – Brokerage services from the Northern California Bank
  – Credit card and auto insurance services from the North American Bank

Important: The general guideline during data integration is that when there are mismatches between the sources and the new CRM system, that the CRM system design be modified to accommodate the existing system functions that are considered essential.

- A decision on less essential services such as Human Resources systems in each bank is deferred until the completion of the migration and CRM system implementation.
- The migration and CRM implementation efforts are to proceed in parallel with no dependency of one over the other.

  Therefore, the CRM system is designed to be sourced from the core services of the individual banks even as migration of these systems is occurring from one bank to the other. At the completion of migration of the core business services, it would be easy to have the CRM sourced entirely from the migrated system only.

The data models of the systems in our scenario are as follows:

- The data model of the North American Bank is shown in Figure 2-1 on page 483. The DDL and field in the VSAM file are shown in Example B-2 on page 897 and Example B-3 on page 903.
- The data model of the Northern California Bank is shown in Figure 2-2 on page 484. The DDL is shown in Example B-1 on page 888.
- The “off-the-shelf” data model of the CRM is shown in Figure 2-3 on page 485. The DDL is shown in Example B-4 on page 904.

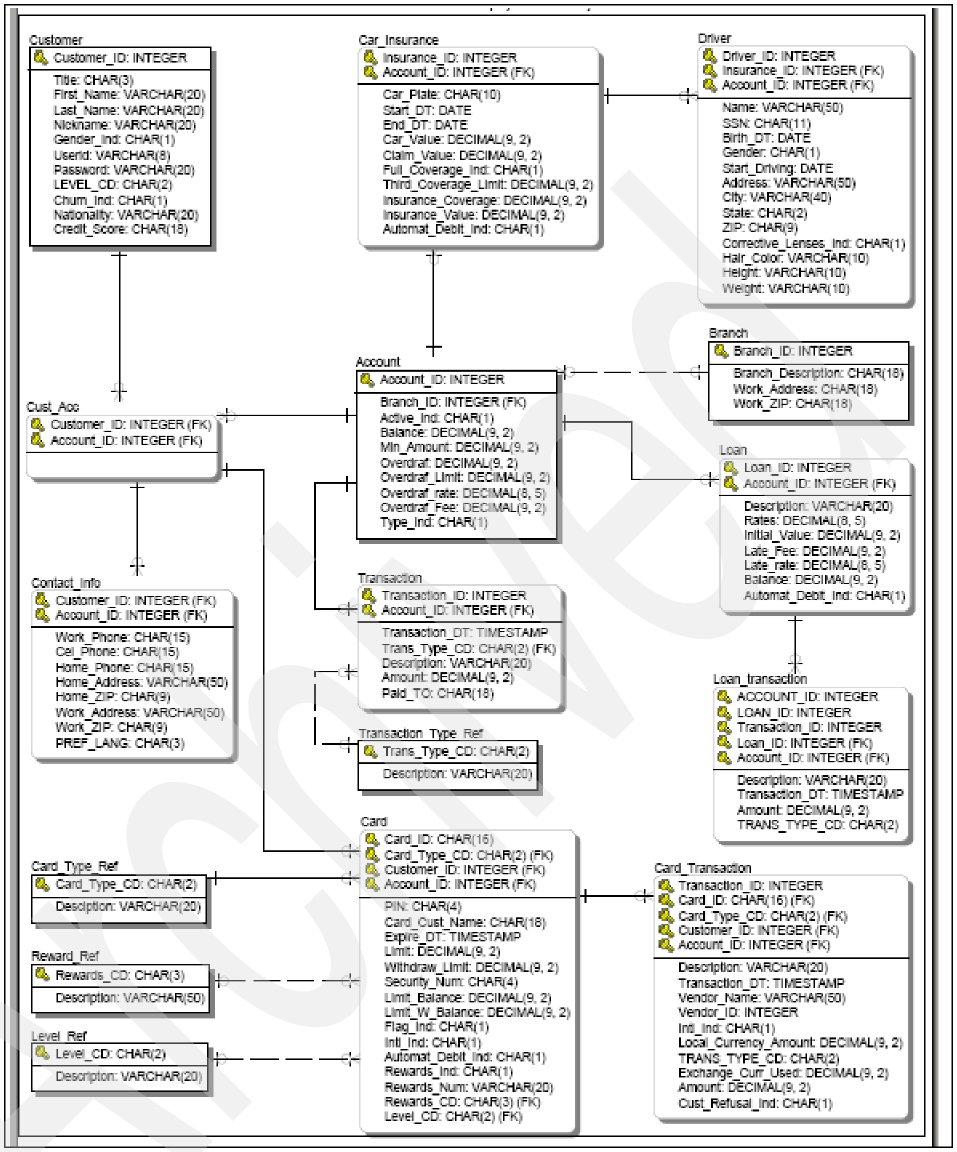


Figure 2-1 Data model of the North American Bank

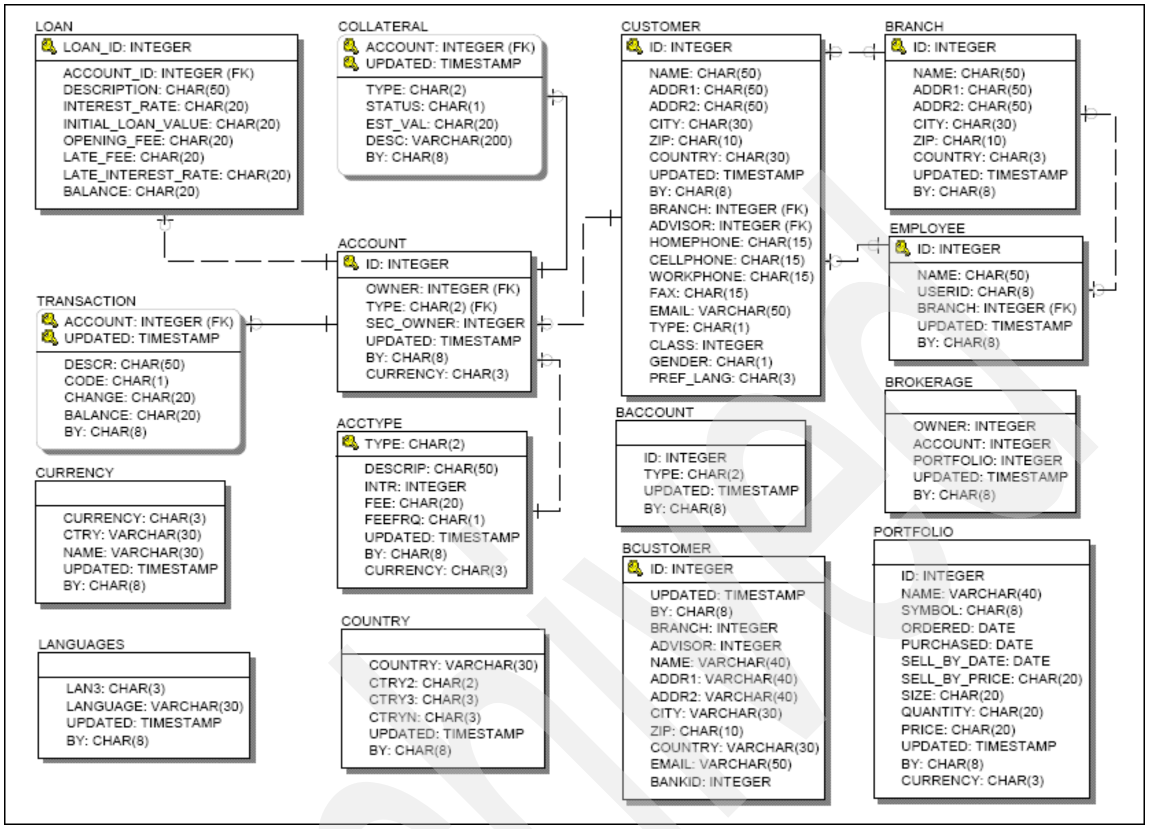


Figure 2-2 Data model of the Northern California Bank

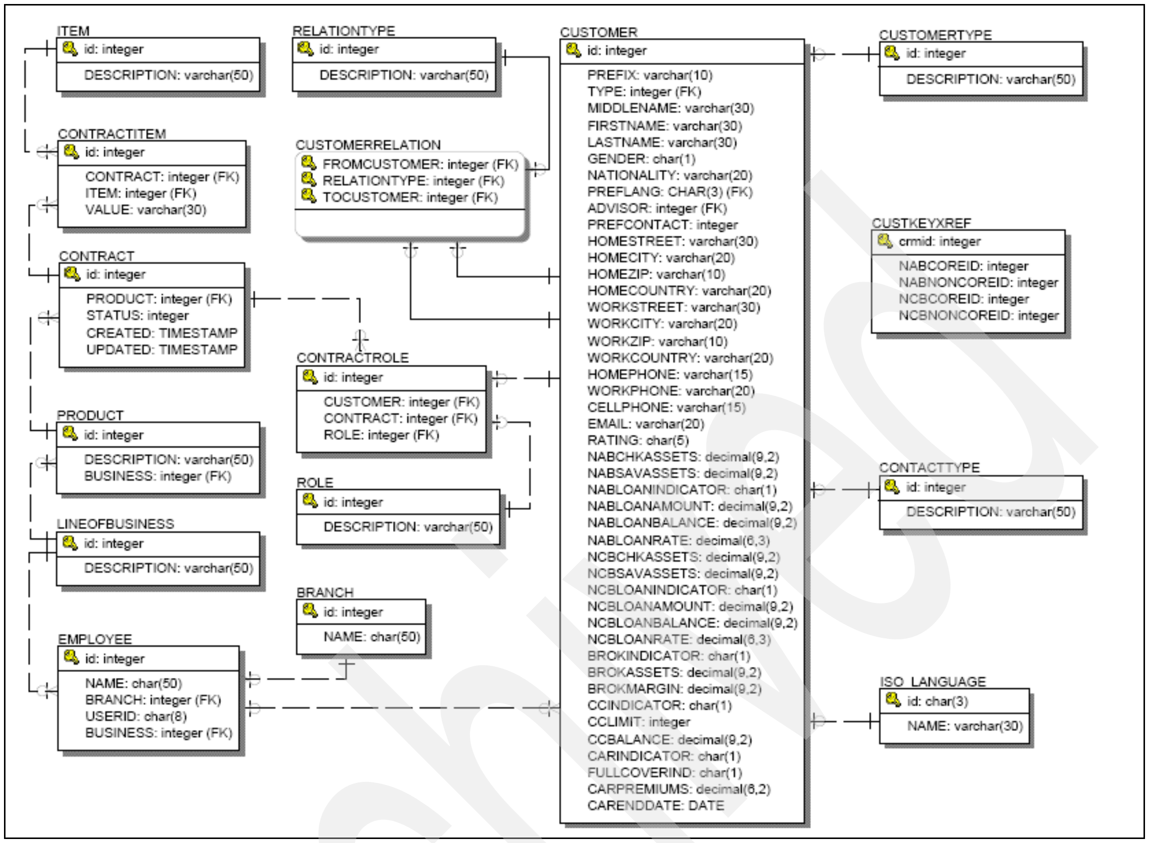


Figure 2-3 "Off-the-shelf" data model of the CRM

2.3 Environment configuration

Figure 2-4 shows the configuration of the merged environment with the new CRM system.

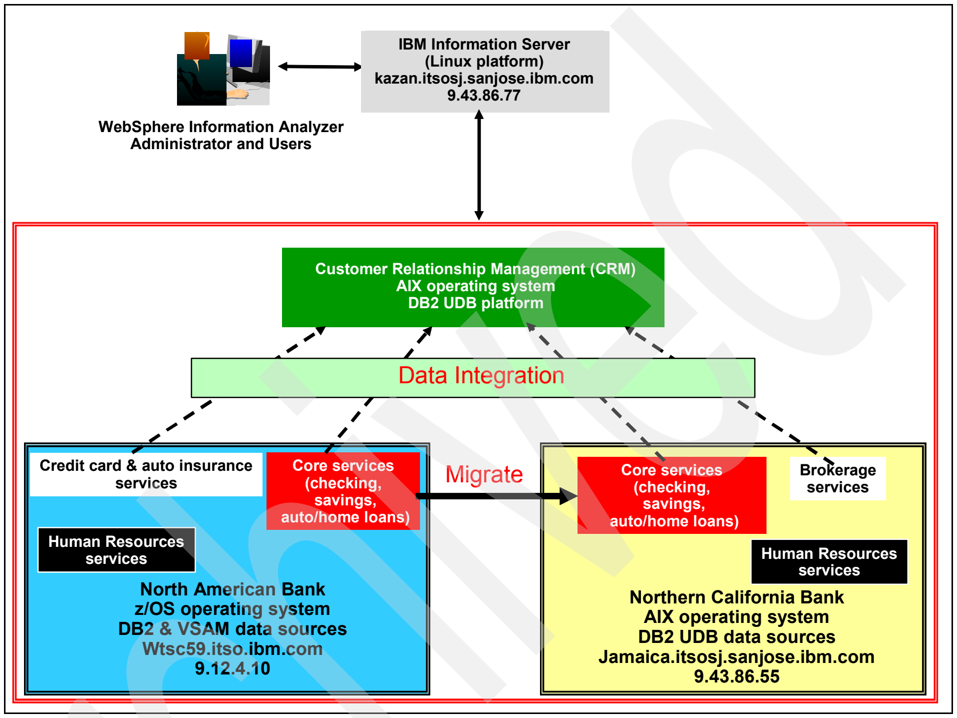

WebSphere Information Analyzer Administrator and Users
IBM Information Server (Linux platform) kazan.itsosj.sanjose.ibm.com 9.43.86.77
Customer Relationship Management (CRM) AIX operating system DB2 UDB platform
Data Integration
Credit card & auto insurance services
Core services (checking, savings, auto/home loans)
Human Resources services
North American Bank z/OS operating system DB2 & VSAM data sources Wtsc59.itso.ibm.com 9.12.4.10
Migrate
Core services (checking, savings, auto/home loans)
Brokerage services
Human Resources services
Northern California Bank AIX operating system DB2 UDB data sources Jamaica.itsosj.sanjose.ibm.com 9.43.86.55

Figure 2-4 Merged banks' environment configuration

Figure 2-4 shows:

- A Red Hat Enterprise Linux 4 server (kazan.itsosj.sanjose.ibm.com — 9.43.86.77) that has IBM Information Server and IBM WebSphere Information Analyzer installed.
- A single IBM AIX 5.2 server (Jamaica.itsosj.sanjose.ibm.com — 9.43.86.55) runs the Northern California Bank's IT systems, including the core services, brokerage services, and human resources services. DB2 V9.1 is used for the data sources.

Attention: The CRM system is meant to be implemented on the IBM AIX 5.2 server. In this publication, we do not actually build the CRM system—we merely take note of the CRM data model to ensure that it is capable of supporting the data content, data types, precision, and scale of the data to be integrated from the core services, credit card, auto insurance services, and brokerage services of the two banks.

- A single z/OS image (wtsc59.itso.ibm.com — 9.12.4.10) that runs the North American Bank's IT systems including the core services, credit card, auto insurance services, and human resources services. DB2 for z/OS V8 and VSAM is used for the data sources. WebSphere Information Integrator Classic Federation (IICF) V9.1 provides the connectivity from IBM WebSphere Information Analyzer to the VSAM data source and the DB2 API stage provides connectivity to DV2 for z/OS V8.

Attention: Our configuration in Figure 2-4 is meant to represent the eclectic mix of operating systems and platforms that are typical of mergers between multiple organizations and describes how IBM Information Server and IBM WebSphere Information Analyzer integrate into such an environment. We are *not* making recommendations about how to configure your environment in this manner or that the configuration will deliver the scalability and performance requirements of your business solution.

2.4 General approach

Figure 2-5 shows the recommended sequence of steps in a migration or data integration scenario. IBM WebSphere Information Analyzer plays the key role in the second step that involves identifying the differences (both structure and content) between the sources and targets.

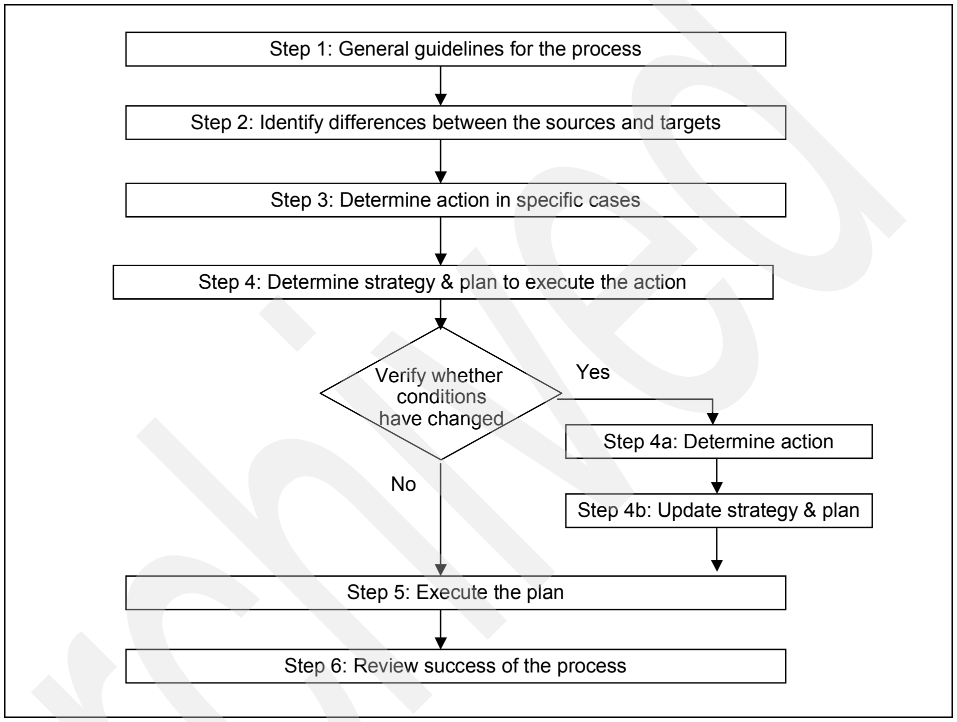

Step 1: General guidelines for the process
Step 2: Identify differences between the sources and targets
Step 3: Determine action in specific cases
Step 4: Determine strategy & plan to execute the action
Verify whether conditions have changed
Yes
No
Step 4a: Determine action
Step 4b: Update strategy & plan
Step 5: Execute the plan
Step 6: Review success of the process

Figure 2-5 General approach

We describe each of the steps shown in Figure 2-5 briefly in this section.

2.4.1 Step 1: General guidelines for the process

In a migration or a data integration effort, structural and content differences between the sources and targets will need to be resolved. Table 2-1 shows many of the commonly encountered differences and potential actions. These action are self-explanatory.

We recommend that you define broad action guidelines for mismatches in critical and non-critical elements between the data sources and targets for the commonly encountered differences. Consequently, if unexpected differences are subsequently discovered, you can focus your energies on addressing them in the inevitable time constrained circumstances of migration and data integration projects.

Table 2-1 Commonly encountered differences and potential actions

Commonly encountered differences	**Potential actions**
Data elements in the source not found in the target	► Add the data elements to the target; significant impact on target applications likely ► Ignore it; loss of function ► Combination of the above depending upon the data element
Data type mismatch between the source and target data elements ► Compatible ► Incompatible ► Coarse to fine ► Fine to coarse	► Compatible – Simple mapping ► Incompatible – Use a cross reference table to map from the source data type to the target data type ► Coarse to fine – No action other than possible transformation ► Fine to coarse – Modify target definition to match fine granularity of the source; significant impact on target applications likely – Transform with loss of granularity; loss of function
Precision, and scale mismatch between the source and the target data elements ► Coarse to fine ► Fine to coarse	► Coarse to fine – No action other than possible transformation ► Fine to coarse – Modify target definition to match fine granularity of the source; significant impact on target applications likely – Transform with loss of granularity; loss of function
Data elements in the target not found in the source, and target data element not nullable or has no default values	► Define default values in the target; some impact on target applications

Commonly encountered differences	Potential actions
Code mismatch between the source and the target data elements; for example, a salutation can be *Mr*, *Mrs*, *Dr*, *Miss*, and *Ms*, and source and target do not have corresponding codes ► Coarse to fine ► Fine to coarse	► Coarse to fine – Use transformation to perform the mapping ► Fine to coarse – Transform with loss of granularity; loss of function – Modify target definition to match fine granularity of the source; significant impact on target applications likely
Multiple data elements in the source maps to a single data element in the target ► For example, an address	► Use transformation to perform the mapping – Might require standardization software
Single data element in the source maps to multiple data elements in the target ► For example, an address	► Use transformation to perform the mapping – Might require standardization software
Different character maps such as Unicode and ASCII	Transformation to perform the mapping

2.4.2 Step 2: Identify differences between the sources and targets

Differences between the source and target can be determined as follows:

- ► From active relational database catalogs, dictionaries, repositories, and other documentation that provide details about the metadata of the various data sources and targets.

  More often than not, metadata information stored in sources (other than the active relational database catalogs) tends to be out of date because it is not maintained regularly as systems evolve.

- ► From an analysis of the data itself using tools such as IBM WebSphere Information Analyzer. Metadata is deduced from the data and presented to the data analyst for review and affirmation or denial of the deduction.

These are complementary approaches, essential to achieving a fuller understanding of how synchronized the definition of the metadata is with the data content. It also enables you to keep the metadata about data sources current, which is critical for building new systems that require data integration from existing systems.

Attention: We strongly recommend that tools such as IBM WebSphere Information Analyzer sharply focus analyses on data content in specific tables and columns based on metadata information obtained from active relational database catalogs, dictionaries, repositories, and other documentation. This will avoid unnecessary data analysis by data analysts of data that is not appropriate for such an analysis. An added benefit is limiting unnecessary and irrelevant processing that could consume valuable processing power that would be best consumed by other applications.

IBM WebSphere Information Analyzer identifies differences between the defined metadata for a data source and the inferences made from the actual data content in these data sources.

Comparing independently generated IBM WebSphere Information Analyzer analyses of different data sources is beyond the scope of IBM WebSphere Information Analyzer and is likely a manual process.

2.4.3 Step 3: Determine action in specific cases

After you compare the metadata and inferred metadata from the data content, you can choose to synchronize them with appropriate actions, such as modifying the metadata definitions, cleansing the data, or ignoring them at your own peril.

In the migration and data integration scenarios that we discuss in this chapter, we compare the metadata and data content between multiple data sources manually. Based on the identified differences with specific data elements, we choose the most acceptable action from the available options, some of which are described in Table 2-1 on page 489.

2.4.4 Step 4: Determine strategy and plan to execute action

After you have chosen the action, you need to design the appropriate tools and procedures to effect the chosen action for each data element. This design will most likely involve the use of data cleansing tools (such as IBM WebSphere QualityStage), ETL[1] tools (such as IBM WebSphere DataStage), data management and database tools (such as DB2 Data Warehouse Edition) where stored procedures or specific database functions are involved, and user-written code.

The execution of the plan would most likely take an extended period of many hours or days. To ensure that migration or data integration occurs without

[1] Extract, transform, and load

disrupting the availability of the source applications, you will most likely synchronize the source and target in multiple phases. A snapshot of the source is initially bulk loaded into the target, followed by an incremental update of the target with changes occurring in the source during the bulk load.

A well planned and rigorously tested set of procedures is essential for a smooth and successful migration or data integration project.

A discussion of this topic is beyond the scope of this book. However, it would be useful to note that IBM WebSphere Information Analyzer can play an active part in the rigorous testing of the process, which can occur at multiple points as follows:

- Assessment of extracted test data to ensure conformity to defined test and transformation objectives.
- Evaluation and comparison of outputs from the development process to ensure broad and quick review.
- Validation of test/QA output against test objectives, particularly for cross-domain comparison of source input to target output and proper data mapping.
- Review of target load files to ensure completeness, comprehensiveness, and consistency to expected results.

In all cases, IBM WebSphere Information Analyzer can facilitate this work by providing rapid insight into the data.

Important: The analysis of data can be expected to take several weeks given the fact that multiple personnel IT Data Analysts (DA) and subject matter experts (SME) are involved in ensuring data quality. Therefore, it is conceivable that changes could occur to the metadata and data content of the data sources and targets during this interval. Therefore, prior to executing the plan, you should check if structure or content changes have occurred. If so, you should determine its impact on the existing plan, update the strategy and plan if required, and test the revised plan before execution.

2.4.5 Step 5: Execute the plan

This step involves executing the plan that was designed in 2.4.4, “Step 4: Determine strategy and plan to execute action” on page 491.

2.4.6 Step 6: Review success of the process

After the designed plan is executed, you need to verify that the process was successful by comparing the metadata and content in the sources and targets.

The process for doing this varies depending upon whether a migration[2] or data integration[3] was involved.

A discussion of this topic is beyond the scope of this book.

2.5 Scope of this book

In this book, we assume that you have completed data profiling successfully and that you have performed an analysis of the results. We also assume that you have specified the actions to be taken when (metadata and content) differences between source and target are identified. When the differences are identified, appropriate actions to be taken are discussed and decided. For details on the data profiling activities and analyses performed, refer to *IBM WebSphere Information Analyzer and Data Quality Assessment*, SG24-7508.

In this book, we describe a step-by-step approach to standardizing names and addresses that are involved in the data integration and post migration scenarios as follows:

- **Data integration**

  After completing data profiling and analyses of North American Bank and Northern California Bank systems, the cleansing of unstructured fields that contain names and addresses and subsequent matching needs to be completed before the data can be integrated into the Customer Relationship Management (CRM) system using IBM WebSphere DataStage.

  In this book, we describe the steps that are involved in standardizing the names and addresses and matching records in order to eliminate duplicates as well as to establish relationships between customers to determine high and low value customers.

  Note: The actual IBM WebSphere DataStage jobs to integrate cleansed data from the North American Bank and Northern California Bank systems into the CRM system using IBM WebSphere DataStage will be covered in an upcoming IBM Redbooks publication.

[2] One-time activity

[3] On-going activity, because data is fed continuously to the target and changes to both the metadata and data content occur at the sources

- **Post migration**

  After the migration of the core services of the North American Bank (z/OS platform) to that of the Northern California Bank (AIX), we show how to use IBM WebSphere QualityStage to cleanse names and addresses in the migrated system.

  This post migration step involves standardizing names and addresses and identifying potential duplicates.

  Note: The actual modification of the migrated system occurs using IBM WebSphere DataStage and will be covered in an upcoming IBM Redbooks publication.

2.6 Data integration of North American Bank and Northern California Bank systems

A business decision was made to integrate the core (savings, checking, and loans) and non-core services (credit card and auto insurance) of the North American Bank on the z/OS platform with the core (savings, checking, and loans) and non-core services (brokerage) of the Northern California Bank on the AIX platform into the CRM system.

The CRM system provides an integrated view of all the customers in the merged bank, including details of the core services (such as checking, savings, and loans) and non-core services products (such as credit card or brokerage services) consumed. The objective of the CRM system is to upsell and cross sell the merged banks' products and services to the customer base. The management of the products and services sold continues to be in the original source system. In other words, the systems that represent the core and non-core services remain in their existing environment and are merely referenced and invoked from the CRM system. Other details include:

- The CRM system does not contain any transactions from the core and non-core services of the North American Bank and the Northern California Bank.
- New data elements exist in the CRM data model that do not exist in the existing systems of the North American Bank and Northern California Bank.
- Surrogate keys are used in the CRM—the keys in the existing systems are mapped to the surrogate keys.
- Customers in the North American Bank might have core and non-core services accounts.

- Customers in the Northern California Bank might have core and non-core services accounts.
- There is some marginal overlap of customers in both the North American Bank and Northern California Bank.

Figure 2-6 on page 496 through Figure 2-8 on page 500 show the jobs that are used to cleanse and match name and address information in the North American Bank and Northern California Bank core and non-core services. It describes:

- North American Bank processing shown in Figure 2-6 on page 496

  Name and address information in the North American Bank core and non-core services is spread across multiple tables such as CUSTOMER, CONTACT_INFO, and DRIVER. It also has two addresses: a home address and a work address.

  – Name information primarily exists in the CUSTOMER and DRIVER tables.

  – Address information primarily exists in the CONTACT_INFO (home and work addresses) and DRIVER (home address only) tables.

  Note: The steps and jobs used to cleanse the North American Bank's names and addresses are similar to those described in 1.10, "Mailing list scenario" on page 117. Therefore, we cross reference these steps as much as possible to avoid duplication. However, we do discuss any differences, such as configuration options and report highlights.

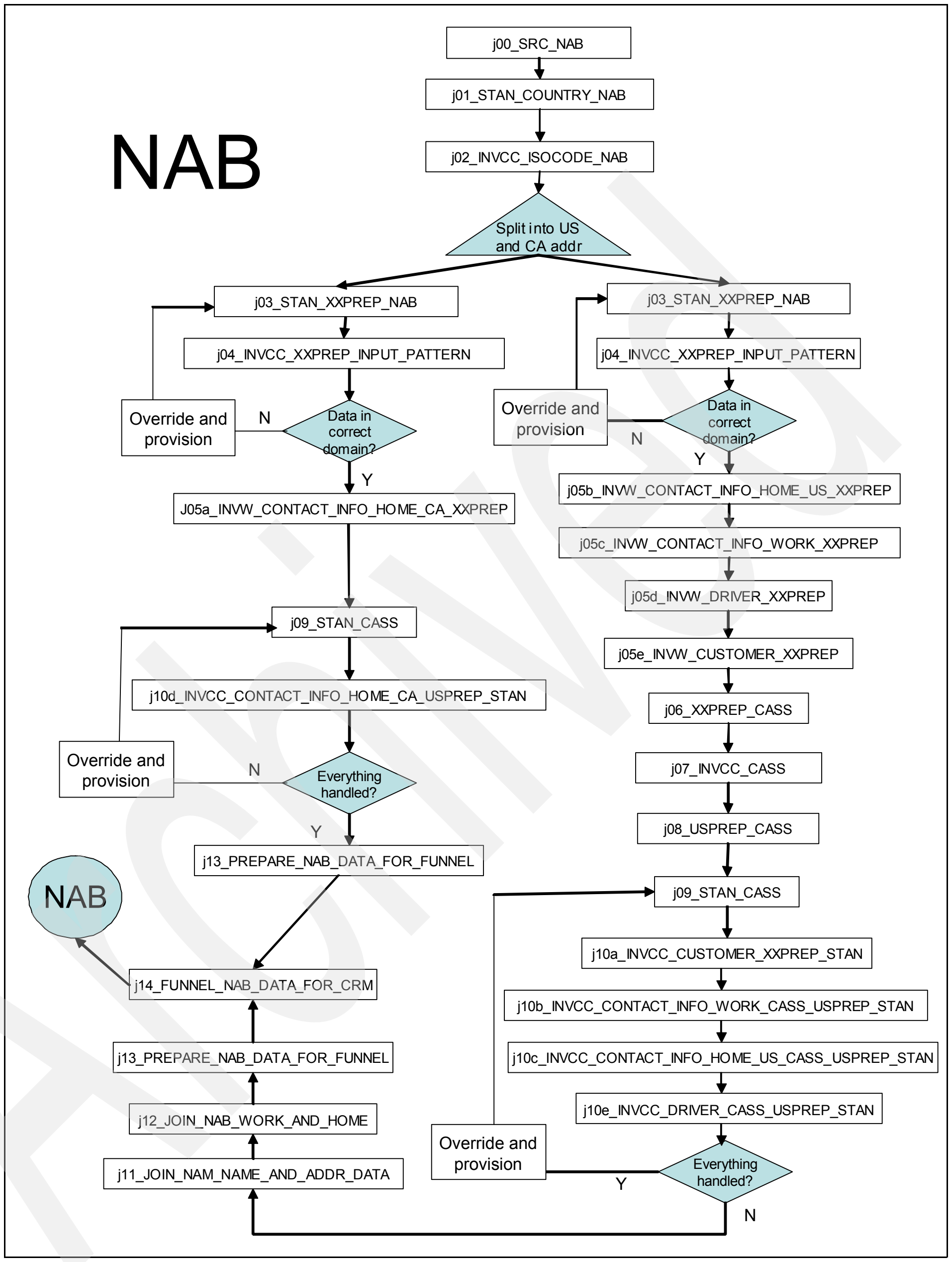

NAB
j00_SRC_NAB
j01_STAN_COUNTRY_NAB
j02_INVCC_ISOCODE_NAB
Split into US and CA addr
j03_STAN_XXPREP_NAB
j04_INVCC_XXPREP_INPUT_PATTERN
Data in correct domain?
N
Y
Override and provision
J05a_INVW_CONTACT_INFO_HOME_CA_XXPREP
j09_STAN_CASS
j10d_INVCC_CONTACT_INFO_HOME_CA_USPREP_STAN
Everything handled?
j13_PREPARE_NAB_DATA_FOR_FUNNEL
j14_FUNNEL_NAB_DATA_FOR_CRM
j12_JOIN_NAB_WORK_AND_HOME
j11_JOIN_NAM_NAME_AND_ADDR_DATA
j03_STAN_XXPREP_NAB
j04_INVCC_XXPREP_INPUT_PATTERN
j05b_INVW_CONTACT_INFO_HOME_US_XXPREP
j05c_INVW_CONTACT_INFO_WORK_XXPREP
j05d_INVW_DRIVER_XXPREP
j05e_INVW_CUSTOMER_XXPREP
j06_XXPREP_CASS
j07_INVCC_CASS
j08_USPREP_CASS
j09_STAN_CASS
j10a_INVCC_CUSTOMER_XXPREP_STAN
j10b_INVCC_CONTACT_INFO_WORK_CASS_USPREP_STAN
j10c_INVCC_CONTACT_INFO_HOME_US_CASS_USPREP_STAN
j10e_INVCC_DRIVER_CASS_USPREP_STAN
Everything handled?
Override and provision

Figure 2-6 Data integration cleansing tasks for North American Bank

- Northern California Bank processing shown in Figure 2-7 on page 498

  Name and address information in the Northern California Bank core and non-core services is spread across multiple tables such as BRANCH, CUSTOMER, and BCUSTOMER. The BRANCH table has the branch name and business address, while the CUSTOMER and BCUSTOMER tables have individual name and home addresses.

  Note: The steps and jobs used to cleanse the Northern California Bank's names and addresses are similar to those described in 1.10, "Mailing list scenario" on page 117. Therefore, we cross reference these steps as much as possible to avoid duplication. However, we do discuss any differences, such as configuration options and report highlights.

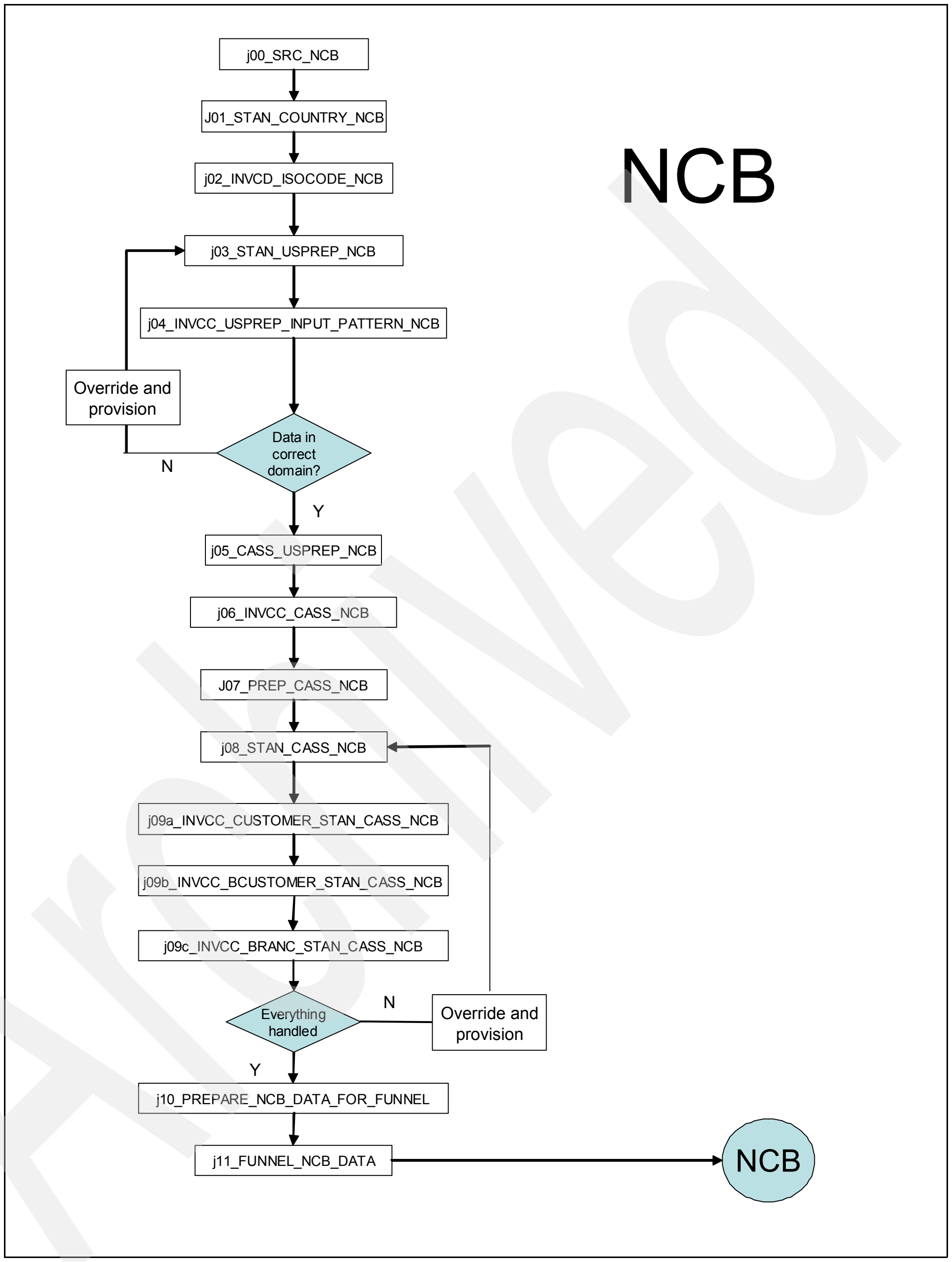


Figure 2-7 Data integration cleansing tasks for Northern California Bank

- Matching North American Bank and Northern California Bank cleansed data and surviving the “best”[4] data from these two banks into the CRM system as shown in Figure 2-8 on page 500

Note: The steps and jobs used to integrate the information from the North American Bank and Northern California Bank into the CRM system are similar to those described in 1.10, “Mailing list scenario” on page 117. Therefore, we cross reference these steps as much as possible to avoid duplication. However, we do discuss any differences, such as configuration options and report highlights.

4 Table 2-2 on page 501, Table 2-3 on page 507, and Table 2-4 on page 507 highlight the differences between the various systems and the elements from which the “best” data should be derived. The information in this table was taken from *IBM WebSphere Information Analyzer and Data Quality Assessment*, SG24-7508.

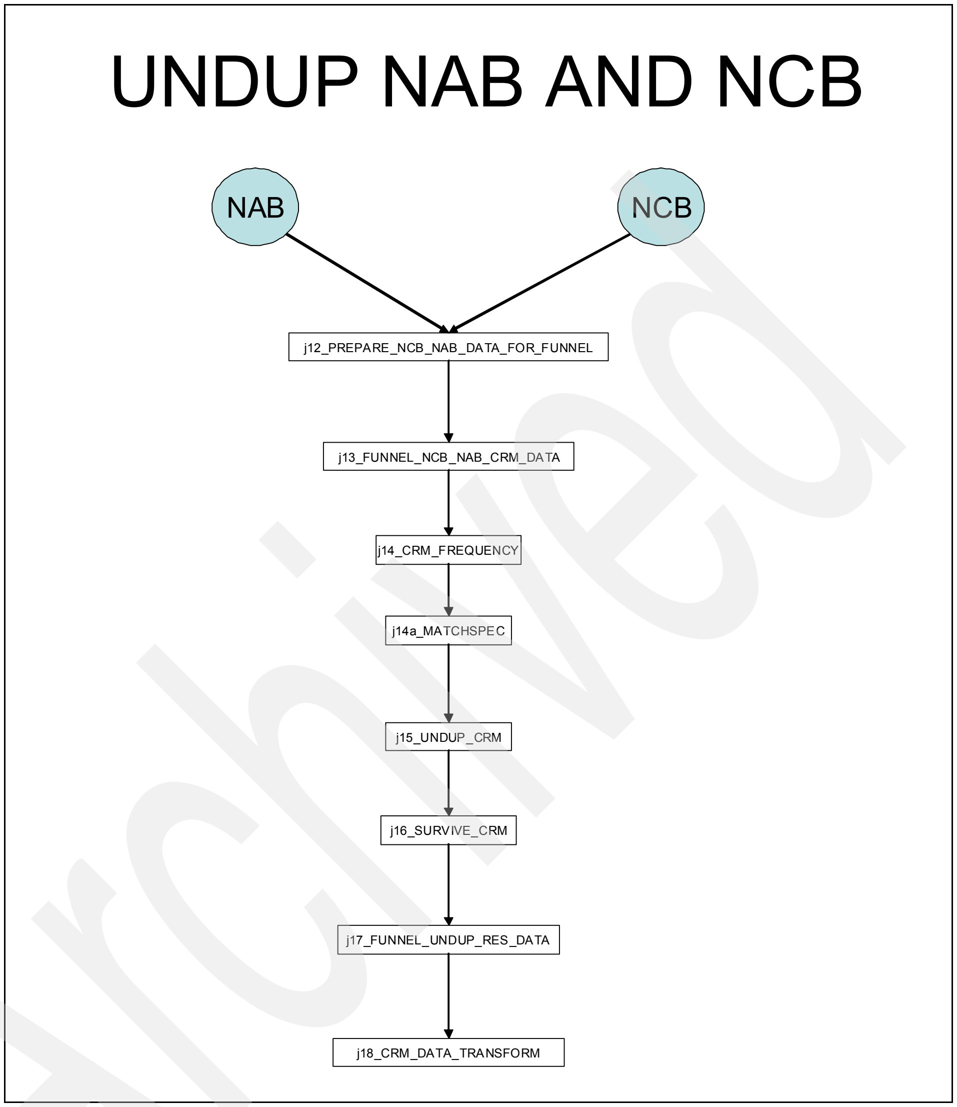


Figure 2-8 Data integration into the CRM system

Table 2-2 Summary of differences between source and target and the action to be taken

North American Bank (NAB) and Northern California Bank (NCB) (source)				CRM system (target)			Action to be taken
Column in the table	**Metadata**		**Data Content example**	**Column in the table**	**Metadata Defined**	**Data Content example**	
	Defined	**Inferred**					
Best of				PREFIX in the CUSTOMER table	VARCHAR(10)	DR.	Use IBM WebSphere QualityStage to standardize, match, and survive the best source of this information and map/transform it to the target
— TITLE in CUSTOMER (NAB)	CHAR(3)	CHAR(3)	MR.				
— NAME in DRIVER (NAB)	VARCHAR(50)	VARCHAR(28)	MR. JOHN DOE				
— NAME in CUSTOMER (NCB)	CHAR(50)	CHAR(22)	JON DOW				
— NAME in BCUSTOMER (NCB)	VARCHAR(40)	VARCHAR(40)	DR. JOHN DOE				
Best of				FIRSTNAME in the CUSTOMER table	VARCHAR(30)	MICHAEL	Use IBM WebSphere QualityStage to standardize, match, and survive the best source of this information and map/transform it to the target
— FIRST_NAME in CUSTOMER (NAB)	VARCHAR(20)	VARCHAR(12)	MICHAEL				
— NAME in DRIVER (NAB)	VARCHAR(50)	VARCHAR(28)	MR. MIKE HUIS				
— NAME in CUSTOMER (NCB)	CHAR(50)	CHAR(22)	MR. MICHAEL HUIS				
— NAME in BCUSTOMER (NCB)	VARCHAR(40)	VARCHAR(40)					
Best of				MIDDLENAME in the CUSTOMER table	VARCHAR(30)	JANE	Use IBM WebSphere QualityStage to standardize, match, and survive the best source of this information and map/transform it to the target
— NAME in DRIVER (NAB)	VARCHAR(50)	VARCHAR(28)	MS. CHRISTINE JANE DAY				
— NAME in CUSTOMER (NCB)	CHAR(50)	CHAR(22)	CHRISTINE DAY				
— NAME in BCUSTOMER (NCB)	VARCHAR(40)	VARCHAR(40)	MS. CHRISTINE J DAY				
— FIRST_NAME in CUSTOMER (NAB)	VARCHAR(20)	VARCHAR(12)	JOHN JACK				
— LAST_NAME in CUSTOMER (NAB)	VARCHAR(20)	VARCHAR(16)	JOSEPH WATSON				
Best of				LASTNAME in the CUSTOMER table	VARCHAR(30)	PESCO	Use IBM WebSphere QualityStage to standardize, match, and survive the best source of this information and map/transform it to the target
— LAST_NAME in CUSTOMER (NAB)	VARCHAR(20)	VARCHAR(16)	PESCO				
— NAME in DRIVER (NAB)	VARCHAR(50)	VARCHAR(28)					
— NAME in CUSTOMER (NCB)	CHAR(50)	CHAR(22)	JOSEPH PESCO				
— NAME in BCUSTOMER (NCB)	VARCHAR(40)	VARCHAR(40)					
Best of				GENDER in the CUSTOMER table	CHAR(1)	M	Use IBM WebSphere QualityStage to standardize, match, and survive the best source of this information and map/transform it to the target
— GENDER in CUSTOMER (NAB)	CHAR(1)	CHAR(1)	M				
— GENDER in DRIVER (NAB)	CHAR(1)	CHAR(1)					
— GENDER in CUSTOMER (NCB)	CHAR(1)	CHAR(1)	0				

North American Bank (NAB) and Northern California Bank (NCB) (source)				CRM system (target)			Action to be taken
NATIONALITY in CUSTOMER (NAB)	VARCHAR(20)	VARCHAR(2)	AU	NATIONALITY in the CUSTOMER table	VARCHAR(20)	AU	Direct mapping
TYPE in CUSTOMER (NCB)	CHAR(1)	CHAR(1)	P	TYPE in the CUSTOMER table	INTEGER	1	Map using CUSTOMERTYPE reference table
PREF_LANG in CUSTOMER (NCB)	CHAR(3)	CHAR(3)	ENG	PREFLANG in the CUSTOMER table	CHAR(3)	ENG	Map using ISO_LANGUAGE reference table
ADVISOR in CUSTOMER (NCB)	INT32	INT32	555110	ADVISOR in the CUSTOMER table	INTEGER	555110	Map using EMPLOYEE table
				PREFCONTACT in the CUSTOMER table	INTEGER		

Best of
— HOME_ADDRESS in CONTACT_INFO (NAB)
— ADDRESS in DRIVER (NAB)
— ADDR1 in CUSTOMER (NCB)
— ADDR2 in CUSTOMER (NCB)
— ADDR1 in BCUSTOMER (NCB)
— ADDR2 in BCUSTOMER (NCB)
	
	VARCHAR(50)
	VARCHAR(50)
	CHAR(50)
	CHAR(50)
	VARCHAR(40)
	VARCHAR(40)
		
		VARCHAR(36)
		VARCHAR(36)
		CHAR(35)
		CHAR(50)
		CHAR(40)
		CHAR(1)
			
			63 KALINDA RD
			63, KALINDA
				HOMESTREET in the CUSTOMER table	VARCHAR(30)	63, KALINDA ROAD	Use IBM WebSphere QualityStage to standardize, match, and survive the best source of this information and map/transform it to the target

Best of
— HOME_ADDRESS in CONTACT_INFO (NAB)
— CITY in DRIVER (NAB)
— CITY in CUSTOMER (NCB)
— CITY in BCUSTOMER (NCB)
	
	VARCHAR(50)
	VARCHAR(40)
	CHAR(30)
	VARCHAR(30)
		
		VARCHAR(36)
		VARCHAR(7)
		CHAR(22)
		VARCHAR(30)
			
			BRENTWOOD
			BRENTWOOD
				HOMECITY in the CUSTOMER table	VARCHAR(20)	BRENTWOOD	Use IBM WebSphere QualityStage to standardize, match, and survive the best source of this information and map/transform it to the target

Best of
— HOME_ADDRESS in CONTACT_INFO (NAB)
— STATE in DRIVER (NAB)
— ADDR1 in CUSTOMER (NCB)
— ADDR2 in CUSTOMER (NCB)
— ADDR1 in BCUSTOMER (NCB)
— ADDR2 in BCUSTOMER (NCB)
	
	VARCHAR(50)
	CHAR(2)
	CHAR(50)
	CHAR(50)
	VARCHAR(40)
	VARCHAR(40)
		
		VARCHAR(36)
		CHAR(2)
		CHAR(35)
		CHAR(50)
		VARCHAR(40)
		VARCHAR(1)
			
			CALIFORNIA
			
			CA
				HOMESTATE in the CUSTOMER table	CHAR(2)	CA	Use IBM WebSphere QualityStage to standardize, match, and survive the best source of this information and map/transform it to the target

North American Bank (NAB) and Northern California Bank (NCB) (source)				CRM system (target)			Action to be taken

Best of
— HOME_ZIP in CONTACT_INFO (NAB)
— ZIP in DRIVER (NAB)
— ZIP in CUSTOMER (NCB)
— ZIP in BCUSTOMER (NCB)
	
	CHAR(9)
	CHAR(9)
	CHAR(10)
	CHAR(10)
		
		CHAR(5)
		INT32 length 5
		INT32 length 5
		INT32 length 5
			95123	HOMEZIP in the CUSTOMER table	VARCHAR(10)	95123-4865	Use IBM WebSphere QualityStage to standardize, match, and survive the best source of this information and map/transform it to the target

Best of
— COUNTRY in CUSTOMER (NCB)
— COUNTRY in BCUSTOMER (NCB)
	
	CHAR(30)
	VARCHAR(30)
		
		CHAR(1)
		VARCHAR(1)
				HOMECOUNTRY in the CUSTOMER table	VARCHAR(20)	U.S.A.	Use IBM WebSphere QualityStage to standardize, match, and survive the best source of this information and map/transform it to the target

WORK_ADDRESS in CONTACT_INFO (NAB)	VARCHAR(50)	VARCHAR(36)	555 BAILEY AVENUE	WORKSTREET in the CUSTOMER table	VARCHAR(30)	555 BAILEY AVE	Use IBM WebSphere QualityStage to standardize and map/transform this information to the target
WORK_ADDRESS in CONTACT_INFO (NAB)	VARCHAR(50)	VARCHAR(36)	SAN JOSE	WORKCITY in the CUSTOMER table	VARCHAR(20)	SAN JOSE	Use IBM WebSphere QualityStage to standardize and map/transform this information to the target
WORK_ADDRESS in CONTACT_INFO (NAB)	VARCHAR(50)	VARCHAR(36)	CALIFORNIA	WORKSTATE in the CUSTOMER table	CHAR(2)	CA	Use IBM WebSphere QualityStage to standardize and map/transform this information to the target
WORK_ZIP in CONTACT_INFO (NAB)	CHAR(9)	INT32 length 5	95123	WORKZIP in the CUSTOMER table	VARCHAR(10)	95123-4865	Mapping with hyphen included
				WORKCOUNTRY in the CUSTOMER table	VARCHAR(20)		
Best of — HOME_PHONE in CONTACT_INFO (NAB) — HOMEPHONE in CUSTOMER (NCB)	 CHAR(15) CHAR(15)	 INT64 length 10 INT64 length 10	4085551234	HOMEPHONE in the CUSTOMER table	VARCHAR(15)	408-5551234	Use IBM WebSphere QualityStage to standardize, match, and survive the best source of this information and map/transform it to the target

Best of
— WORK_PHONE in CONTACT_INFO (NAB)
— WORKPHONE in CUSTOMER (NCB)
	
	CHAR(15)
	CHAR(15)
		
		CHAR(10)
		CHAR(12)
			6505555678	WORKPHONE in the CUSTOMER table	VARCHAR(20)	650-5555678	Use IBM WebSphere QualityStage to standardize, match, and survive the best source of this information and map/transform it to the target

North American Bank (NAB) and Northern California Bank (NCB) (source)				CRM system (target)			Action to be taken

Best of
— CELL_PHONE in CONTACT_INFO (NAB)
—CELLPHONE in CUSTOMER (NCB)
	CHAR(15)
	CHAR(15)
		CHAR(10)
		CHAR(12)
			4155553456	CELLPHONE in the CUSTOMER table	VARCHAR(15)	4155553456	Use IBM WebSphere QualityStage to standardize, match, and survive the best source of this information and map/transform it to the target

Best of
— EMAIL in CUSTOMER (NCB)
— EMAIL in BCUSTOMER (NCB)
	VARCHAR(50)
	VARCHAR(50)
		VARCHAR(5)
		VARCHAR(1)
			IADM	EMAIL in the CUSTOMER table	VARCHAR(20)	IADM	Use IBM WebSphere QualityStage to standardize, match, and survive the best source of this information and map/transform it to the target

				RATING in the CUSTOMER table	CHAR(5)		
				NABCHKASSETS in the CUSTOMER table	DECIMAL(9,2)		
				NABSAVASSETS in the CUSTOMER table	DECIMAL(9,2)		
				NABLOANINDICATOR in the CUSTOMER table	CHAR(1)		
INITIAL_VALUE in LOAN (NAB)	DECIMAL(9,2)	DECIMAL(8,2)	400000.00	NABLOANAMOUNT in the CUSTOMER table	DECIMAL(9,2)	400000.00	Direct mapping
BALANCE in LOAN (NAB)	DECIMAL(9,2)	DECIMAL(8,2)	350000.00	NABLOANBALANCE in the CUSTOMER table	DECIMAL(9,2)	350000.00	Direct mapping
RATES in LOAN (NAB)	DECIMAL(8,5)	DECIMAL(7,5)	6.35	NABLOANRATE in the CUSTOMER table	DECIMAL(6,3)	6.35	Mapping with possible truncation. Should consider increasing precision and scale of target
				NCBCHKASSETS in the CUSTOMER table	DECIMAL(9,2)		
				NCBSAVASSETS in the CUSTOMER table	DECIMAL(9,2)		
				NCBLOANINDICATOR in the CUSTOMER table	CHAR(1)		
INITIAL_LOAN_VALUE in LOAN (NCB)	CHAR(20)	INT32 length 8	500000.00	NCBLOANAMOUNT in the CUSTOMER table	DECIMAL(9,2)	500000.00	Mapping with data type transformation
BALANCE in LOAN (NCB)	CHAR(20)	INT32 length 9	495000.00	NCBLOANBALANCE in the CUSTOMER table	DECIMAL(9,2)	495000.00	Mapping with data type transformation
INTEREST_RATE in LOAN (NCB)	CHAR(20)	SFLOAT length 4	5.35	NCBLOANRATE in the CUSTOMER table	DECIMAL(6,3)	5.35	Mapping with data type transformation
				BROKINDICATOR in the CUSTOMER table	CHAR(1)		

North American Bank (NAB) and Northern California Bank (NCB) (source)				CRM system (target)			Action to be taken
				BROKASSETS in the CUSTOMER table	DECIMAL(9,2)		
				BROKMARGIN in the CUSTOMER table	DECIMAL(9,2)		
				CCINDICATOR in the CUSTOMER table	CHAR(1)		
LIMIT in CARD (NAB)	DECIMAL(9,2)	DECIMAL(8,2)	21000.00	CCLIMIT in the CUSTOMER table	INTEGER	21000	Mapping with truncation of scale. Consider adding scale to the target
LIMIT_BALANCE in CARD (NAB)	DECIMAL(9,2)	DECIMAL(8,2)	4971.50	CCBALANCE in the CUSTOMER table	DECIMAL(9,2)	4971.50	Direct mapping
LIMIT_W_BALANCE in CARD (NAB)	DECIMAL(9,2)	DECIMAL(8,2)	15029.50				Ignore this fields because it can be computed
				CARINDICATOR in the CUSTOMER table	CHAR(1)		
FULL_COVERAGE_IND in CAR_INSURANCE	CHAR(1)	CHAR(1)	N	FULLCOVERIND in the CUSTOMER table	CHAR(1)	N	Direct mapping
INSURANCE_VALUE in CAR_INSURANCE	DECIMAL(9,2)	DECIMAL(8,2)	1200.00	CARPREMIUMS in the CUSTOMER table	DECIMAL(6,2)	1200.00	Mapping with possible truncation. Consider increasing precision
END_DT in CAR_INSURANCE	DATE	DATE	12/31/2007	CARENDDATE in the CUSTOMER table	DATE	12/31/2007	Direct mapping
				CRMID in the CUSTKEYXREF table	INTEGER		
CUSTOMER_ID in CUSTOMER (NAB)	INT32	INT16 length 4	1344	NABCOREID in the CUSTKEYXREF table	INTEGER		
DRIVER_ID in DRIVER (NAB)	INT32	INT16 length 4	338	NABNONCOREID in the CUSTKEYXREF table	INTEGER		Direct mapping. Note that this information can be used to get access to credit cards held by this individual using the CUSTKEYXREF table
ID in CUSTOMER (NCB)	INT32	INT32 length 8	10001500	NCBCOREID in the CUSTKEYXREF table	INTEGER	10001500	Direct mapping
ID in BCUSTOMER (NCB)	INT32	INT32 length 8	20001500	NBCNONCOREID in the CUSTKEYXREF table	INTEGER	20001500	Direct mapping
				ID in the CUSTOMERTYPE table	INTEGER		To be generated
				DESCRIPTION in the CUSTOMERTYPE table	VARCHAR(50)	PERSON	To be generated
				ID in the RELATIONTYPE table	INTEGER		To be generated

North American Bank (NAB) and Northern California Bank (NCB) (source)				CRM system (target)			Action to be taken
				DESCRIPTION in the RELATIONTYPE table	VARCHAR(50)	IS SON OF	To be generated
				ID in the ISO_LANGUAGE table	CHAR(3)		ISO standard code
				DESCRIPTION in the ISO_LANGUAGE table	VARCHAR(50)	DANISH	ISO standard description
				ID in the CONTACTTYPE table	INTEGER		To be generated
				DESCRIPTION in the CONTACTTYPE table	VARCHAR(50)	MAILING ADDRESS	To be generated
				ID in the LINEOFBUSINESS table	INTEGER		To be generated
				DESCRIPTION in the LINEOFBUSINESS table	VARCHAR(50)	BROKERAGE	To be generated
				ID in the ROLE table	INTEGER		To be generated
				DESCRIPTION in the ROLE table	VARCHAR(50)	BENIFICIARY	To be generated
				ID in the ITEM table	INTEGER		To be generated
				DESCRIPTION in the ITEM table	VARCHAR(50)	INTEREST RATE	To be generated
ID in BRANCH (NCB)	INT32	INT8 length 2	12001500	ID in the BRANCH table	INTEGER	12001500	Direct mapping
NAME in BRANCH (NCB)	CHAR(50)	CHAR(28)	SANTA TERESA	NAME in the BRANCH table	VARCHAR(50)	SANTA TERESA	Direct mapping
				FROMCUSTOMER in the CUSTOMERRELATION table	INTEGER		To be generated
				TOCUSTOMER in the CUSTOMERRELATION table	INTEGER		To be generated
				RELATIONTYPE in the CUSTOMERRELATION table	INTEGER		To be generated
				ID in the CONTRACT table	INTEGER		To be generated
				PRODUCT in the CONTRACT table	INTEGER		To be generated
				STATUS in the CONTRACT table	INTEGER		To be generated
				CREATED in the CONTRACT table	TIMESTAMP		To be generated
				UPDATED in the CONTRACT table	TIMESTAMP		To be generated
				ID in the CONTRACTITEM table	INTEGER		To be generated
				CONTRACT in the CONTRACTITEM table	INTEGER		To be generated

North American Bank (NAB) and Northern California Bank (NCB) (source)				CRM system (target)			Action to be taken
				ITEM in the CONTRACTITEM table	INTEGER		To be generated
				VALUE in the CONTRACTITEM table	VARCHAR(30)		To be generated
				ID in the CONTRACTROLE table	INTEGER		To be generated
				CUSTOMER in the CONTRACTROLE table	INTEGER		To be generated
				CONTRACT in the CONTRACTROLE table	INTEGER		To be generated
				ROLE in the CONTRACTROLE table	INTEGER		To be generated
ID in EMPLOYEE (NCB)	INT32	INT32 length 8	13001500	ID in the EMPLOYEE table	INTEGER	13001500	Direct mapping
NAME in EMPLOYEE (NCB)	CHAR(50)	CHAR(20)	JEFFREY JONES	NAME in the EMPLOYEE table	CHAR(50)	JEFFREY JONES	Direct mapping
USERID in EMPLOYEE (NCB)	CHAR(8)	CHAR(8)	JJONES	USERID in the EMPLOYEE table	CHAR(8)	JJONES	Direct mapping
BRANCH in EMPLOYEE (NCB)	INT32	INT32 length 8	12001500	BRANCH in the EMPLOYEE table	INTEGER	12001500	Direct mapping
				BUSINESS in the EMPLOYEE table	INTEGER		To be generated

Table 2-3 North American Bank information missing in the CRM data model

Data element	Action to be taken
NICKNAME in CUSTOMER	Ignore because it is not considered relevant to customer relationship management
CHURN_IND in CUSTOMER	Include this information in the CRM data model
CREDIT_SCORE in CUSTOMER	Include this information in the CRM data model
WORK_ADDRESS in BRANCH	Ignore because the ID and NAME fields in the CRM data model as adequate to obtain this information
WORK_ZIP in BRANCH	Ignore because the ID and NAME fields in the CRM data model as adequate to obtain this information

Table 2-4 Northern California Bank information missing in the CRM data model

Data element	Action to be taken
FAX IN CUSTOMER	Include this information in the CRM data model

The following sections describe the name and address cleansing, and data integration process:

- Cleansing North American Bank's core and non-core services

- Cleansing Northern California Bank's core and non-core services
- Matching and surviving Northern California Bank and Northern California Bank information

2.6.1 Cleansing North American Bank's core and non-core services

Figure 2-6 on page 496 shows the processing flow and jobs that we used for cleansing name and addresses in North American Bank's core and non-core services.

We describe the steps briefly here:

1. We began by extracting all the CUSTOMER, CONTACT_INFO, and DRIVER data from the DB2 database and loading it into data sets to isolate it from changes during analysis. We also introduced a placeholder Transformer stage to provide for pre-processing the source for analysis such as changing default values to nulls. In this case however, no transforms were required.

   Job "j00_SRC_NAB" on page 512 performs this step.

2. Next, we analyzed the addresses in appropriate data sets to determine the (ISO code) country using the COUNTRY rule set in the Standardize stage.

   Job "j01_STAN_COUNTRY_NAB" on page 518 performs this step.

3. We analyzed the ISO codes that were generated by the previous step by the Investigate stage using the character concatenate option with the *C* mask to obtain frequency distribution. This step identified whether the addresses in the appropriate data sets belonged to more than one country and identified the codes of the countries in the addresses.

   In this case, the home addresses in the CONTACT_INFO data set were a mix of U.S. and Canadian addresses, while the work addresses in the CONTACT_INFO data set were all U.S. addresses. The addresses in the DRIVER data set were all U.S. addresses.

   Job "j02_INVCC_ISOCODE_NAB" on page 525 performs this step.

4. Then, we split the records in the CONTACT_INFO data set into two files—one that contains Canadian home addresses and the other that contains U.S. home addresses. We needed to split the records because different rule sets

must be applied to process addresses of different countries. Then, we used the Standardize stage with the following domain-preprocessor rule sets to move name and address data into Name, Address, and Area domains:

- CAPREP for the Canadian home address records in the CONTACT_INFO data set
- USPREP for the name (CUSTOMER and DRIVER data) and address (home and work addresses in CONTACT_INFO, and home addresses in DRIVER) records

Job “j03_STAN_XXPREP_NAB” on page 530 performs this step.

5. We used the Investigate stage using the character concatenate option on the input patterns that were generated by the Standardize stage in the previous step to determine the degree of success that is achieved by the domain-preprocessor rule sets CAPREP and USPREP in parsing the tokens in the name and address fields into the correct domains.

   A visual analysis of the pattern reports that were generated by the Investigate stage indicated a very large number of error input patterns. Therefore, we needed to define Input Pattern overrides and reran the Standardize jobs (“j03_STAN_XXPREP_NAB” on page 530) and the Investigate stage to confirm that the overrides had the desired effect.

   Job “j04_INVCC_XXPREP_INPUT_PATTERN” on page 541 performs this step.

6. After all the name and address data was moved to the correct domain buckets, we used the Investigate stage with the word investigate option to determine the effectiveness of token handling in the NAME, ADDRESS, and AREA domains. Potential Classification and Input Pattern overrides are identified at this point.

   The following jobs perform these steps:

   - “j05a_INVW_CONTACT_INFO_HOME_CA_XXPREP” on page 572
   - “j05b_INVW_CONTACT_INFO_HOME_US_XXPREP” on page 577
   - “j05c_INVW_CONTACT_INFO_WORK_XXPREP” on page 584
   - “j05d_INVW_DRIVER_XXPREP” on page 592
   - “j05e_INVW_CUSTOMER_XXPREP” on page 603

7. For the records with U.S. home and work addresses, we used the CASS stage to validate, correct, and standardize the U.S. addresses in the Address domain. We included a Transformer stage to add a second address line column to customer file because CASS requires two address lines as input for its processing.

   Note: CASS is a separately priced module that requires installation of the CASS module.

Note: The Canadian home addresses in the CONTACT_INFO data set were not processed by SERP (which is the CASS equivalent) due to time constraints.

Job “j06_XXPREP_CASS” on page 605 performs this step.

8. Then we ran the Investigate stage with character concatenate option on the (home address related columns in CONTACT_INFO and DRIVER) results of job “j06_XXPREP_CASS” on page 605 step to determine addresses that were not recognized by CASS (delivery point verification or DPV).

   Note: Due to a bug with handling nulls, we introduced a Transformer stage to convert nulls to a space using column derivation.

   We investigated the generated columns DPVMATCHFLAG_CASS and DPVCODE1_CASS using a *C* mask. A value of `A1` in the DPVCODE1_CASS field indicates a potential problem.

   Job “j07_INVCC_CASS” on page 615 performs this step.

9. The next step was to standardize the name and address contents of the output of job “j06_XXPREP_CASS” on page 605 using the domain-preprocessor USPREP rule set. This step moves name, address, and area content from selected input columns to the NameDomain_USPREP, AddressDomain_USPREP, and AreaDomain_USPREP columns. We also added a column to the output that only had the first three characters of the ZIP code using a Transformer stage. This new column (ZIP3) was used as a blocking variable in an upcoming matching stage.

   Job “j08_USPREP_CASS” on page 619 performs this step.

10. In this step, we standardized the name and address contents of the output of jobs “j03_STAN_XXPREP_NAB” on page 530 and “j08_USPREP_CASS” on page 619 using the domain-specific rule sets USNAME (with column NameDomain_USPREP), USADDR (with column AddressDomain_USPREP), USAREA (with column AreaDomain_USPREP), CAADDR (with column AddressDomain_CAPREP), CAAREA (with column AreaDomain_CAPREP). Separate processes were defined—one for each rule set.

    Note: Here again, we had to introduce a null handling Transformer stage to circumvent a bug.

    Job “j09_STAN_CASS” on page 631 performs this step.

11. The next step identified unhandled patterns and classifications in the previous step. We ran a series of Investigate stage with the character concatenate option using the *C* mask on the unhandled pattern column from the results of “j09_STAN_CASS” on page 631. The columns that were investigated corresponded to the name, address, and area domains. Unhandled patterns were reviewed and classification and input pattern overrides were generated to rectify the problem. We then reran the Standardize stage with the overrides in place and verified using Investigate that all the unhandled patterns were resolved.

    The following jobs perform these steps:

    – “j10a_INVCC_CUSTOMER_XXPREP_STAN” on page 637
    – “j10b_INVCC_CONTACT_INFO_WORK_CASS_USPREP_STAN” on page 640
    – “j10c_INVCC_CONTACT_INFO_HOME_US_CASS_USPREP_STAN” on page 645
    – “j10d_INVCC_CONTACT_INFO_HOME_CA_XXPREP_STAN” on page 647
    – “j10e_INVCC_DRIVER_CASS_USPREP_STAN” on page 652

12. After the unhandled patterns are handled, we reconstructed two files:

    – One with the standardized name (CUSTOMER file) and standardized U.S. home address information (CONTACT_INFO)

    – The other with the standardized name (CUSTOMER file) and standardized Canadian home address information (CONTACT_INFO) from the standardized output by joining them on the CUSTOMER_ID.

    We also renamed certain columns in the standardized work address file that are created in “j09_STAN_CASS” on page 631 to avoid column name conflicts in the upcoming job “j12_JOIN_NAB_WORK_AND_HOME” on page 674.

    Job “j11_JOIN_NAB_NAME_AND_ADDR_DATA” on page 657 performs this step.

13. In the next step, we appended the work address information to the two files that are created in “j11_JOIN_NAB_NAME_AND_ADDR_DATA” on page 657. All the addresses in the work address were U.S. addresses.

    “j12_JOIN_NAB_WORK_AND_HOME” on page 674 performs this step.

14. At this point, we had two reconstructed files with standardized names and home and work addresses—one containing U.S. home addresses and the other containing Canadian home addresses.

    Now, we needed to merge these files with the standardized name and address file of the DRIVER to identify duplicates (with data from the Northern California Bank’s standardized name and addresses from their core and non-core services as described in 2.6.2, “Cleansing Northern California Bank’s core and non-core services” on page 706) and survive the “best” data into the CRM system as described in 2.6.3, “Matching and surviving Northern California Bank and Northern California Bank information” on page 783.

Prior to merging the two files that are created in “j12_JOIN_NAB_WORK_AND_HOME” on page 674 with the standardized name and address DRIVER file that is created in “j09_STAN_CASS” on page 631, we needed to prepare the files for merging in the Funnel stage by ensuring that all the files have the same number of columns and same names of columns using a Transformer stage.

Job “j13_PREPARE_NAB_DATA_FOR_FUNNEL” on page 687 performs this step.

15. In this step, we actually merged the three files that were created in the “j13_PREPARE_NAB_DATA_FOR_FUNNEL” on page 687 step using the Funnel stage.

    Job “j14_FUNNEL_NAB_DATA_FOR_CRM” on page 705 performs this step.

    The output of this step was then included for matching and surviving the “best” data as described in 2.6.3, “Matching and surviving Northern California Bank and Northern California Bank information” on page 783.

We describe these jobs in more detail in the following sections.

j00_SRC_NAB

We begin by extracting all the CUSTOMER, CONTACT_INFO, and DRIVER data of the North American Bank from the DB2 database and loading it into data sets to isolate it from changes during analysis. We also introduce a placeholder Transformer stage to provide for pre-processing the source for analysis such as changing default values to nulls.

Figure 2-9 on page 513 shows the various stages that were used in this job, including a DB2 UDB API stage that was used to access the data in the CUSTOMER, CONTACT_INFO, and DRIVER tables. In this case, however, no transforms were required. We modified the names of the stages as shown.

Because the configuration of this job is very similar to that described in “J00_SRC_CUSTOMER” on page 142, we do not repeated the process here. However, some of the configurations of interest are as follows:

- Figure 2-10 on page 514 shows the generated SQL to retrieve all the rows in the CUSTOMER table. We do not show the SQL for CONTACT_INFO and DRIVER here.
- Figure 2-11 on page 515 shows the configuration of the Transformer Stage that mapped the columns retrieved from the CUSTOMER table directly to the

output with no transforms. We do not show the corresponding configurations for CONTACT_INFO and DRIVER.

- After saving, compiling, and running the job (not shown here), the results are shown in Figure 2-12 on page 516 through Figure 2-14 on page 518. The CUSTOMER, CONTACT_INFO, and DRIVER files all had 34 records.

Now, proceed to “j01_STAN_COUNTRY_NAB” on page 518.

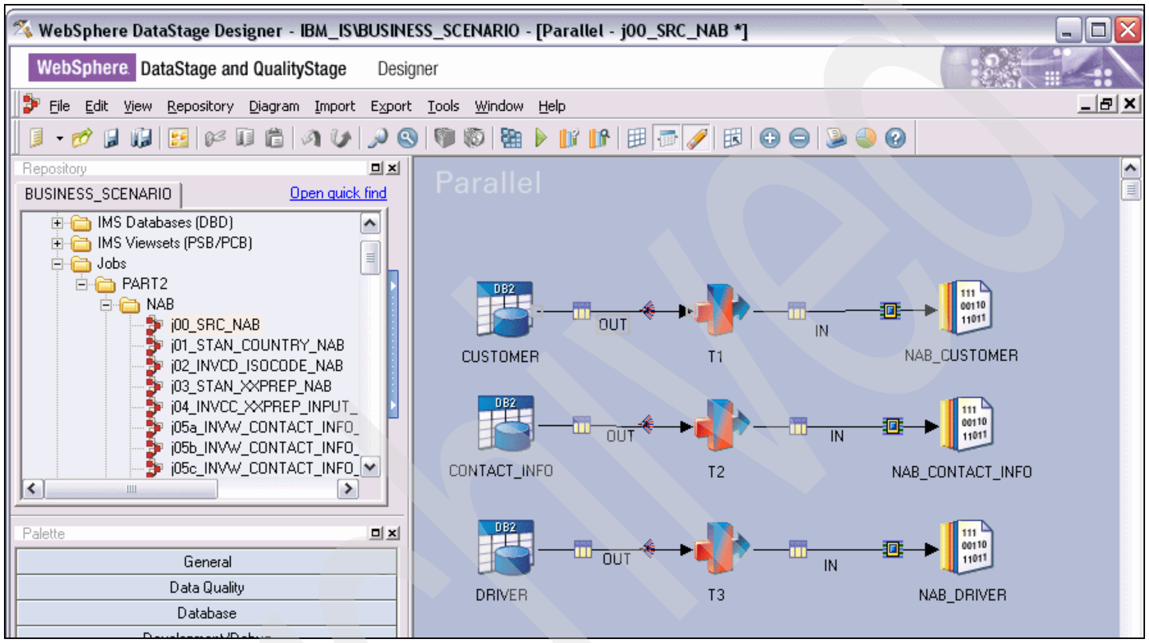
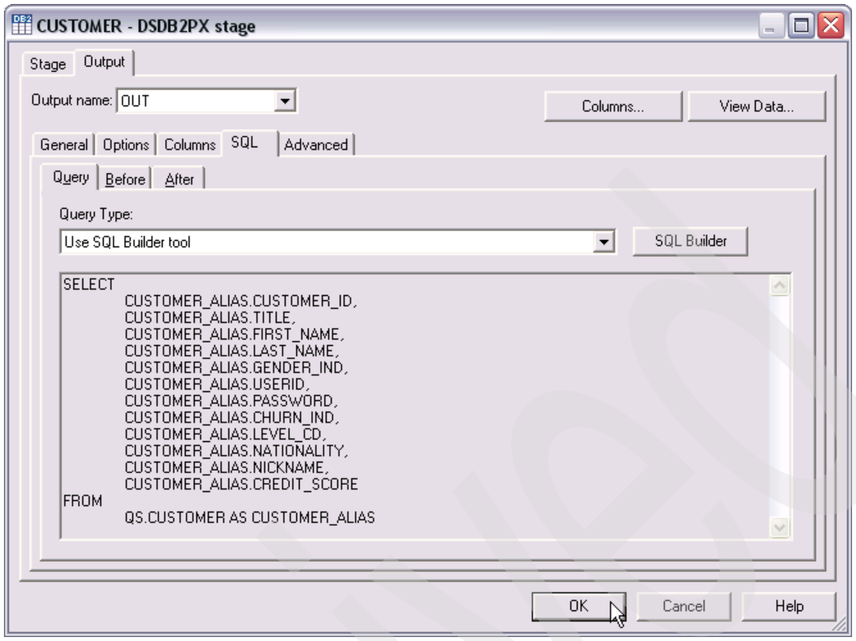


Figure 2-9 Create j00_SRC_NAB job 1/6

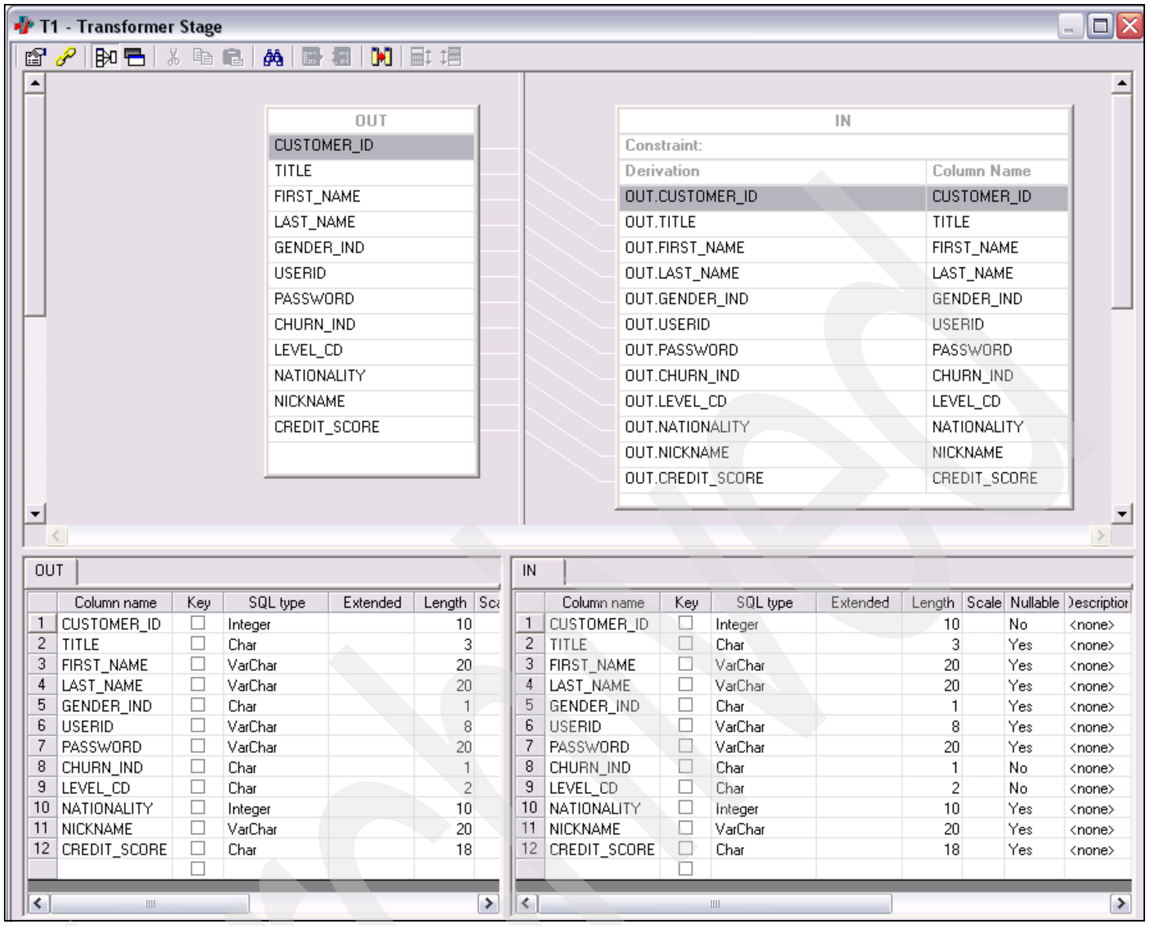


Output name: OUT

Query Type: Use SQL Builder tool

```
SELECT
        CUSTOMER_ALIAS.CUSTOMER_ID,
        CUSTOMER_ALIAS.TITLE,
        CUSTOMER_ALIAS.FIRST_NAME,
        CUSTOMER_ALIAS.LAST_NAME,
        CUSTOMER_ALIAS.GENDER_IND,
        CUSTOMER_ALIAS.USERID,
        CUSTOMER_ALIAS.PASSWORD,
        CUSTOMER_ALIAS.CHURN_IND,
        CUSTOMER_ALIAS.LEVEL_CD,
        CUSTOMER_ALIAS.NATIONALITY,
        CUSTOMER_ALIAS.NICKNAME,
        CUSTOMER_ALIAS.CREDIT_SCORE
FROM
        QS.CUSTOMER AS CUSTOMER_ALIAS
```

Figure 2-10 Create j00_SRC_NAB job 2/6



OUT
CUSTOMER_ID
TITLE
FIRST_NAME
LAST_NAME
GENDER_IND
USERID
PASSWORD
CHURN_IND
LEVEL_CD
NATIONALITY
NICKNAME
CREDIT_SCORE

IN

Constraint:

Derivation	Column Name
OUT.CUSTOMER_ID	CUSTOMER_ID
OUT.TITLE	TITLE
OUT.FIRST_NAME	FIRST_NAME
OUT.LAST_NAME	LAST_NAME
OUT.GENDER_IND	GENDER_IND
OUT.USERID	USERID
OUT.PASSWORD	PASSWORD
OUT.CHURN_IND	CHURN_IND
OUT.LEVEL_CD	LEVEL_CD
OUT.NATIONALITY	NATIONALITY
OUT.NICKNAME	NICKNAME
OUT.CREDIT_SCORE	CREDIT_SCORE

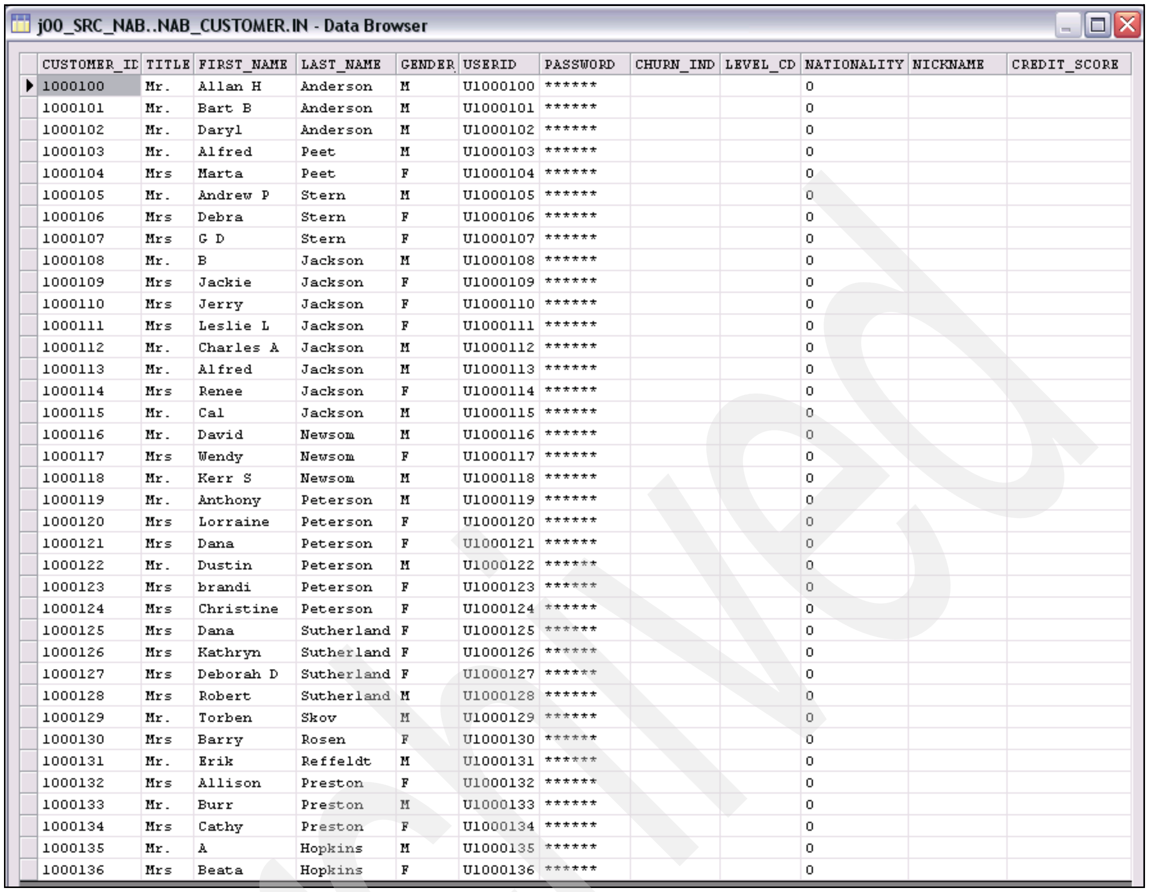
OUT

	Column name	Key	SQL type	Extended	Length	Sca
1	CUSTOMER_ID	☐	Integer		10	
2	TITLE	☐	Char		3	
3	FIRST_NAME	☐	VarChar		20	
4	LAST_NAME	☐	VarChar		20	
5	GENDER_IND	☐	Char		1	
6	USERID	☐	VarChar		8	
7	PASSWORD	☐	VarChar		20	
8	CHURN_IND	☐	Char		1	
9	LEVEL_CD	☐	Char		2	
10	NATIONALITY	☐	Integer		10	
11	NICKNAME	☐	VarChar		20	
12	CREDIT_SCORE	☐	Char		18	
		☐				

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	Descriptior
1	CUSTOMER_ID	☐	Integer		10		No	<none>
2	TITLE	☐	Char		3		Yes	<none>
3	FIRST_NAME	☐	VarChar		20		Yes	<none>
4	LAST_NAME	☐	VarChar		20		Yes	<none>
5	GENDER_IND	☐	Char		1		Yes	<none>
6	USERID	☐	VarChar		8		Yes	<none>
7	PASSWORD	☐	VarChar		20		Yes	<none>
8	CHURN_IND	☐	Char		1		No	<none>
9	LEVEL_CD	☐	Char		2		No	<none>
10	NATIONALITY	☐	Integer		10		Yes	<none>
11	NICKNAME	☐	VarChar		20		Yes	<none>
12	CREDIT_SCORE	☐	Char		18		Yes	<none>
		☐						

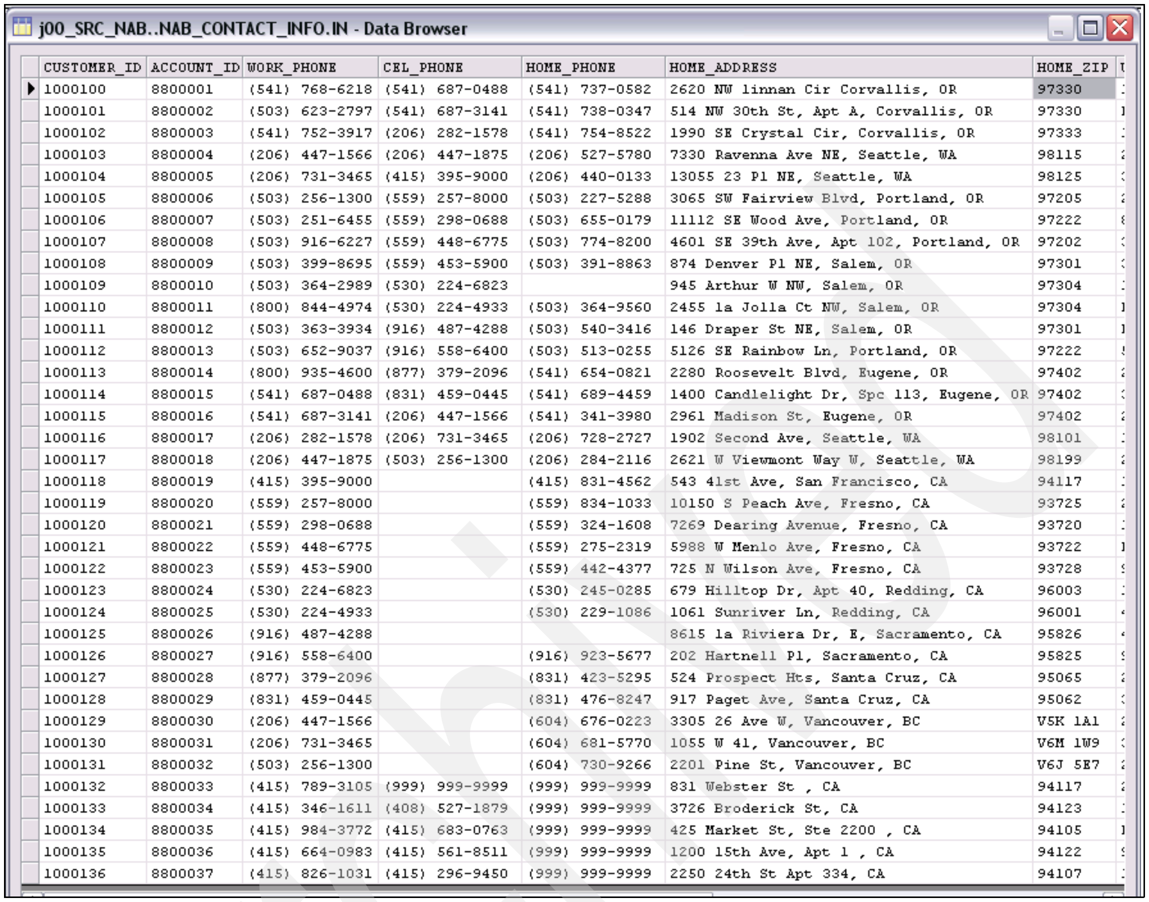
Figure 2-11 Create j00_SRC_NAB job 3/6



j00_SRC_NAB..NAB_CUSTOMER.IN - Data Browser

CUSTOMER_ID	TITLE	FIRST_NAME	LAST_NAME	GENDER	USERID	PASSWORD	CHURN_IND	LEVEL_CD	NATIONALITY	NICKNAME	CREDIT_SCORE
1000100	Mr.	Allan H	Anderson	M	U1000100	******			0		
1000101	Mr.	Bart B	Anderson	M	U1000101	******			0		
1000102	Mr.	Daryl	Anderson	M	U1000102	******			0		
1000103	Mr.	Alfred	Peet	M	U1000103	******			0		
1000104	Mrs	Marta	Peet	F	U1000104	******			0		
1000105	Mr.	Andrew P	Stern	M	U1000105	******			0		
1000106	Mrs	Debra	Stern	F	U1000106	******			0		
1000107	Mrs	G D	Stern	F	U1000107	******			0		
1000108	Mr.	B	Jackson	M	U1000108	******			0		
1000109	Mrs	Jackie	Jackson	F	U1000109	******			0		
1000110	Mrs	Jerry	Jackson	F	U1000110	******			0		
1000111	Mrs	Leslie L	Jackson	F	U1000111	******			0		
1000112	Mr.	Charles A	Jackson	M	U1000112	******			0		
1000113	Mr.	Alfred	Jackson	M	U1000113	******			0		
1000114	Mrs	Renee	Jackson	F	U1000114	******			0		
1000115	Mr.	Cal	Jackson	M	U1000115	******			0		
1000116	Mr.	David	Newsom	M	U1000116	******			0		
1000117	Mrs	Wendy	Newsom	F	U1000117	******			0		
1000118	Mr.	Kerr S	Newsom	M	U1000118	******			0		
1000119	Mr.	Anthony	Peterson	M	U1000119	******			0		
1000120	Mrs	Lorraine	Peterson	F	U1000120	******			0		
1000121	Mrs	Dana	Peterson	F	U1000121	******			0		
1000122	Mr.	Dustin	Peterson	M	U1000122	******			0		
1000123	Mrs	brandi	Peterson	F	U1000123	******			0		
1000124	Mrs	Christine	Peterson	F	U1000124	******			0		
1000125	Mrs	Dana	Sutherland	F	U1000125	******			0		
1000126	Mrs	Kathryn	Sutherland	F	U1000126	******			0		
1000127	Mrs	Deborah D	Sutherland	F	U1000127	******			0		
1000128	Mrs	Robert	Sutherland	M	U1000128	******			0		
1000129	Mr.	Torben	Skov	M	U1000129	******			0		
1000130	Mrs	Barry	Rosen	F	U1000130	******			0		
1000131	Mr.	Erik	Reffeldt	M	U1000131	******			0		
1000132	Mrs	Allison	Preston	F	U1000132	******			0		
1000133	Mr.	Burr	Preston	M	U1000133	******			0		
1000134	Mrs	Cathy	Preston	F	U1000134	******			0		
1000135	Mr.	A	Hopkins	M	U1000135	******			0		
1000136	Mrs	Beata	Hopkins	F	U1000136	******			0		

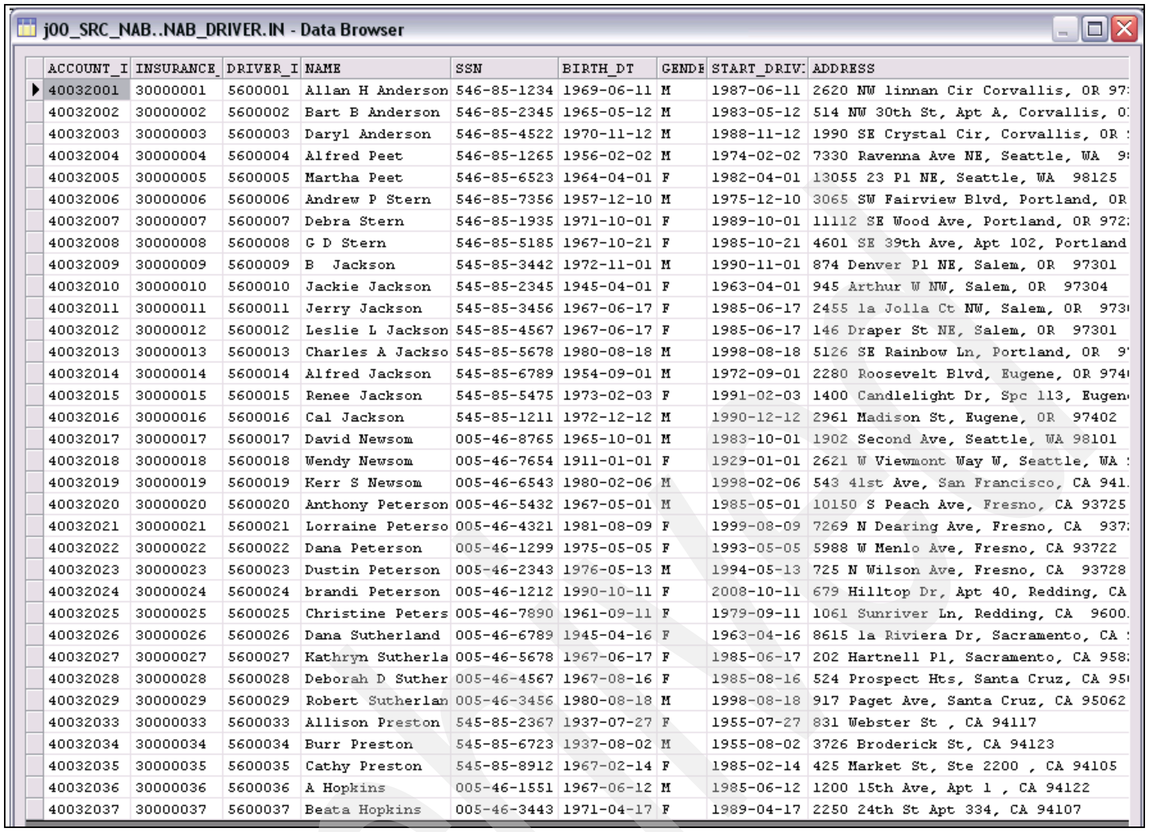
Figure 2-12 Create j00_SRC_NAB job 4/6



j00_SRC_NAB..NAB_CONTACT_INFO.IN - Data Browser

CUSTOMER_ID	ACCOUNT_ID	WORK_PHONE	CEL_PHONE	HOME_PHONE	HOME_ADDRESS	HOME_ZIP
1000100	8800001	(541) 768-6218	(541) 687-0488	(541) 737-0582	2620 NW linnan Cir Corvallis, OR	97330
1000101	8800002	(503) 623-2797	(541) 687-3141	(541) 738-0347	514 NW 30th St, Apt A, Corvallis, OR	97330
1000102	8800003	(541) 752-3917	(206) 282-1578	(541) 754-8522	1990 SE Crystal Cir, Corvallis, OR	97333
1000103	8800004	(206) 447-1566	(206) 447-1875	(206) 527-5780	7330 Ravenna Ave NE, Seattle, WA	98115
1000104	8800005	(206) 731-3465	(415) 395-9000	(206) 440-0133	13055 23 Pl NE, Seattle, WA	98125
1000105	8800006	(503) 256-1300	(559) 257-8000	(503) 227-5288	3065 SW Fairview Blvd, Portland, OR	97205
1000106	8800007	(503) 251-6455	(559) 298-0688	(503) 655-0179	11112 SE Wood Ave, Portland, OR	97222
1000107	8800008	(503) 916-6227	(559) 448-6775	(503) 774-8200	4601 SE 39th Ave, Apt 102, Portland, OR	97202
1000108	8800009	(503) 399-8695	(559) 453-5900	(503) 391-8863	874 Denver Pl NE, Salem, OR	97301
1000109	8800010	(503) 364-2989	(530) 224-6823		945 Arthur W NW, Salem, OR	97304
1000110	8800011	(800) 844-4974	(530) 224-4933	(503) 364-9560	2455 la Jolla Ct NW, Salem, OR	97304
1000111	8800012	(503) 363-3934	(916) 487-4288	(503) 540-3416	146 Draper St NE, Salem, OR	97301
1000112	8800013	(503) 652-9037	(916) 558-6400	(503) 513-0255	5126 SE Rainbow Ln, Portland, OR	97222
1000113	8800014	(800) 935-4600	(877) 379-2096	(541) 654-0821	2280 Roosevelt Blvd, Eugene, OR	97402
1000114	8800015	(541) 687-0488	(831) 459-0445	(541) 689-4459	1400 Candlelight Dr, Spc 113, Eugene, OR	97402
1000115	8800016	(541) 687-3141	(206) 447-1566	(541) 341-3980	2961 Madison St, Eugene, OR	97402
1000116	8800017	(206) 282-1578	(206) 731-3465	(206) 728-2727	1902 Second Ave, Seattle, WA	98101
1000117	8800018	(206) 447-1875	(503) 256-1300	(206) 284-2116	2621 W Viewmont Way W, Seattle, WA	98199
1000118	8800019	(415) 395-9000		(415) 831-4562	543 41st Ave, San Francisco, CA	94117
1000119	8800020	(559) 257-8000		(559) 834-1033	10150 S Peach Ave, Fresno, CA	93725
1000120	8800021	(559) 298-0688		(559) 324-1608	7269 Dearing Avenue, Fresno, CA	93720
1000121	8800022	(559) 448-6775		(559) 275-2319	5988 W Menlo Ave, Fresno, CA	93722
1000122	8800023	(559) 453-5900		(559) 442-4377	725 N Wilson Ave, Fresno, CA	93728
1000123	8800024	(530) 224-6823		(530) 245-0285	679 Hilltop Dr, Apt 40, Redding, CA	96003
1000124	8800025	(530) 224-4933		(530) 229-1086	1061 Sunriver Ln, Redding, CA	96001
1000125	8800026	(916) 487-4288			8615 la Riviera Dr, E, Sacramento, CA	95826
1000126	8800027	(916) 558-6400		(916) 923-5677	202 Hartnell Pl, Sacramento, CA	95825
1000127	8800028	(877) 379-2096		(831) 423-5295	524 Prospect Hts, Santa Cruz, CA	95065
1000128	8800029	(831) 459-0445		(831) 476-8247	917 Paget Ave, Santa Cruz, CA	95062
1000129	8800030	(206) 447-1566		(604) 676-0223	3305 26 Ave W, Vancouver, BC	V5K 1A1
1000130	8800031	(206) 731-3465		(604) 681-5770	1055 W 41, Vancouver, BC	V6M 1W9
1000131	8800032	(503) 256-1300		(604) 730-9266	2201 Pine St, Vancouver, BC	V6J 5E7
1000132	8800033	(415) 789-3105	(999) 999-9999	(999) 999-9999	831 Webster St , CA	94117
1000133	8800034	(415) 346-1611	(408) 527-1879	(999) 999-9999	3726 Broderick St, CA	94123
1000134	8800035	(415) 984-3772	(415) 683-0763	(999) 999-9999	425 Market St, Ste 2200 , CA	94105
1000135	8800036	(415) 664-0983	(415) 561-8511	(999) 999-9999	1200 15th Ave, Apt 1 , CA	94122
1000136	8800037	(415) 826-1031	(415) 296-9450	(999) 999-9999	2250 24th St Apt 334, CA	94107

Figure 2-13 Create j00_SRC_NAB job 5/6



j00_SRC_NAB..NAB_DRIVER.IN - Data Browser

ACCOUNT_I	INSURANCE_	DRIVER_I	NAME	SSN	BIRTH_DT	GENDE	START_DRIV	ADDRESS
40032001	30000001	5600001	Allan H Anderson	546-85-1234	1969-06-11	M	1987-06-11	2620 NW linnan Cir Corvallis, OR 97
40032002	30000002	5600002	Bart B Anderson	546-85-2345	1965-05-12	M	1983-05-12	514 NW 30th St, Apt A, Corvallis, O
40032003	30000003	5600003	Daryl Anderson	546-85-4522	1970-11-12	M	1988-11-12	1990 SE Crystal Cir, Corvallis, OR
40032004	30000004	5600004	Alfred Peet	546-85-1265	1956-02-02	M	1974-02-02	7330 Ravenna Ave NE, Seattle, WA 9
40032005	30000005	5600005	Martha Peet	546-85-6523	1964-04-01	F	1982-04-01	13055 23 Pl NE, Seattle, WA 98125
40032006	30000006	5600006	Andrew P Stern	546-85-7356	1957-12-10	M	1975-12-10	3065 SW Fairview Blvd, Portland, OR
40032007	30000007	5600007	Debra Stern	546-85-1935	1971-10-01	F	1989-10-01	11112 SE Wood Ave, Portland, OR 972
40032008	30000008	5600008	G D Stern	546-85-5185	1967-10-21	F	1985-10-21	4601 SE 39th Ave, Apt 102, Portland
40032009	30000009	5600009	B Jackson	545-85-3442	1972-11-01	M	1990-11-01	874 Denver Pl NE, Salem, OR 97301
40032010	30000010	5600010	Jackie Jackson	545-85-2345	1945-04-01	F	1963-04-01	945 Arthur W NW, Salem, OR 97304
40032011	30000011	5600011	Jerry Jackson	545-85-3456	1967-06-17	F	1985-06-17	2455 la Jolla Ct NW, Salem, OR 973
40032012	30000012	5600012	Leslie L Jackson	545-85-4567	1967-06-17	F	1985-06-17	146 Draper St NE, Salem, OR 97301
40032013	30000013	5600013	Charles A Jackso	545-85-5678	1980-08-18	M	1998-08-18	5126 SE Rainbow Ln, Portland, OR 9
40032014	30000014	5600014	Alfred Jackson	545-85-6789	1954-09-01	M	1972-09-01	2280 Roosevelt Blvd, Eugene, OR 974
40032015	30000015	5600015	Renee Jackson	545-85-5475	1973-02-03	F	1991-02-03	1400 Candlelight Dr, Spc 113, Eugen
40032016	30000016	5600016	Cal Jackson	545-85-1211	1972-12-12	M	1990-12-12	2961 Madison St, Eugene, OR 97402
40032017	30000017	5600017	David Newsom	005-46-8765	1965-10-01	M	1983-10-01	1902 Second Ave, Seattle, WA 98101
40032018	30000018	5600018	Wendy Newsom	005-46-7654	1911-01-01	F	1929-01-01	2621 W Viewmont Way W, Seattle, WA
40032019	30000019	5600019	Kerr S Newsom	005-46-6543	1980-02-06	M	1998-02-06	543 41st Ave, San Francisco, CA 941
40032020	30000020	5600020	Anthony Peterson	005-46-5432	1967-05-01	M	1985-05-01	10150 S Peach Ave, Fresno, CA 93725
40032021	30000021	5600021	Lorraine Peterso	005-46-4321	1981-08-09	F	1999-08-09	7269 N Dearing Ave, Fresno, CA 937
40032022	30000022	5600022	Dana Peterson	005-46-1299	1975-05-05	F	1993-05-05	5988 W Menlo Ave, Fresno, CA 93722
40032023	30000023	5600023	Dustin Peterson	005-46-2343	1976-05-13	M	1994-05-13	725 N Wilson Ave, Fresno, CA 93728
40032024	30000024	5600024	brandi Peterson	005-46-1212	1990-10-11	F	2008-10-11	679 Hilltop Dr, Apt 40, Redding, CA
40032025	30000025	5600025	Christine Peters	005-46-7890	1961-09-11	F	1979-09-11	1061 Sunriver Ln, Redding, CA 9600
40032026	30000026	5600026	Dana Sutherland	005-46-6789	1945-04-16	F	1963-04-16	8615 la Riviera Dr, Sacramento, CA
40032027	30000027	5600027	Kathryn Sutherla	005-46-5678	1967-06-17	F	1985-06-17	202 Hartnell Pl, Sacramento, CA 958
40032028	30000028	5600028	Deborah D Suther	005-46-4567	1967-08-16	F	1985-08-16	524 Prospect Hts, Santa Cruz, CA 95
40032029	30000029	5600029	Robert Sutherlan	005-46-3456	1980-08-18	M	1998-08-18	917 Paget Ave, Santa Cruz, CA 95062
40032033	30000033	5600033	Allison Preston	545-85-2367	1937-07-27	F	1955-07-27	831 Webster St , CA 94117
40032034	30000034	5600034	Burr Preston	545-85-6723	1937-08-02	M	1955-08-02	3726 Broderick St, CA 94123
40032035	30000035	5600035	Cathy Preston	545-85-8912	1967-02-14	F	1985-02-14	425 Market St, Ste 2200 , CA 94105
40032036	30000036	5600036	A Hopkins	005-46-1551	1967-06-12	M	1985-06-12	1200 15th Ave, Apt 1 , CA 94122
40032037	30000037	5600037	Beata Hopkins	005-46-3443	1971-04-17	F	1989-04-17	2250 24th St Apt 334, CA 94107

Figure 2-14 Create j00_SRC_NAB job 6/6

j01_STAN_COUNTRY_NAB

In this step, we analyze the addresses in appropriate data sets to determine the (ISO code) country using the COUNTRY rule set in the Standardize stage, which produces two additional columns ISOCountryCode_COUNTRY and IdentifierFlag_COUNTRY.

Figure 2-15 on page 520 shows the various stages that are used in this job, including the data sets that were created in “j00_SRC_NAB” on page 512, a Copy stage, three Standardize stages (one for each address field), and three output Data Set stages (one for each address field). The Copy stage is used to create two copies of the same input CONTACT_INFO file in order for separate Standardize stages to process the home address information (which can contain U.S. and foreign addresses) and work address (U.S. only). The DRIVER file only has a single address information. We modified the names of the stages as shown.

Because the configuration of this job is very similar to that described in “J01_STAN_COUNTRY” on page 168, we do not repeat it here. However, some of the configurations of interest are as follows:

- Figure 2-16 on page 520 shows the Standardize Rule Process window with the configured COUNTRY rule set and the literal ZQUSZQ followed by the HOME_ADDRESS and HOME_ZIP columns of the CONTACT_INFO file in the Selected Columns list.
- Figure 2-17 on page 521 shows the Standardize Rule Process window with the configured COUNTRY rule set and the literal ZQUSZQ followed by the WORK_ADDRESS and WORK_ZIP columns of the CONTACT_INFO file in the Selected Columns list.
- Figure 2-18 on page 521 shows the Standardize Rule Process window with the configured COUNTRY rule set and the literal ZQUSZQ followed by the CITY, STATE and ZIP columns of the DRIVER file in the Selected Columns list.
- After saving, compiling, and running the job, Figure 2-19 on page 522 shows the results of the execution.
- Figure 2-20 on page 523, Figure 2-21 on page 524, and Figure 2-22 on page 525 show the reports that are produced for the three sources. Because the volumes are small in this case, you can view these reports to see the distribution of country addresses. In the real word where large volumes of data are involved, you need to run Investigate to determine the countries detected as described in “j02_INVCC_ISOCODE_NAB” on page 525.

Now proceed to “j02_INVCC_ISOCODE_NAB” on page 525.

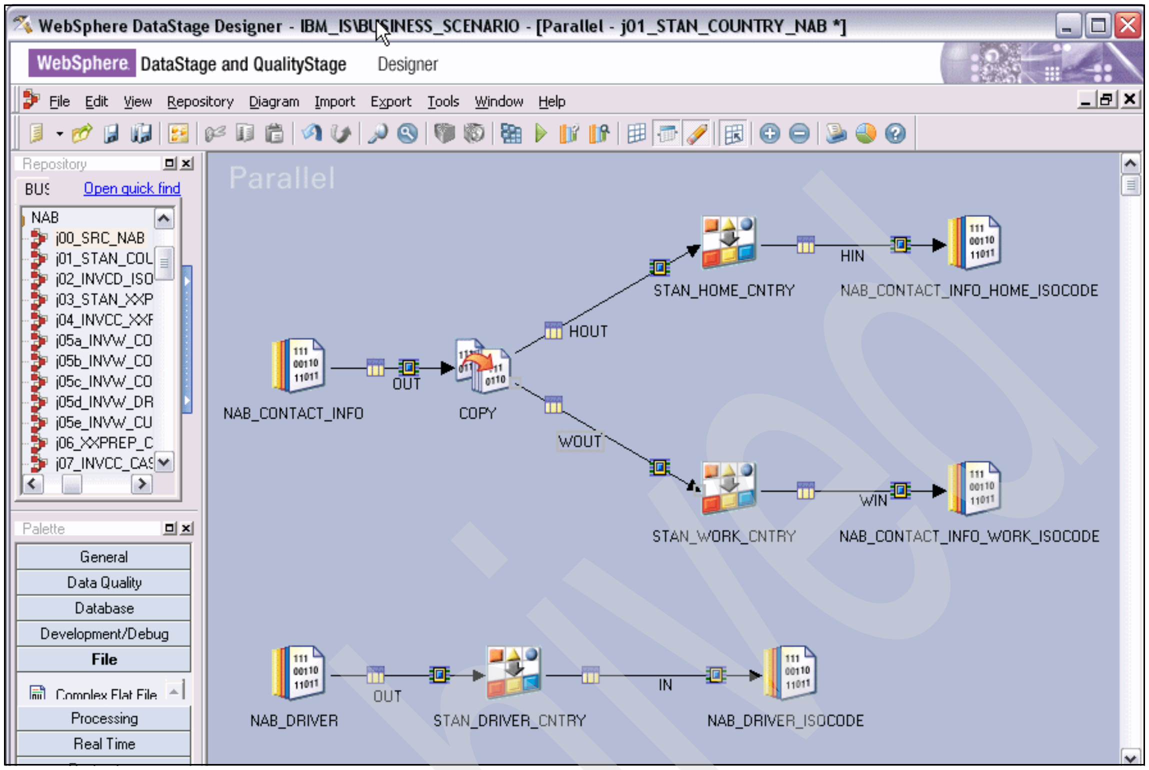
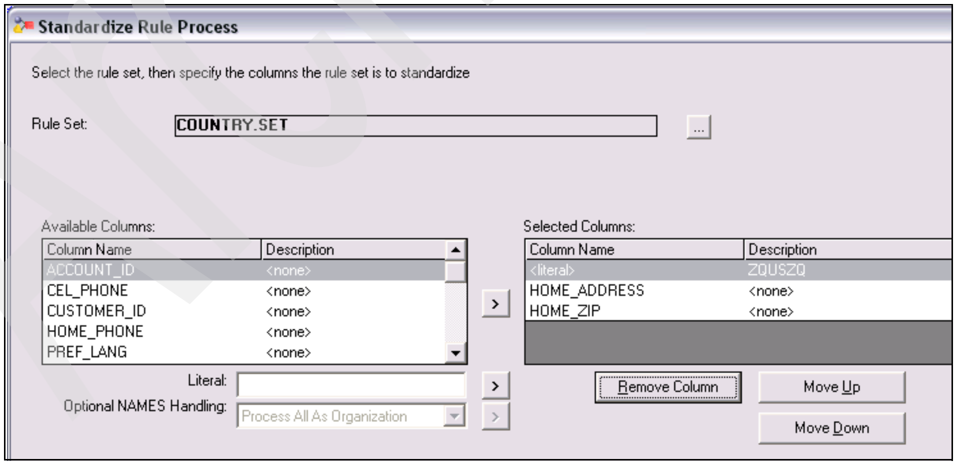


Figure 2-15 Create j01_STAN_COUNTRY_NAB 1/8



Select the rule set, then specify the columns the rule set is to standardize

Rule Set: COUNTRY.SET

Available Columns:

Column Name	Description
ACCOUNT_ID	<none>
CEL_PHONE	<none>
CUSTOMER_ID	<none>
HOME_PHONE	<none>
PREF_LANG	<none>

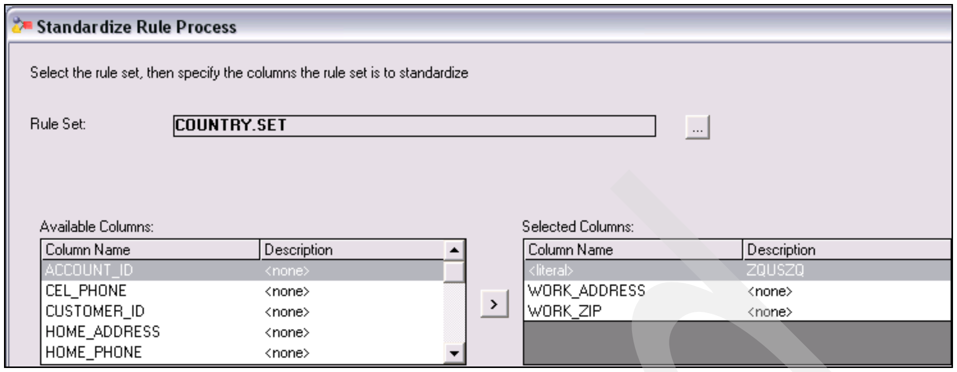
Selected Columns:

Column Name	Description
<literal>	ZQUSZQ
HOME_ADDRESS	<none>
HOME_ZIP	<none>

Literal:

Optional NAMES Handling: Process All As Organization

Figure 2-16 Create j01_STAN_COUNTRY_NAB 2/8


Standardize Rule Process

Select the rule set, then specify the columns the rule set is to standardize

Rule Set: COUNTRY.SET

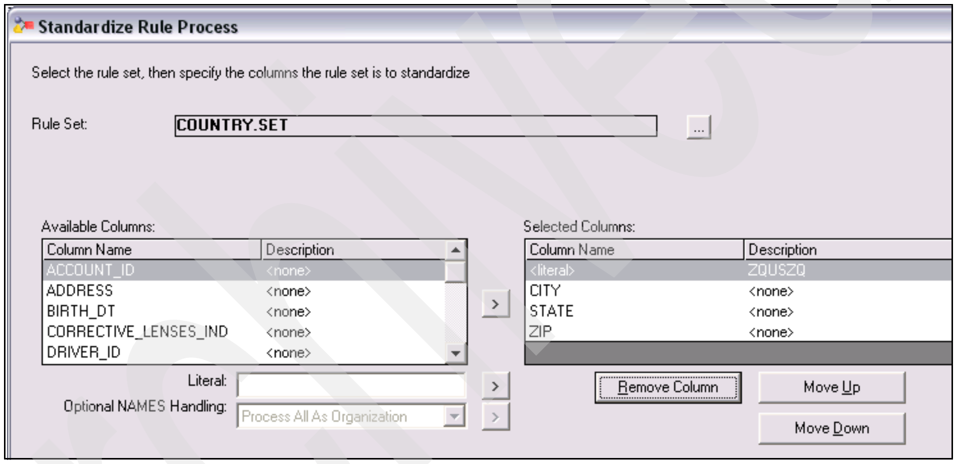
Available Columns:

Column Name	Description
ACCOUNT_ID	<none>
CEL_PHONE	<none>
CUSTOMER_ID	<none>
HOME_ADDRESS	<none>
HOME_PHONE	<none>

Selected Columns:

Column Name	Description
<literal>	ZQUSZQ
WORK_ADDRESS	<none>
WORK_ZIP	<none>

Figure 2-17 Create j01_STAN_COUNTRY_NAB 3/8


Standardize Rule Process

Select the rule set, then specify the columns the rule set is to standardize

Rule Set: COUNTRY.SET

Available Columns:

Column Name	Description
ACCOUNT_ID	<none>
ADDRESS	<none>
BIRTH_DT	<none>
CORRECTIVE_LENSES_IND	<none>
DRIVER_ID	<none>

Selected Columns:

Column Name	Description
<literal>	ZQUSZQ
CITY	<none>
STATE	<none>
ZIP	<none>

Literal:

Optional NAMES Handling: Process All As Organization

Figure 2-18 Create j01_STAN_COUNTRY_NAB 4/8

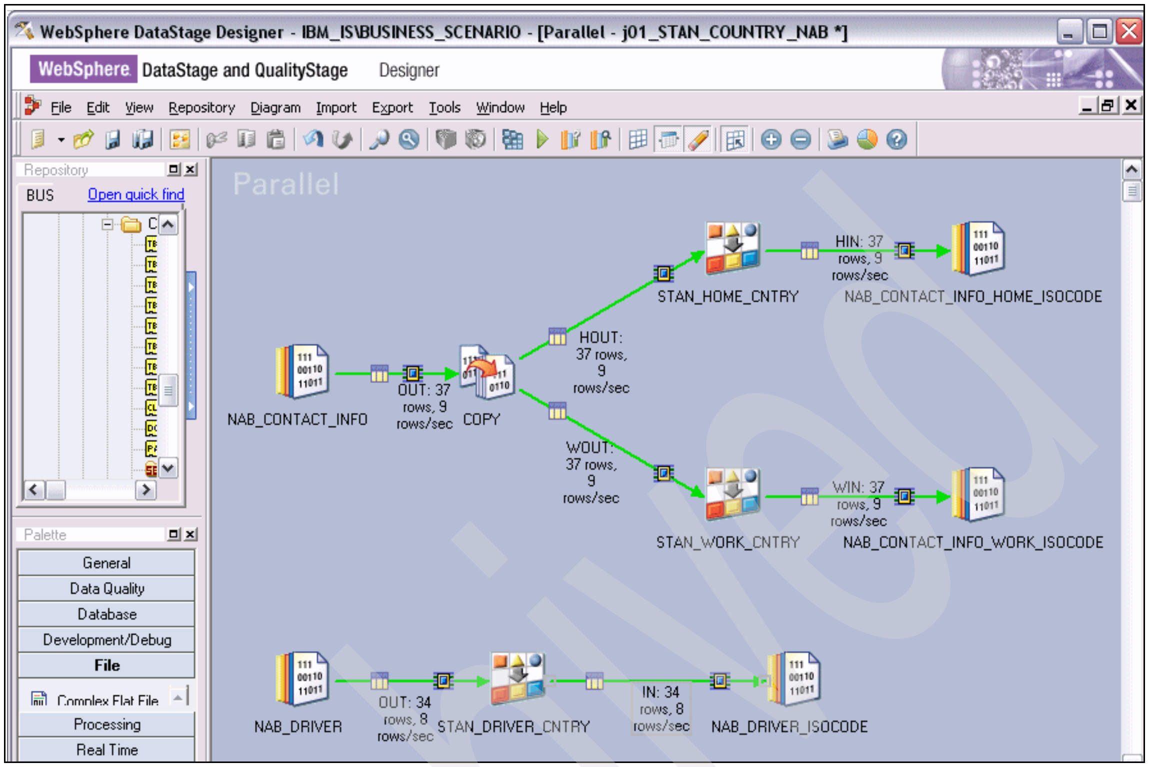
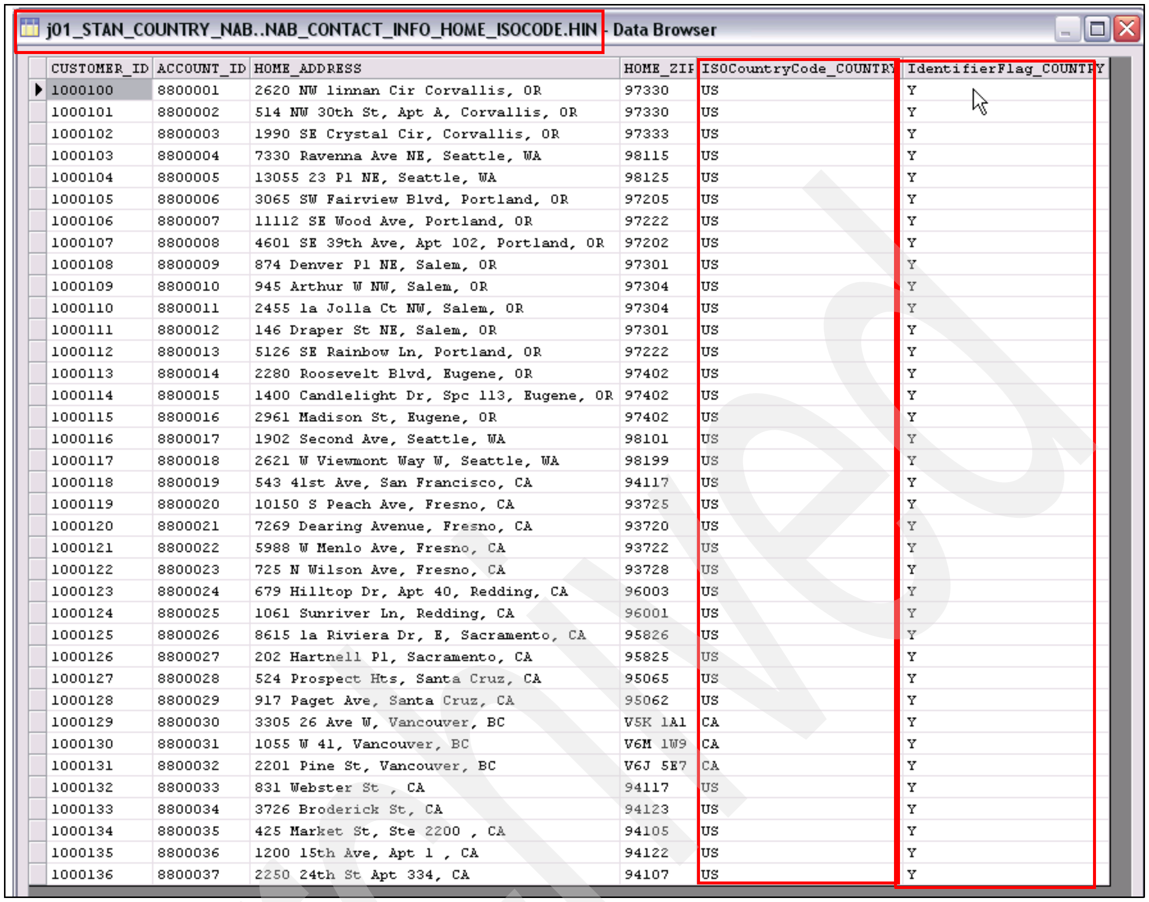


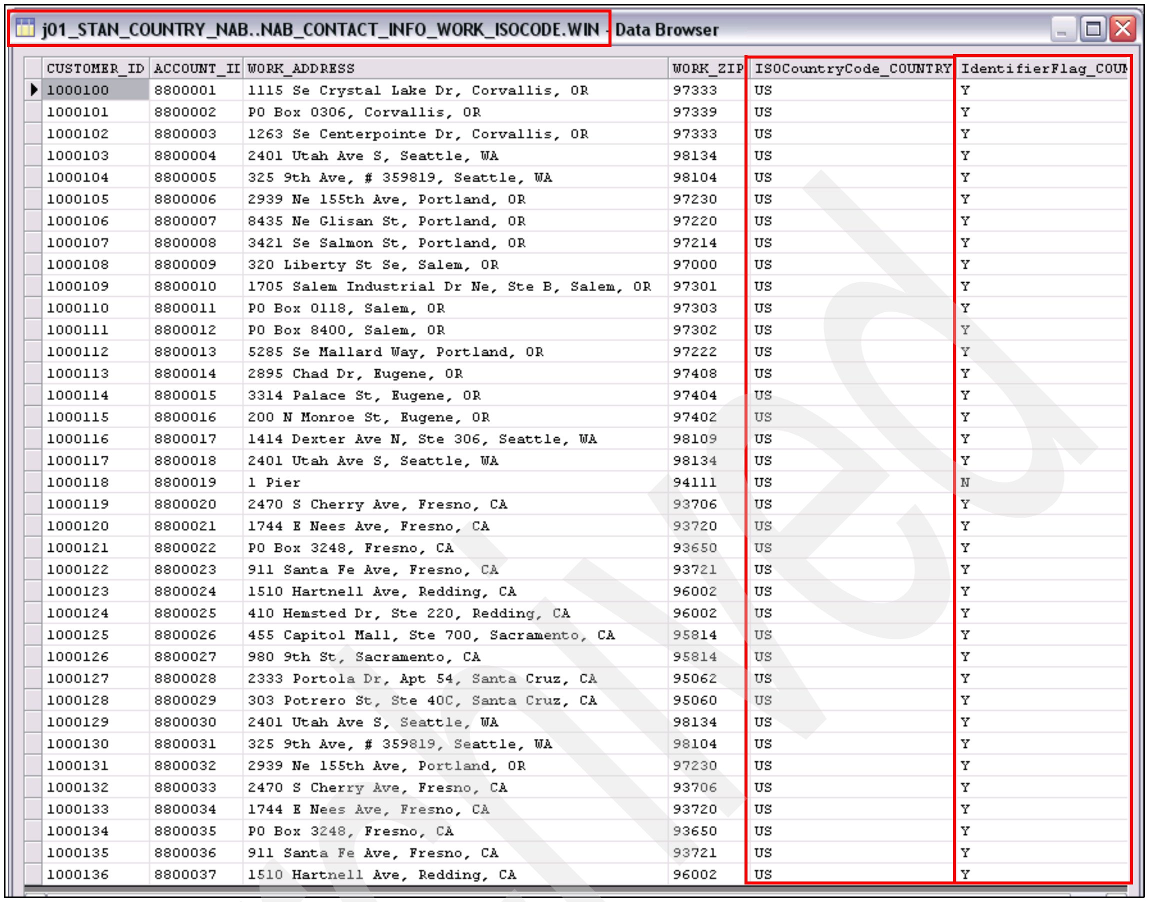
Figure 2-19 Create j01_STAN_COUNTRY_NAB 5/8



j01_STAN_COUNTRY_NAB..NAB_CONTACT_INFO_HOME_ISOCODE.HIN - Data Browser

CUSTOMER_ID	ACCOUNT_ID	HOME_ADDRESS	HOME_ZIP	ISOCountryCode_COUNTRY	IdentifierFlag_COUNTRY
1000100	8800001	2620 NW linnan Cir Corvallis, OR	97330	US	Y
1000101	8800002	514 NW 30th St, Apt A, Corvallis, OR	97330	US	Y
1000102	8800003	1990 SE Crystal Cir, Corvallis, OR	97333	US	Y
1000103	8800004	7330 Ravenna Ave NE, Seattle, WA	98115	US	Y
1000104	8800005	13055 23 Pl NE, Seattle, WA	98125	US	Y
1000105	8800006	3065 SW Fairview Blvd, Portland, OR	97205	US	Y
1000106	8800007	11112 SE Wood Ave, Portland, OR	97222	US	Y
1000107	8800008	4601 SE 39th Ave, Apt 102, Portland, OR	97202	US	Y
1000108	8800009	874 Denver Pl NE, Salem, OR	97301	US	Y
1000109	8800010	945 Arthur W NW, Salem, OR	97304	US	Y
1000110	8800011	2455 la Jolla Ct NW, Salem, OR	97304	US	Y
1000111	8800012	146 Draper St NE, Salem, OR	97301	US	Y
1000112	8800013	5126 SE Rainbow Ln, Portland, OR	97222	US	Y
1000113	8800014	2280 Roosevelt Blvd, Eugene, OR	97402	US	Y
1000114	8800015	1400 Candlelight Dr, Spc 113, Eugene, OR	97402	US	Y
1000115	8800016	2961 Madison St, Eugene, OR	97402	US	Y
1000116	8800017	1902 Second Ave, Seattle, WA	98101	US	Y
1000117	8800018	2621 W Viewmont Way W, Seattle, WA	98199	US	Y
1000118	8800019	543 41st Ave, San Francisco, CA	94117	US	Y
1000119	8800020	10150 S Peach Ave, Fresno, CA	93725	US	Y
1000120	8800021	7269 Dearing Avenue, Fresno, CA	93720	US	Y
1000121	8800022	5988 W Menlo Ave, Fresno, CA	93722	US	Y
1000122	8800023	725 N Wilson Ave, Fresno, CA	93728	US	Y
1000123	8800024	679 Hilltop Dr, Apt 40, Redding, CA	96003	US	Y
1000124	8800025	1061 Sunriver Ln, Redding, CA	96001	US	Y
1000125	8800026	8615 la Riviera Dr, E, Sacramento, CA	95826	US	Y
1000126	8800027	202 Hartnell Pl, Sacramento, CA	95825	US	Y
1000127	8800028	524 Prospect Hts, Santa Cruz, CA	95065	US	Y
1000128	8800029	917 Paget Ave, Santa Cruz, CA	95062	US	Y
1000129	8800030	3305 26 Ave W, Vancouver, BC	V5K 1A1	CA	Y
1000130	8800031	1055 W 41, Vancouver, BC	V6M 1W9	CA	Y
1000131	8800032	2201 Pine St, Vancouver, BC	V6J 5E7	CA	Y
1000132	8800033	831 Webster St , CA	94117	US	Y
1000133	8800034	3726 Broderick St, CA	94123	US	Y
1000134	8800035	425 Market St, Ste 2200 , CA	94105	US	Y
1000135	8800036	1200 15th Ave, Apt 1 , CA	94122	US	Y
1000136	8800037	2250 24th St Apt 334, CA	94107	US	Y

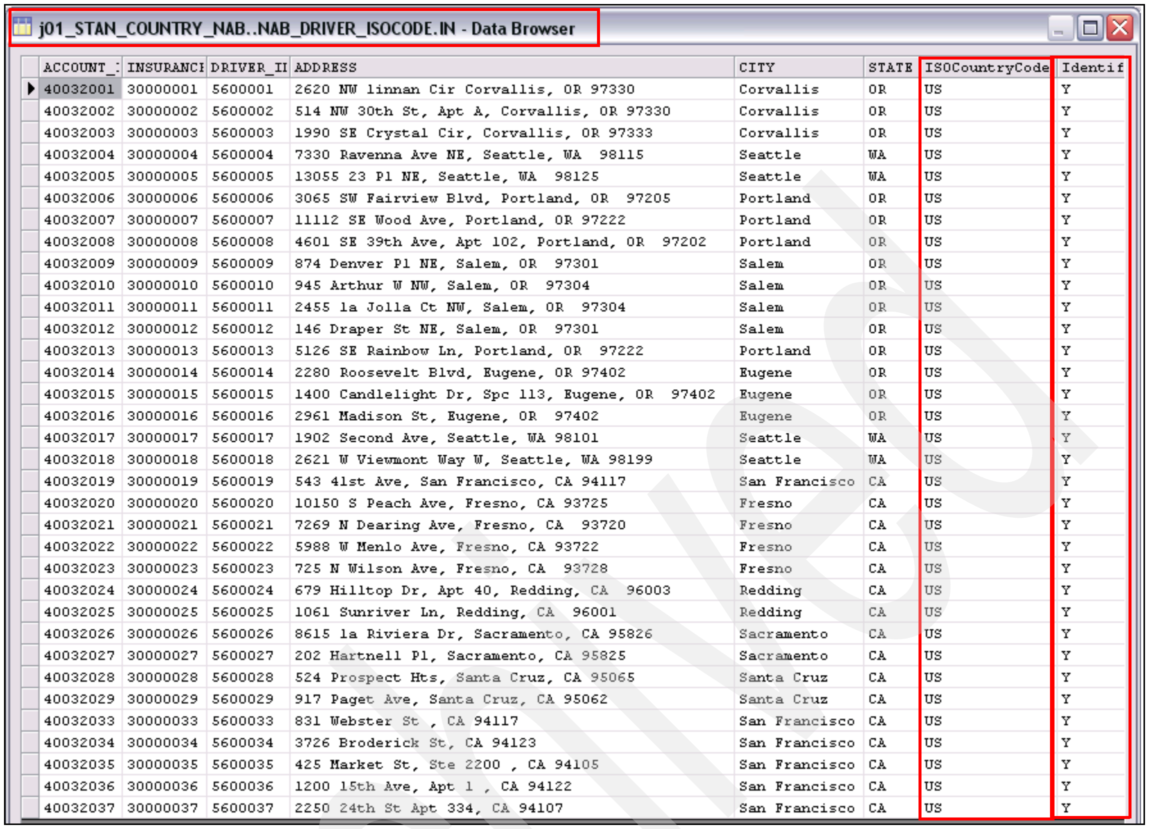
Figure 2-20 Create j01_STAN_COUNTRY_NAB 6/8



j01_STAN_COUNTRY_NAB..NAB_CONTACT_INFO_WORK_ISOCODE.WIN - Data Browser

CUSTOMER_ID	ACCOUNT_II	WORK_ADDRESS	WORK_ZIP	ISOCountryCode_COUNTRY	IdentifierFlag_COUN
1000100	8800001	1115 Se Crystal Lake Dr, Corvallis, OR	97333	US	Y
1000101	8800002	PO Box 0306, Corvallis, OR	97339	US	Y
1000102	8800003	1263 Se Centerpointe Dr, Corvallis, OR	97333	US	Y
1000103	8800004	2401 Utah Ave S, Seattle, WA	98134	US	Y
1000104	8800005	325 9th Ave, # 359819, Seattle, WA	98104	US	Y
1000105	8800006	2939 Ne 155th Ave, Portland, OR	97230	US	Y
1000106	8800007	8435 Ne Glisan St, Portland, OR	97220	US	Y
1000107	8800008	3421 Se Salmon St, Portland, OR	97214	US	Y
1000108	8800009	320 Liberty St Se, Salem, OR	97000	US	Y
1000109	8800010	1705 Salem Industrial Dr Ne, Ste B, Salem, OR	97301	US	Y
1000110	8800011	PO Box 0118, Salem, OR	97303	US	Y
1000111	8800012	PO Box 8400, Salem, OR	97302	US	Y
1000112	8800013	5285 Se Mallard Way, Portland, OR	97222	US	Y
1000113	8800014	2895 Chad Dr, Eugene, OR	97408	US	Y
1000114	8800015	3314 Palace St, Eugene, OR	97404	US	Y
1000115	8800016	200 N Monroe St, Eugene, OR	97402	US	Y
1000116	8800017	1414 Dexter Ave N, Ste 306, Seattle, WA	98109	US	Y
1000117	8800018	2401 Utah Ave S, Seattle, WA	98134	US	Y
1000118	8800019	1 Pier	94111	US	N
1000119	8800020	2470 S Cherry Ave, Fresno, CA	93706	US	Y
1000120	8800021	1744 E Nees Ave, Fresno, CA	93720	US	Y
1000121	8800022	PO Box 3248, Fresno, CA	93650	US	Y
1000122	8800023	911 Santa Fe Ave, Fresno, CA	93721	US	Y
1000123	8800024	1510 Hartnell Ave, Redding, CA	96002	US	Y
1000124	8800025	410 Hemsted Dr, Ste 220, Redding, CA	96002	US	Y
1000125	8800026	455 Capitol Mall, Ste 700, Sacramento, CA	95814	US	Y
1000126	8800027	980 9th St, Sacramento, CA	95814	US	Y
1000127	8800028	2333 Portola Dr, Apt 54, Santa Cruz, CA	95062	US	Y
1000128	8800029	303 Potrero St, Ste 40C, Santa Cruz, CA	95060	US	Y
1000129	8800030	2401 Utah Ave S, Seattle, WA	98134	US	Y
1000130	8800031	325 9th Ave, # 359819, Seattle, WA	98104	US	Y
1000131	8800032	2939 Ne 155th Ave, Portland, OR	97230	US	Y
1000132	8800033	2470 S Cherry Ave, Fresno, CA	93706	US	Y
1000133	8800034	1744 E Nees Ave, Fresno, CA	93720	US	Y
1000134	8800035	PO Box 3248, Fresno, CA	93650	US	Y
1000135	8800036	911 Santa Fe Ave, Fresno, CA	93721	US	Y
1000136	8800037	1510 Hartnell Ave, Redding, CA	96002	US	Y

Figure 2-21 Create j01_STAN_COUNTRY_NAB 7/8



j01_STAN_COUNTRY_NAB..NAB_DRIVER_ISOCODE.IN - Data Browser

ACCOUNT_:	INSURANCI	DRIVER_II	ADDRESS	CITY	STATE	ISOCountryCode	Identif
40032001	30000001	5600001	2620 NW linnan Cir Corvallis, OR 97330	Corvallis	OR	US	Y
40032002	30000002	5600002	514 NW 30th St, Apt A, Corvallis, OR 97330	Corvallis	OR	US	Y
40032003	30000003	5600003	1990 SE Crystal Cir, Corvallis, OR 97333	Corvallis	OR	US	Y
40032004	30000004	5600004	7330 Ravenna Ave NE, Seattle, WA 98115	Seattle	WA	US	Y
40032005	30000005	5600005	13055 23 Pl NE, Seattle, WA 98125	Seattle	WA	US	Y
40032006	30000006	5600006	3065 SW Fairview Blvd, Portland, OR 97205	Portland	OR	US	Y
40032007	30000007	5600007	11112 SE Wood Ave, Portland, OR 97222	Portland	OR	US	Y
40032008	30000008	5600008	4601 SE 39th Ave, Apt 102, Portland, OR 97202	Portland	OR	US	Y
40032009	30000009	5600009	874 Denver Pl NE, Salem, OR 97301	Salem	OR	US	Y
40032010	30000010	5600010	945 Arthur W NW, Salem, OR 97304	Salem	OR	US	Y
40032011	30000011	5600011	2455 la Jolla Ct NW, Salem, OR 97304	Salem	OR	US	Y
40032012	30000012	5600012	146 Draper St NE, Salem, OR 97301	Salem	OR	US	Y
40032013	30000013	5600013	5126 SE Rainbow Ln, Portland, OR 97222	Portland	OR	US	Y
40032014	30000014	5600014	2280 Roosevelt Blvd, Eugene, OR 97402	Eugene	OR	US	Y
40032015	30000015	5600015	1400 Candlelight Dr, Spc 113, Eugene, OR 97402	Eugene	OR	US	Y
40032016	30000016	5600016	2961 Madison St, Eugene, OR 97402	Eugene	OR	US	Y
40032017	30000017	5600017	1902 Second Ave, Seattle, WA 98101	Seattle	WA	US	Y
40032018	30000018	5600018	2621 W Viewmont Way W, Seattle, WA 98199	Seattle	WA	US	Y
40032019	30000019	5600019	543 41st Ave, San Francisco, CA 94117	San Francisco	CA	US	Y
40032020	30000020	5600020	10150 S Peach Ave, Fresno, CA 93725	Fresno	CA	US	Y
40032021	30000021	5600021	7269 N Dearing Ave, Fresno, CA 93720	Fresno	CA	US	Y
40032022	30000022	5600022	5988 W Menlo Ave, Fresno, CA 93722	Fresno	CA	US	Y
40032023	30000023	5600023	725 N Wilson Ave, Fresno, CA 93728	Fresno	CA	US	Y
40032024	30000024	5600024	679 Hilltop Dr, Apt 40, Redding, CA 96003	Redding	CA	US	Y
40032025	30000025	5600025	1061 Sunriver Ln, Redding, CA 96001	Redding	CA	US	Y
40032026	30000026	5600026	8615 la Riviera Dr, Sacramento, CA 95826	Sacramento	CA	US	Y
40032027	30000027	5600027	202 Hartnell Pl, Sacramento, CA 95825	Sacramento	CA	US	Y
40032028	30000028	5600028	524 Prospect Hts, Santa Cruz, CA 95065	Santa Cruz	CA	US	Y
40032029	30000029	5600029	917 Paget Ave, Santa Cruz, CA 95062	Santa Cruz	CA	US	Y
40032033	30000033	5600033	831 Webster St , CA 94117	San Francisco	CA	US	Y
40032034	30000034	5600034	3726 Broderick St, CA 94123	San Francisco	CA	US	Y
40032035	30000035	5600035	425 Market St, Ste 2200 , CA 94105	San Francisco	CA	US	Y
40032036	30000036	5600036	1200 15th Ave, Apt 1 , CA 94122	San Francisco	CA	US	Y
40032037	30000037	5600037	2250 24th St Apt 334, CA 94107	San Francisco	CA	US	Y

Figure 2-22 Create j01_STAN_COUNTRY_NAB 8/8

j02_INVCC_ISOCODE_NAB

We analyze the ISO codes that were generated by the previous step by the Investigate stage using the character concatenate option with the *C* mask to obtain frequency distribution. This step identifies whether the addresses in the appropriate data sets belongs to more than one country and identifies the codes of the countries in the addresses.

In this case, the home addresses in the CONTACT_INFO data set are a mix of U.S. and Canadian addresses, while the work addresses in the CONTACT_INFO data set are all U.S. addresses. The addresses in the DRIVER data set are all U.S. addresses.

Figure 2-23 on page 527 shows the various stages that are used in this job, including the data sets that were created in "j01_STAN_COUNTRY_NAB" on page 518, an Investigate stage, and an output Sequential File stage for each of the input files. We modified the names of the stages as shown.

Because the configuration of this job is very similar to that described in “J02_INVCC_ISCODE” on page 186, we do not repeat it here. However, some of the configurations of interest are as follows:

- Figure 2-24 on page 527 shows the INV_CC_ISOCODE1 - Investigate Stage window with the Character Concatenate Investigate option and columns ISOCountryCode_Country and IdentifierFlag_COUNTRY selected with the *C* masks for each character. The *X* masks are used for columns HOME_ADDRESS and HOME_ZIP.
- Figure 2-25 on page 528 shows the INV_CC_ISOCODE2 - Investigate Stage window with the Character Concatenate Investigate option and columns ISOCountryCode_Country and IdentifierFlag_COUNTRY selected with the *C* masks for each character. The *X* masks are used for columns WORK_ADDRESS and WORK_ZIP.
- Figure 2-26 on page 528 shows the INV_CC_ISOCODE3 - Investigate Stage window with the Character Concatenate Investigate option and columns ISOCountryCode_Country and IdentifierFlag_COUNTRY selected with the *C* masks for each character. The *X* masks are used for columns CITY, STATE, and ZIP.
- After saving, compiling, and running this job, the job statistics are shown in Figure 2-27 on page 529.
- The output of the Investigate stage written to the sequential files are shown in Figure 2-28 on page 529 through Figure 2-30 on page 530 as follows:
  - Figure 2-28 on page 529 shows the CONTACT_INFO home address report with a concatenated value of `US Y` in 91.8919% of the records and a concatenated value of `CA Y` in 0.10811% of the records.
  - Figure 2-29 on page 529 shows the CONTACT_INFO work address report with a concatenated value of `US Y` in 97.2973% of the records and a concatenated value of `US N`[5] in 2.7027% of the records.
  - Figure 2-30 on page 530 shows the DRIVER address report with a concatenated value of `US Y` in 100% of the records.

Now proceed to “j03_STAN_XXPREP_NAB” on page 530.

[5] This means that the COUNTRY rule set could not determine the country and therefore took the default value of “US”.

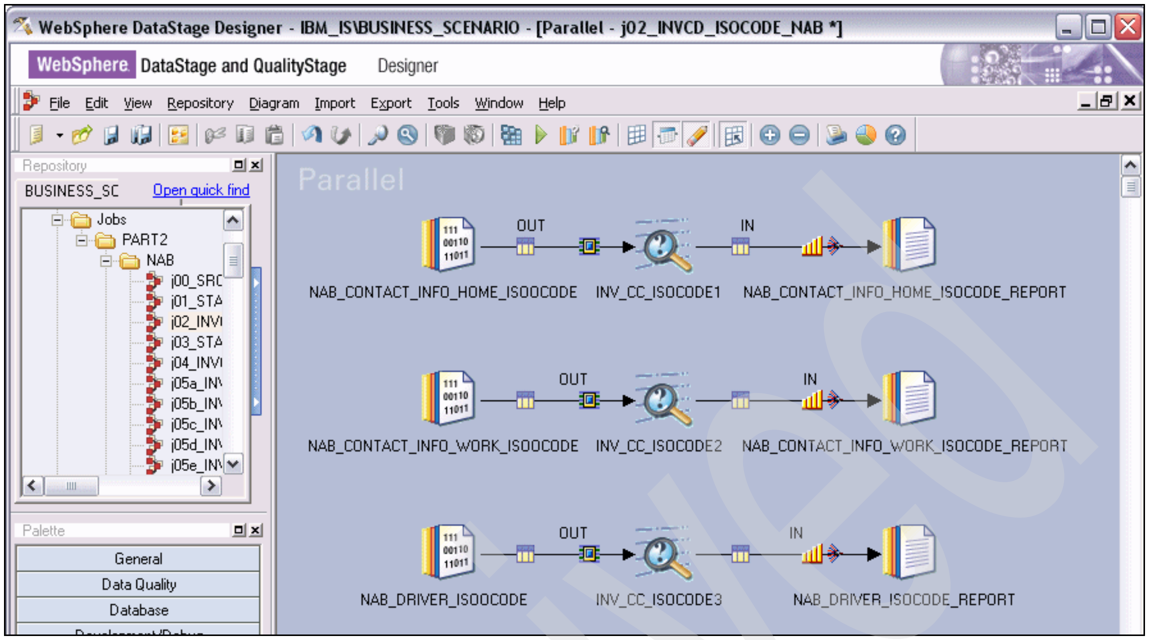
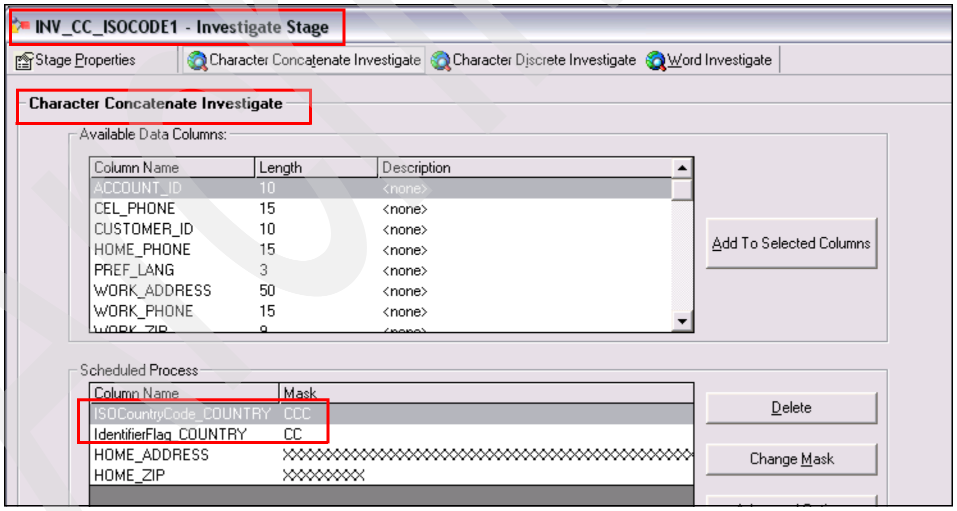

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j02_INVCD_ISOCODE_NAB *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Repository
BUSINESS_SC
Open quick find
Jobs
PART2
NAB
j00_SRC
j01_STA
j02_INVI
j03_STA
j04_INVI
j05a_IN\
j05b_IN\
j05c_IN\
j05d_IN\
j05e_IN\
Palette
General
Data Quality
Database
Parallel
OUT
IN
NAB_CONTACT_INFO_HOME_ISOOCODE
INV_CC_ISOCODE1
NAB_CONTACT_INFO_HOME_ISOCODE_REPORT
NAB_CONTACT_INFO_WORK_ISOOCODE
INV_CC_ISOCODE2
NAB_CONTACT_INFO_WORK_ISOCODE_REPORT
NAB_DRIVER_ISOOCODE
INV_CC_ISOCODE3
NAB_DRIVER_ISOCODE_REPORT

Figure 2-23 Create j02_INVCC_ISOCODE_NAB 1/8


INV_CC_ISOCODE1 - Investigate Stage
Stage Properties
Character Concatenate Investigate
Character Discrete Investigate
Word Investigate
Character Concatenate Investigate
Add To Selected Columns
Delete
Change Mask

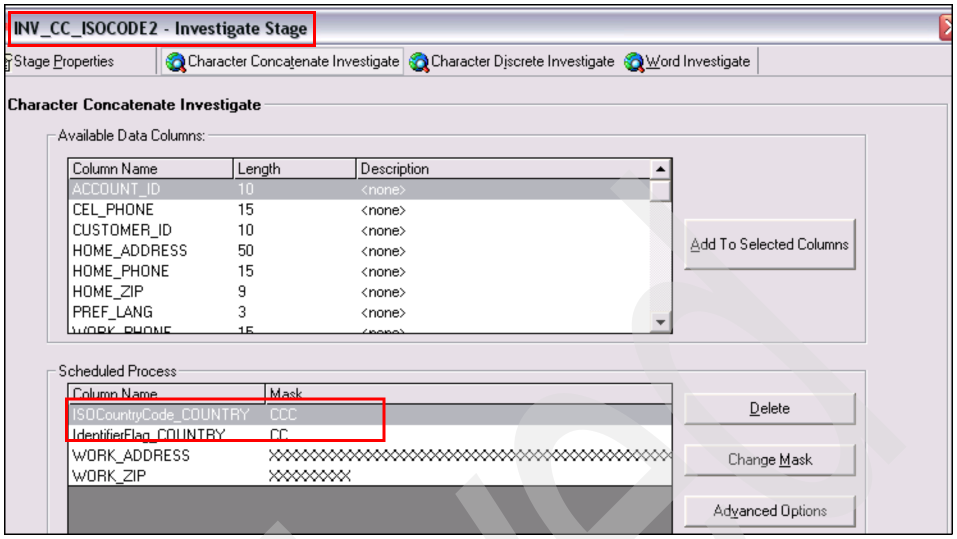
Available Data Columns:

Column Name	Length	Description
ACCOUNT_ID	10	<none>
CEL_PHONE	15	<none>
CUSTOMER_ID	10	<none>
HOME_PHONE	15	<none>
PREF_LANG	3	<none>
WORK_ADDRESS	50	<none>
WORK_PHONE	15	<none>

Scheduled Process

Column Name	Mask
ISOCountryCode_COUNTRY	CCC
IdentifierFlag_COUNTRY	CC
HOME_ADDRESS	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
HOME_ZIP	XXXXXXXXX

Figure 2-24 Create j02_INVCC_ISOCODE_NAB 2/8


INV_CC_ISOCODE2 - Investigate Stage
Stage Properties
Character Concatenate Investigate
Character Discrete Investigate
Word Investigate
Add To Selected Columns
Delete
Change Mask
Advanced Options

Character Concatenate Investigate

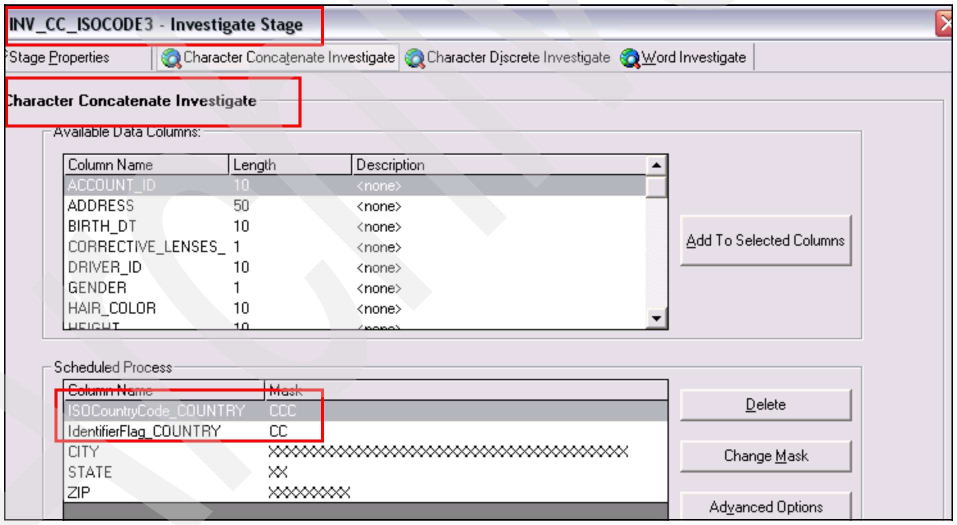
Available Data Columns:

Column Name	Length	Description
ACCOUNT_ID	10	<none>
CEL_PHONE	15	<none>
CUSTOMER_ID	10	<none>
HOME_ADDRESS	50	<none>
HOME_PHONE	15	<none>
HOME_ZIP	9	<none>
PREF_LANG	3	<none>
WORK_PHONE	15	<none>

Scheduled Process

Column Name	Mask
ISOCountryCode_COUNTRY	CCC
IdentifierFlag_COUNTRY	CC
WORK_ADDRESS	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
WORK_ZIP	XXXXXXXXX

Figure 2-25 Create j02_INVCC_ISOCODE_NAB 3/8


INV_CC_ISOCODE3 - Investigate Stage
Stage Properties
Character Concatenate Investigate
Character Discrete Investigate
Word Investigate
Add To Selected Columns
Delete
Change Mask
Advanced Options

Character Concatenate Investigate

Available Data Columns:

Column Name	Length	Description
ACCOUNT_ID	10	<none>
ADDRESS	50	<none>
BIRTH_DT	10	<none>
CORRECTIVE_LENSES_	1	<none>
DRIVER_ID	10	<none>
GENDER	1	<none>
HAIR_COLOR	10	<none>
HEIGHT	10	<none>

Scheduled Process

Column Name	Mask
ISOCountryCode_COUNTRY	CCC
IdentifierFlag_COUNTRY	CC
CITY	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
STATE	XX
ZIP	XXXXXXXXX

Figure 2-26 Create j02_INVCC_ISOCODE_NAB 4/8

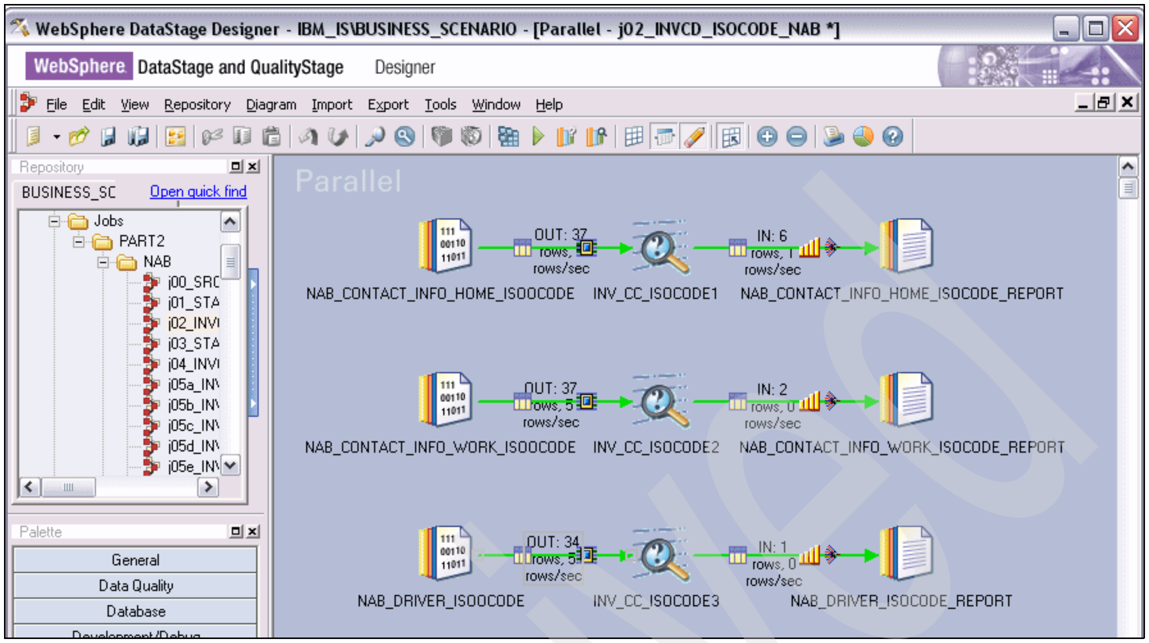
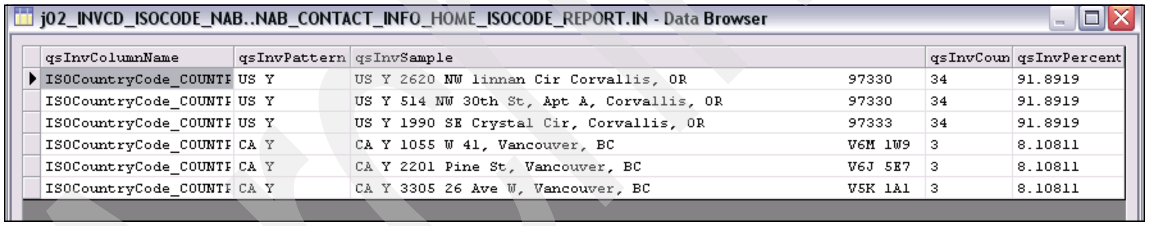
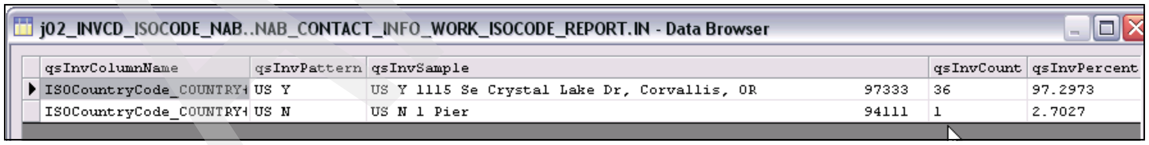


Figure 2-27 Create j02_INVCC_ISOCODE_NAB 5/8


j02_INVCD_ISOCODE_NAB..NAB_CONTACT_INFO_HOME_ISOCODE_REPORT.IN - Data Browser

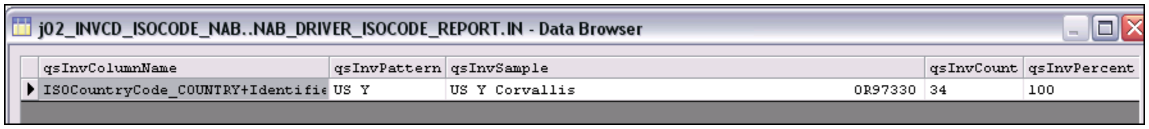
qsInvColumnName	qsInvPattern	qsInvSample	qsInvCoun	qsInvPercent
ISOCountryCode_COUNTI	US Y	US Y 2620 NW linnan Cir Corvallis, OR 97330	34	91.8919
ISOCountryCode_COUNTI	US Y	US Y 514 NW 30th St, Apt A, Corvallis, OR 97330	34	91.8919
ISOCountryCode_COUNTI	US Y	US Y 1990 SE Crystal Cir, Corvallis, OR 97333	34	91.8919
ISOCountryCode_COUNTI	CA Y	CA Y 1055 W 41, Vancouver, BC V6M 1W9	3	8.10811
ISOCountryCode_COUNTI	CA Y	CA Y 2201 Pine St, Vancouver, BC V6J 5E7	3	8.10811
ISOCountryCode_COUNTI	CA Y	CA Y 3305 26 Ave W, Vancouver, BC V5K 1A1	3	8.10811

Figure 2-28 Create j02_INVCC_ISOCODE_NAB 6/8


j02_INVCD_ISOCODE_NAB..NAB_CONTACT_INFO_WORK_ISOCODE_REPORT.IN - Data Browser

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
ISOCountryCode_COUNTRY	US Y	US Y 1115 Se Crystal Lake Dr, Corvallis, OR 97333	36	97.2973
ISOCountryCode_COUNTRY	US N	US N 1 Pier 94111	1	2.7027

Figure 2-29 Create j02_INVCC_ISOCODE_NAB 7/8


j02_INVCD_ISOCODE_NAB..NAB_DRIVER_ISOCODE_REPORT.IN - Data Browser

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
ISOCountryCode_COUNTRY+Identifie	US Y	US Y Corvallis 0R97330	34	100

Figure 2-30 Create j02_INVCD_ISOCODE_NAB 8/8

j03_STAN_XXPREP_NAB

We next split the records in the CONTACT_INFO data set into two files—one containing Canadian home addresses and the other containing U.S. home addresses. We need to split the records because different rule sets must be used to process addresses of different countries. We then use the Standardize stage with the following domain-preprocessor rule sets to move name and address data into Name, Address, and Area domains:

- CAPREP for the Canadian home address records in the CONTACT_INFO data set.
- USPREP for the name (CUSTOMER and DRIVER data) and address (home and work addresses in CONTACT_INFO and home addresses in DRIVER) records.

Figure 2-31 on page 532 shows the various stages that are used in this job, including a Transformer stage to split the home addresses in the CONTACT_INFO file into U.S. and CA address records; Standardize stages for the (work addresses of) CONTACT_INFO, (name and address information of) DRIVER, and (name information of) CUSTOMER files; and one output Data Set stage for each Standardize stage. We modified the names of the stages as shown.

Because the configuration of this job is very similar to that described in "J03_STAN_USPREP" on page 196, we do not repeat it here. However, some of the configurations of interest are as follows:

- Figure 2-32 on page 533 shows the FILTER_ISOCODE - Transformer Stage window which has the constraints TRIM(OUT.ISOCountryCode_COUNTRY) ='US' and TRIM(OUT.ISOCountryCode_COUNTRY) ='CA' to split the records based on the country code.
- The Standardize Rule Process windows of Figure 2-33 on page 534 through Figure 2-36 on page 535 show the USPREP and CAPREP rule sets and the literals and columns selected for analysis as follows:
  - Figure 2-33 on page 534 shows the Standardize Rule Process window with the configured USPREP rule set, and the literal ZQMIXAZQ followed by the HOME_ADDRESS column, followed by the literal ZQAREAZQ followed by the HOME_ZIP column of the CONTACT_INFO file (containing only U.S. home addresses) in the Selected Columns list.

- Figure 2-34 on page 534 shows the Standardize Rule Process window with the configured CAPREP rule set, and the literal ZQMIXAZQ followed by the HOME_ADDRESS column, followed by the literal ZQAREAZQ followed by the HOME_ZIP column of the CONTACT_INFO file (containing only CA home addresses) in the Selected Columns list.
- Figure 2-35 on page 535 shows the Standardize Rule Process window with the configured USPREP rule set, and the literal ZQMIXAZQ followed by the WORK_ADDRESS column, followed by the literal ZQAREAZQ followed by the WORK_ZIP column of the CONTACT_INFO file (containing only U.S. work addresses) in the Selected Columns list.
- Figure 2-36 on page 535 shows the Standardize Rule Process window with the configured USPREP rule set, and the literal ZQNAMEZQ followed by the NAME column, followed the literal ZQMIXAZQ followed by the ADDRESS column, followed by the literal ZQAREAZQ followed by the CITY, STATE, and ZIP columns of the DRIVER file (containing only U.S. work addresses) in the Selected Columns list.
- Figure 2-37 on page 536 shows the Standardize Rule Process window with the configured USPREP rule set, and the literal ZQNAMEZQ followed by the TITLE, FIRST_NAME, and LAST_NAME columns of the CUSTOMER file (containing only U.S. work addresses) in the Selected Columns list.

- After saving, compiling, and running this job, view the results as shown in Figure 2-38 on page 537.
- Figure 2-39 on page 538 through Figure 2-43 on page 541 show the contents of the standardized output for each of the five data sets.

  The partial reports show the columns NameDomain_USPREP (contains prefix, first name, last name, suffix tokens), AddressDomain_USPREP (contains apartment, street name and street type tokens), and AreaDomain_USPREP (contains city, state, ZIP code tokens) that were parsed from the input columns.

  The AddressDomain_USPREP column shows city names in it (as highlighted), which indicates that the parsing of the input columns into the correct domain buckets was not totally successful. These problems need to be resolved after reviewing the input patterns that were generated by the Standardize stage.

Now proceed to “j04_INVCC_XXPREP_INPUT_PATTERN” on page 541.

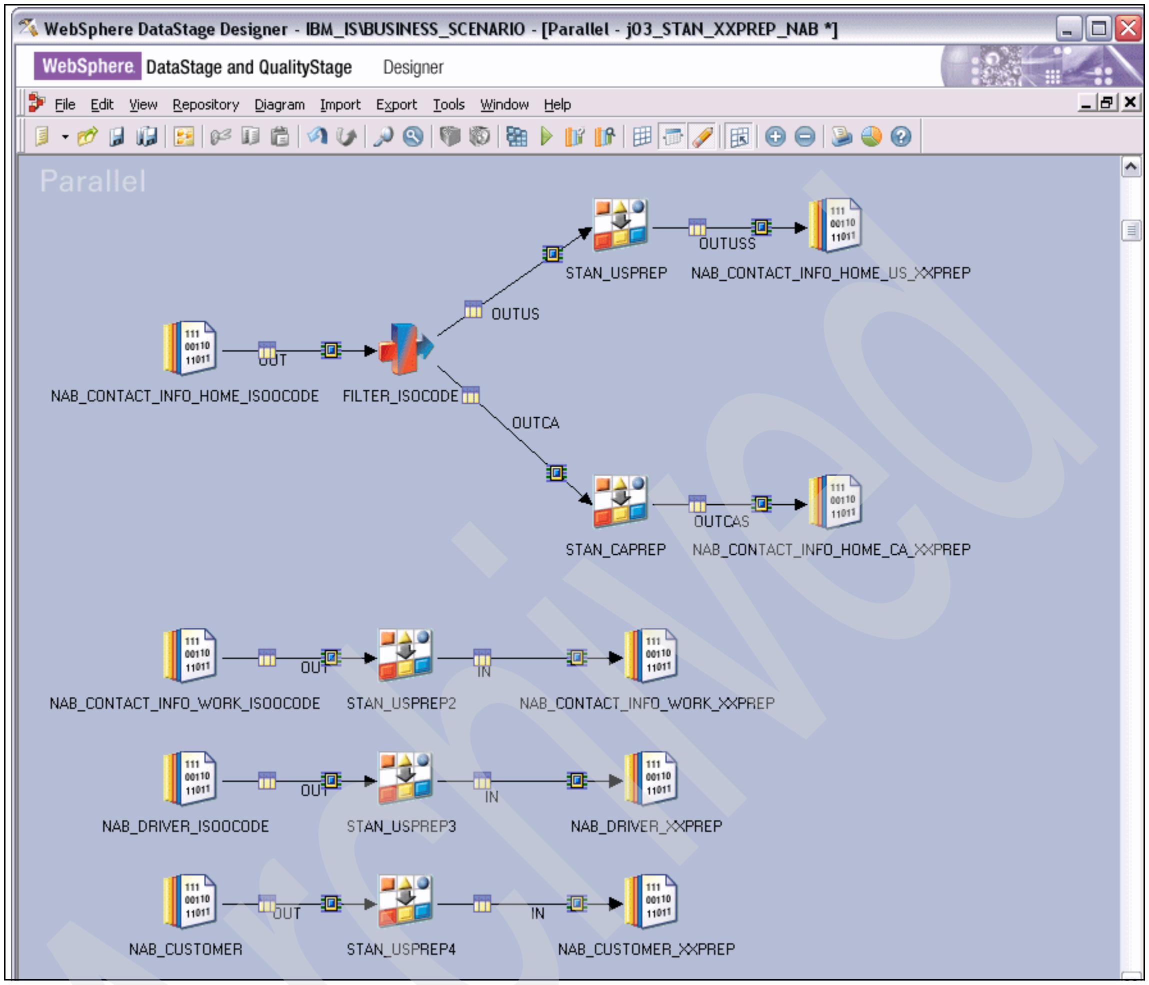
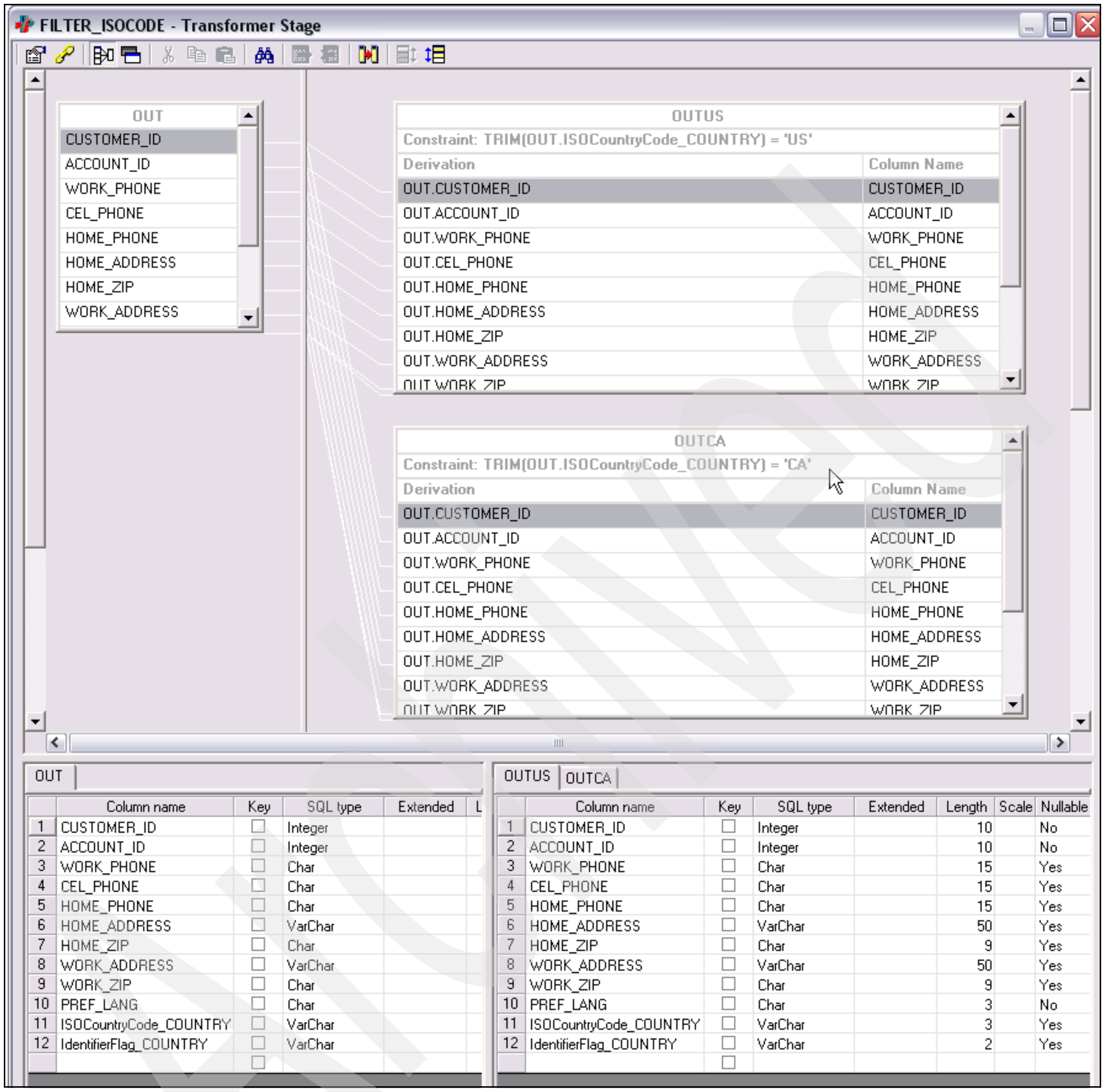

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j03_STAN_XXPREP_NAB *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
STAN_USPREP
OUTUSS
NAB_CONTACT_INFO_HOME_US_XXPREP
OUTUS
NAB_CONTACT_INFO_HOME_ISOCODE
OUT
FILTER_ISOCODE
OUTCA
STAN_CAPREP
OUTCAS
NAB_CONTACT_INFO_HOME_CA_XXPREP
NAB_CONTACT_INFO_WORK_ISOCODE
OUT
STAN_USPREP2
IN
NAB_CONTACT_INFO_WORK_XXPREP
NAB_DRIVER_ISOCODE
OUT
STAN_USPREP3
IN
NAB_DRIVER_XXPREP
NAB_CUSTOMER
OUT
STAN_USPREP4
IN
NAB_CUSTOMER_XXPREP

Figure 2-31 Create j03_STAN_XXPREP_NAB 1/13


FILTER_ISOCODE - Transformer Stage

OUT

CUSTOMER_ID
ACCOUNT_ID
WORK_PHONE
CEL_PHONE
HOME_PHONE
HOME_ADDRESS
HOME_ZIP
WORK_ADDRESS

OUTUS

Constraint: TRIM(OUT.ISOCountryCode_COUNTRY) = 'US'

Derivation	Column Name
OUT.CUSTOMER_ID	CUSTOMER_ID
OUT.ACCOUNT_ID	ACCOUNT_ID
OUT.WORK_PHONE	WORK_PHONE
OUT.CEL_PHONE	CEL_PHONE
OUT.HOME_PHONE	HOME_PHONE
OUT.HOME_ADDRESS	HOME_ADDRESS
OUT.HOME_ZIP	HOME_ZIP
OUT.WORK_ADDRESS	WORK_ADDRESS
OUT.WORK_ZIP	WORK_ZIP

OUTCA

Constraint: TRIM(OUT.ISOCountryCode_COUNTRY) = 'CA'

Derivation	Column Name
OUT.CUSTOMER_ID	CUSTOMER_ID
OUT.ACCOUNT_ID	ACCOUNT_ID
OUT.WORK_PHONE	WORK_PHONE
OUT.CEL_PHONE	CEL_PHONE
OUT.HOME_PHONE	HOME_PHONE
OUT.HOME_ADDRESS	HOME_ADDRESS
OUT.HOME_ZIP	HOME_ZIP
OUT.WORK_ADDRESS	WORK_ADDRESS
OUT.WORK_ZIP	WORK_ZIP

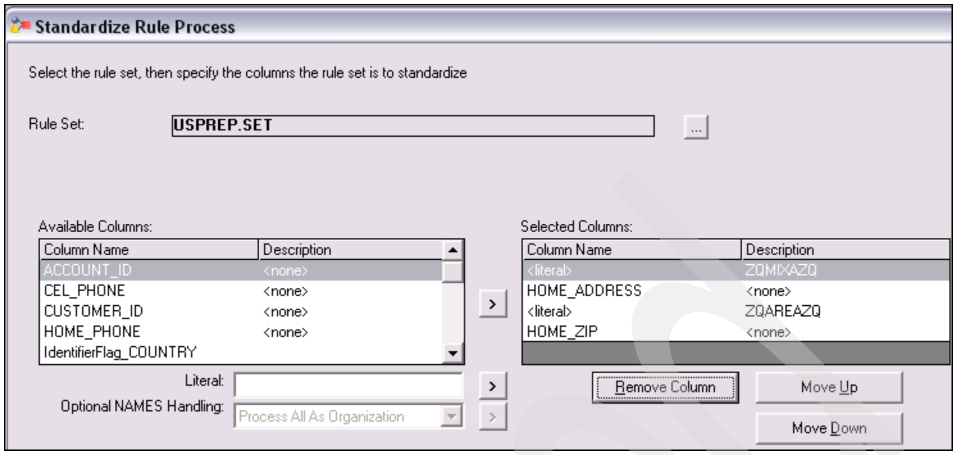
OUT

	Column name	Key	SQL type	Extended	L
1	CUSTOMER_ID	☐	Integer		
2	ACCOUNT_ID	☐	Integer		
3	WORK_PHONE	☐	Char		
4	CEL_PHONE	☐	Char		
5	HOME_PHONE	☐	Char		
6	HOME_ADDRESS	☐	VarChar		
7	HOME_ZIP	☐	Char		
8	WORK_ADDRESS	☐	VarChar		
9	WORK_ZIP	☐	Char		
10	PREF_LANG	☐	Char		
11	ISOCountryCode_COUNTRY	☐	VarChar		
12	IdentifierFlag_COUNTRY	☐	VarChar		
		☐			

OUTUS | OUTCA

	Column name	Key	SQL type	Extended	Length	Scale	Nullable
1	CUSTOMER_ID	☐	Integer		10		No
2	ACCOUNT_ID	☐	Integer		10		No
3	WORK_PHONE	☐	Char		15		Yes
4	CEL_PHONE	☐	Char		15		Yes
5	HOME_PHONE	☐	Char		15		Yes
6	HOME_ADDRESS	☐	VarChar		50		Yes
7	HOME_ZIP	☐	Char		9		Yes
8	WORK_ADDRESS	☐	VarChar		50		Yes
9	WORK_ZIP	☐	Char		9		Yes
10	PREF_LANG	☐	Char		3		No
11	ISOCountryCode_COUNTRY	☐	VarChar		3		Yes
12	IdentifierFlag_COUNTRY	☐	VarChar		2		Yes
		☐					

Figure 2-32 Create j03_STAN_XXPREP_NAB 2/13


Standardize Rule Process

Select the rule set, then specify the columns the rule set is to standardize

Rule Set: USPREP.SET

Available Columns:

Column Name	Description
ACCOUNT_ID	<none>
CEL_PHONE	<none>
CUSTOMER_ID	<none>
HOME_PHONE	<none>
IdentifierFlag_COUNTRY	

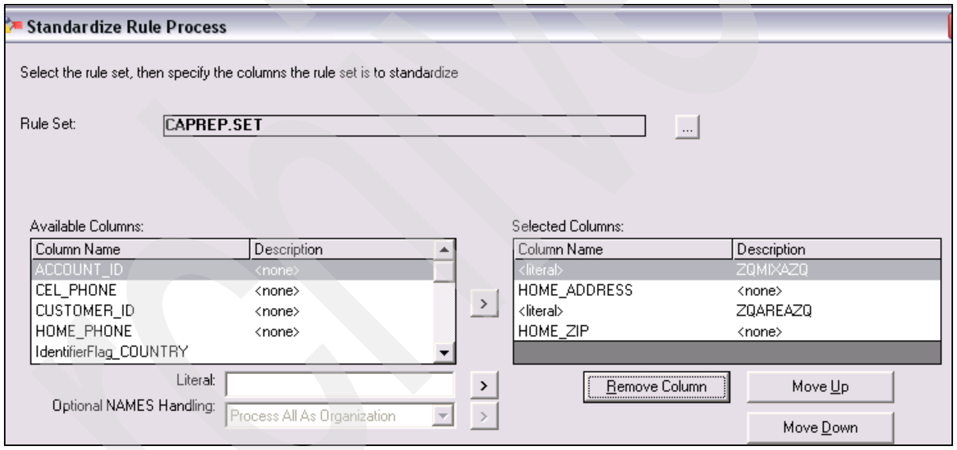
Selected Columns:

Column Name	Description
<literal>	ZQMIXAZQ
HOME_ADDRESS	<none>
<literal>	ZQAREAZQ
HOME_ZIP	<none>

Literal:

Optional NAMES Handling: Process All As Organization

Figure 2-33 Create j03_STAN_XXPREP_NAB 3/13


Standardize Rule Process

Select the rule set, then specify the columns the rule set is to standardize

Rule Set: CAPREP.SET

Available Columns:

Column Name	Description
ACCOUNT_ID	<none>
CEL_PHONE	<none>
CUSTOMER_ID	<none>
HOME_PHONE	<none>
IdentifierFlag_COUNTRY	

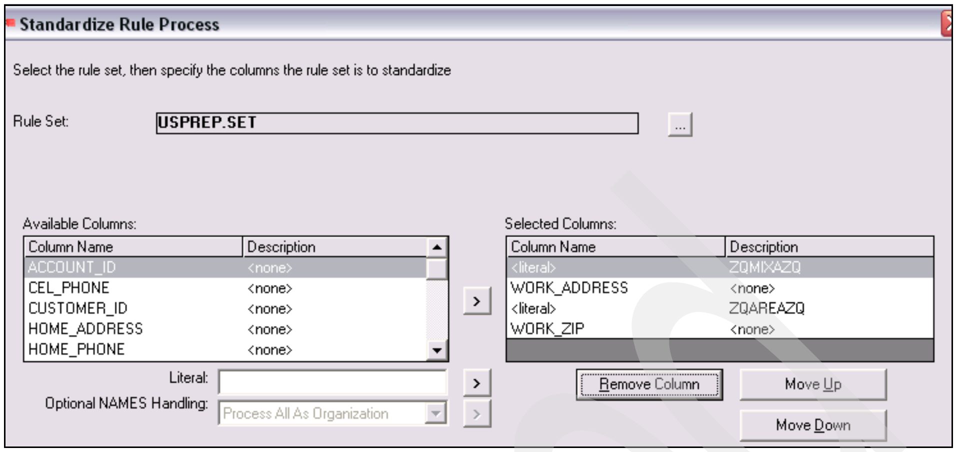
Selected Columns:

Column Name	Description
<literal>	ZQMIXAZQ
HOME_ADDRESS	<none>
<literal>	ZQAREAZQ
HOME_ZIP	<none>

Literal:

Optional NAMES Handling: Process All As Organization

Figure 2-34 Create j03_STAN_XXPREP_NAB 4/13


Standardize Rule Process

Select the rule set, then specify the columns the rule set is to standardize

Rule Set: USPREP.SET

Available Columns:

Column Name	Description
ACCOUNT_ID	<none>
CEL_PHONE	<none>
CUSTOMER_ID	<none>
HOME_ADDRESS	<none>
HOME_PHONE	<none>

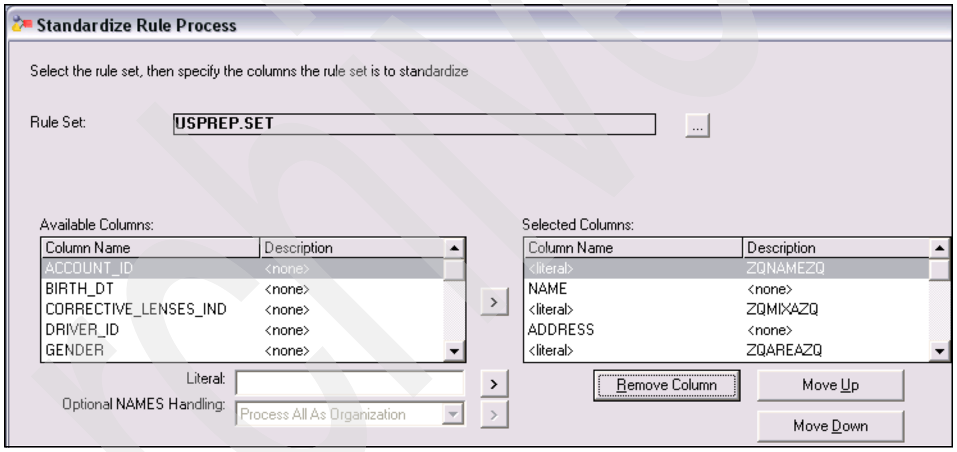
Selected Columns:

Column Name	Description
<literal>	ZQMIXAZQ
WORK_ADDRESS	<none>
<literal>	ZQAREAZQ
WORK_ZIP	<none>

Literal:

Optional NAMES Handling: Process All As Organization

Figure 2-35 Create j03_STAN_XXPREP_NAB 5/13


Standardize Rule Process

Select the rule set, then specify the columns the rule set is to standardize

Rule Set: USPREP.SET

Available Columns:

Column Name	Description
ACCOUNT_ID	<none>
BIRTH_DT	<none>
CORRECTIVE_LENSES_IND	<none>
DRIVER_ID	<none>
GENDER	<none>

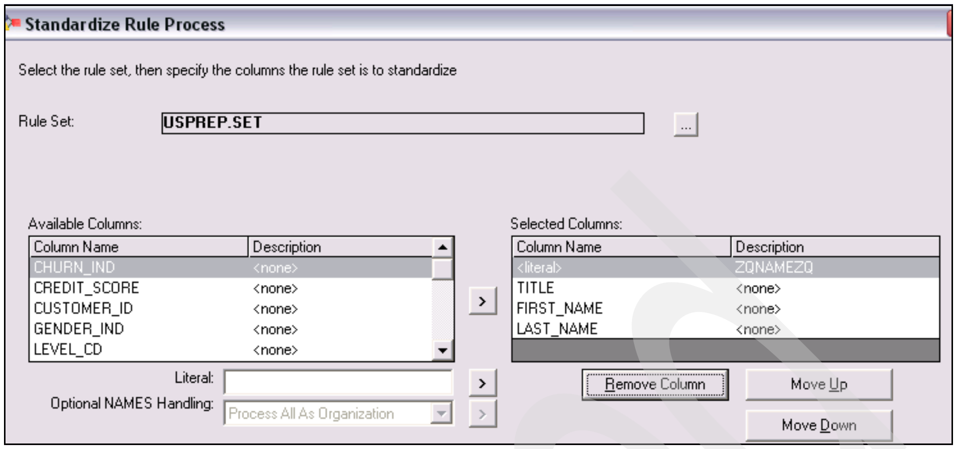
Selected Columns:

Column Name	Description
<literal>	ZQNAMEZQ
NAME	<none>
<literal>	ZQMIXAZQ
ADDRESS	<none>
<literal>	ZQAREAZQ

Literal:

Optional NAMES Handling: Process All As Organization

Figure 2-36 Create j03_STAN_XXPREP_NAB 6/13


Standardize Rule Process

Select the rule set, then specify the columns the rule set is to standardize

Rule Set: USPREP.SET

Available Columns:

Column Name	Description
CHURN_IND	<none>
CREDIT_SCORE	<none>
CUSTOMER_ID	<none>
GENDER_IND	<none>
LEVEL_CD	<none>

Selected Columns:

Column Name	Description
<literal>	ZQNAMEZQ
TITLE	<none>
FIRST_NAME	<none>
LAST_NAME	<none>

Literal:

Optional NAMES Handling: Process All As Organization

Figure 2-37 Create j03_STAN_XXPREP_NAB 7/13

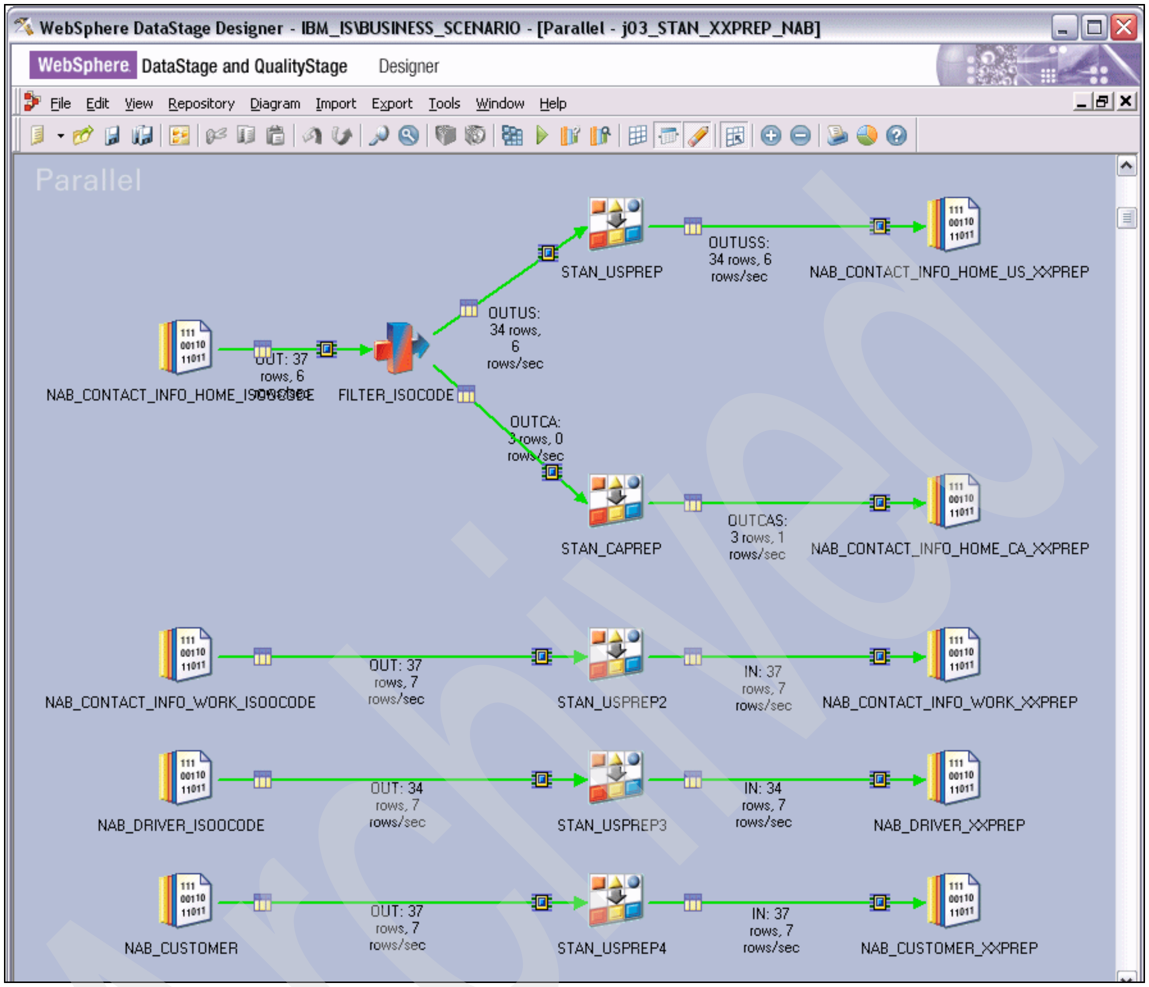
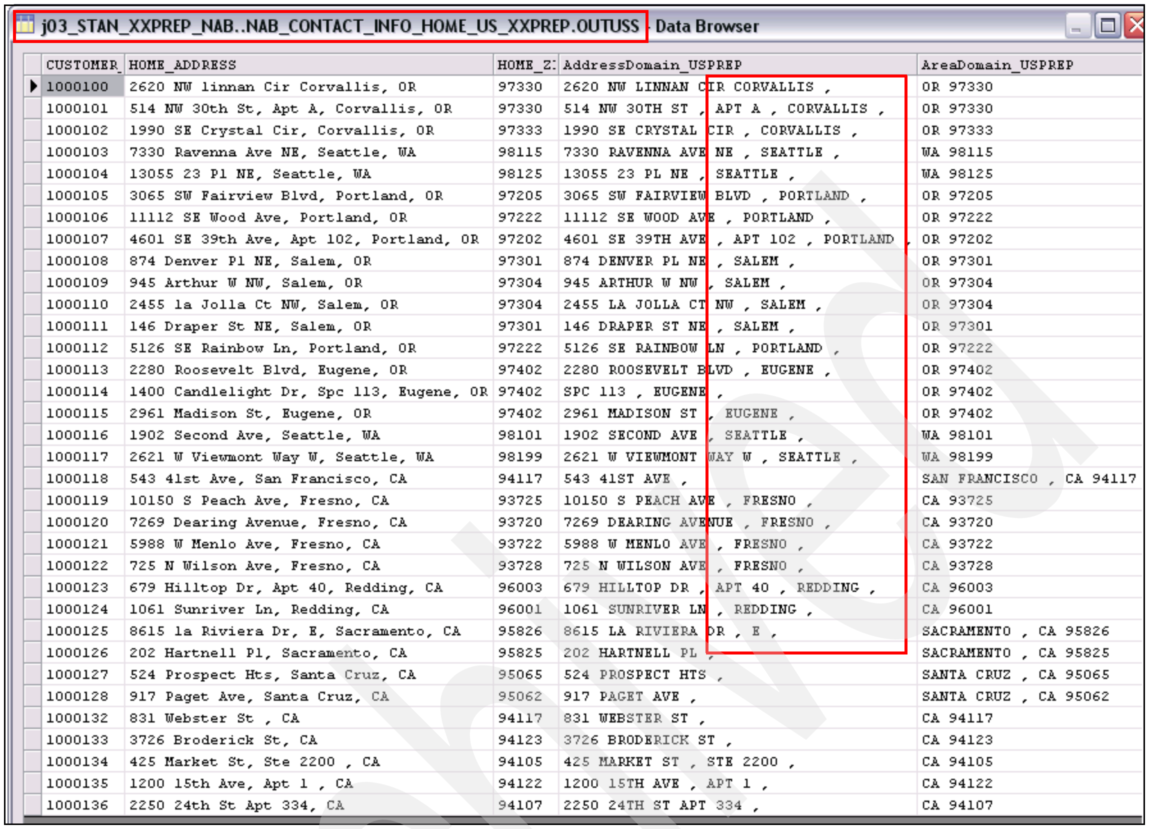

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j03_STAN_XXPREP_NAB]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
STAN_USPREP
OUTUSS: 34 rows, 6 rows/sec
NAB_CONTACT_INFO_HOME_US_XXPREP
OUTUS: 34 rows, 6 rows/sec
NAB_CONTACT_INFO_HOME_ISOCODE
OUT: 37 rows, 6 rows/sec
FILTER_ISOCODE
OUTCA: 3 rows, 0 rows/sec
STAN_CAPREP
OUTCAS: 3 rows, 1 rows/sec
NAB_CONTACT_INFO_HOME_CA_XXPREP
NAB_CONTACT_INFO_WORK_ISOOCODE
OUT: 37 rows, 7 rows/sec
STAN_USPREP2
IN: 37 rows, 7 rows/sec
NAB_CONTACT_INFO_WORK_XXPREP
NAB_DRIVER_ISOOCODE
OUT: 34 rows, 7 rows/sec
STAN_USPREP3
IN: 34 rows, 7 rows/sec
NAB_DRIVER_XXPREP
NAB_CUSTOMER
OUT: 37 rows, 7 rows/sec
STAN_USPREP4
IN: 37 rows, 7 rows/sec
NAB_CUSTOMER_XXPREP

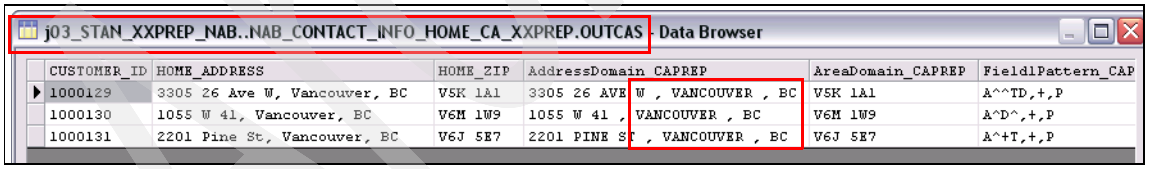
Figure 2-38 Create j03_STAN_XXPREP_NAB 8/13



j03_STAN_XXPREP_NAB..NAB_CONTACT_INFO_HOME_US_XXPREP.OUTUSS - Data Browser

CUSTOMER_	HOME_ADDRESS	HOME_Z:	AddressDomain_USPREP	AreaDomain_USPREP
1000100	2620 NW linnan Cir Corvallis, OR	97330	2620 NW LINNAN CIR CORVALLIS ,	OR 97330
1000101	514 NW 30th St, Apt A, Corvallis, OR	97330	514 NW 30TH ST , APT A , CORVALLIS ,	OR 97330
1000102	1990 SE Crystal Cir, Corvallis, OR	97333	1990 SE CRYSTAL CIR , CORVALLIS ,	OR 97333
1000103	7330 Ravenna Ave NE, Seattle, WA	98115	7330 RAVENNA AVE NE , SEATTLE ,	WA 98115
1000104	13055 23 Pl NE, Seattle, WA	98125	13055 23 PL NE , SEATTLE ,	WA 98125
1000105	3065 SW Fairview Blvd, Portland, OR	97205	3065 SW FAIRVIEW BLVD , PORTLAND ,	OR 97205
1000106	11112 SE Wood Ave, Portland, OR	97222	11112 SE WOOD AVE , PORTLAND ,	OR 97222
1000107	4601 SE 39th Ave, Apt 102, Portland, OR	97202	4601 SE 39TH AVE , APT 102 , PORTLAND ,	OR 97202
1000108	874 Denver Pl NE, Salem, OR	97301	874 DENVER PL NE , SALEM ,	OR 97301
1000109	945 Arthur W NW, Salem, OR	97304	945 ARTHUR W NW , SALEM ,	OR 97304
1000110	2455 la Jolla Ct NW, Salem, OR	97304	2455 LA JOLLA CT NW , SALEM ,	OR 97304
1000111	146 Draper St NE, Salem, OR	97301	146 DRAPER ST NE , SALEM ,	OR 97301
1000112	5126 SE Rainbow Ln, Portland, OR	97222	5126 SE RAINBOW LN , PORTLAND ,	OR 97222
1000113	2280 Roosevelt Blvd, Eugene, OR	97402	2280 ROOSEVELT BLVD , EUGENE ,	OR 97402
1000114	1400 Candlelight Dr, Spc 113, Eugene, OR	97402	SPC 113 , EUGENE ,	OR 97402
1000115	2961 Madison St, Eugene, OR	97402	2961 MADISON ST , EUGENE ,	OR 97402
1000116	1902 Second Ave, Seattle, WA	98101	1902 SECOND AVE , SEATTLE ,	WA 98101
1000117	2621 W Viewmont Way W, Seattle, WA	98199	2621 W VIEWMONT WAY W , SEATTLE ,	WA 98199
1000118	543 41st Ave, San Francisco, CA	94117	543 41ST AVE ,	SAN FRANCISCO , CA 94117
1000119	10150 S Peach Ave, Fresno, CA	93725	10150 S PEACH AVE , FRESNO ,	CA 93725
1000120	7269 Dearing Avenue, Fresno, CA	93720	7269 DEARING AVENUE , FRESNO ,	CA 93720
1000121	5988 W Menlo Ave, Fresno, CA	93722	5988 W MENLO AVE , FRESNO ,	CA 93722
1000122	725 N Wilson Ave, Fresno, CA	93728	725 N WILSON AVE , FRESNO ,	CA 93728
1000123	679 Hilltop Dr, Apt 40, Redding, CA	96003	679 HILLTOP DR , APT 40 , REDDING ,	CA 96003
1000124	1061 Sunriver Ln, Redding, CA	96001	1061 SUNRIVER LN , REDDING ,	CA 96001
1000125	8615 la Riviera Dr, E, Sacramento, CA	95826	8615 LA RIVIERA DR , E ,	SACRAMENTO , CA 95826
1000126	202 Hartnell Pl, Sacramento, CA	95825	202 HARTNELL PL ,	SACRAMENTO , CA 95825
1000127	524 Prospect Hts, Santa Cruz, CA	95065	524 PROSPECT HTS ,	SANTA CRUZ , CA 95065
1000128	917 Paget Ave, Santa Cruz, CA	95062	917 PAGET AVE ,	SANTA CRUZ , CA 95062
1000132	831 Webster St , CA	94117	831 WEBSTER ST ,	CA 94117
1000133	3726 Broderick St, CA	94123	3726 BRODERICK ST ,	CA 94123
1000134	425 Market St, Ste 2200 , CA	94105	425 MARKET ST , STE 2200 ,	CA 94105
1000135	1200 15th Ave, Apt 1 , CA	94122	1200 15TH AVE , APT 1 ,	CA 94122
1000136	2250 24th St Apt 334, CA	94107	2250 24TH ST APT 334 ,	CA 94107

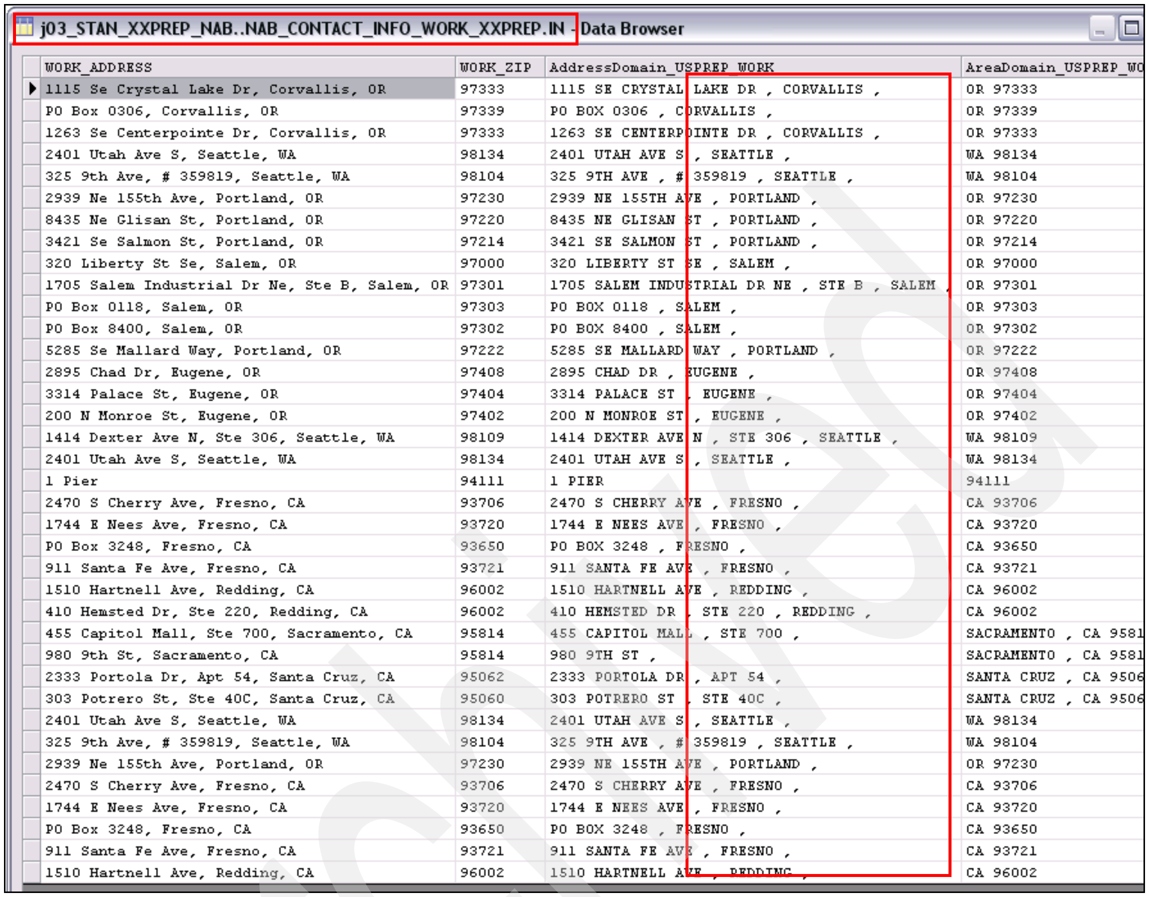
Figure 2-39 Create j03_STAN_XXPREP_NAB 9/13



j03_STAN_XXPREP_NAB..NAB_CONTACT_INFO_HOME_CA_XXPREP.OUTCAS - Data Browser

CUSTOMER_ID	HOME_ADDRESS	HOME_ZIP	AddressDomain_CAPREP	AreaDomain_CAPREP	Field1Pattern_CAP
1000129	3305 26 Ave W, Vancouver, BC	V5K 1A1	3305 26 AVE W , VANCOUVER , BC	V5K 1A1	A^^TD,+,P
1000130	1055 W 41, Vancouver, BC	V6M 1W9	1055 W 41 , VANCOUVER , BC	V6M 1W9	A^D^,+,P
1000131	2201 Pine St, Vancouver, BC	V6J 5E7	2201 PINE ST , VANCOUVER , BC	V6J 5E7	A^+T,+,P

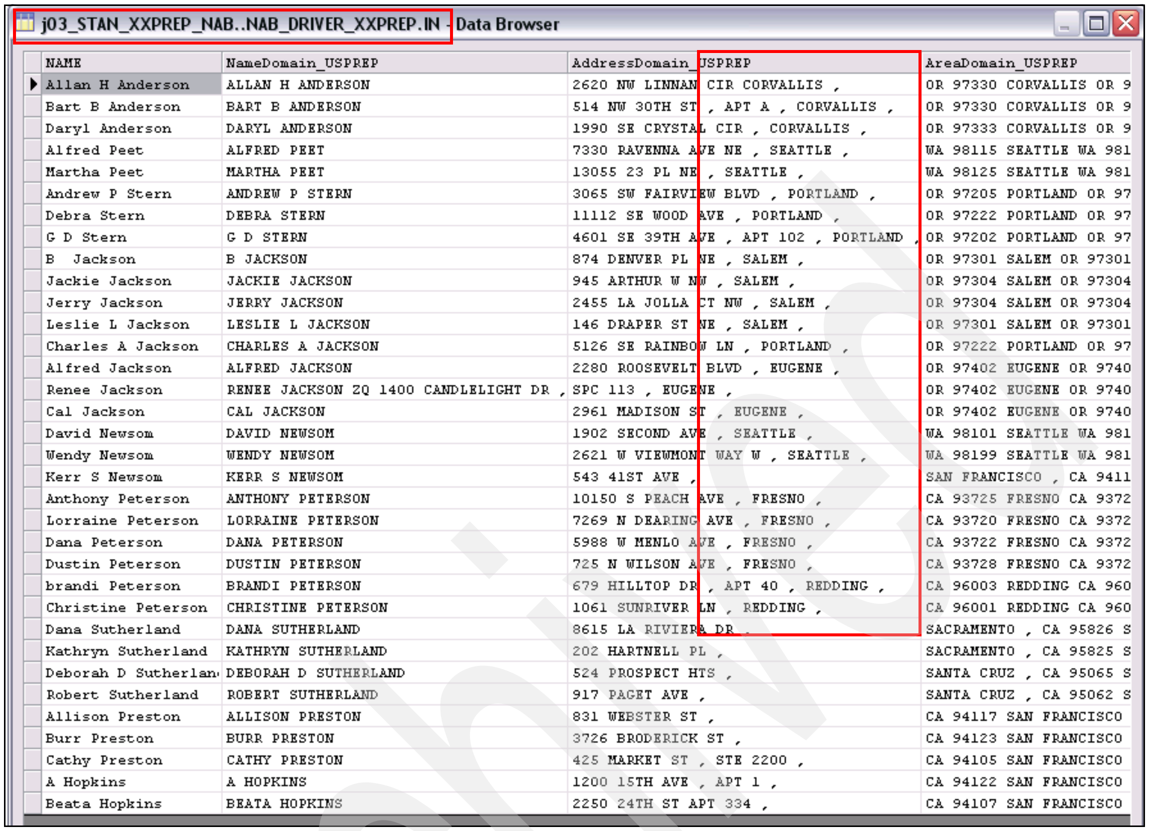
Figure 2-40 Create j03_STAN_XXPREP_NAB 10/13



j03_STAN_XXPREP_NAB..NAB_CONTACT_INFO_WORK_XXPREP.IN - Data Browser

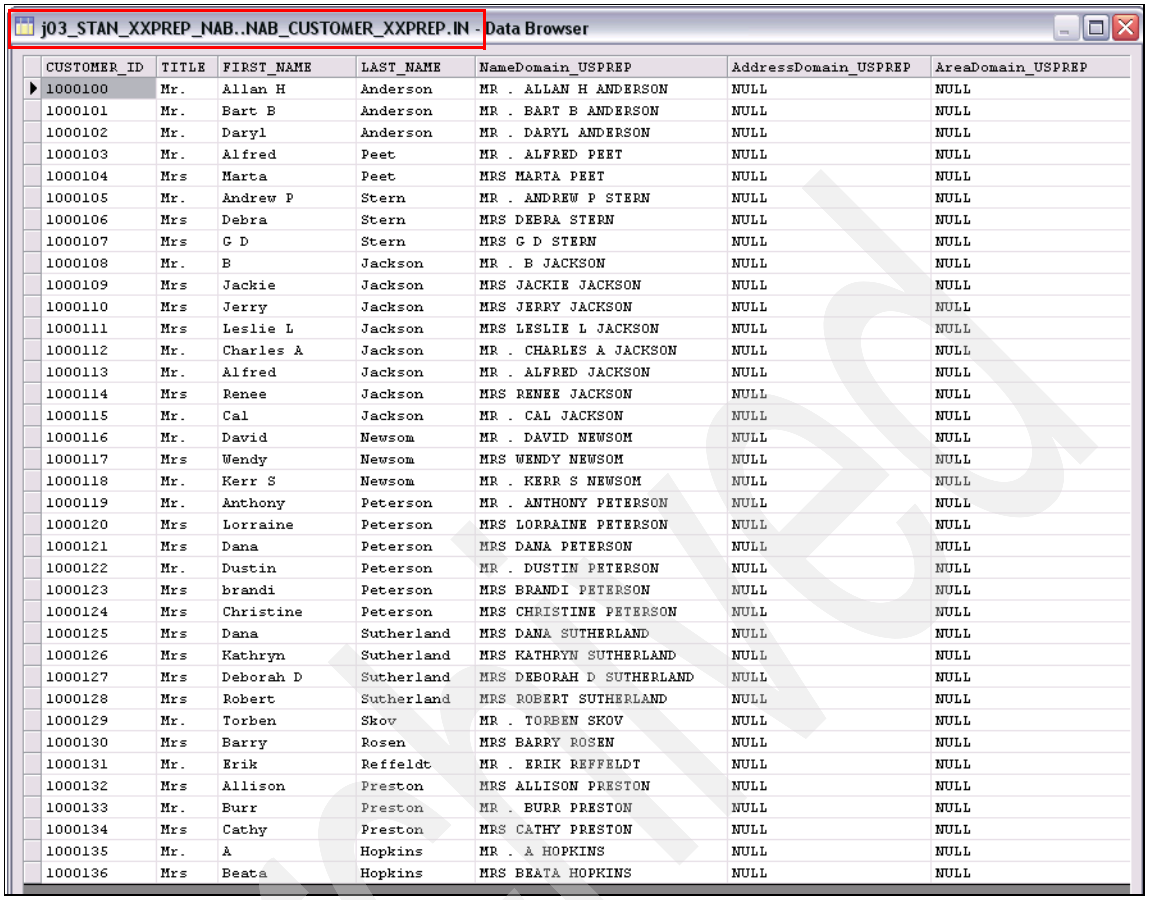
WORK_ADDRESS	WORK_ZIP	AddressDomain_USPREP_WORK	AreaDomain_USPREP_WO
1115 Se Crystal Lake Dr, Corvallis, OR	97333	1115 SE CRYSTAL LAKE DR , CORVALLIS ,	OR 97333
PO Box 0306, Corvallis, OR	97339	PO BOX 0306 , CORVALLIS ,	OR 97339
1263 Se Centerpointe Dr, Corvallis, OR	97333	1263 SE CENTERPOINTE DR , CORVALLIS ,	OR 97333
2401 Utah Ave S, Seattle, WA	98134	2401 UTAH AVE S , SEATTLE ,	WA 98134
325 9th Ave, # 359819, Seattle, WA	98104	325 9TH AVE , # 359819 , SEATTLE ,	WA 98104
2939 Ne 155th Ave, Portland, OR	97230	2939 NE 155TH AVE , PORTLAND ,	OR 97230
8435 Ne Glisan St, Portland, OR	97220	8435 NE GLISAN ST , PORTLAND ,	OR 97220
3421 Se Salmon St, Portland, OR	97214	3421 SE SALMON ST , PORTLAND ,	OR 97214
320 Liberty St Se, Salem, OR	97000	320 LIBERTY ST SE , SALEM ,	OR 97000
1705 Salem Industrial Dr Ne, Ste B, Salem, OR	97301	1705 SALEM INDUSTRIAL DR NE , STE B , SALEM ,	OR 97301
PO Box 0118, Salem, OR	97303	PO BOX 0118 , SALEM ,	OR 97303
PO Box 8400, Salem, OR	97302	PO BOX 8400 , SALEM ,	OR 97302
5285 Se Mallard Way, Portland, OR	97222	5285 SE MALLARD WAY , PORTLAND ,	OR 97222
2895 Chad Dr, Eugene, OR	97408	2895 CHAD DR , EUGENE ,	OR 97408
3314 Palace St, Eugene, OR	97404	3314 PALACE ST , EUGENE ,	OR 97404
200 N Monroe St, Eugene, OR	97402	200 N MONROE ST , EUGENE ,	OR 97402
1414 Dexter Ave N, Ste 306, Seattle, WA	98109	1414 DEXTER AVE N , STE 306 , SEATTLE ,	WA 98109
2401 Utah Ave S, Seattle, WA	98134	2401 UTAH AVE S , SEATTLE ,	WA 98134
1 Pier	94111	1 PIER	94111
2470 S Cherry Ave, Fresno, CA	93706	2470 S CHERRY AVE , FRESNO ,	CA 93706
1744 E Nees Ave, Fresno, CA	93720	1744 E NEES AVE , FRESNO ,	CA 93720
PO Box 3248, Fresno, CA	93650	PO BOX 3248 , FRESNO ,	CA 93650
911 Santa Fe Ave, Fresno, CA	93721	911 SANTA FE AVE , FRESNO ,	CA 93721
1510 Hartnell Ave, Redding, CA	96002	1510 HARTNELL AVE , REDDING ,	CA 96002
410 Hemsted Dr, Ste 220, Redding, CA	96002	410 HEMSTED DR , STE 220 , REDDING ,	CA 96002
455 Capitol Mall, Ste 700, Sacramento, CA	95814	455 CAPITOL MALL , STE 700 ,	SACRAMENTO , CA 9581
980 9th St, Sacramento, CA	95814	980 9TH ST ,	SACRAMENTO , CA 9581
2333 Portola Dr, Apt 54, Santa Cruz, CA	95062	2333 PORTOLA DR , APT 54 ,	SANTA CRUZ , CA 9506
303 Potrero St, Ste 40C, Santa Cruz, CA	95060	303 POTRERO ST , STE 40C ,	SANTA CRUZ , CA 9506
2401 Utah Ave S, Seattle, WA	98134	2401 UTAH AVE S , SEATTLE ,	WA 98134
325 9th Ave, # 359819, Seattle, WA	98104	325 9TH AVE , # 359819 , SEATTLE ,	WA 98104
2939 Ne 155th Ave, Portland, OR	97230	2939 NE 155TH AVE , PORTLAND ,	OR 97230
2470 S Cherry Ave, Fresno, CA	93706	2470 S CHERRY AVE , FRESNO ,	CA 93706
1744 E Nees Ave, Fresno, CA	93720	1744 E NEES AVE , FRESNO ,	CA 93720
PO Box 3248, Fresno, CA	93650	PO BOX 3248 , FRESNO ,	CA 93650
911 Santa Fe Ave, Fresno, CA	93721	911 SANTA FE AVE , FRESNO ,	CA 93721
1510 Hartnell Ave, Redding, CA	96002	1510 HARTNELL AVE , REDDING ,	CA 96002

Figure 2-41 Create j03_STAN_XXPREP_NAB 11/13


j03_STAN_XXPREP_NAB..NAB_DRIVER_XXPREP.IN - Data Browser

NAME	NameDomain_USPREP	AddressDomain_USPREP	AreaDomain_USPREP
Allan H Anderson	ALLAN H ANDERSON	2620 NW LINNAN CIR CORVALLIS ,	OR 97330 CORVALLIS OR 9
Bart B Anderson	BART B ANDERSON	514 NW 30TH ST , APT A , CORVALLIS ,	OR 97330 CORVALLIS OR 9
Daryl Anderson	DARYL ANDERSON	1990 SE CRYSTAL CIR , CORVALLIS ,	OR 97333 CORVALLIS OR 9
Alfred Peet	ALFRED PEET	7330 RAVENNA AVE NE , SEATTLE ,	WA 98115 SEATTLE WA 981
Martha Peet	MARTHA PEET	13055 23 PL NE , SEATTLE ,	WA 98125 SEATTLE WA 981
Andrew P Stern	ANDREW P STERN	3065 SW FAIRVIEW BLVD , PORTLAND ,	OR 97205 PORTLAND OR 97
Debra Stern	DEBRA STERN	11112 SE WOOD AVE , PORTLAND ,	OR 97222 PORTLAND OR 97
G D Stern	G D STERN	4601 SE 39TH AVE , APT 102 , PORTLAND ,	OR 97202 PORTLAND OR 97
B Jackson	B JACKSON	874 DENVER PL NE , SALEM ,	OR 97301 SALEM OR 97301
Jackie Jackson	JACKIE JACKSON	945 ARTHUR W NW , SALEM ,	OR 97304 SALEM OR 97304
Jerry Jackson	JERRY JACKSON	2455 LA JOLLA CT NW , SALEM ,	OR 97304 SALEM OR 97304
Leslie L Jackson	LESLIE L JACKSON	146 DRAPER ST NE , SALEM ,	OR 97301 SALEM OR 97301
Charles A Jackson	CHARLES A JACKSON	5126 SE RAINBOW LN , PORTLAND ,	OR 97222 PORTLAND OR 97
Alfred Jackson	ALFRED JACKSON	2280 ROOSEVELT BLVD , EUGENE ,	OR 97402 EUGENE OR 9740
Renee Jackson	RENEE JACKSON ZQ 1400 CANDLELIGHT DR ,	SPC 113 , EUGENE ,	OR 97402 EUGENE OR 9740
Cal Jackson	CAL JACKSON	2961 MADISON ST , EUGENE ,	OR 97402 EUGENE OR 9740
David Newsom	DAVID NEWSOM	1902 SECOND AVE , SEATTLE ,	WA 98101 SEATTLE WA 981
Wendy Newsom	WENDY NEWSOM	2621 W VIEWMONT WAY W , SEATTLE ,	WA 98199 SEATTLE WA 981
Kerr S Newsom	KERR S NEWSOM	543 41ST AVE ,	SAN FRANCISCO , CA 9411
Anthony Peterson	ANTHONY PETERSON	10150 S PEACH AVE , FRESNO ,	CA 93725 FRESNO CA 9372
Lorraine Peterson	LORRAINE PETERSON	7269 N DEARING AVE , FRESNO ,	CA 93720 FRESNO CA 9372
Dana Peterson	DANA PETERSON	5988 W MENLO AVE , FRESNO ,	CA 93722 FRESNO CA 9372
Dustin Peterson	DUSTIN PETERSON	725 N WILSON AVE , FRESNO ,	CA 93728 FRESNO CA 9372
brandi Peterson	BRANDI PETERSON	679 HILLTOP DR , APT 40 , REDDING ,	CA 96003 REDDING CA 960
Christine Peterson	CHRISTINE PETERSON	1061 SUNRIVER LN , REDDING ,	CA 96001 REDDING CA 960
Dana Sutherland	DANA SUTHERLAND	8615 LA RIVIERA DR ,	SACRAMENTO , CA 95826 S
Kathryn Sutherland	KATHRYN SUTHERLAND	202 HARTNELL PL ,	SACRAMENTO , CA 95825 S
Deborah D Sutherlan	DEBORAH D SUTHERLAND	524 PROSPECT HTS ,	SANTA CRUZ , CA 95065 S
Robert Sutherland	ROBERT SUTHERLAND	917 PAGET AVE ,	SANTA CRUZ , CA 95062 S
Allison Preston	ALLISON PRESTON	831 WEBSTER ST ,	CA 94117 SAN FRANCISCO
Burr Preston	BURR PRESTON	3726 BRODERICK ST ,	CA 94123 SAN FRANCISCO
Cathy Preston	CATHY PRESTON	425 MARKET ST , STE 2200 ,	CA 94105 SAN FRANCISCO
A Hopkins	A HOPKINS	1200 15TH AVE , APT 1 ,	CA 94122 SAN FRANCISCO
Beata Hopkins	BEATA HOPKINS	2250 24TH ST APT 334 ,	CA 94107 SAN FRANCISCO

Figure 2-42 Create j03_STAN_XXPREP_NAB 12/13



j03_STAN_XXPREP_NAB..NAB_CUSTOMER_XXPREP.IN - Data Browser

CUSTOMER_ID	TITLE	FIRST_NAME	LAST_NAME	NameDomain_USPREP	AddressDomain_USPREP	AreaDomain_USPREP
1000100	Mr.	Allan H	Anderson	MR . ALLAN H ANDERSON	NULL	NULL
1000101	Mr.	Bart B	Anderson	MR . BART B ANDERSON	NULL	NULL
1000102	Mr.	Daryl	Anderson	MR . DARYL ANDERSON	NULL	NULL
1000103	Mr.	Alfred	Peet	MR . ALFRED PEET	NULL	NULL
1000104	Mrs	Marta	Peet	MRS MARTA PEET	NULL	NULL
1000105	Mr.	Andrew P	Stern	MR . ANDREW P STERN	NULL	NULL
1000106	Mrs	Debra	Stern	MRS DEBRA STERN	NULL	NULL
1000107	Mrs	G D	Stern	MRS G D STERN	NULL	NULL
1000108	Mr.	B	Jackson	MR . B JACKSON	NULL	NULL
1000109	Mrs	Jackie	Jackson	MRS JACKIE JACKSON	NULL	NULL
1000110	Mrs	Jerry	Jackson	MRS JERRY JACKSON	NULL	NULL
1000111	Mrs	Leslie L	Jackson	MRS LESLIE L JACKSON	NULL	NULL
1000112	Mr.	Charles A	Jackson	MR . CHARLES A JACKSON	NULL	NULL
1000113	Mr.	Alfred	Jackson	MR . ALFRED JACKSON	NULL	NULL
1000114	Mrs	Renee	Jackson	MRS RENEE JACKSON	NULL	NULL
1000115	Mr.	Cal	Jackson	MR . CAL JACKSON	NULL	NULL
1000116	Mr.	David	Newsom	MR . DAVID NEWSOM	NULL	NULL
1000117	Mrs	Wendy	Newsom	MRS WENDY NEWSOM	NULL	NULL
1000118	Mr.	Kerr S	Newsom	MR . KERR S NEWSOM	NULL	NULL
1000119	Mr.	Anthony	Peterson	MR . ANTHONY PETERSON	NULL	NULL
1000120	Mrs	Lorraine	Peterson	MRS LORRAINE PETERSON	NULL	NULL
1000121	Mrs	Dana	Peterson	MRS DANA PETERSON	NULL	NULL
1000122	Mr.	Dustin	Peterson	MR . DUSTIN PETERSON	NULL	NULL
1000123	Mrs	brandi	Peterson	MRS BRANDI PETERSON	NULL	NULL
1000124	Mrs	Christine	Peterson	MRS CHRISTINE PETERSON	NULL	NULL
1000125	Mrs	Dana	Sutherland	MRS DANA SUTHERLAND	NULL	NULL
1000126	Mrs	Kathryn	Sutherland	MRS KATHRYN SUTHERLAND	NULL	NULL
1000127	Mrs	Deborah D	Sutherland	MRS DEBORAH D SUTHERLAND	NULL	NULL
1000128	Mrs	Robert	Sutherland	MRS ROBERT SUTHERLAND	NULL	NULL
1000129	Mr.	Torben	Skov	MR . TORBEN SKOV	NULL	NULL
1000130	Mrs	Barry	Rosen	MRS BARRY ROSEN	NULL	NULL
1000131	Mr.	Erik	Reffeldt	MR . ERIK REFFELDT	NULL	NULL
1000132	Mrs	Allison	Preston	MRS ALLISON PRESTON	NULL	NULL
1000133	Mr.	Burr	Preston	MR . BURR PRESTON	NULL	NULL
1000134	Mrs	Cathy	Preston	MRS CATHY PRESTON	NULL	NULL
1000135	Mr.	A	Hopkins	MR . A HOPKINS	NULL	NULL
1000136	Mrs	Beata	Hopkins	MRS BEATA HOPKINS	NULL	NULL

Figure 2-43 Create j03_STAN_XXPREP_NAB 13/13

j04_INVCC_XXPREP_INPUT_PATTERN

We use the Investigate stage using the character concatenate option on the input patterns that were generated by the Standardize stages in the previous step to determine the degree of success that is achieved by the domain-preprocessor rule sets CAPREP and USPREP in parsing the tokens in the name and address fields into the correct domains.

Figure 2-44 on page 544 shows the various stages that are used in this job, including the five data sets that were created in "j03_STAN_XXPREP_NAB" on page 530, five Investigate stages, and five corresponding output Sequential File stages. We modified the names of the stages as shown.

Because the configuration of this job is very similar to that described in "J02_INVCC_ISCODE" on page 186, we do not repeat it here. However, some of the configurations of interest are as follows:

- Figure 2-45 on page 545 shows the INVESTIGATE_CC - Investigate Stage window with the Character Concatenate Investigate option and columns InputPattern_USPREP with the *C* mask for each character and *X* masks on the AddressDomain_USPREP, AreaDomain_USPREP, OutboundPattern_USPREP, HOME_ADDRESS, and HOME_ZIP in the U.S. home addresses CONTACT_INFO file.
- Figure 2-46 on page 545 shows the INVESTIGATE_CC2 - Investigate Stage window with the Character Concatenate Investigate option and columns InputPattern_CAPREP with the *C* mask for each character and *X* masks on the AddressDomain_CAPREP, AreaDomain_CAPREP, OutboundPattern_CAPREP, HOME_ADDRESS, HOME_HONE, and HOME_ZIP in the Canadian home addresses CONTACT_INFO file.
- Figure 2-47 on page 546 shows the INVESTIGATE_CC3 - Investigate Stage window with the Character Concatenate Investigate option and columns InputPattern_USPREP_WORK with the *C* mask for each character and *X* masks on the AddressDomain_USPREP_WORK, AreaDomain_USPREP_WORK, OutboundPattern_USPREP_WORK, WORK_ADDRESS, WORK_PHONE, and WORK_ZIP in the U.S. work addresses CONTACT_INFO file.
- Figure 2-48 on page 546 shows the INVESTIGATE_CC4 - Investigate Stage window with the Character Concatenate Investigate option and columns InputPattern_USPREP with the *C* mask for each character and *X* masks on the NameDomain_USPREP, AddressDomain_USPREP, AreaDomain_USPREP, OutboundPattern_USPREP, NAME, and ADDRESS in the U.S. work addresses DRIVER file.
- Figure 2-49 on page 547 shows the INVESTIGATE_CC5 - Investigate Stage window with the Character Concatenate Investigate option and columns InputPattern_USPREP with the *C* mask for each character and *X* masks on the NameDomain_USPREP, and OutboundPattern_USPREP in the U.S. work addresses CUSTOMER file.
- After saving, compiling, and running this job, the job statistics are shown in Figure 2-50 on page 548.
- The output of the Investigate stages written to the sequential files are shown in Figure 2-51 on page 549 through Figure 2-65 on page 558 as follows:
  - Figure 2-51 on page 549 through Figure 2-54 on page 550 show the report corresponding to the U.S. home addresses of the CONTACT_INFO file. It shows the input patterns that correspond to the erroneous parsing

that left city names in the AddressDomain_USPREP column as highlighted.

– Figure 2-55 on page 551 and Figure 2-56 on page 551 show the report corresponding to the Canadian home addresses of the CONTACT_INFO file. It shows the input patterns that correspond to the erroneous parsing that left city names in the AddressDomain_CAPREP column as highlighted.
– Figure 2-57 on page 552 through Figure 2-59 on page 554 show the report corresponding to the U.S. work addresses of the CONTACT_INFO file. It shows the input patterns that correspond to the erroneous parsing that left city names in the AddressDomain_USPREP_WORK column as highlighted.
– Figure 2-60 on page 555 through Figure 2-63 on page 557 show the report corresponding to the U.S. addresses of the DRIVER file. It shows the input patterns that correspond to the erroneous parsing that left address details in the NameDomain_USPREP column as highlighted.
– Figure 2-64 on page 557 and Figure 2-65 on page 558 show the report corresponding to the names in the CUSTOMER file. All the names seem to have been properly passed.

Given the number of erroneous input patterns, we define input pattern overrides for them in the USPREP rule set to generate the correct output (shown in Figure 2-66 on page 558 through Figure 2-75 on page 567) to generate the proper outbound patterns to move name/address data to the correct domain buckets.

After provisioning the USPREP rule set (not shown here), we rerun the“j03_STAN_XXPREP_NAB” on page 530 and “j04_INVCC_XXPREP_INPUT_PATTERN” on page 541 to verify the success of the overrides as shown in Investigate reports Figure 2-78 on page 570 through Figure 2-81 on page 572.

Now proceed to “j05a_INVW_CONTACT_INFO_HOME_CA_XXPREP” on page 572.

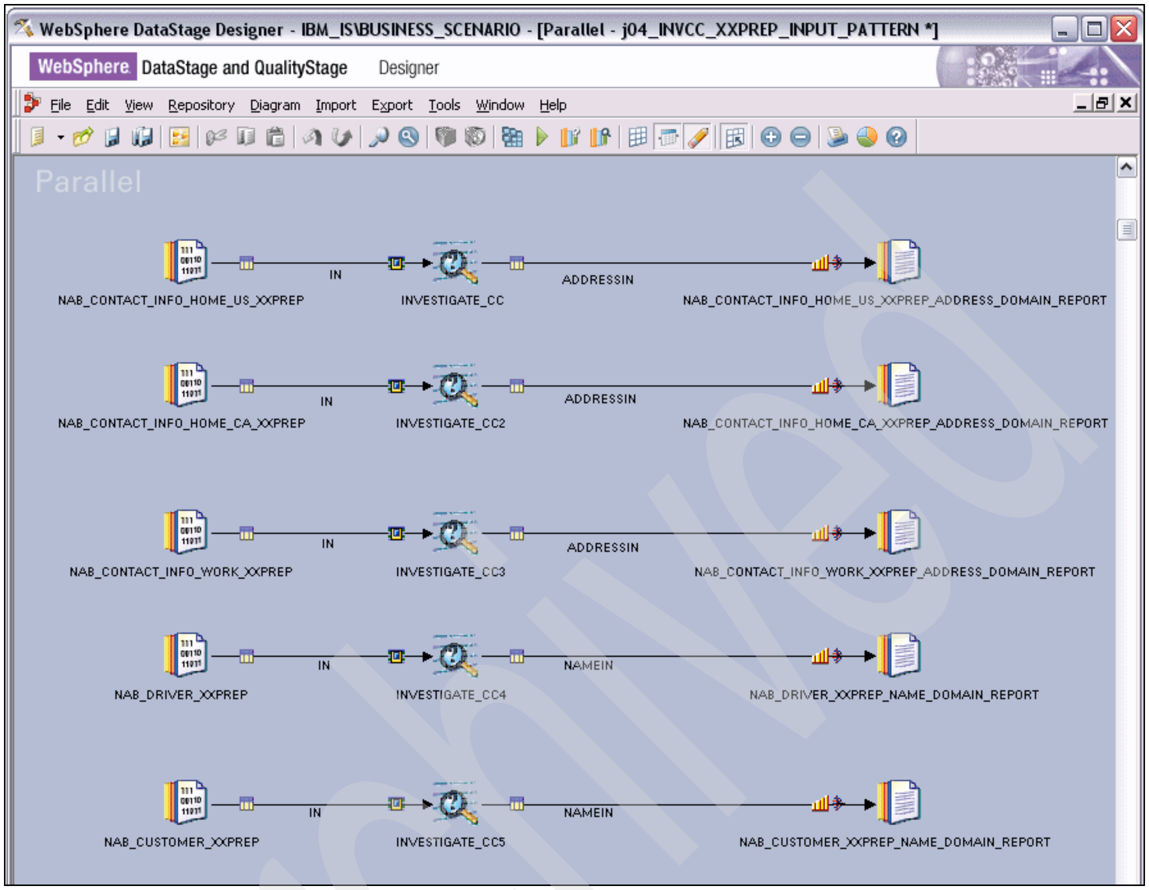
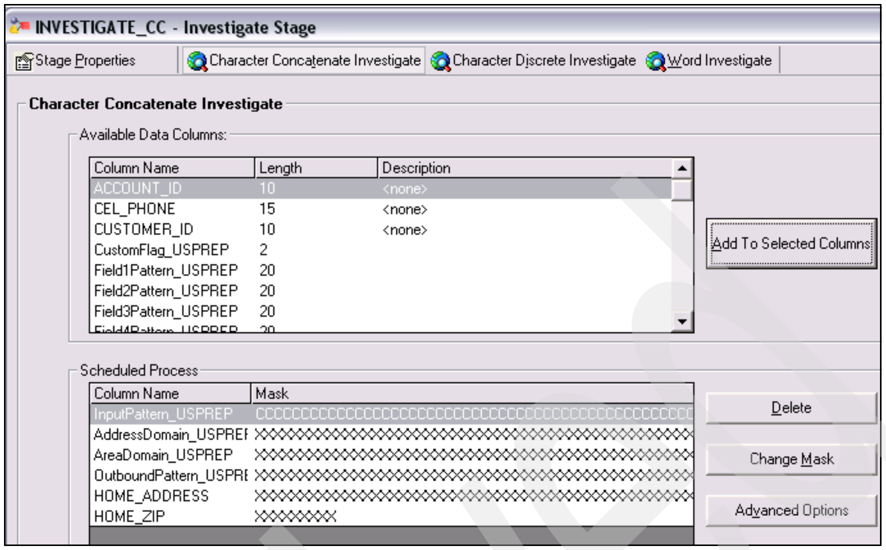

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j04_INVCC_XXPREP_INPUT_PATTERN *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NAB_CONTACT_INFO_HOME_US_XXPREP
IN
INVESTIGATE_CC
ADDRESSIN
NAB_CONTACT_INFO_HOME_US_XXPREP_ADDRESS_DOMAIN_REPORT
NAB_CONTACT_INFO_HOME_CA_XXPREP
IN
INVESTIGATE_CC2
ADDRESSIN
NAB_CONTACT_INFO_HOME_CA_XXPREP_ADDRESS_DOMAIN_REPORT
NAB_CONTACT_INFO_WORK_XXPREP
IN
INVESTIGATE_CC3
ADDRESSIN
NAB_CONTACT_INFO_WORK_XXPREP_ADDRESS_DOMAIN_REPORT
NAB_DRIVER_XXPREP
IN
INVESTIGATE_CC4
NAMEIN
NAB_DRIVER_XXPREP_NAME_DOMAIN_REPORT
NAB_CUSTOMER_XXPREP
IN
INVESTIGATE_CC5
NAMEIN
NAB_CUSTOMER_XXPREP_NAME_DOMAIN_REPORT

Figure 2-44 Create j04_INVCC_XXPREP_INPUT_PATTERN 1/38



INVESTIGATE_CC - Investigate Stage

Stage Properties
Character Concatenate Investigate
Character Discrete Investigate
Word Investigate

Character Concatenate Investigate

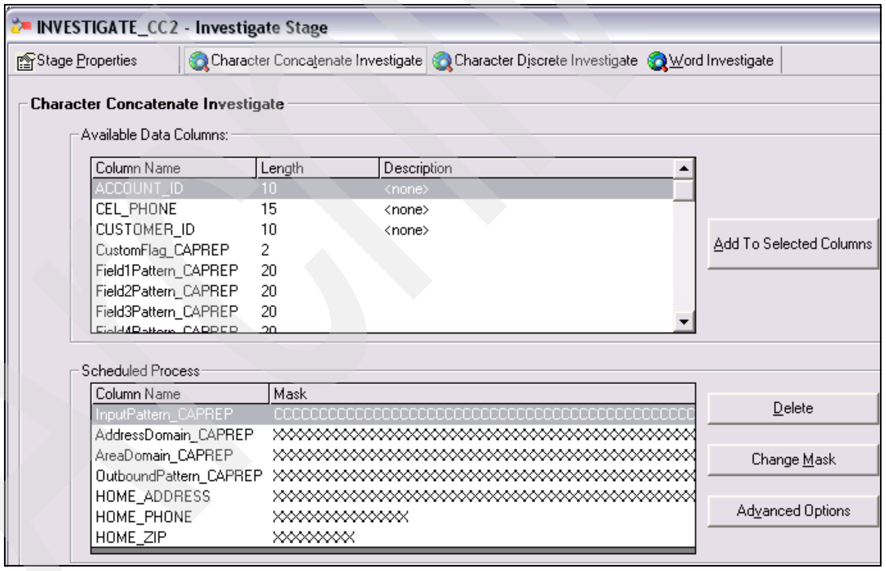
Available Data Columns:

Column Name	Length	Description
ACCOUNT_ID	10	<none>
CEL_PHONE	15	<none>
CUSTOMER_ID	10	<none>
CustomFlag_USPREP	2	
Field1Pattern_USPREP	20	
Field2Pattern_USPREP	20	
Field3Pattern_USPREP	20	

Scheduled Process

Column Name	Mask
InputPattern_USPREP	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
AddressDomain_USPREI	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
AreaDomain_USPREP	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
OutboundPattern_USPRI	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
HOME_ADDRESS	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
HOME_ZIP	XXXXXXXXX

Figure 2-45 Create j04_INVCC_XXPREP_INPUT_PATTERN 2/38



INVESTIGATE_CC2 - Investigate Stage

Stage Properties
Character Concatenate Investigate
Character Discrete Investigate
Word Investigate

Character Concatenate Investigate

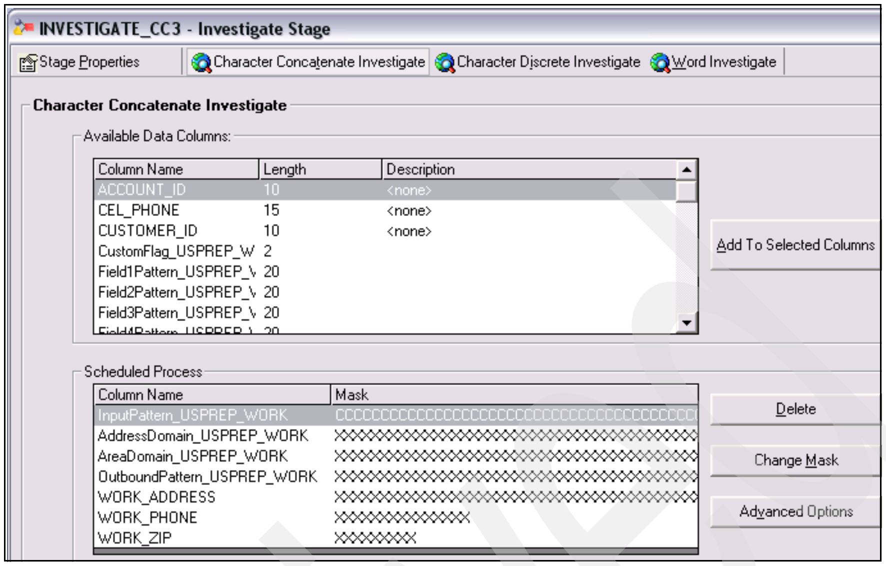
Available Data Columns:

Column Name	Length	Description
ACCOUNT_ID	10	<none>
CEL_PHONE	15	<none>
CUSTOMER_ID	10	<none>
CustomFlag_CAPREP	2	
Field1Pattern_CAPREP	20	
Field2Pattern_CAPREP	20	
Field3Pattern_CAPREP	20	

Scheduled Process

Column Name	Mask
InputPattern_CAPREP	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
AddressDomain_CAPREP	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
AreaDomain_CAPREP	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
OutboundPattern_CAPREP	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
HOME_ADDRESS	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
HOME_PHONE	XXXXXXXXXXXXXXX
HOME_ZIP	XXXXXXXXX

Figure 2-46 Create j04_INVCC_XXPREP_INPUT_PATTERN 3/38


INVESTIGATE_CC3 - Investigate Stage

Stage Properties | Character Concatenate Investigate | Character Discrete Investigate | Word Investigate

Character Concatenate Investigate

Available Data Columns:

Column Name	Length	Description
ACCOUNT_ID	10	<none>
CEL_PHONE	15	<none>
CUSTOMER_ID	10	<none>
CustomFlag_USPREP_W	2	
Field1Pattern_USPREP_\	20	
Field2Pattern_USPREP_\	20	
Field3Pattern_USPREP_\	20	

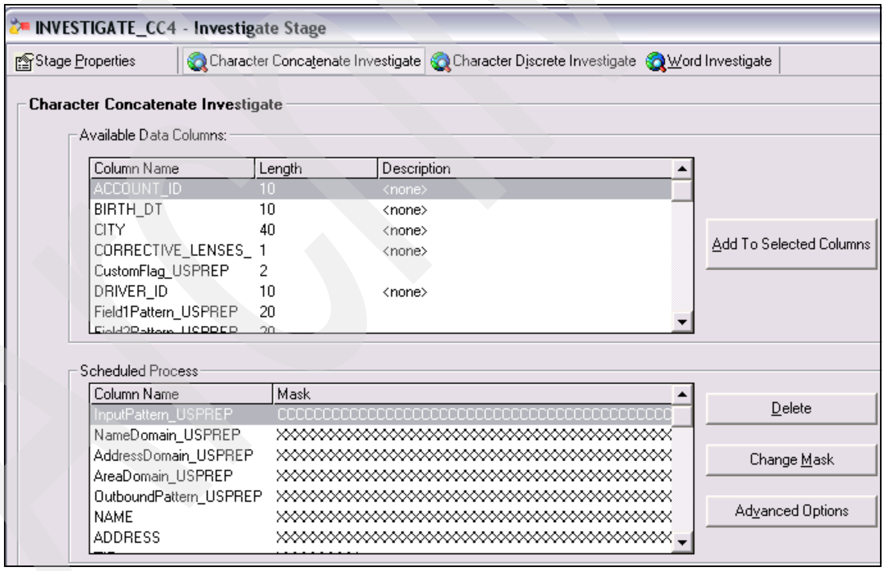
Add To Selected Columns

Scheduled Process

Column Name	Mask
InputPattern_USPREP_WORK	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
AddressDomain_USPREP_WORK	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
AreaDomain_USPREP_WORK	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
OutboundPattern_USPREP_WORK	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
WORK_ADDRESS	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
WORK_PHONE	XXXXXXXXXXXXXXX
WORK_ZIP	XXXXXXXXX

Delete | Change Mask | Advanced Options

Figure 2-47 Create j04_INVCC_XXPREP_INPUT_PATTERN 4/38


INVESTIGATE_CC4 - Investigate Stage

Stage Properties | Character Concatenate Investigate | Character Discrete Investigate | Word Investigate

Character Concatenate Investigate

Available Data Columns:

Column Name	Length	Description
ACCOUNT_ID	10	<none>
BIRTH_DT	10	<none>
CITY	40	<none>
CORRECTIVE_LENSES_	1	<none>
CustomFlag_USPREP	2	
DRIVER_ID	10	<none>
Field1Pattern_USPREP	20	

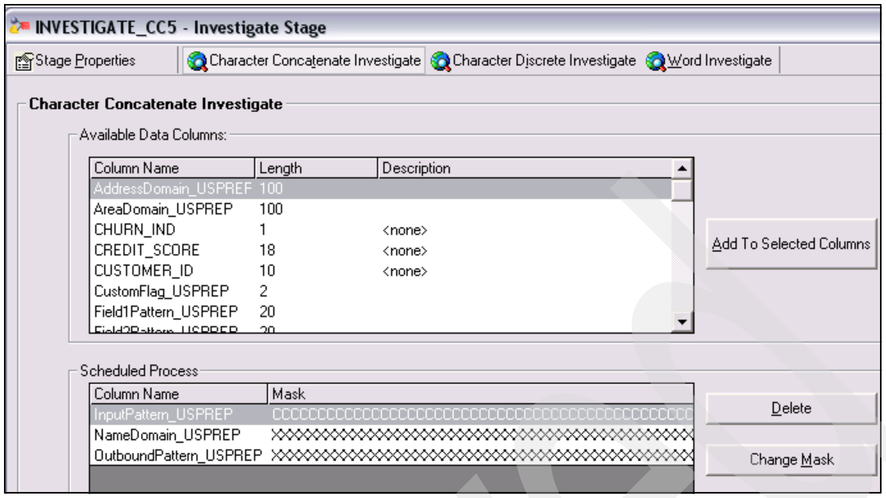
Add To Selected Columns

Scheduled Process

Column Name	Mask
InputPattern_USPREP	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
NameDomain_USPREP	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
AddressDomain_USPREP	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
AreaDomain_USPREP	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
OutboundPattern_USPREP	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
NAME	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
ADDRESS	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Delete | Change Mask | Advanced Options

Figure 2-48 Create j04_INVCC_XXPREP_INPUT_PATTERN 5/38


INVESTIGATE_CC5 - Investigate Stage

Stage Properties
Character Concatenate Investigate
Character Discrete Investigate
Word Investigate

Character Concatenate Investigate

Available Data Columns:

Column Name	Length	Description
AddressDomain_USPREF	100	
AreaDomain_USPREP	100	
CHURN_IND	1	<none>
CREDIT_SCORE	18	<none>
CUSTOMER_ID	10	<none>
CustomFlag_USPREP	2	
Field1Pattern_USPREP	20	
Field2Pattern_USPREP	20	

Scheduled Process

Column Name	Mask
InputPattern_USPREP	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
NameDomain_USPREP	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
OutboundPattern_USPREP	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Figure 2-49 Create j04_INVCC_XXPREP_INPUT_PATTERN 6/38

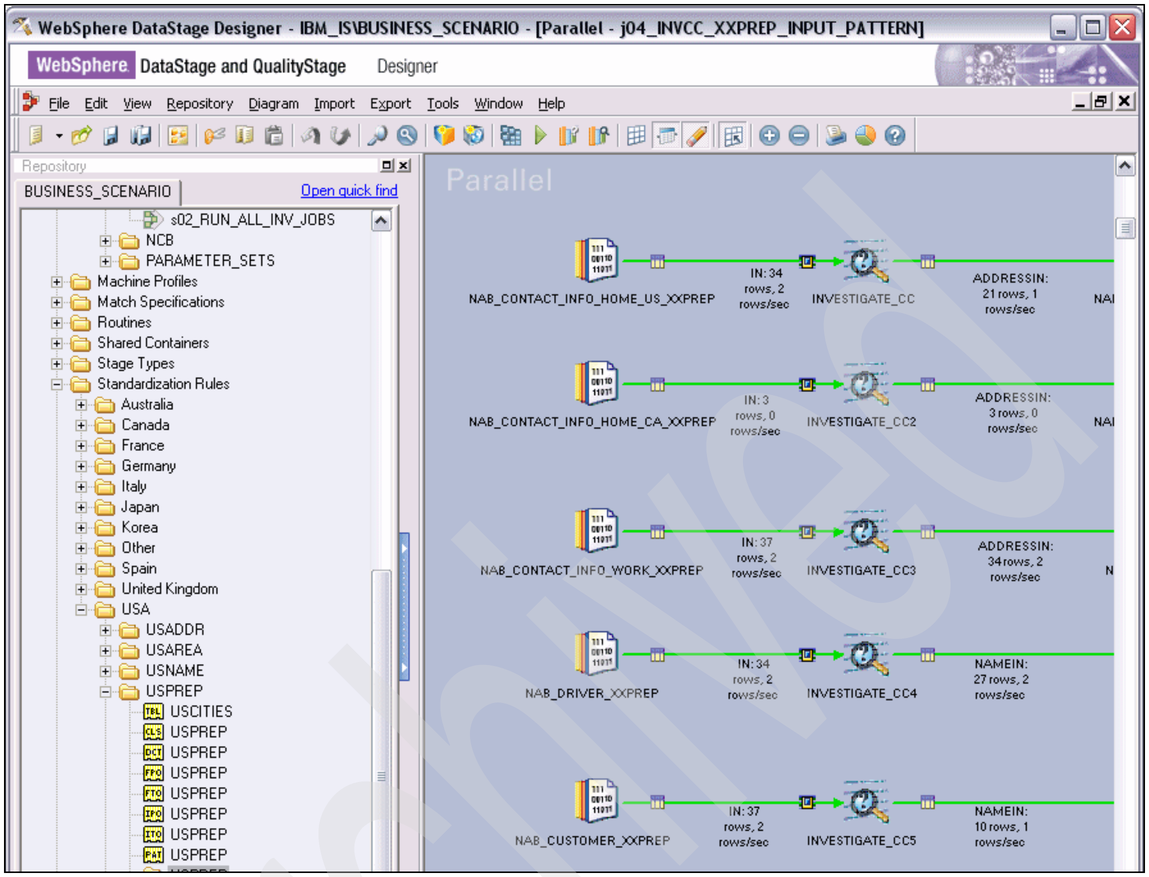
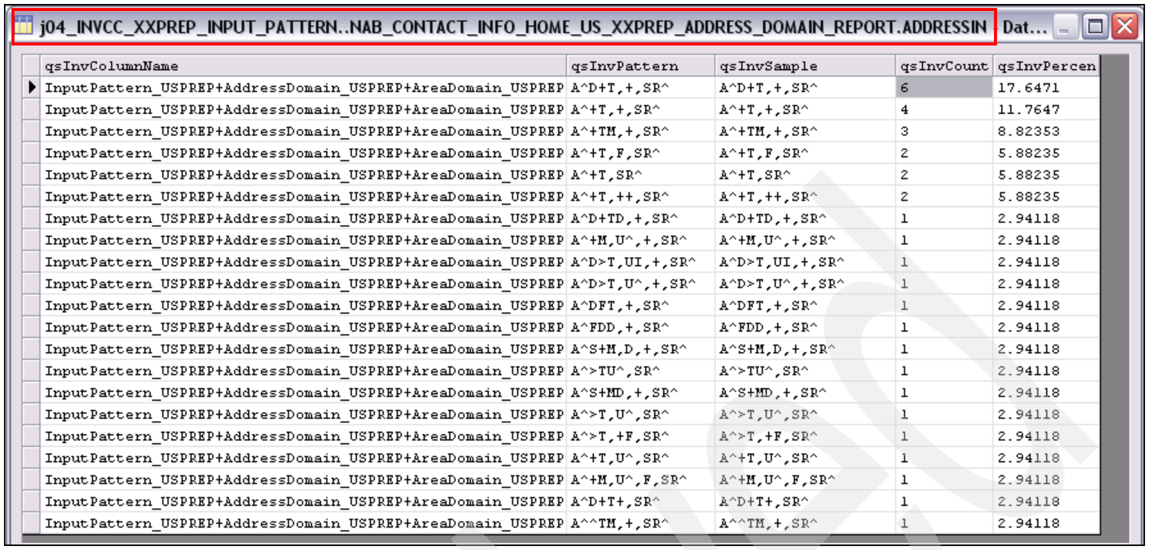
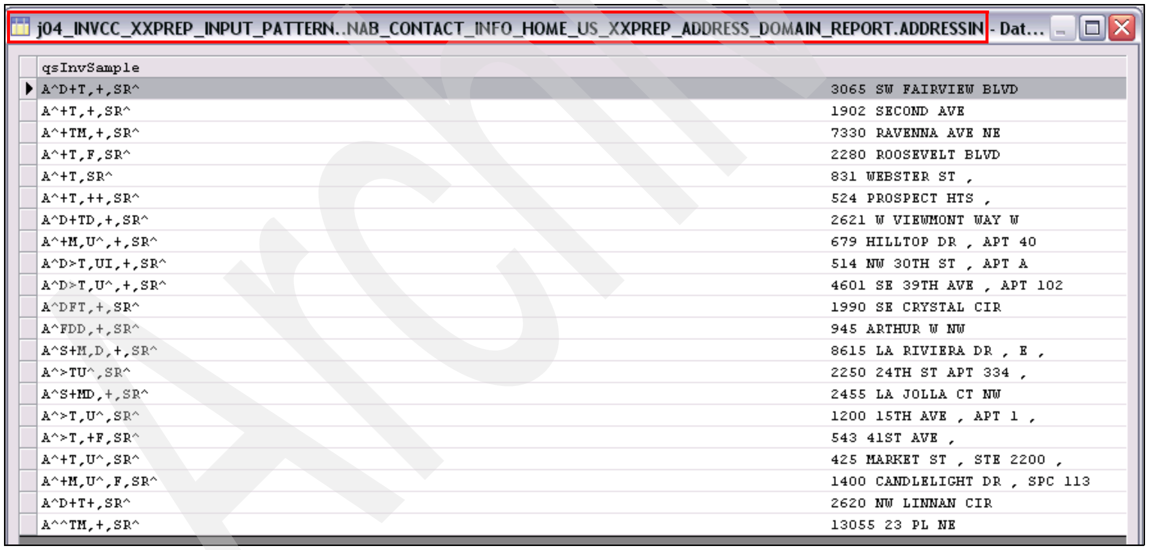


Figure 2-50 Create j04_INVCC_XXPREP_INPUT_PATTERN 7/38


j04_INVCC_XXPREP_INPUT_PATTERN..NAB_CONTACT_INFO_HOME_US_XXPREP_ADDRESS_DOMAIN_REPORT.ADDRESSIN - Dat...

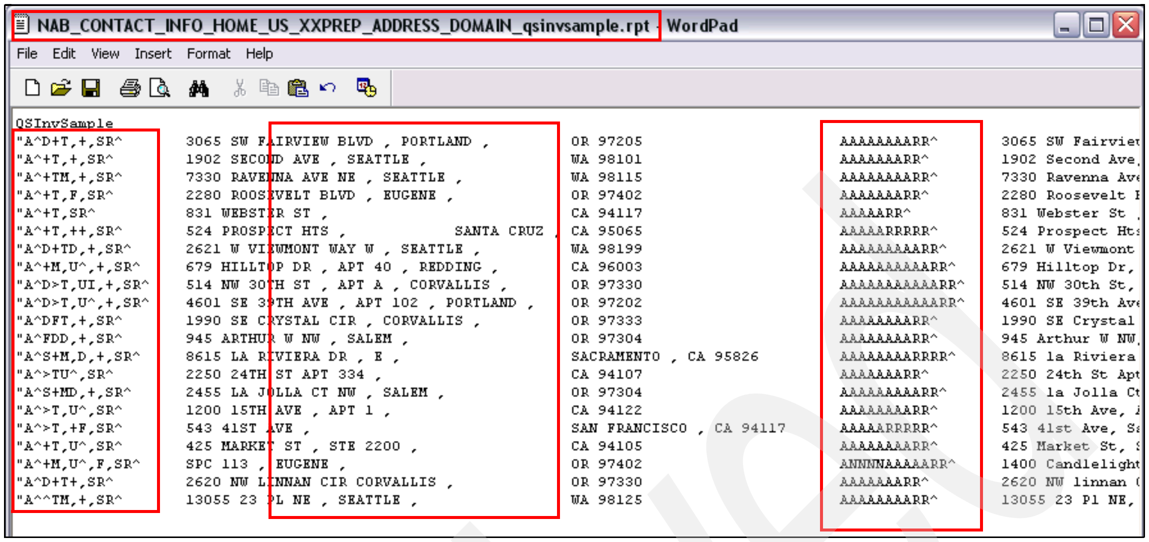
qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercen
InputPattern_USPREP+AddressDomain_USPREP+AreaDomain_USPREP	A^D+T,+,SR^	A^D+T,+,SR^	6	17.6471
InputPattern_USPREP+AddressDomain_USPREP+AreaDomain_USPREP	A^+T,+,SR^	A^+T,+,SR^	4	11.7647
InputPattern_USPREP+AddressDomain_USPREP+AreaDomain_USPREP	A^+TM,+,SR^	A^+TM,+,SR^	3	8.82353
InputPattern_USPREP+AddressDomain_USPREP+AreaDomain_USPREP	A^+T,F,SR^	A^+T,F,SR^	2	5.88235
InputPattern_USPREP+AddressDomain_USPREP+AreaDomain_USPREP	A^+T,SR^	A^+T,SR^	2	5.88235
InputPattern_USPREP+AddressDomain_USPREP+AreaDomain_USPREP	A^+T,++,SR^	A^+T,++,SR^	2	5.88235
InputPattern_USPREP+AddressDomain_USPREP+AreaDomain_USPREP	A^D+TD,+,SR^	A^D+TD,+,SR^	1	2.94118
InputPattern_USPREP+AddressDomain_USPREP+AreaDomain_USPREP	A^+M,U^,+,SR^	A^+M,U^,+,SR^	1	2.94118
InputPattern_USPREP+AddressDomain_USPREP+AreaDomain_USPREP	A^D>T,UI,+,SR^	A^D>T,UI,+,SR^	1	2.94118
InputPattern_USPREP+AddressDomain_USPREP+AreaDomain_USPREP	A^D>T,U^,+,SR^	A^D>T,U^,+,SR^	1	2.94118
InputPattern_USPREP+AddressDomain_USPREP+AreaDomain_USPREP	A^DFT,+,SR^	A^DFT,+,SR^	1	2.94118
InputPattern_USPREP+AddressDomain_USPREP+AreaDomain_USPREP	A^FDD,+,SR^	A^FDD,+,SR^	1	2.94118
InputPattern_USPREP+AddressDomain_USPREP+AreaDomain_USPREP	A^S+M,D,+,SR^	A^S+M,D,+,SR^	1	2.94118
InputPattern_USPREP+AddressDomain_USPREP+AreaDomain_USPREP	A^>TU^,SR^	A^>TU^,SR^	1	2.94118
InputPattern_USPREP+AddressDomain_USPREP+AreaDomain_USPREP	A^S+MD,+,SR^	A^S+MD,+,SR^	1	2.94118
InputPattern_USPREP+AddressDomain_USPREP+AreaDomain_USPREP	A^>T,U^,SR^	A^>T,U^,SR^	1	2.94118
InputPattern_USPREP+AddressDomain_USPREP+AreaDomain_USPREP	A^>T,+F,SR^	A^>T,+F,SR^	1	2.94118
InputPattern_USPREP+AddressDomain_USPREP+AreaDomain_USPREP	A^+T,U^,SR^	A^+T,U^,SR^	1	2.94118
InputPattern_USPREP+AddressDomain_USPREP+AreaDomain_USPREP	A^+M,U^,F,SR^	A^+M,U^,F,SR^	1	2.94118
InputPattern_USPREP+AddressDomain_USPREP+AreaDomain_USPREP	A^D+T+,SR^	A^D+T+,SR^	1	2.94118
InputPattern_USPREP+AddressDomain_USPREP+AreaDomain_USPREP	A^^TM,+,SR^	A^^TM,+,SR^	1	2.94118

Figure 2-51 Create j04_INVCC_XXPREP_INPUT_PATTERN 8/38


j04_INVCC_XXPREP_INPUT_PATTERN..NAB_CONTACT_INFO_HOME_US_XXPREP_ADDRESS_DOMAIN_REPORT.ADDRESSIN - Dat...

qsInvSample	
A^D+T,+,SR^	3065 SW FAIRVIEW BLVD
A^+T,+,SR^	1902 SECOND AVE
A^+TM,+,SR^	7330 RAVENNA AVE NE
A^+T,F,SR^	2280 ROOSEVELT BLVD
A^+T,SR^	831 WEBSTER ST ,
A^+T,++,SR^	524 PROSPECT HTS ,
A^D+TD,+,SR^	2621 W VIEWMONT WAY W
A^+M,U^,+,SR^	679 HILLTOP DR , APT 40
A^D>T,UI,+,SR^	514 NW 30TH ST , APT A
A^D>T,U^,+,SR^	4601 SE 39TH AVE , APT 102
A^DFT,+,SR^	1990 SE CRYSTAL CIR
A^FDD,+,SR^	945 ARTHUR W NW
A^S+M,D,+,SR^	8615 LA RIVIERA DR , E ,
A^>TU^,SR^	2250 24TH ST APT 334 ,
A^S+MD,+,SR^	2455 LA JOLLA CT NW
A^>T,U^,SR^	1200 15TH AVE , APT 1 ,
A^>T,+F,SR^	543 41ST AVE ,
A^+T,U^,SR^	425 MARKET ST , STE 2200 ,
A^+M,U^,F,SR^	1400 CANDLELIGHT DR , SPC 113
A^D+T+,SR^	2620 NW LINNAN CIR
A^^TM,+,SR^	13055 23 PL NE

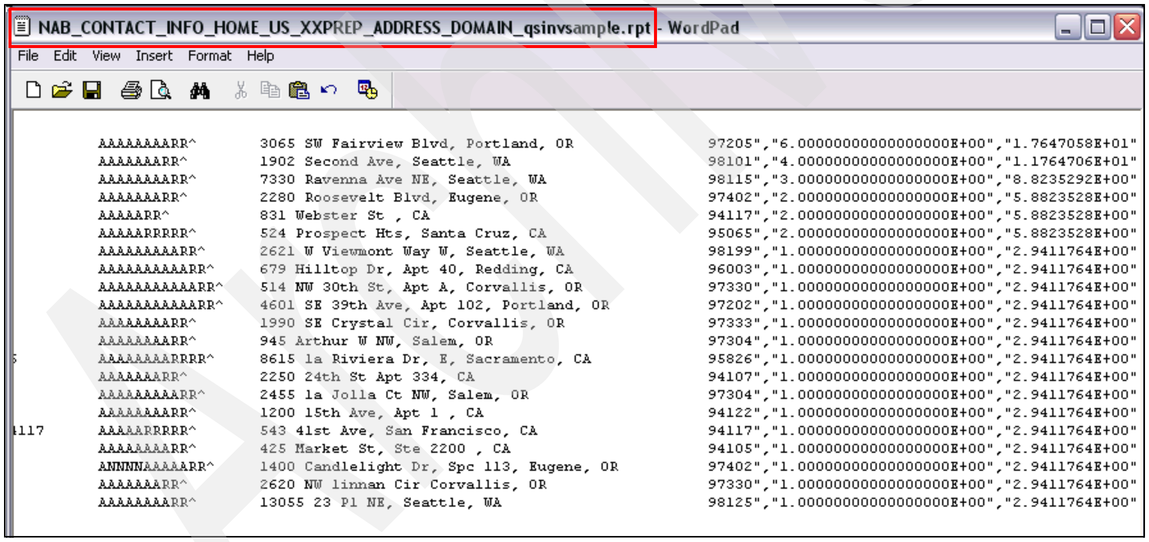
Figure 2-52 Create j04_INVCC_XXPREP_INPUT_PATTERN 9/38



NAB_CONTACT_INFO_HOME_US_XXPREP_ADDRESS_DOMAIN_qsinvsample.rpt - WordPad

```
QSInvSample
"A^D+T,+,SR^        3065 SW FAIRVIEW BLVD , PORTLAND ,          OR 97205                         AAAAAAAAARR^     3065 SW Fairview
"A^+T,+,SR^         1902 SECOND AVE , SEATTLE ,                 WA 98101                         AAAAAAAARR^      1902 Second Ave,
"A^+TM,+,SR^        7330 RAVENNA AVE NE , SEATTLE ,             WA 98115                         AAAAAAAAARR^     7330 Ravenna Ave
"A^+T,F,SR^         2280 ROOSEVELT BLVD , EUGENE ,              OR 97402                         AAAAAAAARR^      2280 Roosevelt B
"A^+T,SR^           831 WEBSTER ST ,                            CA 94117                         AAAAARR^         831 Webster St ,
"A^+T,++,SR^        524 PROSPECT HTS ,               SANTA CRUZ , CA 95065                       AAAAARRRRR^      524 Prospect Hts
"A^D+TD,+,SR^       2621 W VIEWMONT WAY W , SEATTLE ,           WA 98199                         AAAAAAAAAARR^    2621 W Viewmont
"A^+M,U^,+,SR^      679 HILLTOP DR , APT 40 , REDDING ,         CA 96003                         AAAAAAAAAAARR^   679 Hilltop Dr,
"A^D>T,UI,+,SR^     514 NW 30TH ST , APT A , CORVALLIS ,        OR 97330                         AAAAAAAAAAAARR^  514 NW 30th St,
"A^D>T,U^,+,SR^     4601 SE 39TH AVE , APT 102 , PORTLAND ,     OR 97202                         AAAAAAAAAAAARR^  4601 SE 39th Ave
"A^DFT,+,SR^        1990 SE CRYSTAL CIR , CORVALLIS ,           OR 97333                         AAAAAAAAARR^     1990 SE Crystal
"A^FDD,+,SR^        945 ARTHUR W NW , SALEM ,                   OR 97304                         AAAAAAAAARR^     945 Arthur W NW,
"A^S+M,D,+,SR^      8615 LA RIVIERA DR , E ,                    SACRAMENTO , CA 95826            AAAAAAAAARRRR^   8615 la Riviera
"A^>TU^,SR^         2250 24TH ST APT 334 ,                      CA 94107                         AAAAAAAARR^      2250 24th St Apt
"A^S+MD,+,SR^       2455 LA JOLLA CT NW , SALEM ,               OR 97304                         AAAAAAAAAARR^    2455 la Jolla Ct
"A^>T,U^,SR^        1200 15TH AVE , APT 1 ,                     CA 94122                         AAAAAAAAARR^     1200 15th Ave, A
"A^>T,+F,SR^        543 41ST AVE ,                              SAN FRANCISCO , CA 94117         AAAAARRRRR^      543 41st Ave, Sa
"A^+T,U^,SR^        425 MARKET ST , STE 2200 ,                  CA 94105                         AAAAAAAARR^      425 Market St, S
"A^+M,U^,F,SR^      SPC 113 , EUGENE ,                          OR 97402                         ANNNNAAAAAARR^   1400 Candlelight
"A^D+T+,SR^         2620 NW LINNAN CIR CORVALLIS ,              OR 97330                         AAAAAAAARR^      2620 NW linnan C
"A^^TM,+,SR^        13055 23 PL NE , SEATTLE ,                  WA 98125                         AAAAAAAAARR^     13055 23 Pl NE,
```

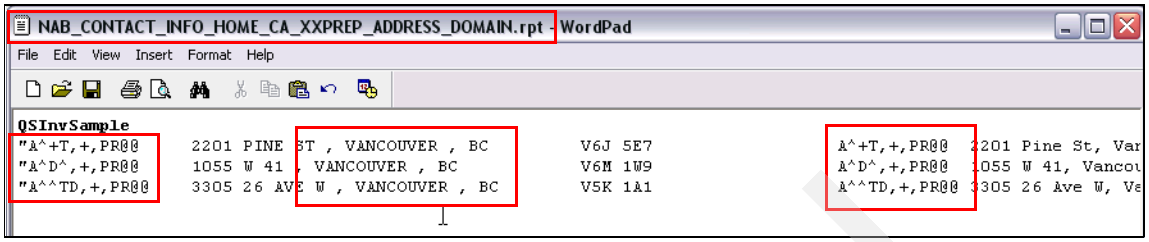
Figure 2-53 Create j04_INVCC_XXPREP_INPUT_PATTERN 10/38



NAB_CONTACT_INFO_HOME_US_XXPREP_ADDRESS_DOMAIN_qsinvsample.rpt - WordPad

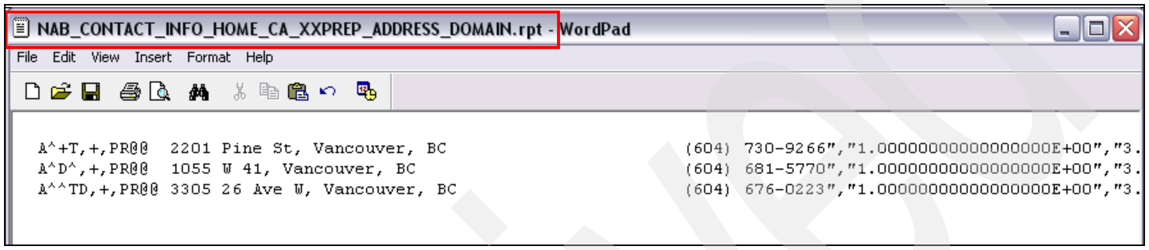
```
         AAAAAAAAARR^       3065 SW Fairview Blvd, Portland, OR         97205","6.0000000000000000E+00","1.7647058E+01"
         AAAAAAAARR^        1902 Second Ave, Seattle, WA                98101","4.0000000000000000E+00","1.1764706E+01"
         AAAAAAAAARR^       7330 Ravenna Ave NE, Seattle, WA            98115","3.0000000000000000E+00","8.8235292E+00"
         AAAAAAAARR^        2280 Roosevelt Blvd, Eugene, OR             97402","2.0000000000000000E+00","5.8823528E+00"
         AAAAARR^           831 Webster St , CA                         94117","2.0000000000000000E+00","5.8823528E+00"
         AAAAARRRRR^        524 Prospect Hts, Santa Cruz, CA            95065","2.0000000000000000E+00","5.8823528E+00"
         AAAAAAAAAARR^      2621 W Viewmont Way W, Seattle, WA          98199","1.0000000000000000E+00","2.9411764E+00"
         AAAAAAAAAAARR^     679 Hilltop Dr, Apt 40, Redding, CA         96003","1.0000000000000000E+00","2.9411764E+00"
         AAAAAAAAAAAARR^    514 NW 30th St, Apt A, Corvallis, OR        97330","1.0000000000000000E+00","2.9411764E+00"
         AAAAAAAAAAAARR^    4601 SE 39th Ave, Apt 102, Portland, OR     97202","1.0000000000000000E+00","2.9411764E+00"
         AAAAAAAAARR^       1990 SE Crystal Cir, Corvallis, OR          97333","1.0000000000000000E+00","2.9411764E+00"
         AAAAAAAAARR^       945 Arthur W NW, Salem, OR                  97304","1.0000000000000000E+00","2.9411764E+00"
5        AAAAAAAAARRRR^     8615 la Riviera Dr, E, Sacramento, CA       95826","1.0000000000000000E+00","2.9411764E+00"
         AAAAAAAARR^        2250 24th St Apt 334, CA                    94107","1.0000000000000000E+00","2.9411764E+00"
         AAAAAAAAAARR^      2455 la Jolla Ct NW, Salem, OR              97304","1.0000000000000000E+00","2.9411764E+00"
         AAAAAAAAARR^       1200 15th Ave, Apt 1 , CA                   94122","1.0000000000000000E+00","2.9411764E+00"
4117     AAAAARRRRR^        543 41st Ave, San Francisco, CA             94117","1.0000000000000000E+00","2.9411764E+00"
         AAAAAAAARR^        425 Market St, Ste 2200 , CA                94105","1.0000000000000000E+00","2.9411764E+00"
         ANNNNAAAAAARR^     1400 Candlelight Dr, Spc 113, Eugene, OR    97402","1.0000000000000000E+00","2.9411764E+00"
         AAAAAAAARR^        2620 NW linnan Cir Corvallis, OR            97330","1.0000000000000000E+00","2.9411764E+00"
         AAAAAAAAARR^       13055 23 Pl NE, Seattle, WA                 98125","1.0000000000000000E+00","2.9411764E+00"
```

Figure 2-54 Create j04_INVCC_XXPREP_INPUT_PATTERN 11/38


NAB_CONTACT_INFO_HOME_CA_XXPREP_ADDRESS_DOMAIN.rpt - WordPad
File Edit View Insert Format Help

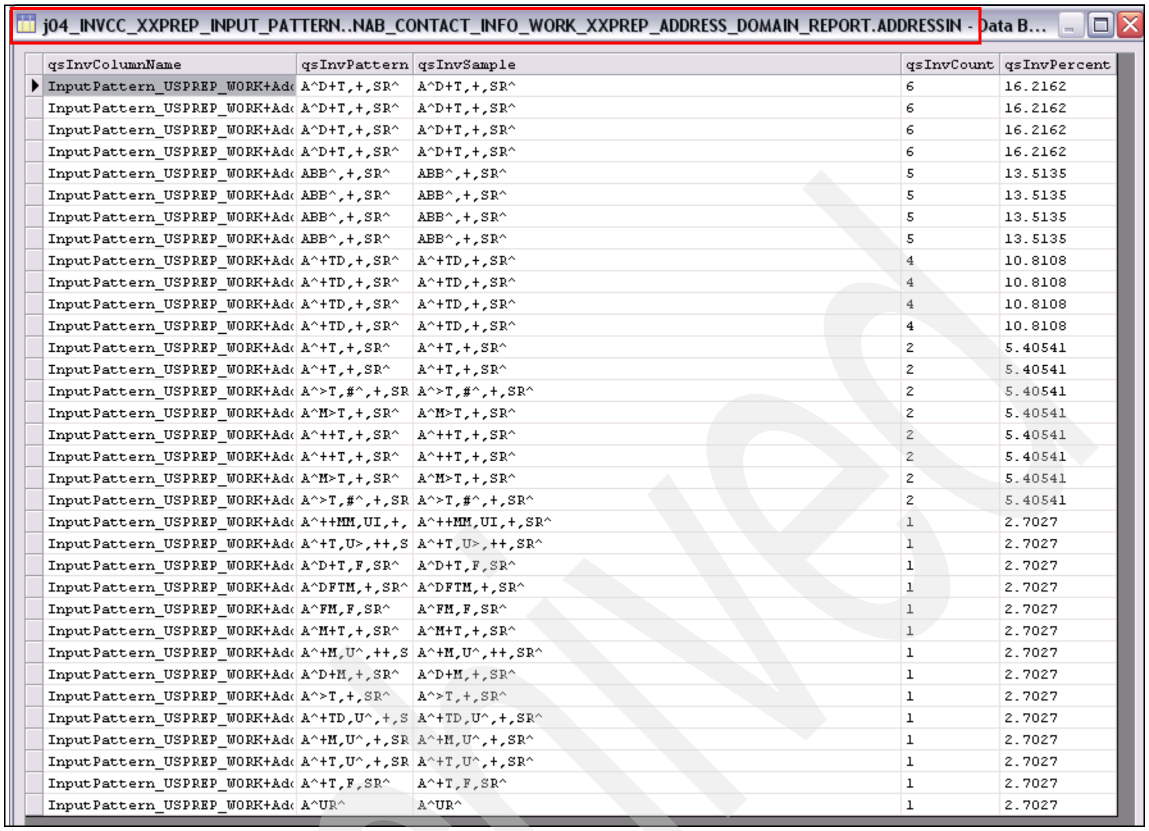
```
QSInvSample
"A^+T,+,PR@@      2201 PINE ST , VANCOUVER , BC        V6J 5E7              A^+T,+,PR@@  2201 Pine St, Var
"A^D^,+,PR@@      1055 W 41 , VANCOUVER , BC           V6M 1W9              A^D^,+,PR@@  1055 W 41, Vancou
"A^^TD,+,PR@@     3305 26 AVE W , VANCOUVER , BC       V5K 1A1              A^^TD,+,PR@@ 3305 26 Ave W, Va
```

Figure 2-55 Create j04_INVCC_XXPREP_INPUT_PATTERN 12/38


NAB_CONTACT_INFO_HOME_CA_XXPREP_ADDRESS_DOMAIN.rpt - WordPad
File Edit View Insert Format Help

```
A^+T,+,PR@@  2201 Pine St, Vancouver, BC                (604) 730-9266","1.000000000000000000E+00","3.
A^D^,+,PR@@  1055 W 41, Vancouver, BC                   (604) 681-5770","1.000000000000000000E+00","3.
A^^TD,+,PR@@ 3305 26 Ave W, Vancouver, BC               (604) 676-0223","1.000000000000000000E+00","3.
```

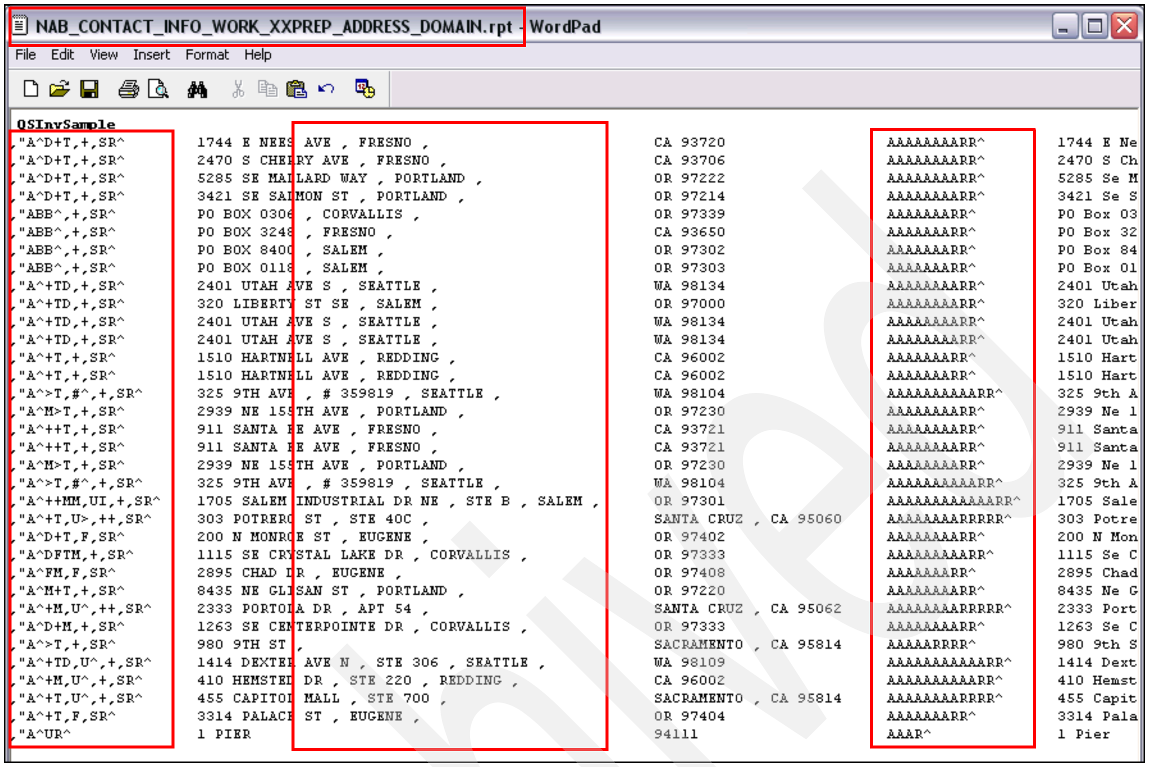
Figure 2-56 Create j04_INVCC_XXPREP_INPUT_PATTERN 13/38



j04_INVCC_XXPREP_INPUT_PATTERN..NAB_CONTACT_INFO_WORK_XXPREP_ADDRESS_DOMAIN_REPORT.ADDRESSIN - Data B...

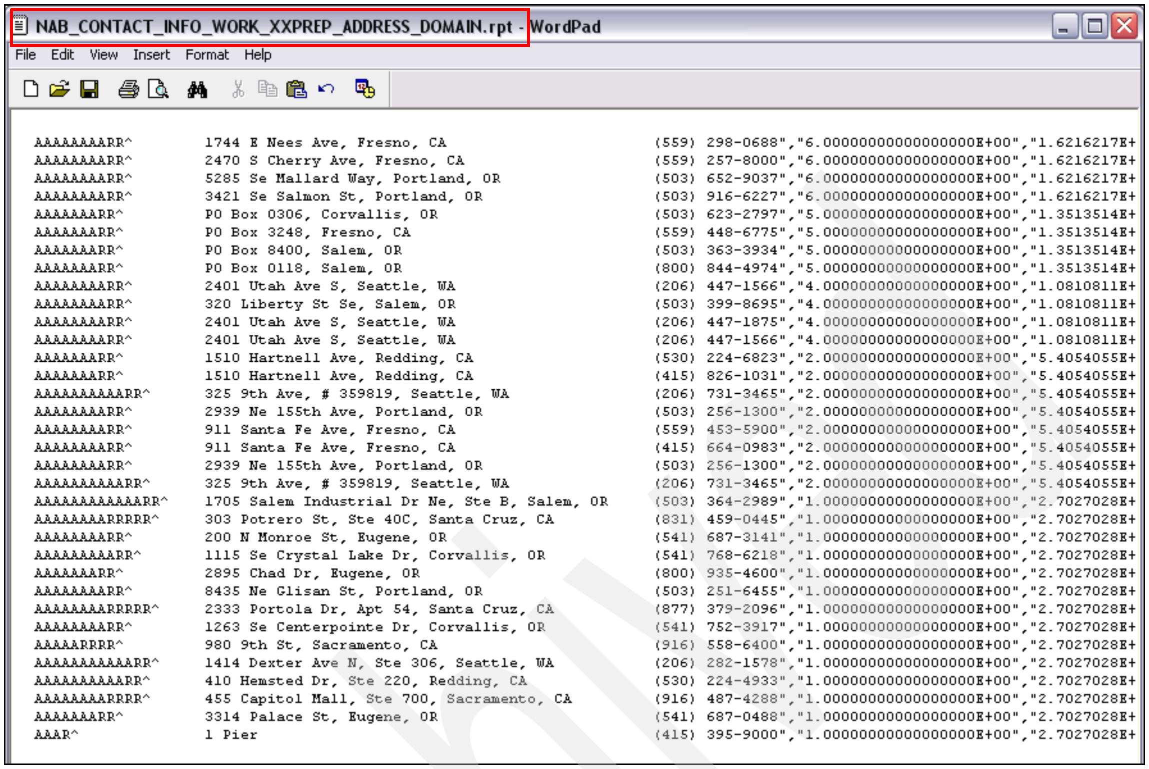
qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
InputPattern_USPREP_WORK+Ad	A^D+T,+,SR^	A^D+T,+,SR^	6	16.2162
InputPattern_USPREP_WORK+Ad	A^D+T,+,SR^	A^D+T,+,SR^	6	16.2162
InputPattern_USPREP_WORK+Ad	A^D+T,+,SR^	A^D+T,+,SR^	6	16.2162
InputPattern_USPREP_WORK+Ad	A^D+T,+,SR^	A^D+T,+,SR^	6	16.2162
InputPattern_USPREP_WORK+Ad	ABB^,+,SR^	ABB^,+,SR^	5	13.5135
InputPattern_USPREP_WORK+Ad	ABB^,+,SR^	ABB^,+,SR^	5	13.5135
InputPattern_USPREP_WORK+Ad	ABB^,+,SR^	ABB^,+,SR^	5	13.5135
InputPattern_USPREP_WORK+Ad	ABB^,+,SR^	ABB^,+,SR^	5	13.5135
InputPattern_USPREP_WORK+Ad	A^+TD,+,SR^	A^+TD,+,SR^	4	10.8108
InputPattern_USPREP_WORK+Ad	A^+TD,+,SR^	A^+TD,+,SR^	4	10.8108
InputPattern_USPREP_WORK+Ad	A^+TD,+,SR^	A^+TD,+,SR^	4	10.8108
InputPattern_USPREP_WORK+Ad	A^+TD,+,SR^	A^+TD,+,SR^	4	10.8108
InputPattern_USPREP_WORK+Ad	A^+T,+,SR^	A^+T,+,SR^	2	5.40541
InputPattern_USPREP_WORK+Ad	A^+T,+,SR^	A^+T,+,SR^	2	5.40541
InputPattern_USPREP_WORK+Ad	A^>T,#^,+,SR	A^>T,#^,+,SR^	2	5.40541
InputPattern_USPREP_WORK+Ad	A^M>T,+,SR^	A^M>T,+,SR^	2	5.40541
InputPattern_USPREP_WORK+Ad	A^++T,+,SR^	A^++T,+,SR^	2	5.40541
InputPattern_USPREP_WORK+Ad	A^++T,+,SR^	A^++T,+,SR^	2	5.40541
InputPattern_USPREP_WORK+Ad	A^M>T,+,SR^	A^M>T,+,SR^	2	5.40541
InputPattern_USPREP_WORK+Ad	A^>T,#^,+,SR	A^>T,#^,+,SR^	2	5.40541
InputPattern_USPREP_WORK+Ad	A^++MM,UI,+,	A^++MM,UI,+,SR^	1	2.7027
InputPattern_USPREP_WORK+Ad	A^+T,U>,++,S	A^+T,U>,++,SR^	1	2.7027
InputPattern_USPREP_WORK+Ad	A^D+T,F,SR^	A^D+T,F,SR^	1	2.7027
InputPattern_USPREP_WORK+Ad	A^DFTM,+,SR^	A^DFTM,+,SR^	1	2.7027
InputPattern_USPREP_WORK+Ad	A^FM,F,SR^	A^FM,F,SR^	1	2.7027
InputPattern_USPREP_WORK+Ad	A^M+T,+,SR^	A^M+T,+,SR^	1	2.7027
InputPattern_USPREP_WORK+Ad	A^+M,U^,++,S	A^+M,U^,++,SR^	1	2.7027
InputPattern_USPREP_WORK+Ad	A^D+M,+,SR^	A^D+M,+,SR^	1	2.7027
InputPattern_USPREP_WORK+Ad	A^>T,+,SR^	A^>T,+,SR^	1	2.7027
InputPattern_USPREP_WORK+Ad	A^+TD,U^,+,S	A^+TD,U^,+,SR^	1	2.7027
InputPattern_USPREP_WORK+Ad	A^+M,U^,+,SR	A^+M,U^,+,SR^	1	2.7027
InputPattern_USPREP_WORK+Ad	A^+T,U^,+,SR	A^+T,U^,+,SR^	1	2.7027
InputPattern_USPREP_WORK+Ad	A^+T,F,SR^	A^+T,F,SR^	1	2.7027
InputPattern_USPREP_WORK+Ad	A^UR^	A^UR^	1	2.7027

Figure 2-57 Create j04_INVCC_XXPREP_INPUT_PATTERN 14/38


NAB_CONTACT_INFO_WORK_XXPREP_ADDRESS_DOMAIN.rpt - WordPad
File Edit View Insert Format Help

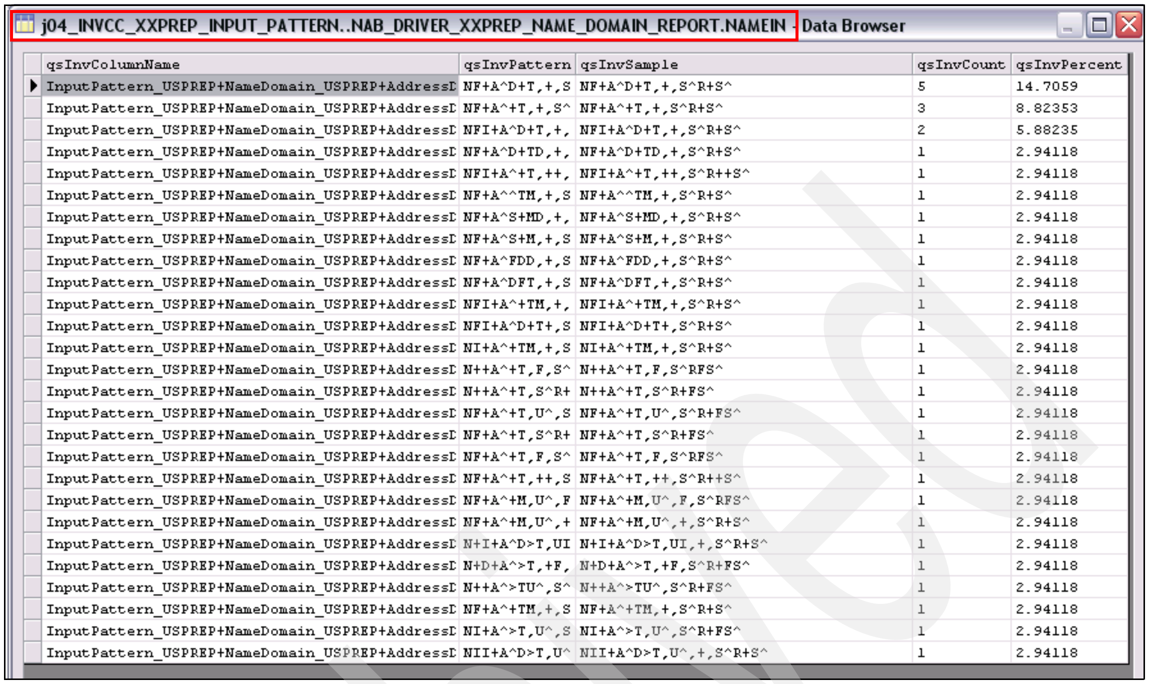
```
QSInvSample
,"A^D+T,+,SR^        1744 E NEES AVE , FRESNO ,                   CA 93720                  AAAAAAAARR^       1744 E Ne
,"A^D+T,+,SR^        2470 S CHERRY AVE , FRESNO ,                 CA 93706                  AAAAAAAARR^       2470 S Ch
,"A^D+T,+,SR^        5285 SE MALLARD WAY , PORTLAND ,             OR 97222                  AAAAAAAARR^       5285 Se M
,"A^D+T,+,SR^        3421 SE SALMON ST , PORTLAND ,               OR 97214                  AAAAAAAARR^       3421 Se S
,"ABB^,+,SR^         PO BOX 0306 , CORVALLIS ,                    OR 97339                  AAAAAAARR^        PO Box 03
,"ABB^,+,SR^         PO BOX 3248 , FRESNO ,                       CA 93650                  AAAAAAARR^        PO Box 32
,"ABB^,+,SR^         PO BOX 8400 , SALEM ,                        OR 97302                  AAAAAAARR^        PO Box 84
,"ABB^,+,SR^         PO BOX 0118 , SALEM ,                        OR 97303                  AAAAAAARR^        PO Box 01
,"A^+TD,+,SR^        2401 UTAH AVE S , SEATTLE ,                  WA 98134                  AAAAAAAARR^       2401 Utah
,"A^+TD,+,SR^        320 LIBERTY ST SE , SALEM ,                  OR 97000                  AAAAAAAARR^       320 Liber
,"A^+TD,+,SR^        2401 UTAH AVE S , SEATTLE ,                  WA 98134                  AAAAAAAARR^       2401 Utah
,"A^+TD,+,SR^        2401 UTAH AVE S , SEATTLE ,                  WA 98134                  AAAAAAAARR^       2401 Utah
,"A^+T,+,SR^         1510 HARTNELL AVE , REDDING ,                CA 96002                  AAAAAAARR^        1510 Hart
,"A^+T,+,SR^         1510 HARTNELL AVE , REDDING ,                CA 96002                  AAAAAAARR^        1510 Hart
,"A^>T,#^,+,SR^      325 9TH AVE , # 359819 , SEATTLE ,           WA 98104                  AAAAAAAAAARR^     325 9th A
,"A^M>T,+,SR^        2939 NE 155TH AVE , PORTLAND ,               OR 97230                  AAAAAAAARR^       2939 Ne 1
,"A^++T,+,SR^        911 SANTA FE AVE , FRESNO ,                  CA 93721                  AAAAAAAARR^       911 Santa
,"A^++T,+,SR^        911 SANTA FE AVE , FRESNO ,                  CA 93721                  AAAAAAAARR^       911 Santa
,"A^M>T,+,SR^        2939 NE 155TH AVE , PORTLAND ,               OR 97230                  AAAAAAAARR^       2939 Ne 1
,"A^>T,#^,+,SR^      325 9TH AVE , # 359819 , SEATTLE ,           WA 98104                  AAAAAAAAAARR^     325 9th A
,"A^++MM,UI,+,SR^    1705 SALEM INDUSTRIAL DR NE , STE B , SALEM , OR 97301                 AAAAAAAAAAAARR^   1705 Sale
,"A^+T,U>,++,SR^     303 POTRERO ST , STE 40C ,                   SANTA CRUZ , CA 95060     AAAAAAAARRRRR^    303 Potre
,"A^D+T,F,SR^        200 N MONROE ST , EUGENE ,                   OR 97402                  AAAAAAAARR^       200 N Mon
,"A^DFTM,+,SR^       1115 SE CRYSTAL LAKE DR , CORVALLIS ,        OR 97333                  AAAAAAAAARR^      1115 Se C
,"A^FM,F,SR^         2895 CHAD DR , EUGENE ,                      OR 97408                  AAAAAAARR^        2895 Chad
,"A^M+T,+,SR^        8435 NE GLISAN ST , PORTLAND ,               OR 97220                  AAAAAAAARR^       8435 Ne G
,"A^+M,U^,++,SR^     2333 PORTOLA DR , APT 54 ,                   SANTA CRUZ , CA 95062     AAAAAAAARRRRR^    2333 Port
,"A^D+M,+,SR^        1263 SE CENTERPOINTE DR , CORVALLIS ,        OR 97333                  AAAAAAAARR^       1263 Se C
,"A^>T,+,SR^         980 9TH ST ,                                 SACRAMENTO , CA 95814     AAAAARRRR^        980 9th S
,"A^+TD,U^,+,SR^     1414 DEXTER AVE N , STE 306 , SEATTLE ,      WA 98109                  AAAAAAAAAAARR^    1414 Dext
,"A^+M,U^,+,SR^      410 HEMSTED DR , STE 220 , REDDING ,         CA 96002                  AAAAAAAAAARR^     410 Hemst
,"A^+T,U^,+,SR^      455 CAPITOL MALL , STE 700 ,                 SACRAMENTO , CA 95814     AAAAAAAARRRR^     455 Capit
,"A^+T,F,SR^         3314 PALACE ST , EUGENE ,                    OR 97404                  AAAAAAARR^        3314 Pala
,"A^UR^              1 PIER                                       94111                     AAAR^             1 Pier
```

Figure 2-58   Create j04_INVCC_XXPREP_INPUT_PATTERN 15/38


NAB_CONTACT_INFO_WORK_XXPREP_ADDRESS_DOMAIN.rpt - WordPad
File Edit View Insert Format Help

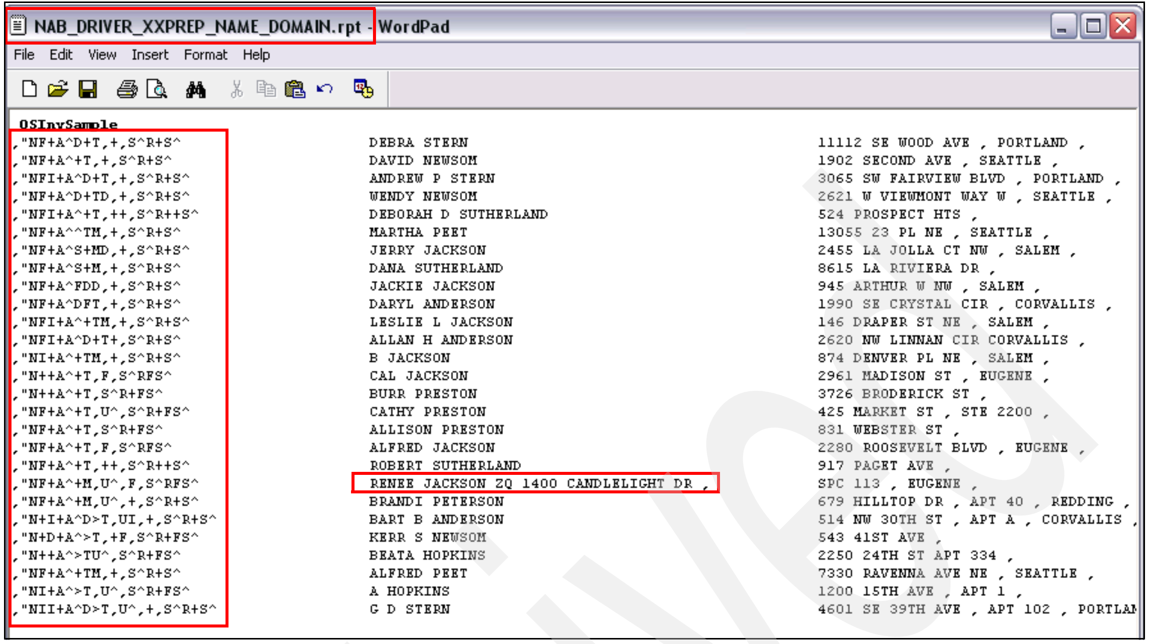
```
AAAAAAAARR^        1744 E Nees Ave, Fresno, CA                       (559) 298-0688","6.00000000000000000E+00","1.6216217E+
AAAAAAAARR^        2470 S Cherry Ave, Fresno, CA                     (559) 257-8000","6.00000000000000000E+00","1.6216217E+
AAAAAAAARR^        5285 Se Mallard Way, Portland, OR                 (503) 652-9037","6.00000000000000000E+00","1.6216217E+
AAAAAAAARR^        3421 Se Salmon St, Portland, OR                   (503) 916-6227","6.00000000000000000E+00","1.6216217E+
AAAAAAARR^         PO Box 0306, Corvallis, OR                        (503) 623-2797","5.00000000000000000E+00","1.3513514E+
AAAAAAARR^         PO Box 3248, Fresno, CA                           (559) 448-6775","5.00000000000000000E+00","1.3513514E+
AAAAAAARR^         PO Box 8400, Salem, OR                            (503) 363-3934","5.00000000000000000E+00","1.3513514E+
AAAAAAARR^         PO Box 0118, Salem, OR                            (800) 844-4974","5.00000000000000000E+00","1.3513514E+
AAAAAAAARR^        2401 Utah Ave S, Seattle, WA                      (206) 447-1566","4.00000000000000000E+00","1.0810811E+
AAAAAAAARR^        320 Liberty St Se, Salem, OR                      (503) 399-8695","4.00000000000000000E+00","1.0810811E+
AAAAAAAARR^        2401 Utah Ave S, Seattle, WA                      (206) 447-1875","4.00000000000000000E+00","1.0810811E+
AAAAAAAARR^        2401 Utah Ave S, Seattle, WA                      (206) 447-1566","4.00000000000000000E+00","1.0810811E+
AAAAAAARR^         1510 Hartnell Ave, Redding, CA                    (530) 224-6823","2.00000000000000000E+00","5.4054055E+
AAAAAAARR^         1510 Hartnell Ave, Redding, CA                    (415) 826-1031","2.00000000000000000E+00","5.4054055E+
AAAAAAAAAARR^      325 9th Ave, # 359819, Seattle, WA                (206) 731-3465","2.00000000000000000E+00","5.4054055E+
AAAAAAAARR^        2939 Ne 155th Ave, Portland, OR                   (503) 256-1300","2.00000000000000000E+00","5.4054055E+
AAAAAAAARR^        911 Santa Fe Ave, Fresno, CA                      (559) 453-5900","2.00000000000000000E+00","5.4054055E+
AAAAAAAARR^        911 Santa Fe Ave, Fresno, CA                      (415) 664-0983","2.00000000000000000E+00","5.4054055E+
AAAAAAAARR^        2939 Ne 155th Ave, Portland, OR                   (503) 256-1300","2.00000000000000000E+00","5.4054055E+
AAAAAAAAAARR^      325 9th Ave, # 359819, Seattle, WA                (206) 731-3465","2.00000000000000000E+00","5.4054055E+
AAAAAAAAAAAARR^    1705 Salem Industrial Dr Ne, Ste B, Salem, OR     (503) 364-2989","1.00000000000000000E+00","2.7027028E+
AAAAAAAARRRRR^     303 Potrero St, Ste 40C, Santa Cruz, CA           (831) 459-0445","1.00000000000000000E+00","2.7027028E+
AAAAAAAARR^        200 N Monroe St, Eugene, OR                       (541) 687-3141","1.00000000000000000E+00","2.7027028E+
AAAAAAAAARR^       1115 Se Crystal Lake Dr, Corvallis, OR            (541) 768-6218","1.00000000000000000E+00","2.7027028E+
AAAAAAARR^         2895 Chad Dr, Eugene, OR                          (800) 935-4600","1.00000000000000000E+00","2.7027028E+
AAAAAAAARR^        8435 Ne Glisan St, Portland, OR                   (503) 251-6455","1.00000000000000000E+00","2.7027028E+
AAAAAAAARRRRR^     2333 Portola Dr, Apt 54, Santa Cruz, CA           (877) 379-2096","1.00000000000000000E+00","2.7027028E+
AAAAAAAARR^        1263 Se Centerpointe Dr, Corvallis, OR            (541) 752-3917","1.00000000000000000E+00","2.7027028E+
AAAAARRRR^         980 9th St, Sacramento, CA                        (916) 558-6400","1.00000000000000000E+00","2.7027028E+
AAAAAAAAAAARR^     1414 Dexter Ave N, Ste 306, Seattle, WA           (206) 282-1578","1.00000000000000000E+00","2.7027028E+
AAAAAAAAAARR^      410 Hemsted Dr, Ste 220, Redding, CA              (530) 224-4933","1.00000000000000000E+00","2.7027028E+
AAAAAAAARRRR^      455 Capitol Mall, Ste 700, Sacramento, CA         (916) 487-4288","1.00000000000000000E+00","2.7027028E+
AAAAAAARR^         3314 Palace St, Eugene, OR                        (541) 687-0488","1.00000000000000000E+00","2.7027028E+
AAAR^              1 Pier                                            (415) 395-9000","1.00000000000000000E+00","2.7027028E+
```

Figure 2-59 Create j04_INVCC_XXPREP_INPUT_PATTERN 16/38


j04_INVCC_XXPREP_INPUT_PATTERN..NAB_DRIVER_XXPREP_NAME_DOMAIN_REPORT.NAMEIN - Data Browser

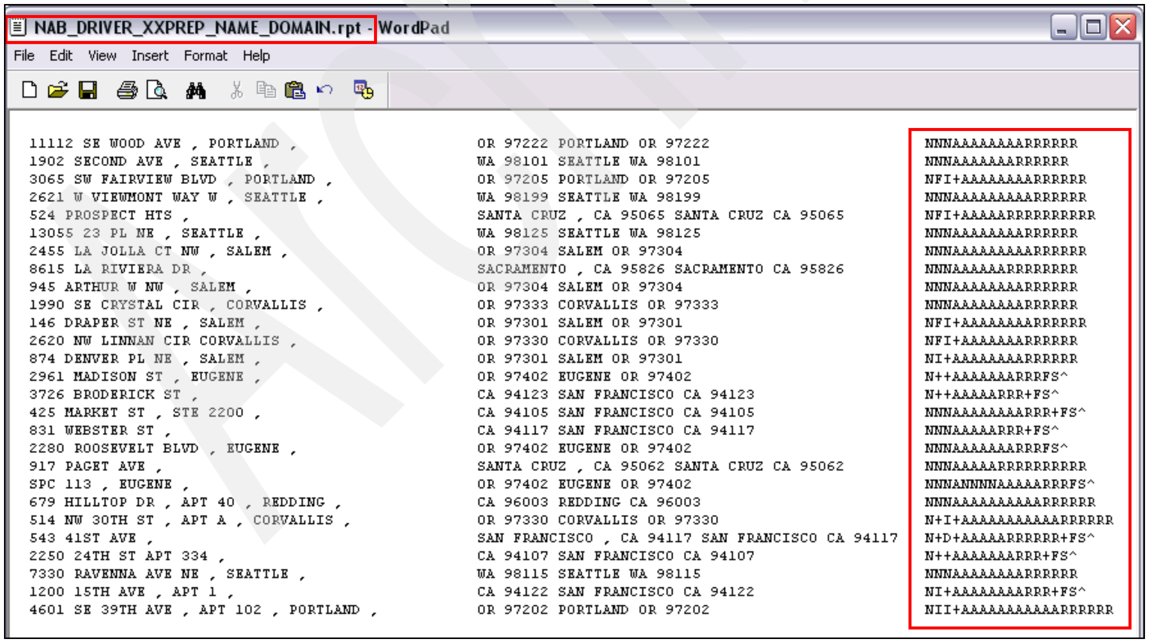
qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
InputPattern_USPREP+NameDomain_USPREP+AddressD	NF+A^D+T,+,S	NF+A^D+T,+,S^R+S^	5	14.7059
InputPattern_USPREP+NameDomain_USPREP+AddressD	NF+A^+T,+,S^	NF+A^+T,+,S^R+S^	3	8.82353
InputPattern_USPREP+NameDomain_USPREP+AddressD	NFI+A^D+T,+,	NFI+A^D+T,+,S^R+S^	2	5.88235
InputPattern_USPREP+NameDomain_USPREP+AddressD	NF+A^D+TD,+,	NF+A^D+TD,+,S^R+S^	1	2.94118
InputPattern_USPREP+NameDomain_USPREP+AddressD	NFI+A^+T,++,	NFI+A^+T,++,S^R++S^	1	2.94118
InputPattern_USPREP+NameDomain_USPREP+AddressD	NF+A^^TM,+,S	NF+A^^TM,+,S^R+S^	1	2.94118
InputPattern_USPREP+NameDomain_USPREP+AddressD	NF+A^S+MD,+,	NF+A^S+MD,+,S^R+S^	1	2.94118
InputPattern_USPREP+NameDomain_USPREP+AddressD	NF+A^S+M,+,S	NF+A^S+M,+,S^R+S^	1	2.94118
InputPattern_USPREP+NameDomain_USPREP+AddressD	NF+A^FDD,+,S	NF+A^FDD,+,S^R+S^	1	2.94118
InputPattern_USPREP+NameDomain_USPREP+AddressD	NF+A^DFT,+,S	NF+A^DFT,+,S^R+S^	1	2.94118
InputPattern_USPREP+NameDomain_USPREP+AddressD	NFI+A^+TM,+,	NFI+A^+TM,+,S^R+S^	1	2.94118
InputPattern_USPREP+NameDomain_USPREP+AddressD	NFI+A^D+T+,S	NFI+A^D+T+,S^R+S^	1	2.94118
InputPattern_USPREP+NameDomain_USPREP+AddressD	NI+A^+TM,+,S	NI+A^+TM,+,S^R+S^	1	2.94118
InputPattern_USPREP+NameDomain_USPREP+AddressD	N++A^+T,F,S^	N++A^+T,F,S^RFS^	1	2.94118
InputPattern_USPREP+NameDomain_USPREP+AddressD	N++A^+T,S^R+	N++A^+T,S^R+FS^	1	2.94118
InputPattern_USPREP+NameDomain_USPREP+AddressD	NF+A^+T,U^,S	NF+A^+T,U^,S^R+FS^	1	2.94118
InputPattern_USPREP+NameDomain_USPREP+AddressD	NF+A^+T,S^R+	NF+A^+T,S^R+FS^	1	2.94118
InputPattern_USPREP+NameDomain_USPREP+AddressD	NF+A^+T,F,S^	NF+A^+T,F,S^RFS^	1	2.94118
InputPattern_USPREP+NameDomain_USPREP+AddressD	NF+A^+T,++,S	NF+A^+T,++,S^R++S^	1	2.94118
InputPattern_USPREP+NameDomain_USPREP+AddressD	NF+A^+M,U^,F	NF+A^+M,U^,F,S^RFS^	1	2.94118
InputPattern_USPREP+NameDomain_USPREP+AddressD	NF+A^+M,U^,+	NF+A^+M,U^,+,S^R+S^	1	2.94118
InputPattern_USPREP+NameDomain_USPREP+AddressD	N+I+A^D>T,UI	N+I+A^D>T,UI,+,S^R+S^	1	2.94118
InputPattern_USPREP+NameDomain_USPREP+AddressD	N+D+A^>T,+F,	N+D+A^>T,+F,S^R+FS^	1	2.94118
InputPattern_USPREP+NameDomain_USPREP+AddressD	N++A^>TU^,S^	N++A^>TU^,S^R+FS^	1	2.94118
InputPattern_USPREP+NameDomain_USPREP+AddressD	NF+A^+TM,+,S	NF+A^+TM,+,S^R+S^	1	2.94118
InputPattern_USPREP+NameDomain_USPREP+AddressD	NI+A^>T,U^,S	NI+A^>T,U^,S^R+FS^	1	2.94118
InputPattern_USPREP+NameDomain_USPREP+AddressD	NII+A^D>T,U^	NII+A^D>T,U^,+,S^R+S^	1	2.94118

Figure 2-60 Create j04_INVCC_XXPREP_INPUT_PATTERN 17/38



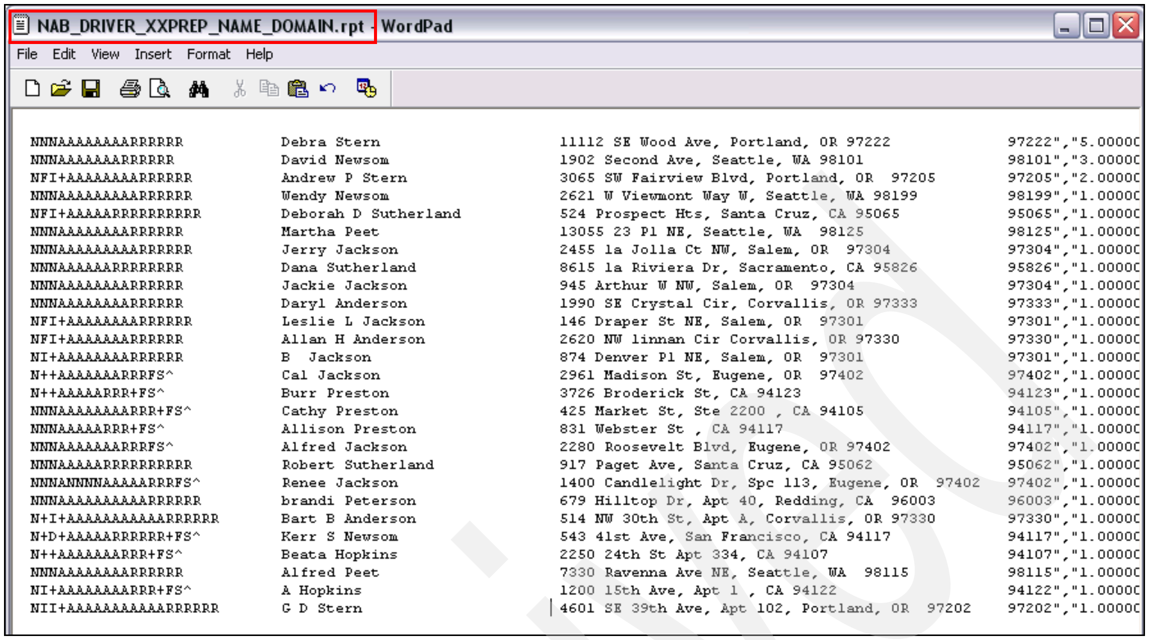
```
OSInvSample
,"NF+A^D+T,+,S^R+S^          DEBRA STERN                               11112 SE WOOD AVE , PORTLAND ,
,"NF+A^+T,+,S^R+S^           DAVID NEWSOM                              1902 SECOND AVE , SEATTLE ,
,"NFI+A^D+T,+,S^R+S^         ANDREW P STERN                            3065 SW FAIRVIEW BLVD , PORTLAND ,
,"NF+A^D+TD,+,S^R+S^         WENDY NEWSOM                              2621 W VIEWMONT WAY W , SEATTLE ,
,"NFI+A^+T,++,S^R++S^        DEBORAH D SUTHERLAND                      524 PROSPECT HTS ,
,"NF+A^^TM,+,S^R+S^          MARTHA PEET                               13055 23 PL NE , SEATTLE ,
,"NF+A^S+MD,+,S^R+S^         JERRY JACKSON                             2455 LA JOLLA CT NW , SALEM ,
,"NF+A^S+M,+,S^R+S^          DANA SUTHERLAND                           8615 LA RIVIERA DR ,
,"NF+A^FDD,+,S^R+S^          JACKIE JACKSON                            945 ARTHUR W NW , SALEM ,
,"NF+A^DFT,+,S^R+S^          DARYL ANDERSON                            1990 SE CRYSTAL CIR , CORVALLIS ,
,"NFI+A^+TM,+,S^R+S^         LESLIE L JACKSON                          146 DRAPER ST NE , SALEM ,
,"NFI+A^D+T+,S^R+S^          ALLAN H ANDERSON                          2620 NW LINNAN CIR CORVALLIS ,
,"NI+A^+TM,+,S^R+S^          B JACKSON                                 874 DENVER PL NE , SALEM ,
,"N++A^+T,F,S^RFS^           CAL JACKSON                               2961 MADISON ST , EUGENE ,
,"N++A^+T,S^R+FS^            BURR PRESTON                              3726 BRODERICK ST ,
,"NF+A^+T,U^,S^R+FS^         CATHY PRESTON                             425 MARKET ST , STE 2200 ,
,"NF+A^+T,S^R+FS^            ALLISON PRESTON                           831 WEBSTER ST ,
,"NF+A^+T,F,S^RFS^           ALFRED JACKSON                            2280 ROOSEVELT BLVD , EUGENE ,
,"NF+A^+T,++,S^R++S^         ROBERT SUTHERLAND                         917 PAGET AVE ,
,"NF+A^+M,U^,F,S^RFS^        RENEE JACKSON ZQ 1400 CANDLELIGHT DR ,    SPC 113 , EUGENE ,
,"NF+A^+M,U^,+,S^R+S^        BRANDI PETERSON                           679 HILLTOP DR , APT 40 , REDDING ,
,"N+I+A^D>T,UI,+,S^R+S^      BART B ANDERSON                           514 NW 30TH ST , APT A , CORVALLIS ,
,"N+D+A^>T,+F,S^R+FS^        KERR S NEWSOM                             543 41ST AVE ,
,"N++A^>TU^,S^R+FS^          BEATA HOPKINS                             2250 24TH ST APT 334 ,
,"NF+A^+TM,+,S^R+S^          ALFRED PEET                               7330 RAVENNA AVE NE , SEATTLE ,
,"NI+A^>T,U^,S^R+FS^         A HOPKINS                                 1200 15TH AVE , APT 1 ,
,"NII+A^D>T,U^,+,S^R+S^      G D STERN                                 4601 SE 39TH AVE , APT 102 , PORTLAN
```

Figure 2-61 Create j04_INVCC_XXPREP_INPUT_PATTERN 18/38



```
11112 SE WOOD AVE , PORTLAND ,            OR 97222 PORTLAND OR 97222                   NNNAAAAAAAARRRRRR
1902 SECOND AVE , SEATTLE ,               WA 98101 SEATTLE WA 98101                    NNNAAAAAAAARRRRR
3065 SW FAIRVIEW BLVD , PORTLAND ,        OR 97205 PORTLAND OR 97205                   NFI+AAAAAAAARRRRRR
2621 W VIEWMONT WAY W , SEATTLE ,         WA 98199 SEATTLE WA 98199                    NNNAAAAAAAAARRRRRR
524 PROSPECT HTS ,                        SANTA CRUZ , CA 95065 SANTA CRUZ CA 95065    NFI+AAAAARRRRRRRRRR
13055 23 PL NE , SEATTLE ,                WA 98125 SEATTLE WA 98125                    NNNAAAAAAAARRRRRR
2455 LA JOLLA CT NW , SALEM ,             OR 97304 SALEM OR 97304                      NNNAAAAAAAAARRRRRR
8615 LA RIVIERA DR ,                      SACRAMENTO , CA 95826 SACRAMENTO CA 95826    NNNAAAAAAARRRRRRR
945 ARTHUR W NW , SALEM ,                 OR 97304 SALEM OR 97304                      NNNAAAAAAAARRRRRR
1990 SE CRYSTAL CIR , CORVALLIS ,         OR 97333 CORVALLIS OR 97333                  NNNAAAAAAAARRRRRR
146 DRAPER ST NE , SALEM ,                OR 97301 SALEM OR 97301                      NFI+AAAAAAAARRRRRR
2620 NW LINNAN CIR CORVALLIS ,            OR 97330 CORVALLIS OR 97330                  NFI+AAAAAAAARRRRR
874 DENVER PL NE , SALEM ,                OR 97301 SALEM OR 97301                      NI+AAAAAAAARRRRRR
2961 MADISON ST , EUGENE ,                OR 97402 EUGENE OR 97402                     N++AAAAAAARRRFS^
3726 BRODERICK ST ,                       CA 94123 SAN FRANCISCO CA 94123              N++AAAAARRR+FS^
425 MARKET ST , STE 2200 ,                CA 94105 SAN FRANCISCO CA 94105              NNNAAAAAAAARRR+FS^
831 WEBSTER ST ,                          CA 94117 SAN FRANCISCO CA 94117              NNNAAAAARRR+FS^
2280 ROOSEVELT BLVD , EUGENE ,            OR 97402 EUGENE OR 97402                     NNNAAAAAAARRRFS^
917 PAGET AVE ,                           SANTA CRUZ , CA 95062 SANTA CRUZ CA 95062    NNNAAAAARRRRRRRRRR
SPC 113 , EUGENE ,                        OR 97402 EUGENE OR 97402                     NNNANNNNAAAAARRRFS^
679 HILLTOP DR , APT 40 , REDDING ,       CA 96003 REDDING CA 96003                    NNNAAAAAAAAAARRRRRR
514 NW 30TH ST , APT A , CORVALLIS ,      OR 97330 CORVALLIS OR 97330                  N+I+AAAAAAAAAAARRRRRR
543 41ST AVE ,                            SAN FRANCISCO , CA 94117 SAN FRANCISCO CA 94117  N+D+AAAAARRRRRR+FS^
2250 24TH ST APT 334 ,                    CA 94107 SAN FRANCISCO CA 94107              N++AAAAAAARRR+FS^
7330 RAVENNA AVE NE , SEATTLE ,           WA 98115 SEATTLE WA 98115                    NNNAAAAAAAARRRRRR
1200 15TH AVE , APT 1 ,                   CA 94122 SAN FRANCISCO CA 94122              NI+AAAAAAAARRR+FS^
4601 SE 39TH AVE , APT 102 , PORTLAND ,   OR 97202 PORTLAND OR 97202                   NII+AAAAAAAAAAARRRRRR
```

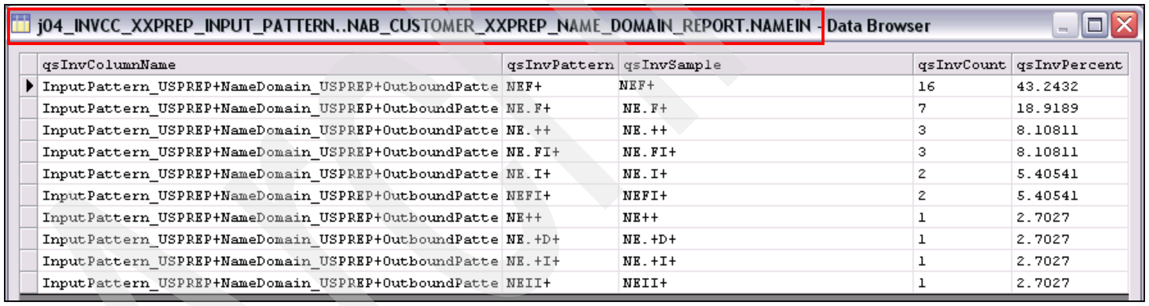
Figure 2-62 Create j04_INVCC_XXPREP_INPUT_PATTERN 19/38



NAB_DRIVER_XXPREP_NAME_DOMAIN.rpt - WordPad

```
NNNAAAAAAAAARRRRRR        Debra Stern               11112 SE Wood Ave, Portland, OR 97222            97222","5.0000C
NNNAAAAAAAARRRRRR         David Newsom              1902 Second Ave, Seattle, WA 98101               98101","3.0000C
NFI+AAAAAAAARRRRRR        Andrew P Stern            3065 SW Fairview Blvd, Portland, OR  97205       97205","2.0000C
NNNAAAAAAAAAARRRRRR       Wendy Newsom              2621 W Viewmont Way W, Seattle, WA 98199         98199","1.0000C
NFI+AAAAARRRRRRRRRRR      Deborah D Sutherland      524 Prospect Hts, Santa Cruz, CA 95065           95065","1.0000C
NNNAAAAAAAAARRRRRR        Martha Peet               13055 23 Pl NE, Seattle, WA  98125               98125","1.0000C
NNNAAAAAAAAAARRRRRR       Jerry Jackson             2455 la Jolla Ct NW, Salem, OR  97304            97304","1.0000C
NNNAAAAAARRRRRRRRR        Dana Sutherland           8615 la Riviera Dr, Sacramento, CA 95826         95826","1.0000C
NNNAAAAAAAAARRRRRR        Jackie Jackson            945 Arthur W NW, Salem, OR  97304                97304","1.0000C
NNNAAAAAAAAARRRRRR        Daryl Anderson            1990 SE Crystal Cir, Corvallis, OR 97333         97333","1.0000C
NFI+AAAAAAAARRRRRR        Leslie L Jackson          146 Draper St NE, Salem, OR  97301               97301","1.0000C
NFI+AAAAAAARRRRRR         Allan H Anderson          2620 NW linnan Cir Corvallis, OR 97330           97330","1.0000C
NI+AAAAAAAARRRRRR         B  Jackson                874 Denver Pl NE, Salem, OR  97301               97301","1.0000C
N++AAAAAAARRRFS^          Cal Jackson               2961 Madison St, Eugene, OR  97402               97402","1.0000C
N++AAAAARRR+FS^           Burr Preston              3726 Broderick St, CA 94123                      94123","1.0000C
NNNAAAAAAAARRR+FS^        Cathy Preston             425 Market St, Ste 2200 , CA 94105               94105","1.0000C
NNNAAAAARRR+FS^           Allison Preston           831 Webster St , CA 94117                        94117","1.0000C
NNNAAAAAAAARRRFS^         Alfred Jackson            2280 Roosevelt Blvd, Eugene, OR 97402            97402","1.0000C
NNNAAAAARRRRRRRRRR        Robert Sutherland         917 Paget Ave, Santa Cruz, CA 95062              95062","1.0000C
NNNANNNNAAAAARRRFS^       Renee Jackson             1400 Candlelight Dr, Spc 113, Eugene, OR  97402  97402","1.0000C
NNNAAAAAAAAAAARRRRRR      brandi Peterson           679 Hilltop Dr, Apt 40, Redding, CA  96003       96003","1.0000C
N+I+AAAAAAAAAAARRRRRR     Bart B Anderson           514 NW 30th St, Apt A, Corvallis, OR 97330       97330","1.0000C
N+D+AAAAARRRRRR+FS^       Kerr S Newsom             543 41st Ave, San Francisco, CA 94117            94117","1.0000C
N++AAAAAAARRR+FS^         Beata Hopkins             2250 24th St Apt 334, CA 94107                   94107","1.0000C
NNNAAAAAAAARRRRRR         Alfred Peet               7330 Ravenna Ave NE, Seattle, WA  98115          98115","1.0000C
NI+AAAAAAARRR+FS^         A Hopkins                 1200 15th Ave, Apt 1 , CA 94122                  94122","1.0000C
NII+AAAAAAAAAAARRRRRR     G D Stern                 4601 SE 39th Ave, Apt 102, Portland, OR  97202   97202","1.0000C
```

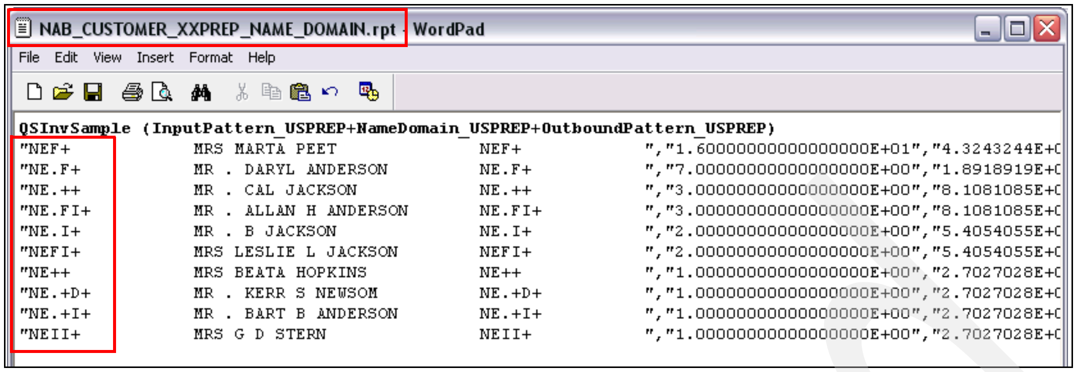
Figure 2-63   Create j04_INVCC_XXPREP_INPUT_PATTERN 20/38



j04_INVCC_XXPREP_INPUT_PATTERN..NAB_CUSTOMER_XXPREP_NAME_DOMAIN_REPORT.NAMEIN - Data Browser

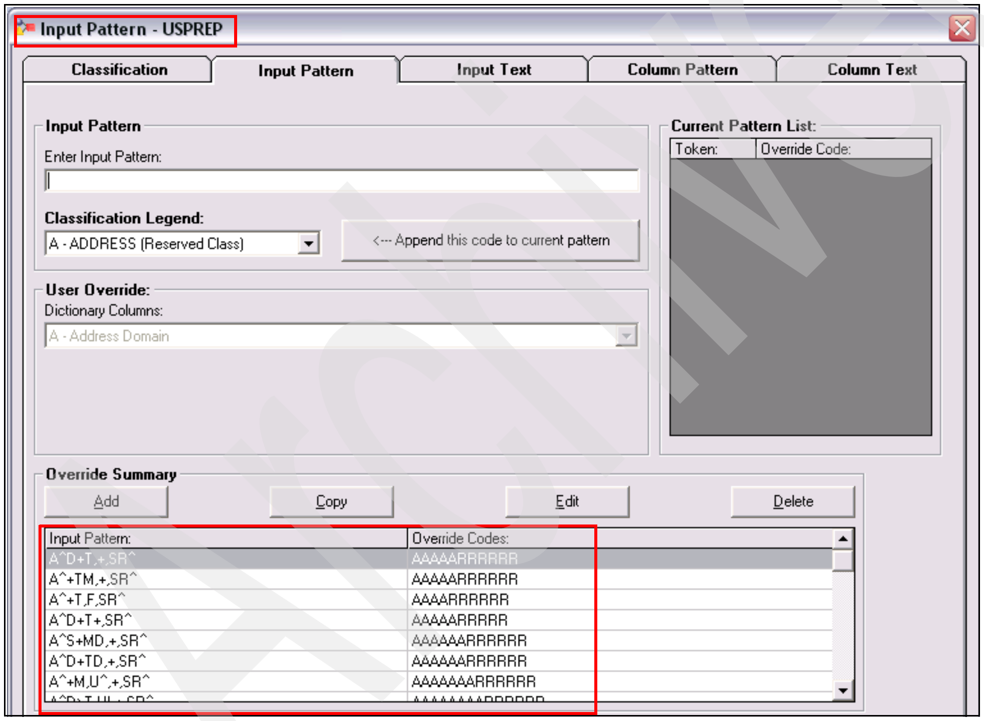
qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
InputPattern_USPREP+NameDomain_USPREP+OutboundPatte	NEF+	NEF+	16	43.2432
InputPattern_USPREP+NameDomain_USPREP+OutboundPatte	NE.F+	NE.F+	7	18.9189
InputPattern_USPREP+NameDomain_USPREP+OutboundPatte	NE.++	NE.++	3	8.10811
InputPattern_USPREP+NameDomain_USPREP+OutboundPatte	NE.FI+	NE.FI+	3	8.10811
InputPattern_USPREP+NameDomain_USPREP+OutboundPatte	NE.I+	NE.I+	2	5.40541
InputPattern_USPREP+NameDomain_USPREP+OutboundPatte	NEFI+	NEFI+	2	5.40541
InputPattern_USPREP+NameDomain_USPREP+OutboundPatte	NE++	NE++	1	2.7027
InputPattern_USPREP+NameDomain_USPREP+OutboundPatte	NE.+D+	NE.+D+	1	2.7027
InputPattern_USPREP+NameDomain_USPREP+OutboundPatte	NE.+I+	NE.+I+	1	2.7027
InputPattern_USPREP+NameDomain_USPREP+OutboundPatte	NEII+	NEII+	1	2.7027

Figure 2-64   Create j04_INVCC_XXPREP_INPUT_PATTERN 21/38


NAB_CUSTOMER_XXPREP_NAME_DOMAIN.rpt - WordPad
File Edit View Insert Format Help

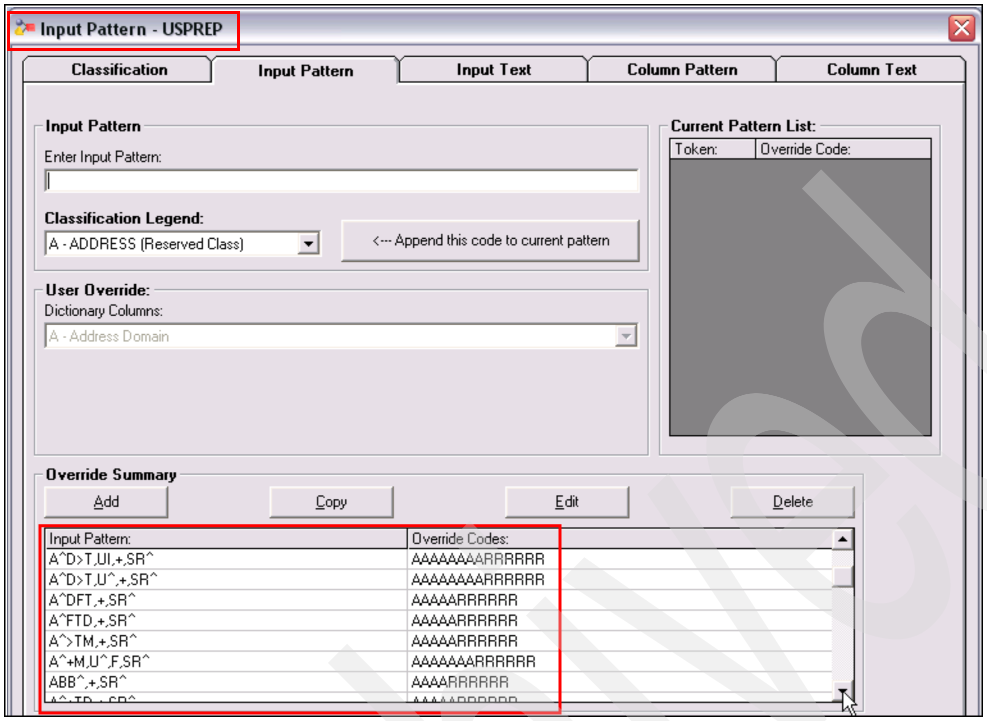
```
QSInvSample (InputPattern_USPREP+NameDomain_USPREP+OutboundPattern_USPREP)
"NEF+           MRS MARTA PEET           NEF+           ","1.6000000000000000E+01","4.3243244E+0
"NE.F+          MR . DARYL ANDERSON      NE.F+          ","7.0000000000000000E+00","1.8918919E+0
"NE.++          MR . CAL JACKSON         NE.++          ","3.0000000000000000E+00","8.1081085E+0
"NE.FI+         MR . ALLAN H ANDERSON    NE.FI+         ","3.0000000000000000E+00","8.1081085E+0
"NE.I+          MR . B JACKSON           NE.I+          ","2.0000000000000000E+00","5.4054055E+0
"NEFI+          MRS LESLIE L JACKSON     NEFI+          ","2.0000000000000000E+00","5.4054055E+0
"NE++           MRS BEATA HOPKINS        NE++           ","1.0000000000000000E+00","2.7027028E+0
"NE.+D+         MR . KERR S NEWSOM       NE.+D+         ","1.0000000000000000E+00","2.7027028E+0
"NE.+I+         MR . BART B ANDERSON     NE.+I+         ","1.0000000000000000E+00","2.7027028E+0
"NEII+          MRS G D STERN            NEII+          ","1.0000000000000000E+00","2.7027028E+0
```

Figure 2-65 Create j04_INVCC_XXPREP_INPUT_PATTERN 22/38



Input Pattern:	Override Codes:
A^D+T,+,SR^	AAAAARRRRRR
A^+TM,+,SR^	AAAAARRRRRR
A^+T,F,SR^	AAAARRRRRR
A^D+T+,SR^	AAAAARRRRR
A^S+MD,+,SR^	AAAAAARRRRRR
A^D+TD,+,SR^	AAAAAARRRRRR
A^+M,U^,+,SR^	AAAAAAARRRRRR

Figure 2-66 Create j04_INVCC_XXPREP_INPUT_PATTERN 23/38


Input Pattern - USPREP

Classification | Input Pattern | Input Text | Column Pattern | Column Text

Input Pattern

Enter Input Pattern:

Classification Legend: A - ADDRESS (Reserved Class)

<--- Append this code to current pattern

User Override:

Dictionary Columns: A - Address Domain

Current Pattern List:

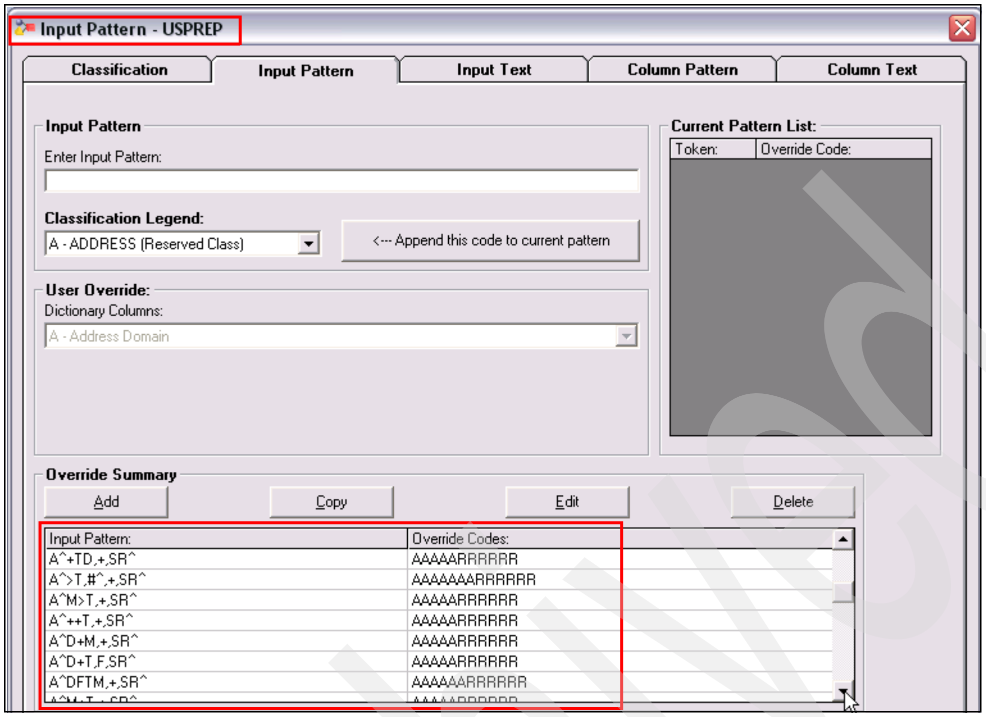
Token:	Override Code:

Override Summary

Add | Copy | Edit | Delete

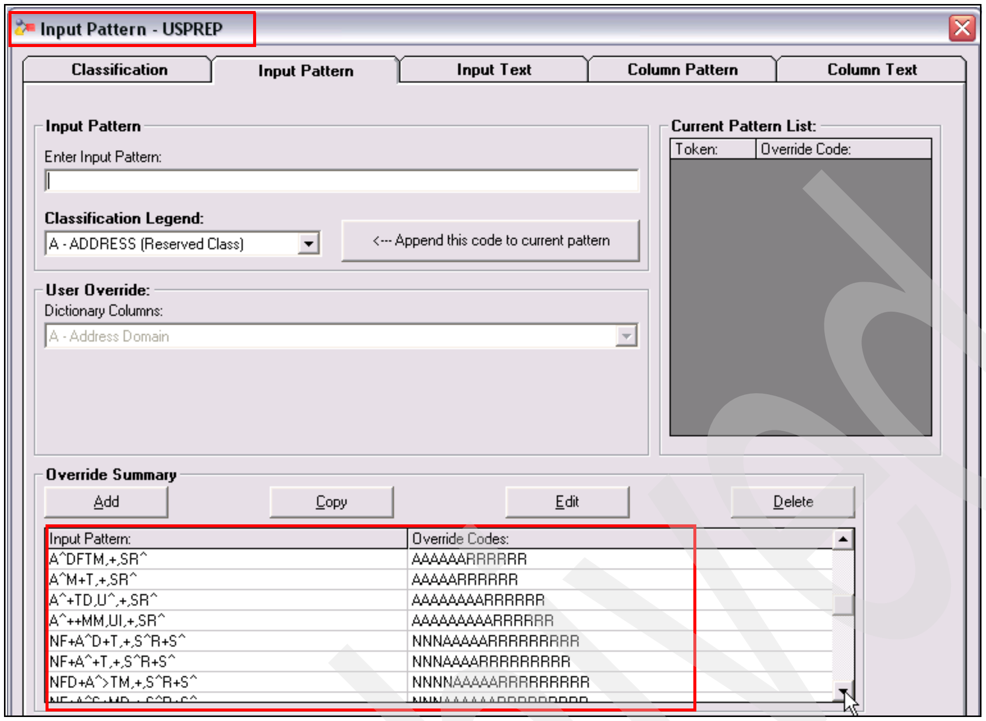
Input Pattern:	Override Codes:
A^D>T,UI,+,SR^	AAAAAAAARRRRRR
A^D>T,U^,+,SR^	AAAAAAAARRRRRR
A^DFT,+,SR^	AAAAARRRRRR
A^FTD,+,SR^	AAAAARRRRRR
A^>TM,+,SR^	AAAAARRRRRR
A^+M,U^,F,SR^	AAAAAAARRRRRR
ABB^,+,SR^	AAAARRRRRR

Figure 2-67 Create j04_INVCC_XXPREP_INPUT_PATTERN 24/38



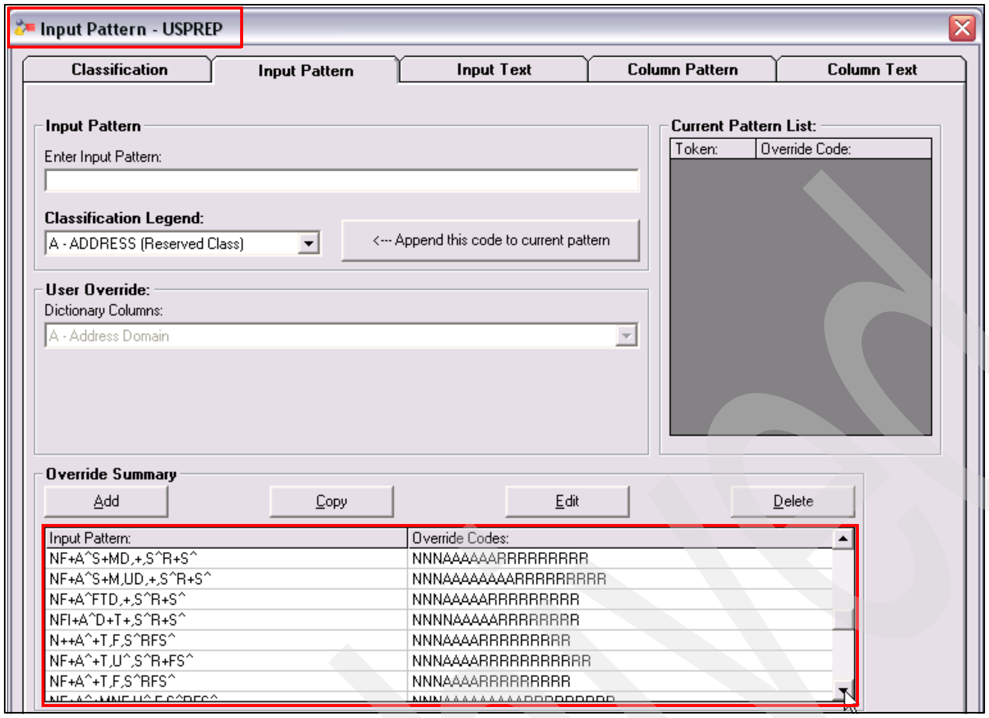
Input Pattern:	Override Codes:
A^+TD,+,SR^	AAAAARRRRRR
A^>T,#^,+,SR^	AAAAAAARRRRRR
A^M>T,+,SR^	AAAAARRRRRR
A^++T,+,SR^	AAAAARRRRRR
A^D+M,+,SR^	AAAAARRRRRR
A^D+T,F,SR^	AAAAARRRRRR
A^DFTM,+,SR^	AAAAAARRRRRR

Figure 2-68 Create j04_INVCC_XXPREP_INPUT_PATTERN 25/38



Input Pattern:	Override Codes:
A^DFTM,+,SR^	AAAAAARRRRRR
A^M+T,+,SR^	AAAAARRRRRR
A^+TD,U^,+,SR^	AAAAAAAARRRRRR
A^++MM,UI,+,SR^	AAAAAAAAARRRRRR
NF+A^D+T,+,S^R+S^	NNNAAAAARRRRRRRRR
NF+A^+T,+,S^R+S^	NNNAAAARRRRRRRRR
NFD+A^>TM,+,S^R+S^	NNNNAAAAARRRRRRRRR

Figure 2-69 Create j04_INVCC_XXPREP_INPUT_PATTERN 26/38


Input Pattern - USPREP

Classification | Input Pattern | Input Text | Column Pattern | Column Text

Input Pattern

Enter Input Pattern:

Classification Legend: A - ADDRESS (Reserved Class)

<--- Append this code to current pattern

User Override:

Dictionary Columns: A - Address Domain

Current Pattern List:

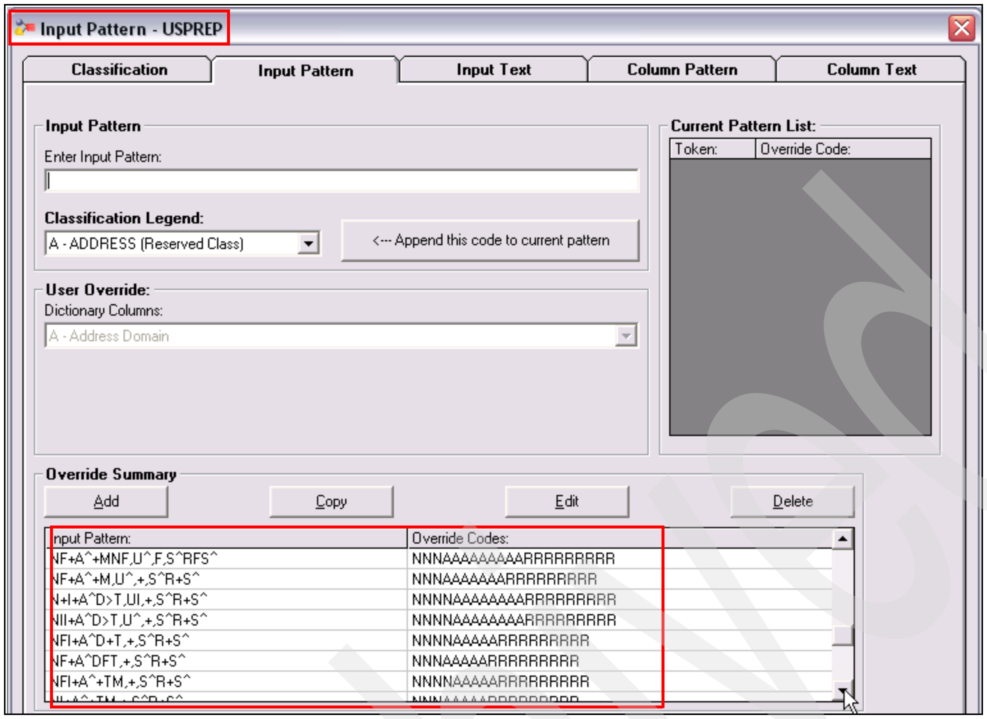
Token:	Override Code:

Override Summary

Add | Copy | Edit | Delete

Input Pattern:	Override Codes:
NF+A^S+MD,+,S^R+S^	NNNAAAAAARRRRRRRRR
NF+A^S+M,UD,+,S^R+S^	NNNAAAAAAAARRRRRRRRR
NF+A^FTD,+,S^R+S^	NNNAAAAARRRRRRRRR
NFI+A^D+T+,S^R+S^	NNNNAAAAARRRRRRRR
N++A^+T,F,S^RFS^	NNNAAAARRRRRRRRR
NF+A^+T,U^,S^R+FS^	NNNAAAARRRRRRRRRRR
NF+A^+T,F,S^RFS^	NNNAAAARRRRRRRRR

Figure 2-70 Create j04_INVCC_XXPREP_INPUT_PATTERN 27/38



Input Pattern

Enter Input Pattern:

Classification Legend: A - ADDRESS (Reserved Class)

User Override:

Dictionary Columns: A - Address Domain

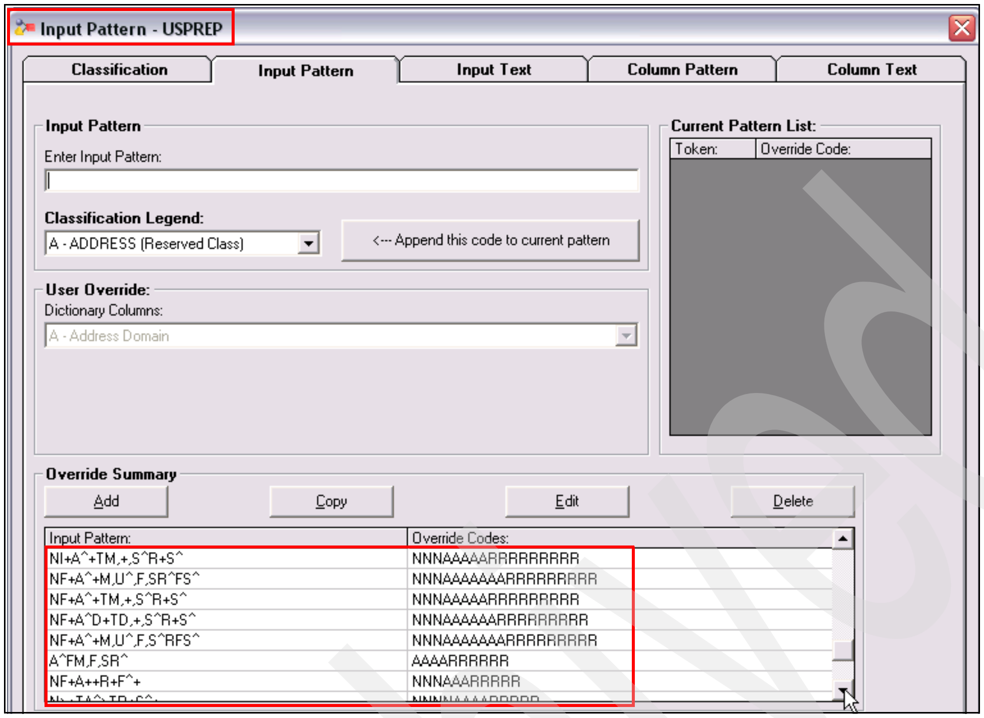
Current Pattern List:

Token:	Override Code:

Override Summary

nput Pattern:	Override Codes:
NF+A^+MNF,U^,F,S^RFS^	NNNAAAAAAAAARRRRRRRRR
NF+A^+M,U^,+,S^R+S^	NNNAAAAAAAARRRRRRRRR
N+I+A^D>T,UI,+,S^R+S^	NNNNAAAAAAAAARRRRRRRRR
NII+A^D>T,U^,+,S^R+S^	NNNNAAAAAAAAARRRRRRRRR
NFI+A^D+T,+,S^R+S^	NNNNAAAAARRRRRRRRR
NF+A^DFT,+,S^R+S^	NNNAAAAARRRRRRRRR
NFI+A^+TM,+,S^R+S^	NNNNAAAAARRRRRRRRR
[illegible]	[illegible]

Figure 2-71 Create j04_INVCC_XXPREP_INPUT_PATTERN 28/38


Input Pattern - USPREP

Input Pattern

Enter Input Pattern:

Classification Legend: A - ADDRESS (Reserved Class)

User Override:

Dictionary Columns: A - Address Domain

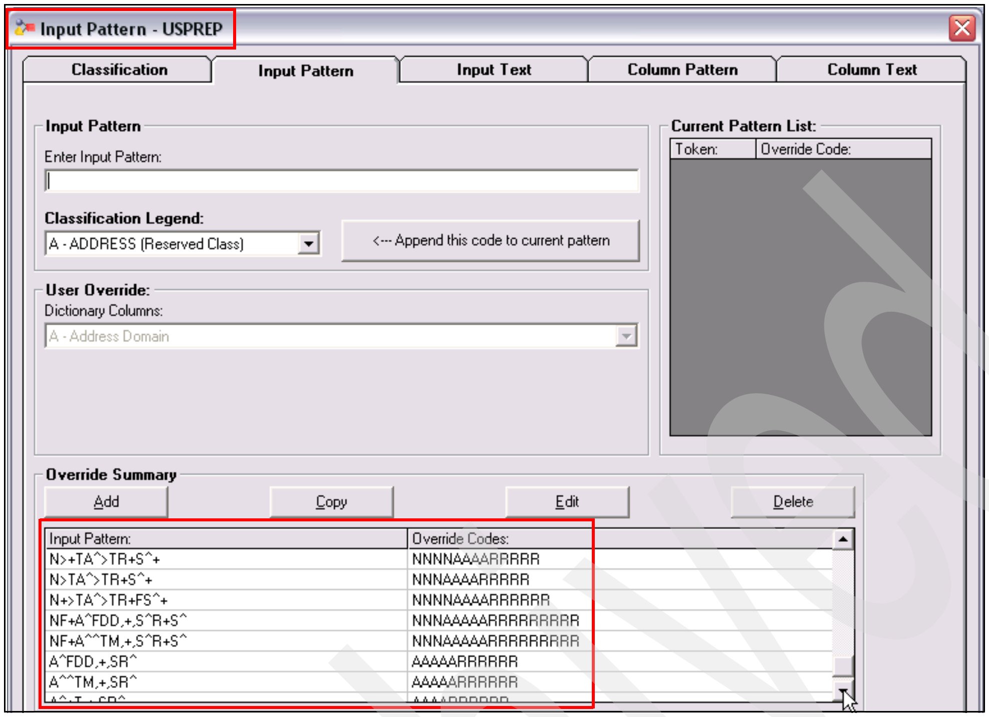
Current Pattern List:

Token:	Override Code:

Override Summary

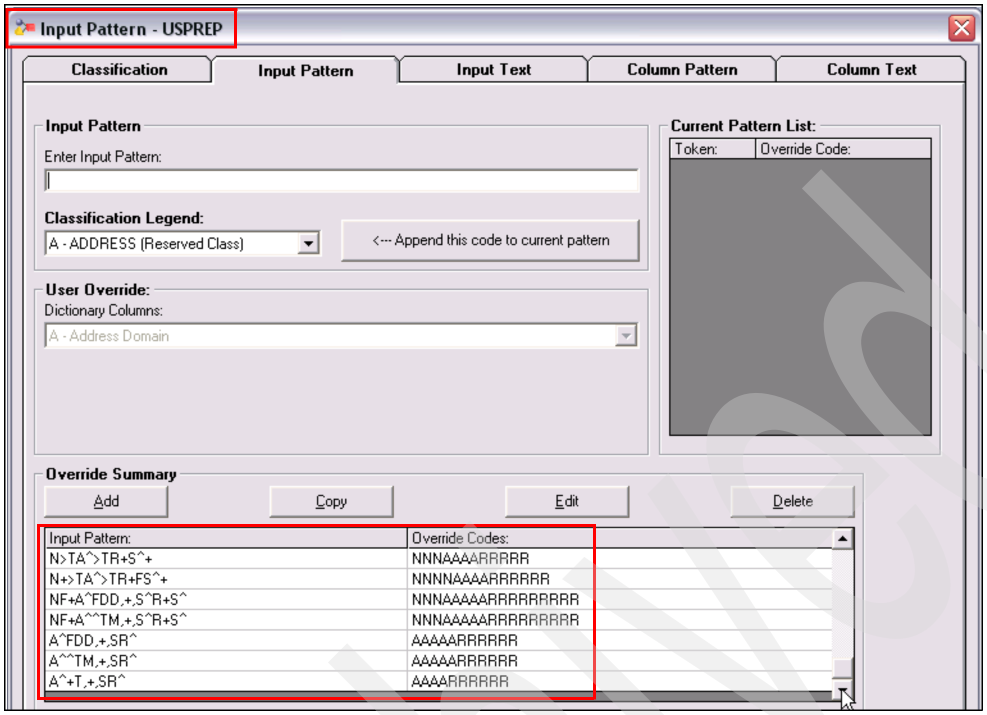
Input Pattern:	Override Codes:
NI+A^+TM,+,S^R+S^	NNNAAAAARRRRRRRRR
NF+A^+M,U^,F,SR^FS^	NNNAAAAAAAARRRRRRRR
NF+A^+TM,+,S^R+S^	NNNAAAAARRRRRRRRR
NF+A^D+TD,+,S^R+S^	NNNAAAAAARRRRRRRRR
NF+A^+M,U^,F,S^RFS^	NNNAAAAAAAARRRRRRRR
A^FM,F,SR^	AAAARRRRRR
NF+A++R+F^+	NNNAARRRRRR

Figure 2-72   Create j04_INVCC_XXPREP_INPUT_PATTERN 29/38



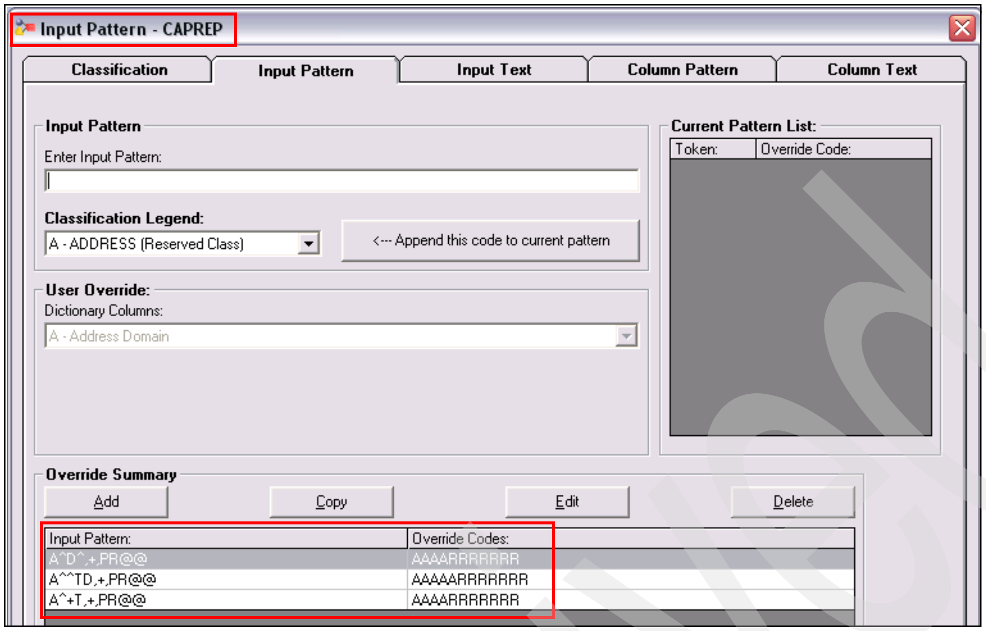
Input Pattern:	Override Codes:
N>+TA^>TR+S^+	NNNNAAAARRRRR
N>TA^>TR+S^+	NNNAAAARRRRR
N+>TA^>TR+FS^+	NNNNAAAARRRRRR
NF+A^FDD,+,S^R+S^	NNNAAAAARRRRRRRR
NF+A^^TM,+,S^R+S^	NNNAAAAARRRRRRRRR
A^FDD,+,SR^	AAAAARRRRRR
A^^TM,+,SR^	AAAAARRRRRR
[illegible]	[illegible]

Figure 2-73   Create j04_INVCC_XXPREP_INPUT_PATTERN 30/38



Input Pattern:	Override Codes:
N>TA^>TR+S^+	NNNAAAARRRRR
N+>TA^>TR+FS^+	NNNNAAAARRRRRR
NF+A^FDD,+,S^R+S^	NNNAAAAARRRRRRRRR
NF+A^^TM,+,S^R+S^	NNNAAAAARRRRRRRRR
A^FDD,+,SR^	AAAAARRRRRR
A^^TM,+,SR^	AAAAARRRRRR
A^+T,+,SR^	AAAARRRRRR

Figure 2-74 Create j04_INVCC_XXPREP_INPUT_PATTERN 31/38



Classification Legend: A - ADDRESS (Reserved Class)

Dictionary Columns: A - Address Domain

Input Pattern:	Override Codes:
A^D^,+,PR@@	AAAARRRRRRR
A^^TD,+,PR@@	AAAAARRRRRRR
A^+T,+,PR@@	AAAARRRRRRR

Figure 2-75 Create j04_INVCC_XXPREP_INPUT_PATTERN 32/38

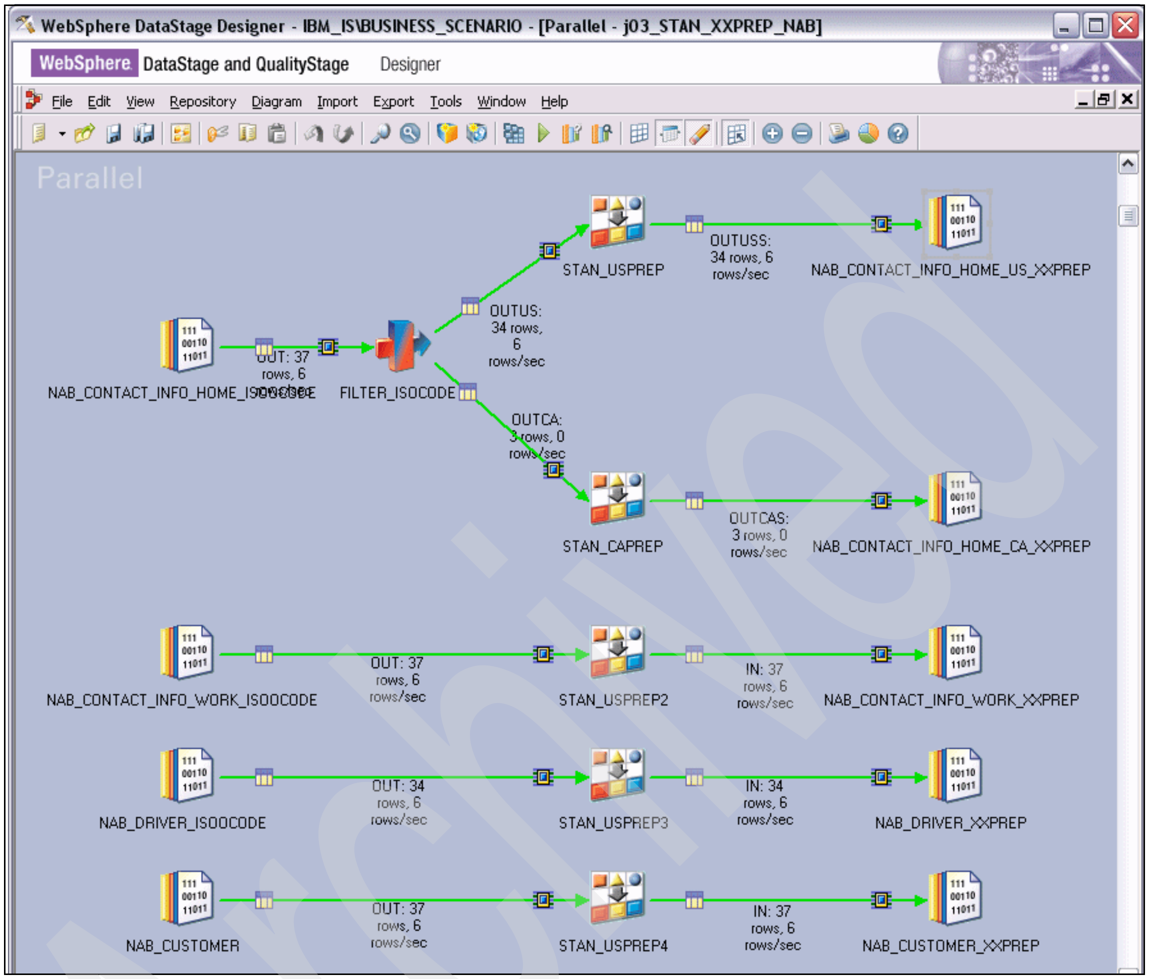

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j03_STAN_XXPREP_NAB]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
STAN_USPREP
OUTUSS: 34 rows, 6 rows/sec
NAB_CONTACT_INFO_HOME_US_XXPREP
OUTUS: 34 rows, 6 rows/sec
NAB_CONTACT_INFO_HOME_ISOCODE
OUT: 37 rows, 6 rows/sec
FILTER_ISOCODE
OUTCA: 3 rows, 0 rows/sec
STAN_CAPREP
OUTCAS: 3 rows, 0 rows/sec
NAB_CONTACT_INFO_HOME_CA_XXPREP
NAB_CONTACT_INFO_WORK_ISOOCODE
OUT: 37 rows, 6 rows/sec
STAN_USPREP2
IN: 37 rows, 6 rows/sec
NAB_CONTACT_INFO_WORK_XXPREP
NAB_DRIVER_ISOOCODE
OUT: 34 rows, 6 rows/sec
STAN_USPREP3
IN: 34 rows, 6 rows/sec
NAB_DRIVER_XXPREP
NAB_CUSTOMER
OUT: 37 rows, 6 rows/sec
STAN_USPREP4
IN: 37 rows, 6 rows/sec
NAB_CUSTOMER_XXPREP

Figure 2-76 Create j04_INVCC_XXPREP_INPUT_PATTERN 33/38

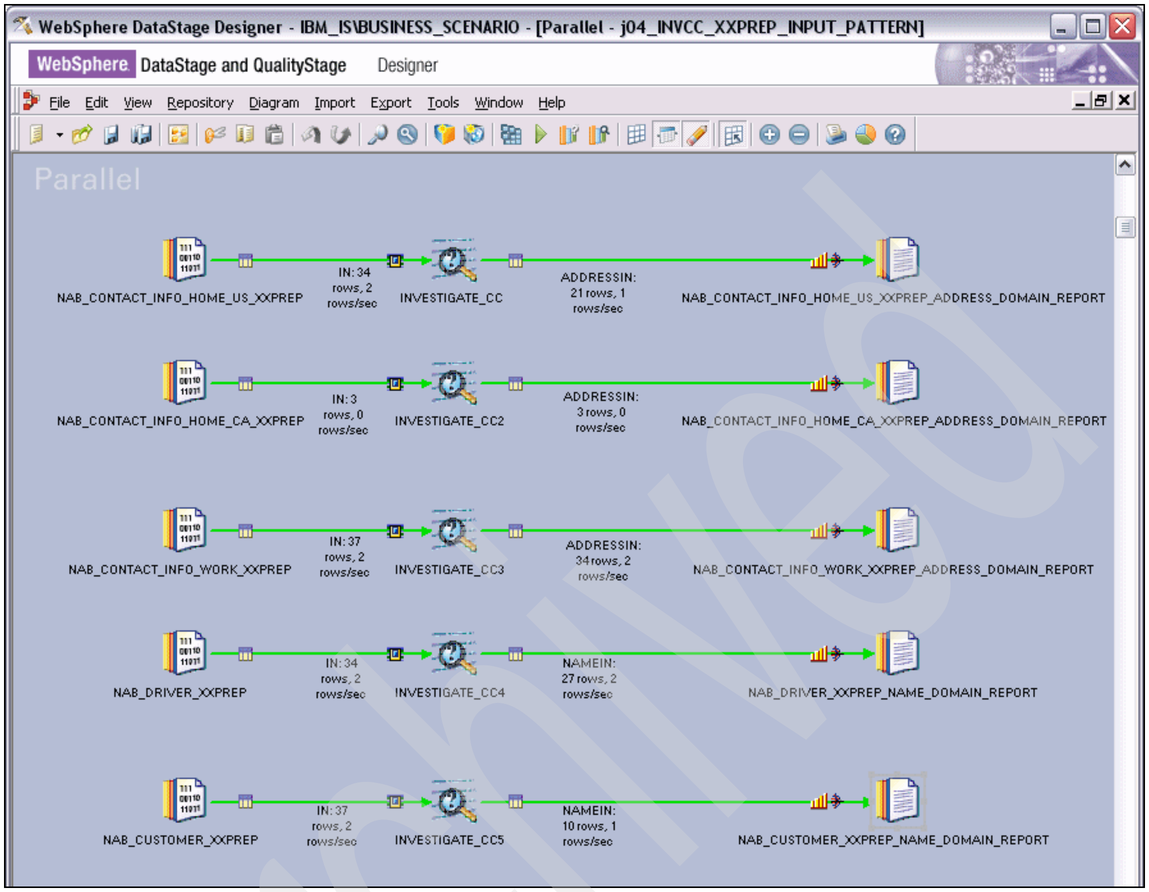
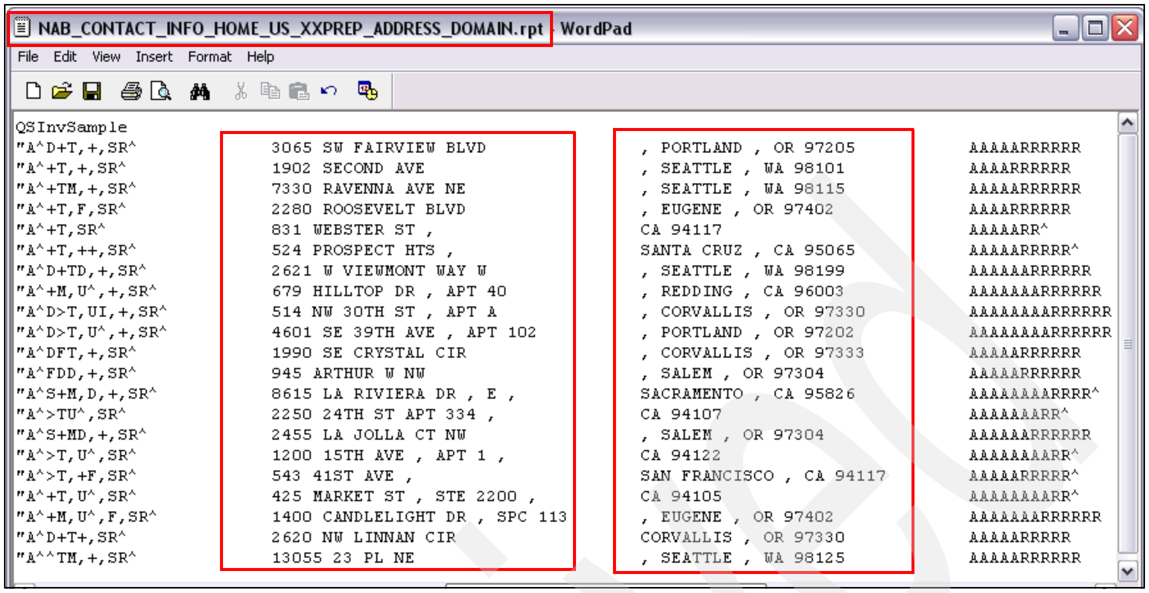

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j04_INVCC_XXPREP_INPUT_PATTERN]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NAB_CONTACT_INFO_HOME_US_XXPREP
IN: 34 rows, 2 rows/sec
INVESTIGATE_CC
ADDRESSIN: 21 rows, 1 rows/sec
NAB_CONTACT_INFO_HOME_US_XXPREP_ADDRESS_DOMAIN_REPORT
NAB_CONTACT_INFO_HOME_CA_XXPREP
IN: 3 rows, 0 rows/sec
INVESTIGATE_CC2
ADDRESSIN: 3 rows, 0 rows/sec
NAB_CONTACT_INFO_HOME_CA_XXPREP_ADDRESS_DOMAIN_REPORT
NAB_CONTACT_INFO_WORK_XXPREP
IN: 37 rows, 2 rows/sec
INVESTIGATE_CC3
ADDRESSIN: 34 rows, 2 rows/sec
NAB_CONTACT_INFO_WORK_XXPREP_ADDRESS_DOMAIN_REPORT
NAB_DRIVER_XXPREP
IN: 34 rows, 2 rows/sec
INVESTIGATE_CC4
NAMEIN: 27 rows, 2 rows/sec
NAB_DRIVER_XXPREP_NAME_DOMAIN_REPORT
NAB_CUSTOMER_XXPREP
IN: 37 rows, 2 rows/sec
INVESTIGATE_CC5
NAMEIN: 10 rows, 1 rows/sec
NAB_CUSTOMER_XXPREP_NAME_DOMAIN_REPORT

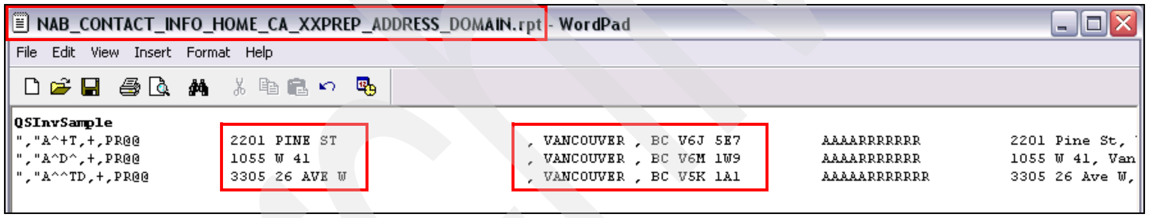
Figure 2-77 Create j04_INVCC_XXPREP_INPUT_PATTERN 34/38



NAB_CONTACT_INFO_HOME_US_XXPREP_ADDRESS_DOMAIN.rpt - WordPad

```
QSInvSample
"A^D+T,+,SR^          3065 SW FAIRVIEW BLVD        , PORTLAND , OR 97205         AAAAARRRRRR
"A^+T,+,SR^           1902 SECOND AVE              , SEATTLE , WA 98101          AAAARRRRRR
"A^+TM,+,SR^          7330 RAVENNA AVE NE          , SEATTLE , WA 98115          AAAAARRRRRR
"A^+T,F,SR^           2280 ROOSEVELT BLVD          , EUGENE , OR 97402           AAAARRRRRR
"A^+T,SR^             831 WEBSTER ST ,             CA 94117                      AAAAARR^
"A^+T,++,SR^          524 PROSPECT HTS ,           SANTA CRUZ , CA 95065         AAAAARRRRR^
"A^D+TD,+,SR^         2621 W VIEWMONT WAY W        , SEATTLE , WA 98199          AAAAAARRRRRR
"A^+M,U^,+,SR^        679 HILLTOP DR , APT 40      , REDDING , CA 96003          AAAAAAARRRRRR
"A^D>T,UI,+,SR^       514 NW 30TH ST , APT A       , CORVALLIS , OR 97330        AAAAAAAARRRRRR
"A^D>T,U^,+,SR^       4601 SE 39TH AVE , APT 102   , PORTLAND , OR 97202         AAAAAAAARRRRRR
"A^DFT,+,SR^          1990 SE CRYSTAL CIR          , CORVALLIS , OR 97333        AAAAARRRRRR
"A^FDD,+,SR^          945 ARTHUR W NW              , SALEM , OR 97304            AAAAARRRRRR
"A^S+M,D,+,SR^        8615 LA RIVIERA DR , E ,     SACRAMENTO , CA 95826         AAAAAAAARRRR^
"A^>TU^,SR^           2250 24TH ST APT 334 ,       CA 94107                      AAAAAAARR^
"A^S+MD,+,SR^         2455 LA JOLLA CT NW          , SALEM , OR 97304            AAAAAARRRRRR
"A^>T,U^,SR^          1200 15TH AVE , APT 1 ,      CA 94122                      AAAAAAAARR^
"A^>T,+F,SR^          543 41ST AVE ,               SAN FRANCISCO , CA 94117      AAAAARRRRR^
"A^+T,U^,SR^          425 MARKET ST , STE 2200 ,   CA 94105                      AAAAAAAARR^
"A^+M,U^,F,SR^        1400 CANDLELIGHT DR , SPC 113 , EUGENE , OR 97402          AAAAAAARRRRRR
"A^D+T+,SR^           2620 NW LINNAN CIR           CORVALLIS , OR 97330          AAAAARRRRRR
"A^^TM,+,SR^          13055 23 PL NE               , SEATTLE , WA 98125          AAAAARRRRRR
```

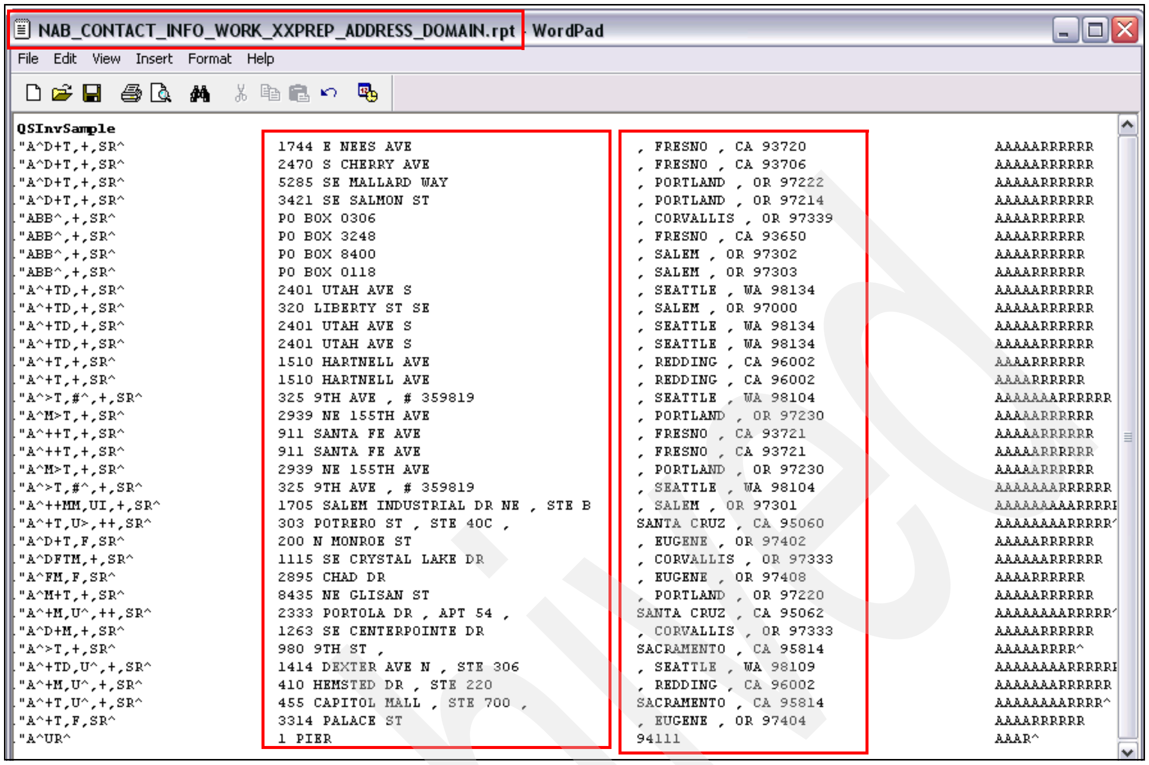
Figure 2-78 Create j04_INVCC_XXPREP_INPUT_PATTERN 35/38



NAB_CONTACT_INFO_HOME_CA_XXPREP_ADDRESS_DOMAIN.rpt - WordPad

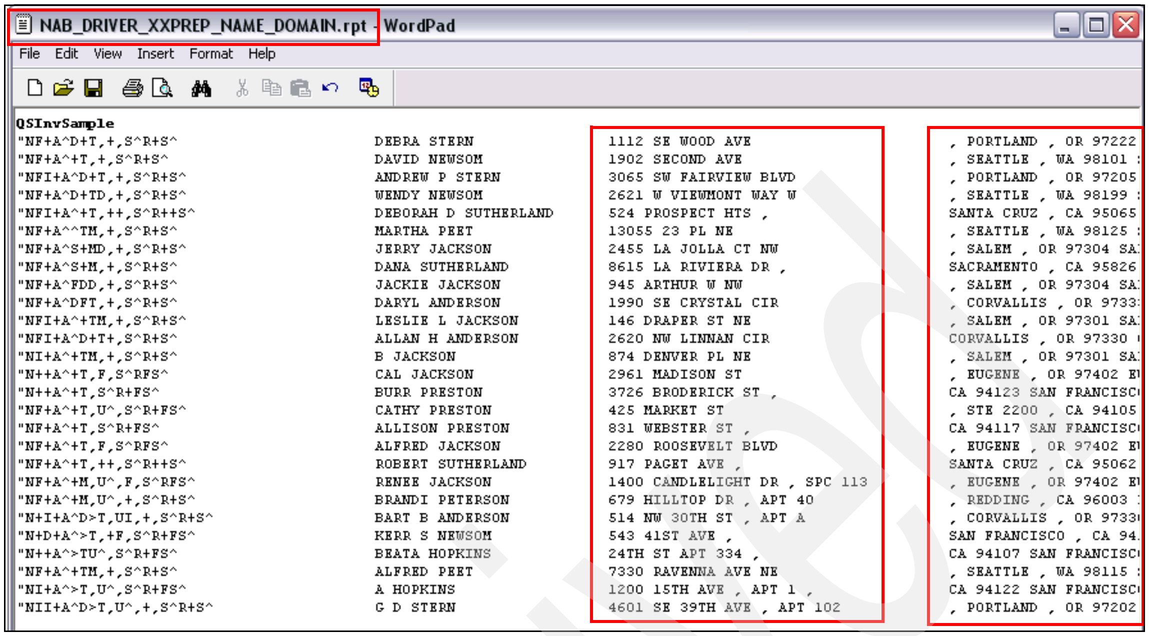
```
QSInvSample
","A^+T,+,PR@@      2201 PINE ST           , VANCOUVER , BC V6J 5E7     AAAARRRRRRR      2201 Pine St,
","A^D^,+,PR@@      1055 W 41              , VANCOUVER , BC V6M 1W9     AAAARRRRRRR      1055 W 41, Van
","A^^TD,+,PR@@     3305 26 AVE W          , VANCOUVER , BC V5K 1A1     AAAAARRRRRRR     3305 26 Ave W,
```

Figure 2-79 Create j04_INVCC_XXPREP_INPUT_PATTERN 36/38



```
QSInvSample
"A^D+T,+,SR^                1744 E NEES AVE                         , FRESNO , CA 93720                     AAAAARRRRRR
"A^D+T,+,SR^                2470 S CHERRY AVE                       , FRESNO , CA 93706                     AAAAARRRRRR
"A^D+T,+,SR^                5285 SE MALLARD WAY                     , PORTLAND , OR 97222                   AAAAARRRRRR
"A^D+T,+,SR^                3421 SE SALMON ST                       , PORTLAND , OR 97214                   AAAAARRRRRR
"ABB^,+,SR^                 PO BOX 0306                             , CORVALLIS , OR 97339                  AAAARRRRRR
"ABB^,+,SR^                 PO BOX 3248                             , FRESNO , CA 93650                     AAAARRRRRR
"ABB^,+,SR^                 PO BOX 8400                             , SALEM , OR 97302                      AAAARRRRRR
"ABB^,+,SR^                 PO BOX 0118                             , SALEM , OR 97303                      AAAARRRRRR
"A^+TD,+,SR^                2401 UTAH AVE S                         , SEATTLE , WA 98134                    AAAAARRRRRR
"A^+TD,+,SR^                320 LIBERTY ST SE                       , SALEM , OR 97000                      AAAAARRRRRR
"A^+TD,+,SR^                2401 UTAH AVE S                         , SEATTLE , WA 98134                    AAAAARRRRRR
"A^+TD,+,SR^                2401 UTAH AVE S                         , SEATTLE , WA 98134                    AAAAARRRRRR
"A^+T,+,SR^                 1510 HARTNELL AVE                       , REDDING , CA 96002                    AAAARRRRRR
"A^+T,+,SR^                 1510 HARTNELL AVE                       , REDDING , CA 96002                    AAAARRRRRR
"A^>T,#^,+,SR^              325 9TH AVE , # 359819                  , SEATTLE , WA 98104                    AAAAAAARRRRRR
"A^M>T,+,SR^                2939 NE 155TH AVE                       , PORTLAND , OR 97230                   AAAAARRRRRR
"A^++T,+,SR^                911 SANTA FE AVE                        , FRESNO , CA 93721                     AAAAARRRRRR
"A^++T,+,SR^                911 SANTA FE AVE                        , FRESNO , CA 93721                     AAAAARRRRRR
"A^M>T,+,SR^                2939 NE 155TH AVE                       , PORTLAND , OR 97230                   AAAAARRRRRR
"A^>T,#^,+,SR^              325 9TH AVE , # 359819                  , SEATTLE , WA 98104                    AAAAAAAARRRRRR
"A^++MM,UI,+,SR^            1705 SALEM INDUSTRIAL DR NE , STE B     , SALEM , OR 97301                      AAAAAAAAAARRRRI
"A^+T,U>,++,SR^             303 POTRERO ST , STE 40C ,              SANTA CRUZ , CA 95060                   AAAAAAAAARRRRR^
"A^D+T,F,SR^                200 N MONROE ST                         , EUGENE , OR 97402                     AAAAARRRRRR
"A^DFTM,+,SR^               1115 SE CRYSTAL LAKE DR                 , CORVALLIS , OR 97333                  AAAAAARRRRRR
"A^FM,F,SR^                 2895 CHAD DR                            , EUGENE , OR 97408                     AAAARRRRRR
"A^M+T,+,SR^                8435 NE GLISAN ST                       , PORTLAND , OR 97220                   AAAAARRRRRR
"A^+M,U^,++,SR^             2333 PORTOLA DR , APT 54 ,              SANTA CRUZ , CA 95062                   AAAAAAAAARRRRR^
"A^D+M,+,SR^                1263 SE CENTERPOINTE DR                 , CORVALLIS , OR 97333                  AAAAARRRRRR
"A^>T,+,SR^                 980 9TH ST ,                            SACRAMENTO , CA 95814                   AAAAARRRR^
"A^+TD,U^,+,SR^             1414 DEXTER AVE N , STE 306             , SEATTLE , WA 98109                    AAAAAAAAARRRRRI
"A^+M,U^,+,SR^              410 HEMSTED DR , STE 220                , REDDING , CA 96002                    AAAAAAARRRRRR
"A^+T,U^,+,SR^              455 CAPITOL MALL , STE 700 ,            SACRAMENTO , CA 95814                   AAAAAAAAARRRR^
"A^+T,F,SR^                 3314 PALACE ST                          , EUGENE , OR 97404                     AAAARRRRRR
"A^UR^                      1 PIER                                  94111                                   AAAR^
```

Figure 2-80   Create j04_INVCC_XXPREP_INPUT_PATTERN 37/38


NAB_DRIVER_XXPREP_NAME_DOMAIN.rpt - WordPad
File Edit View Insert Format Help

```
QSInvSample
"NF+A^D+T,+,S^R+S^              DEBRA STERN              1112 SE WOOD AVE                   , PORTLAND , OR 97222
"NF+A^+T,+,S^R+S^               DAVID NEWSOM             1902 SECOND AVE                    , SEATTLE , WA 98101 :
"NFI+A^D+T,+,S^R+S^             ANDREW P STERN           3065 SW FAIRVIEW BLVD              , PORTLAND , OR 97205
"NF+A^D+TD,+,S^R+S^             WENDY NEWSOM             2621 W VIEWMONT WAY W              , SEATTLE , WA 98199 :
"NFI+A^+T,++,S^R++S^            DEBORAH D SUTHERLAND     524 PROSPECT HTS ,                 SANTA CRUZ , CA 95065
"NF+A^^TM,+,S^R+S^              MARTHA PEET              13055 23 PL NE                     , SEATTLE , WA 98125 :
"NF+A^S+MD,+,S^R+S^             JERRY JACKSON            2455 LA JOLLA CT NW                , SALEM , OR 97304 SA
"NF+A^S+M,+,S^R+S^              DANA SUTHERLAND          8615 LA RIVIERA DR ,               SACRAMENTO , CA 95826
"NF+A^FDD,+,S^R+S^              JACKIE JACKSON           945 ARTHUR W NW                    , SALEM , OR 97304 SA
"NF+A^DFT,+,S^R+S^              DARYL ANDERSON           1990 SE CRYSTAL CIR                , CORVALLIS , OR 9733
"NFI+A^+TM,+,S^R+S^             LESLIE L JACKSON         146 DRAPER ST NE                   , SALEM , OR 97301 SA
"NFI+A^D+T+,S^R+S^              ALLAN H ANDERSON         2620 NW LINNAN CIR                 CORVALLIS , OR 97330
"NI+A^+TM,+,S^R+S^              B JACKSON                874 DENVER PL NE                   , SALEM , OR 97301 SA
"N++A^+T,F,S^RFS^               CAL JACKSON              2961 MADISON ST                    , EUGENE , OR 97402 E
"N++A^+T,S^R+FS^                BURR PRESTON             3726 BRODERICK ST ,                CA 94123 SAN FRANCISC
"NF+A^+T,U^,S^R+FS^             CATHY PRESTON            425 MARKET ST                      , STE 2200 , CA 94105
"NF+A^+T,S^R+FS^                ALLISON PRESTON          831 WEBSTER ST ,                   CA 94117 SAN FRANCISC
"NF+A^+T,F,S^RFS^               ALFRED JACKSON           2280 ROOSEVELT BLVD                , EUGENE , OR 97402 E
"NF+A^+T,++,S^R++S^             ROBERT SUTHERLAND        917 PAGET AVE ,                    SANTA CRUZ , CA 95062
"NF+A^+M,U^,F,S^RFS^            RENEE JACKSON            1400 CANDLELIGHT DR , SPC 113      , EUGENE , OR 97402 E
"NF+A^+M,U^,+,S^R+S^            BRANDI PETERSON          679 HILLTOP DR , APT 40            , REDDING , CA 96003
"N+I+A^D>T,UI,+,S^R+S^          BART B ANDERSON          514 NW 30TH ST , APT A             , CORVALLIS , OR 9733
"N+D+A^>T,+F,S^R+FS^            KERR S NEWSOM            543 41ST AVE ,                     SAN FRANCISCO , CA 94.
"N++A^>TU^,S^R+FS^              BEATA HOPKINS            24TH ST APT 334 ,                  CA 94107 SAN FRANCISC
"NF+A^+TM,+,S^R+S^              ALFRED PEET              7330 RAVENNA AVE NE                , SEATTLE , WA 98115 :
"NI+A^>T,U^,S^R+FS^             A HOPKINS                1200 15TH AVE , APT 1 ,            CA 94122 SAN FRANCISC
"NII+A^D>T,U^,+,S^R+S^          G D STERN                4601 SE 39TH AVE , APT 102         , PORTLAND , OR 97202
```

Figure 2-81 Create j04_INVCC_XXPREP_INPUT_PATTERN 38/38

j05a_INVW_CONTACT_INFO_HOME_CA_XXPREP

After all the name and address data is moved to the correct domain buckets, we use the Investigate stage with the word investigate option to determine the effectiveness of token handling in the NAME, ADDRESS, and AREA domains. Potential Classification and Input Pattern overrides were identified at this point.

We analyze the CONTACT_INFO file with the Canadian home addresses containing the name, address, and area buckets that were generated by the "j03_STAN_XXPREP_NAB" on page 530 job by the Investigate stage using word investigate and the domain-specific USADDR and USAREA rule sets to determine the degree of success achieved by the rule sets in identifying the tokens correctly.

Because a single Investigate stage can only have a single rule set associated with it, we needed to split the CONTACT_INFO file with the Canadian home addresses (using a Copy stage) and process the files by two independent Investigate stages, with each stage using a particular domain-specific rule set. Both the pattern and token reports are generated in each Investigate stage.

Figure 2-82 on page 574 shows the various stages that are used in this job, including the data set that was created in "j03_STAN_XXPREP_NAB" on page 530, a Copy stage, two Investigate stages that each use a different domain-specific rule set, and two sequential file stages (one each for the token report and pattern report) for each Investigate stage. We modified the names of the stages as shown.

Because the configuration of this job is very similar to that described in “J04_INVW_USPREP” on page 203, we do not repeated it here. However, some of the configurations of interest are as follows:

- Figure 2-83 on page 575 shows the INVADDR - Investigate Stage window using the USADDR rule set with the Word Investigate option. The selected column is AddressDomain_CAPREP.
- Figure 2-84 on page 575 shows the INVAREA - Investigate Stage window using the USAREA rule set with the Word Investigate option. The selected column is AreaDomain_CAPREP.
- After saving, compiling, and running this job (Figure 2-85 on page 576), the contents of the output of the Investigate stage are written to the sequential files shown in Figure 2-86 on page 576 through Figure 2-89 on page 577.

  The reports show a few tokens (such as PINE and Vancouver) in the various reports not being recognized with proper classifications (code ?). The patterns also appear to need overrides, but you can only be certain of this after the pattern action language is invoked.

Proceed now to “j05b_INVW_CONTACT_INFO_HOME_US_XXPREP” on page 577.

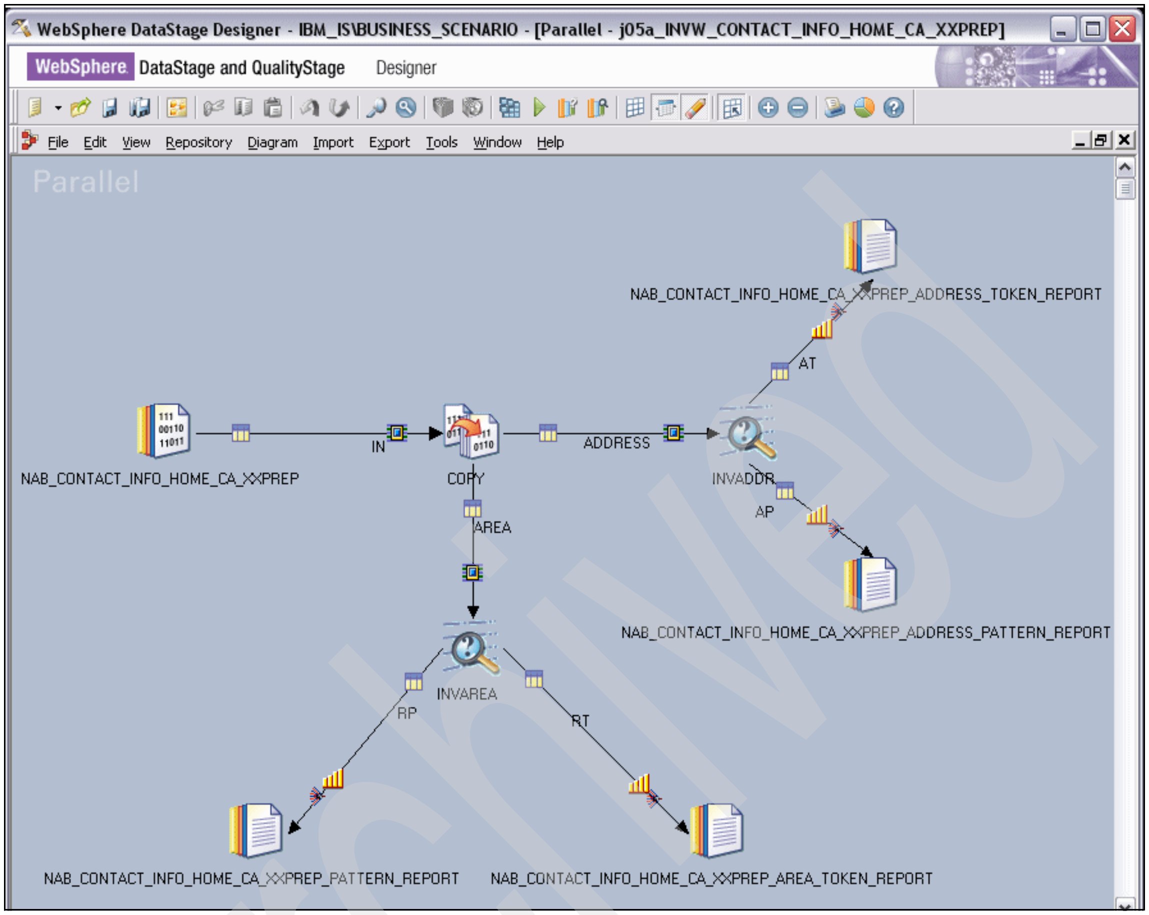
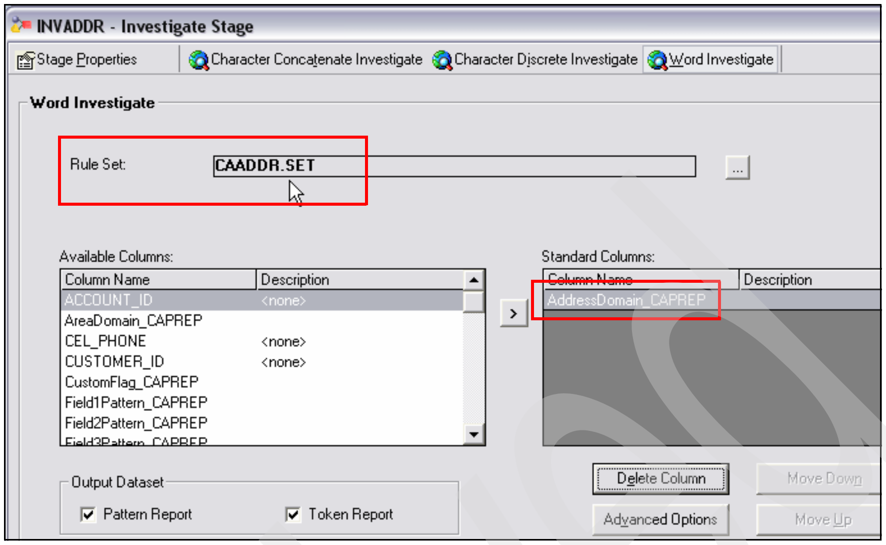

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j05a_INVW_CONTACT_INFO_HOME_CA_XXPREP]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NAB_CONTACT_INFO_HOME_CA_XXPREP_ADDRESS_TOKEN_REPORT
AT
NAB_CONTACT_INFO_HOME_CA_XXPREP
IN
COPY
ADDRESS
INVADDR
AP
AREA
NAB_CONTACT_INFO_HOME_CA_XXPREP_ADDRESS_PATTERN_REPORT
INVAREA
RP
RT
NAB_CONTACT_INFO_HOME_CA_XXPREP_PATTERN_REPORT
NAB_CONTACT_INFO_HOME_CA_XXPREP_AREA_TOKEN_REPORT

Figure 2-82 Create j05a_INVW_CONTACT_INFO_HOME_CA_XXPREP 1/8


INVADDR - Investigate Stage

Word Investigate

Rule Set: CAADDR.SET

Available Columns:

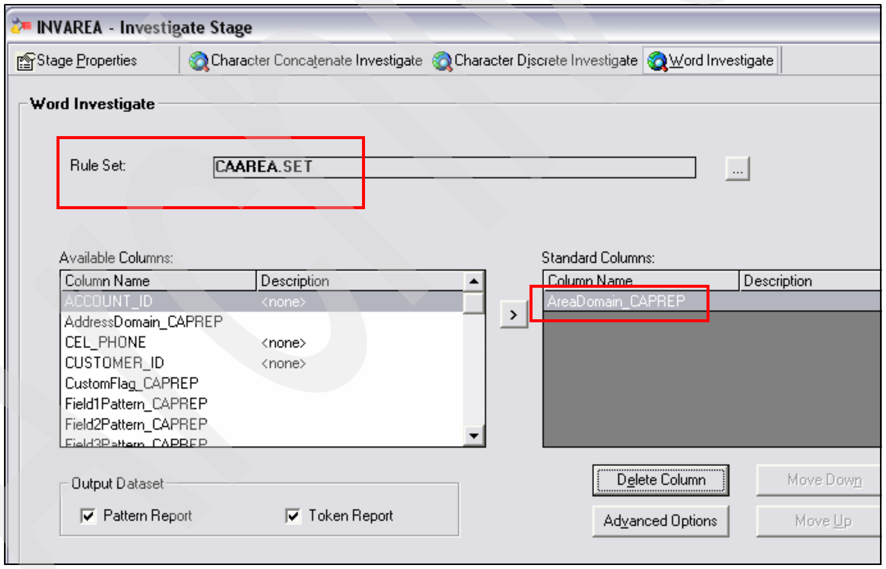
Column Name	Description
ACCOUNT_ID	<none>
AreaDomain_CAPREP	
CEL_PHONE	<none>
CUSTOMER_ID	<none>
CustomFlag_CAPREP	
Field1Pattern_CAPREP	
Field2Pattern_CAPREP	
Field3Pattern_CAPREP	

Standard Columns:

Column Name	Description
AddressDomain_CAPREP	

Output Dataset: ☑ Pattern Report ☑ Token Report

Figure 2-83   Create j05a_INVW_CONTACT_INFO_HOME_CA_XXPREP 2/8


INVAREA - Investigate Stage

Word Investigate

Rule Set: CAAREA.SET

Available Columns:

Column Name	Description
ACCOUNT_ID	<none>
AddressDomain_CAPREP	
CEL_PHONE	<none>
CUSTOMER_ID	<none>
CustomFlag_CAPREP	
Field1Pattern_CAPREP	
Field2Pattern_CAPREP	
Field3Pattern_CAPREP	

Standard Columns:

Column Name	Description
AreaDomain_CAPREP	

Output Dataset: ☑ Pattern Report ☑ Token Report

Figure 2-84   Create j05a_INVW_CONTACT_INFO_HOME_CA_XXPREP 3/8

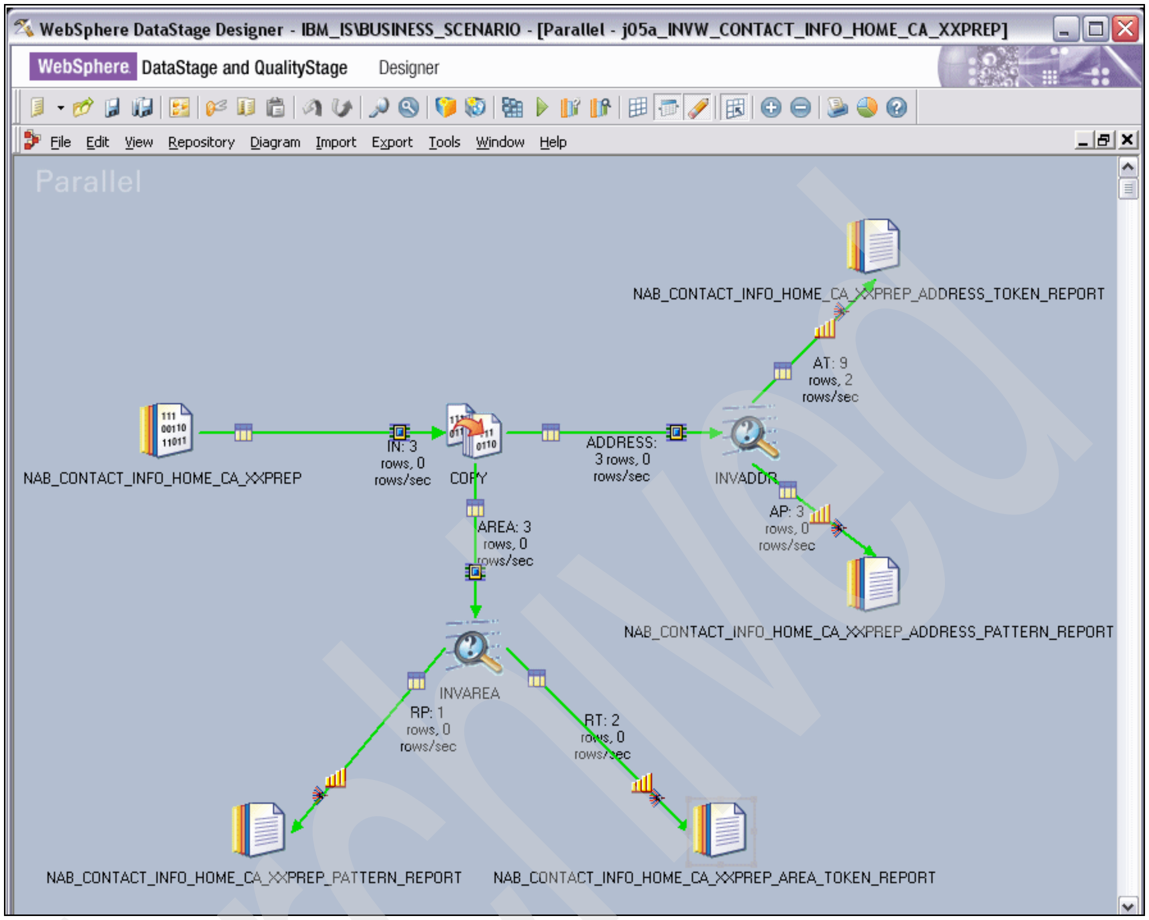
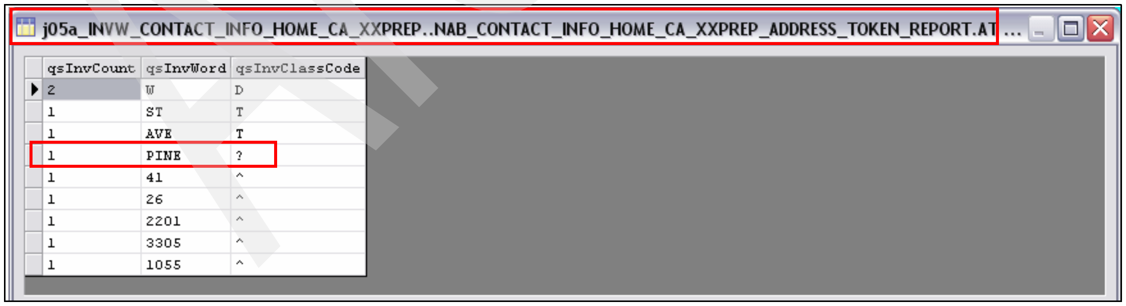

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j05a_INVW_CONTACT_INFO_HOME_CA_XXPREP]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NAB_CONTACT_INFO_HOME_CA_XXPREP_ADDRESS_TOKEN_REPORT
AT: 9 rows, 2 rows/sec
NAB_CONTACT_INFO_HOME_CA_XXPREP
IN: 3 rows, 0 rows/sec
COPY
ADDRESS: 3 rows, 0 rows/sec
INVADDR
AP: 3 rows, 0 rows/sec
AREA: 3 rows, 0 rows/sec
NAB_CONTACT_INFO_HOME_CA_XXPREP_ADDRESS_PATTERN_REPORT
INVAREA
RP: 1 rows, 0 rows/sec
RT: 2 rows, 0 rows/sec
NAB_CONTACT_INFO_HOME_CA_XXPREP_PATTERN_REPORT
NAB_CONTACT_INFO_HOME_CA_XXPREP_AREA_TOKEN_REPORT

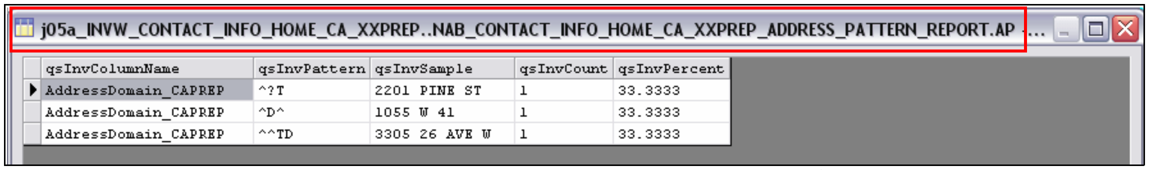
Figure 2-85 Create j05a_INVW_CONTACT_INFO_HOME_CA_XXPREP 4/8



j05a_INVW_CONTACT_INFO_HOME_CA_XXPREP..NAB_CONTACT_INFO_HOME_CA_XXPREP_ADDRESS_TOKEN_REPORT.AT ...

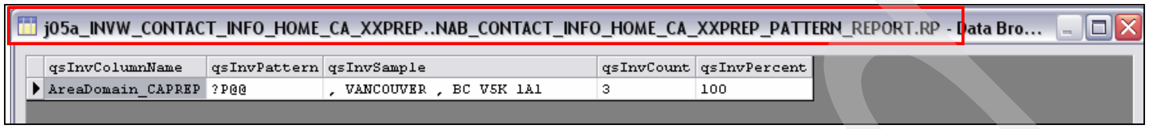
qsInvCount	qsInvWord	qsInvClassCode
2	W	D
1	ST	T
1	AVE	T
1	PINE	?
1	41	^
1	26	^
1	2201	^
1	3305	^
1	1055	^

Figure 2-86 Create j05a_INVW_CONTACT_INFO_HOME_CA_XXPREP 5/8


j05a_INVW_CONTACT_INFO_HOME_CA_XXPREP..NAB_CONTACT_INFO_HOME_CA_XXPREP_ADDRESS_PATTERN_REPORT.AP ...

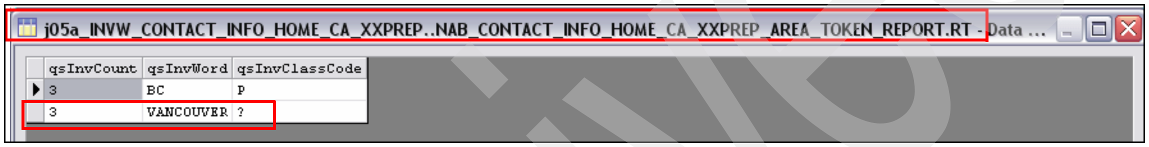
qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
AddressDomain_CAPREP	^?T	2201 PINE ST	1	33.3333
AddressDomain_CAPREP	^D^	1055 W 41	1	33.3333
AddressDomain_CAPREP	^^TD	3305 26 AVE W	1	33.3333

Figure 2-87   Create j05a_INVW_CONTACT_INFO_HOME_CA_XXPREP 6/8


j05a_INVW_CONTACT_INFO_HOME_CA_XXPREP..NAB_CONTACT_INFO_HOME_CA_XXPREP_PATTERN_REPORT.RP - Data Bro...

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
AreaDomain_CAPREP	?P@@	, VANCOUVER , BC V5K 1A1	3	100

Figure 2-88   Create j05a_INVW_CONTACT_INFO_HOME_CA_XXPREP 7/8


j05a_INVW_CONTACT_INFO_HOME_CA_XXPREP..NAB_CONTACT_INFO_HOME_CA_XXPREP_AREA_TOKEN_REPORT.RT - Data ...

qsInvCount	qsInvWord	qsInvClassCode
3	BC	P
3	VANCOUVER	?

Figure 2-89   Create j05a_INVW_CONTACT_INFO_HOME_CA_XXPREP 8/8

j05b_INVW_CONTACT_INFO_HOME_US_XXPREP

In this step, we analyze the CONTACT_INFO file with the U.S. home addresses containing the name, address, and area buckets that were generated by the "j03_STAN_XXPREP_NAB" on page 530 job by the Investigate stage using word investigate and the domain-specific USADDR and USAREA rule sets to determine the degree of success achieved by the Standardize stage in identifying the tokens correctly.

Because a single Investigate stage can only have a single rule set associated with it, we split the CONTACT_INFO file with the U.S. home addresses (using a Copy stage) and process the files by two independent Investigate stages, with each stage using a particular domain-specific rule set. Both the pattern and token reports were generated in each Investigate stage.

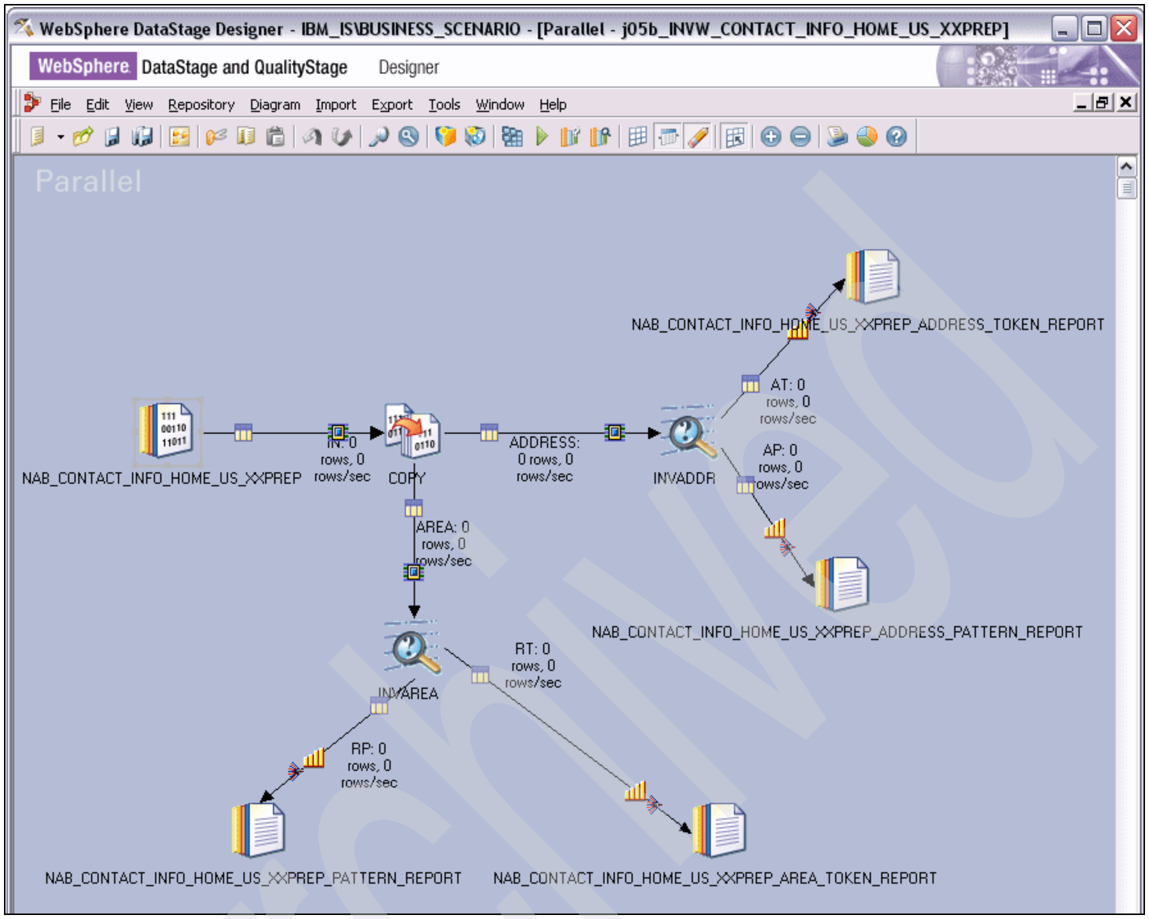
Figure 2-90 on page 579 shows the various stages that are used in this job, including the data set that was created in "j03_STAN_XXPREP_NAB" on page 530, a Copy stage, two Investigate stages that each use a different domain-specific rule set, and two sequential file stages (one each for the token report and pattern report) for each Investigate stage. We modified the names of the stages as shown.

Because the configuration of this job is very similar to that described in “J04_INVW_USPREP” on page 203, we do not repeat it here. However, some of the configurations of interest are as follows:

- Figure 2-91 on page 580 shows the INVADDR - Investigate Stage window using the USADDR rule set with the Word Investigate option. The selected column is AddressDomain_USPREP.
- Figure 2-92 on page 580 shows the INVAREA - Investigate Stage window using the USAREA rule set with the Word Investigate option. The selected column is AreaDomain_USPREP.
- After saving, compiling, and running this job (Figure 2-93 on page 581), the contents of the output of Investigate stage are written to the sequential files shown in Figure 2-94 on page 582 through Figure 2-97 on page 584.

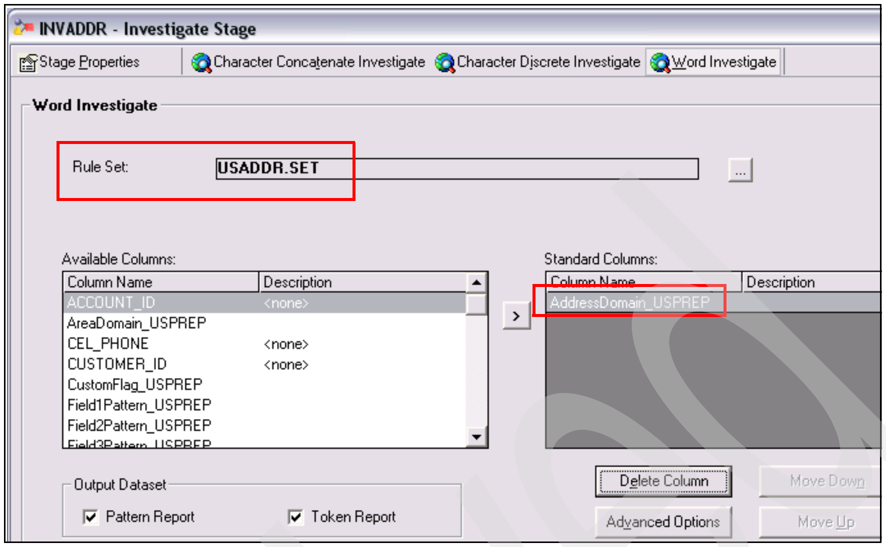
  The reports show a number of tokens in the various reports not being recognized with proper classifications (code `?`). The patterns also appear to need overrides, but you can only be certain of this after the pattern action language is invoked.

Proceed now to “j05c_INVW_CONTACT_INFO_WORK_XXPREP” on page 584.



WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j05b_INVW_CONTACT_INFO_HOME_US_XXPREP]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NAB_CONTACT_INFO_HOME_US_XXPREP
IN: 0 rows, 0 rows/sec
COPY
ADDRESS: 0 rows, 0 rows/sec
INVADDR
AT: 0 rows, 0 rows/sec
AP: 0 rows, 0 rows/sec
NAB_CONTACT_INFO_HOME_US_XXPREP_ADDRESS_TOKEN_REPORT
NAB_CONTACT_INFO_HOME_US_XXPREP_ADDRESS_PATTERN_REPORT
AREA: 0 rows, 0 rows/sec
INVAREA
RT: 0 rows, 0 rows/sec
RP: 0 rows, 0 rows/sec
NAB_CONTACT_INFO_HOME_US_XXPREP_PATTERN_REPORT
NAB_CONTACT_INFO_HOME_US_XXPREP_AREA_TOKEN_REPORT

Figure 2-90 Create j05b_INVW_CONTACT_INFO_HOME_US_XXPREP 1/8


INVADDR - Investigate Stage

Stage Properties | Character Concatenate Investigate | Character Discrete Investigate | Word Investigate

Word Investigate

Rule Set: USADDR.SET

Available Columns:

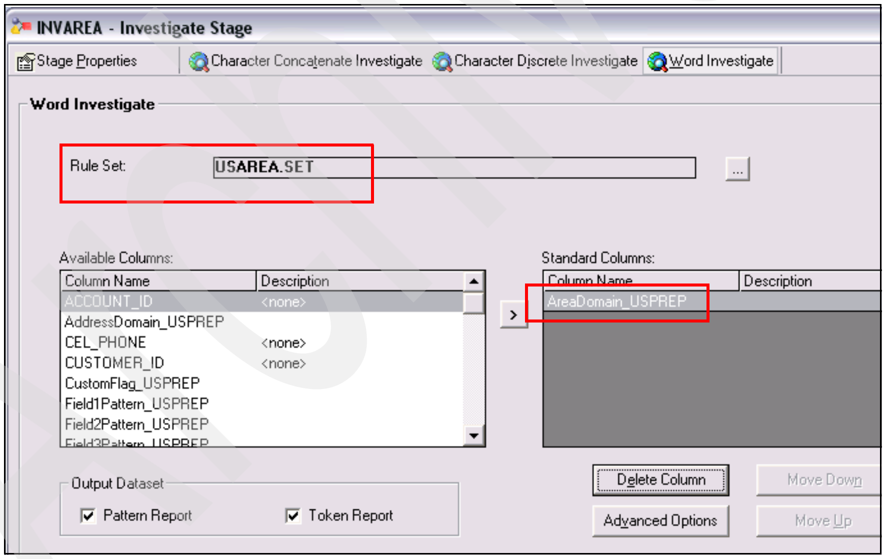
Column Name	Description
ACCOUNT_ID	<none>
AreaDomain_USPREP	
CEL_PHONE	<none>
CUSTOMER_ID	<none>
CustomFlag_USPREP	
Field1Pattern_USPREP	
Field2Pattern_USPREP	
Field3Pattern_USPREP	

Standard Columns:

Column Name	Description
AddressDomain_USPREP	

Output Dataset
- [✓] Pattern Report
- [✓] Token Report

Figure 2-91 Create j05b_INVW_CONTACT_INFO_HOME_US_XXPREP 2/8


INVAREA - Investigate Stage

Stage Properties | Character Concatenate Investigate | Character Discrete Investigate | Word Investigate

Word Investigate

Rule Set: USAREA.SET

Available Columns:

Column Name	Description
ACCOUNT_ID	<none>
AddressDomain_USPREP	
CEL_PHONE	<none>
CUSTOMER_ID	<none>
CustomFlag_USPREP	
Field1Pattern_USPREP	
Field2Pattern_USPREP	
Field3Pattern_USPREP	

Standard Columns:

Column Name	Description
AreaDomain_USPREP	

Output Dataset
- [✓] Pattern Report
- [✓] Token Report

Figure 2-92 Create j05b_INVW_CONTACT_INFO_HOME_US_XXPREP 3/8

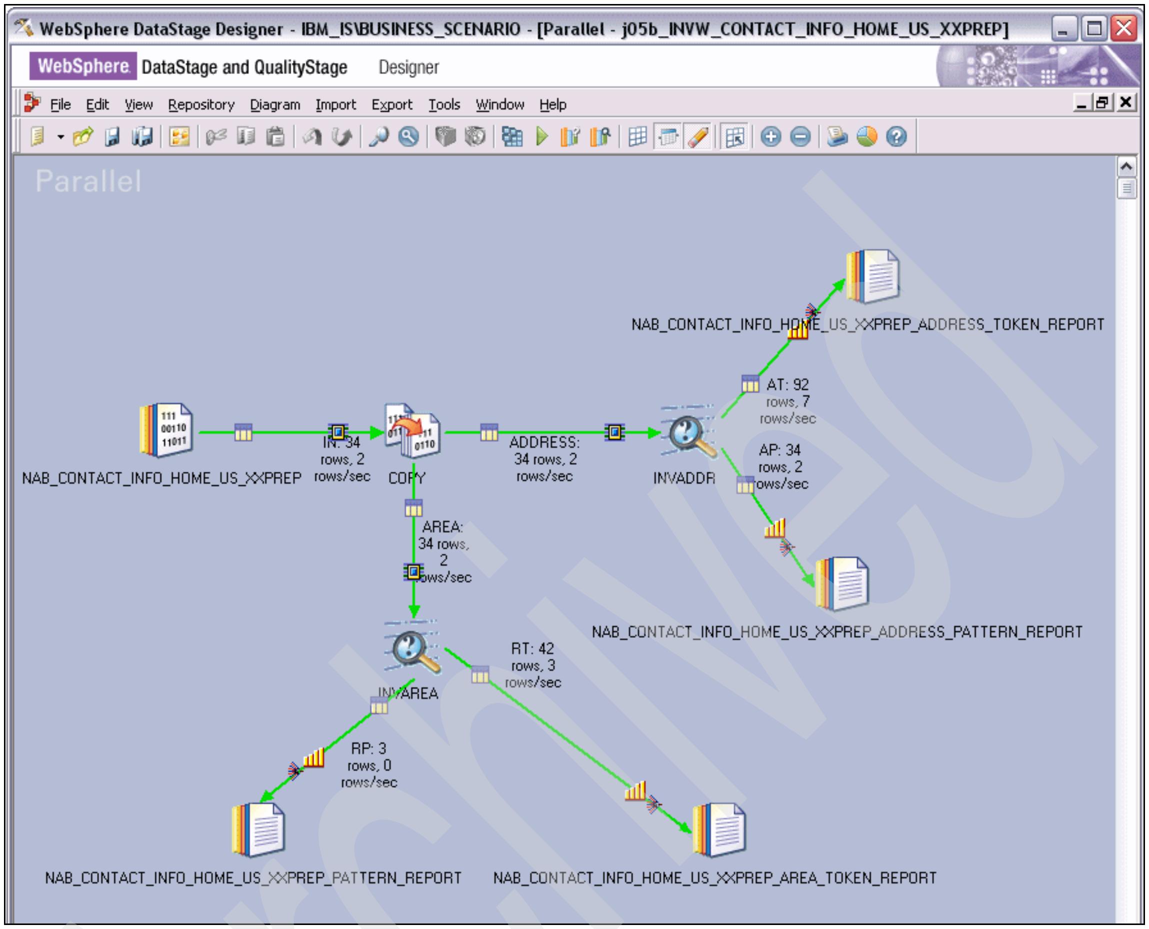
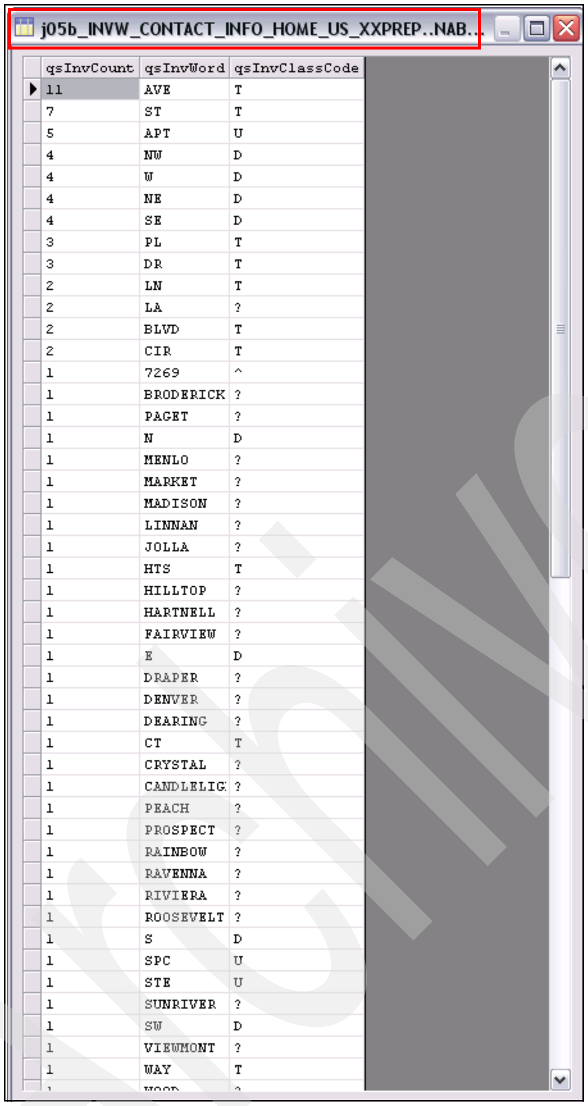

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j05b_INVW_CONTACT_INFO_HOME_US_XXPREP]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NAB_CONTACT_INFO_HOME_US_XXPREP_ADDRESS_TOKEN_REPORT
AT: 92 rows, 7 rows/sec
NAB_CONTACT_INFO_HOME_US_XXPREP
IN: 34 rows, 2 rows/sec
COPY
ADDRESS: 34 rows, 2 rows/sec
INVADDR
AP: 34 rows, 2 rows/sec
AREA: 34 rows, 2 rows/sec
NAB_CONTACT_INFO_HOME_US_XXPREP_ADDRESS_PATTERN_REPORT
INVAREA
RT: 42 rows, 3 rows/sec
RP: 3 rows, 0 rows/sec
NAB_CONTACT_INFO_HOME_US_XXPREP_PATTERN_REPORT
NAB_CONTACT_INFO_HOME_US_XXPREP_AREA_TOKEN_REPORT

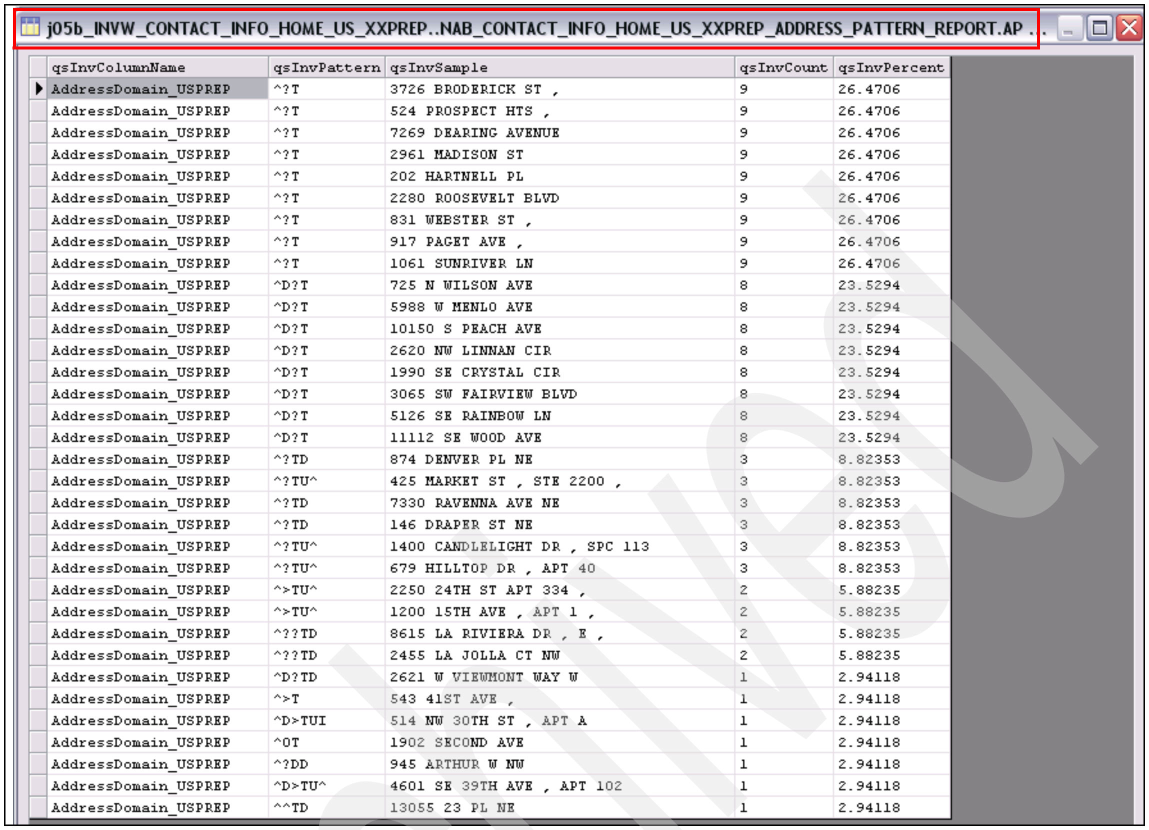
Figure 2-93 Create j05b_INVW_CONTACT_INFO_HOME_US_XXPREP 4/8



j05b_INVW_CONTACT_INFO_HOME_US_XXPREP..NAB...

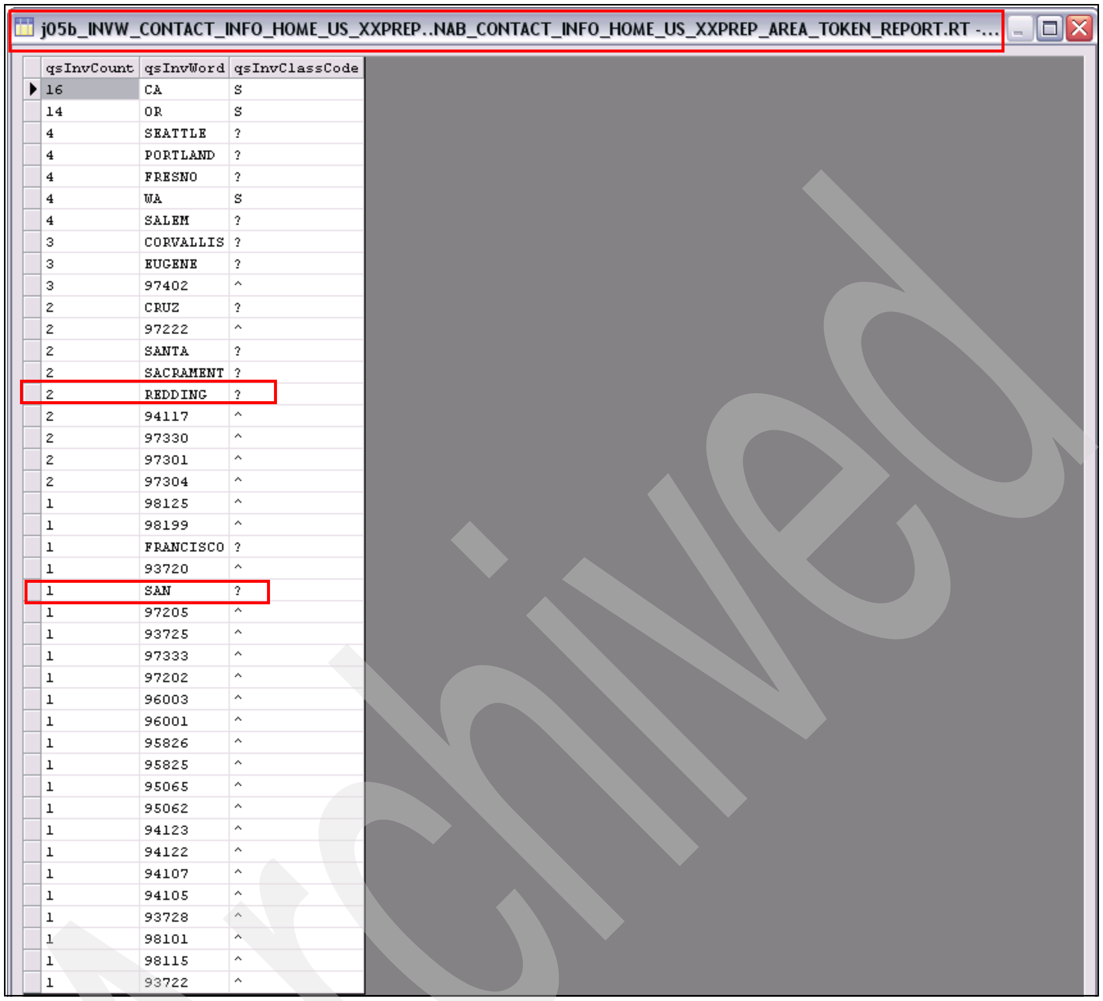
qsInvCount	qsInvWord	qsInvClassCode
11	AVE	T
7	ST	T
5	APT	U
4	NW	D
4	W	D
4	NE	D
4	SE	D
3	PL	T
3	DR	T
2	LN	T
2	LA	?
2	BLVD	T
2	CIR	T
1	7269	^
1	BRODERICK	?
1	PAGET	?
1	N	D
1	MENLO	?
1	MARKET	?
1	MADISON	?
1	LINNAN	?
1	JOLLA	?
1	HTS	T
1	HILLTOP	?
1	HARTNELL	?
1	FAIRVIEW	?
1	E	D
1	DRAPER	?
1	DENVER	?
1	DEARING	?
1	CT	T
1	CRYSTAL	?
1	CANDLELIG	?
1	PEACH	?
1	PROSPECT	?
1	RAINBOW	?
1	RAVENNA	?
1	RIVIERA	?
1	ROOSEVELT	?
1	S	D
1	SPC	U
1	STE	U
1	SUNRIVER	?
1	SW	D
1	VIEWMONT	?
1	WAY	T

Figure 2-94 Create j05b_INVW_CONTACT_INFO_HOME_US_XXPREP 5/8


j05b_INVW_CONTACT_INFO_HOME_US_XXPREP..NAB_CONTACT_INFO_HOME_US_XXPREP_ADDRESS_PATTERN_REPORT.AP ..

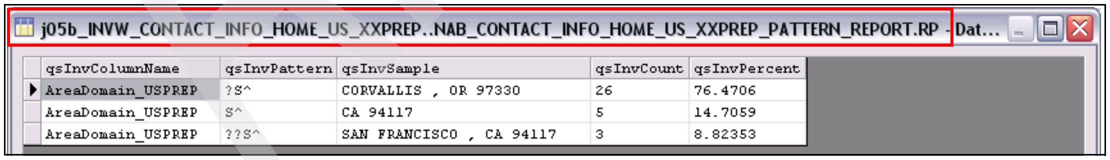
qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
AddressDomain_USPREP	^?T	3726 BRODERICK ST ,	9	26.4706
AddressDomain_USPREP	^?T	524 PROSPECT HTS ,	9	26.4706
AddressDomain_USPREP	^?T	7269 DEARING AVENUE	9	26.4706
AddressDomain_USPREP	^?T	2961 MADISON ST	9	26.4706
AddressDomain_USPREP	^?T	202 HARTNELL PL	9	26.4706
AddressDomain_USPREP	^?T	2280 ROOSEVELT BLVD	9	26.4706
AddressDomain_USPREP	^?T	831 WEBSTER ST ,	9	26.4706
AddressDomain_USPREP	^?T	917 PAGET AVE ,	9	26.4706
AddressDomain_USPREP	^?T	1061 SUNRIVER LN	9	26.4706
AddressDomain_USPREP	^D?T	725 N WILSON AVE	8	23.5294
AddressDomain_USPREP	^D?T	5988 W MENLO AVE	8	23.5294
AddressDomain_USPREP	^D?T	10150 S PEACH AVE	8	23.5294
AddressDomain_USPREP	^D?T	2620 NW LINNAN CIR	8	23.5294
AddressDomain_USPREP	^D?T	1990 SE CRYSTAL CIR	8	23.5294
AddressDomain_USPREP	^D?T	3065 SW FAIRVIEW BLVD	8	23.5294
AddressDomain_USPREP	^D?T	5126 SE RAINBOW LN	8	23.5294
AddressDomain_USPREP	^D?T	11112 SE WOOD AVE	8	23.5294
AddressDomain_USPREP	^?TD	874 DENVER PL NE	3	8.82353
AddressDomain_USPREP	^?TU^	425 MARKET ST , STE 2200 ,	3	8.82353
AddressDomain_USPREP	^?TD	7330 RAVENNA AVE NE	3	8.82353
AddressDomain_USPREP	^?TD	146 DRAPER ST NE	3	8.82353
AddressDomain_USPREP	^?TU^	1400 CANDLELIGHT DR , SPC 113	3	8.82353
AddressDomain_USPREP	^?TU^	679 HILLTOP DR , APT 40	3	8.82353
AddressDomain_USPREP	^>TU^	2250 24TH ST APT 334 ,	2	5.88235
AddressDomain_USPREP	^>TU^	1200 15TH AVE , APT 1 ,	2	5.88235
AddressDomain_USPREP	^??TD	8615 LA RIVIERA DR , E ,	2	5.88235
AddressDomain_USPREP	^??TD	2455 LA JOLLA CT NW	2	5.88235
AddressDomain_USPREP	^D?TD	2621 W VIEWMONT WAY W	1	2.94118
AddressDomain_USPREP	^>T	543 41ST AVE ,	1	2.94118
AddressDomain_USPREP	^D>TUI	514 NW 30TH ST , APT A	1	2.94118
AddressDomain_USPREP	^0T	1902 SECOND AVE	1	2.94118
AddressDomain_USPREP	^?DD	945 ARTHUR W NW	1	2.94118
AddressDomain_USPREP	^D>TU^	4601 SE 39TH AVE , APT 102	1	2.94118
AddressDomain_USPREP	^^TD	13055 23 PL NE	1	2.94118

Figure 2-95 Create j05b_INVW_CONTACT_INFO_HOME_US_XXPREP 6/8


j05b_INVW_CONTACT_INFO_HOME_US_XXPREP..NAB_CONTACT_INFO_HOME_US_XXPREP_AREA_TOKEN_REPORT.RT -...

qsInvCount	qsInvWord	qsInvClassCode
16	CA	S
14	OR	S
4	SEATTLE	?
4	PORTLAND	?
4	FRESNO	?
4	WA	S
4	SALEM	?
3	CORVALLIS	?
3	EUGENE	?
3	97402	^
2	CRUZ	?
2	97222	^
2	SANTA	?
2	SACRAMENT	?
2	REDDING	?
2	94117	^
2	97330	^
2	97301	^
2	97304	^
1	98125	^
1	98199	^
1	FRANCISCO	?
1	93720	^
1	SAN	?
1	97205	^
1	93725	^
1	97333	^
1	97202	^
1	96003	^
1	96001	^
1	95826	^
1	95825	^
1	95065	^
1	95062	^
1	94123	^
1	94122	^
1	94107	^
1	94105	^
1	93728	^
1	98101	^
1	98115	^
1	93722	^

Figure 2-96 Create j05b_INVW_CONTACT_INFO_HOME_US_XXPREP 7/8


j05b_INVW_CONTACT_INFO_HOME_US_XXPREP..NAB_CONTACT_INFO_HOME_US_XXPREP_PATTERN_REPORT.RP - Dat...

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
AreaDomain_USPREP	?S^	CORVALLIS , OR 97330	26	76.4706
AreaDomain_USPREP	S^	CA 94117	5	14.7059
AreaDomain_USPREP	??S^	SAN FRANCISCO , CA 94117	3	8.82353

Figure 2-97 Create j05b_INVW_CONTACT_INFO_HOME_US_XXPREP 8/8

j05c_INVW_CONTACT_INFO_WORK_XXPREP

In this step, we analyze the CONTACT_INFO file with the U.S. work addresses containing the name, address, and area buckets generated by the "j03_STAN_XXPREP_NAB" on page 530 job by the Investigate stage using word investigate and the domain-specific USADDR and USAREA rule sets to determine the degree of success that is achieved by the rule sets in identifying the tokens correctly.

Because a single Investigate stage can only have a single rule set associated with it, we copy the CONTACT_INFO file with the U.S. home addresses (using a Copy stage) and process the file by two independent Investigate stages, with each stage using a particular domain-specific rule set. Both the pattern and token reports are generated in each Investigate stage.

Figure 2-98 on page 586 shows the various stages that are used in this job, including the data set that was created in "j03_STAN_XXPREP_NAB" on page 530, a Copy stage, two Investigate stages that each use a different domain-specific rule set, and two sequential file stages (one each for the token report and pattern report) for each Investigate stage. We modified the names of the stages as shown.

Because the configuration of this job is very similar to that described in "J04_INVW_USPREP" on page 203, we do not repeat it here. However, some of the configurations of interest are as follows:

- Figure 2-99 on page 587 shows the INVADDR - Investigate Stage window using the USADDR rule set with the Word Investigate option. The selected column is AddressDomain_USPREP_WORK.
- Figure 2-100 on page 587 shows the INVAREA - Investigate Stage window using the USAREA rule set with the Word Investigate option. The selected column is AreaDomain_USPREP_WORK.
- After saving, compiling, and running this job (Figure 2-107 on page 594), the contents of the output of the Investigate stage are written to the sequential files shown in Figure 2-102 on page 589 through Figure 2-106 on page 592.

  The reports show a number of tokens in the various reports that are not recognized with proper classifications (code ?).

Proceed to "j05d_INVW_DRIVER_XXPREP" on page 592.

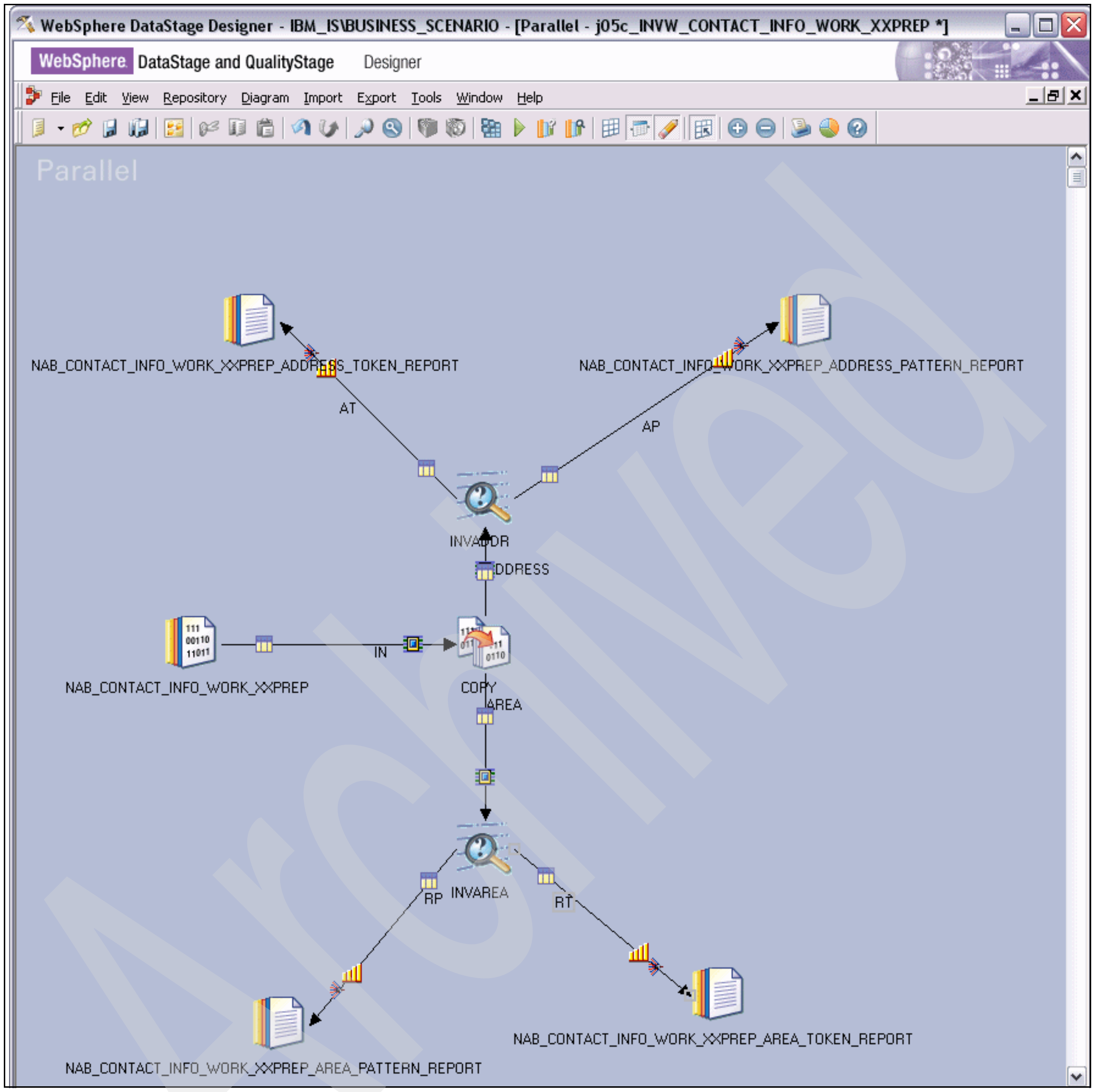
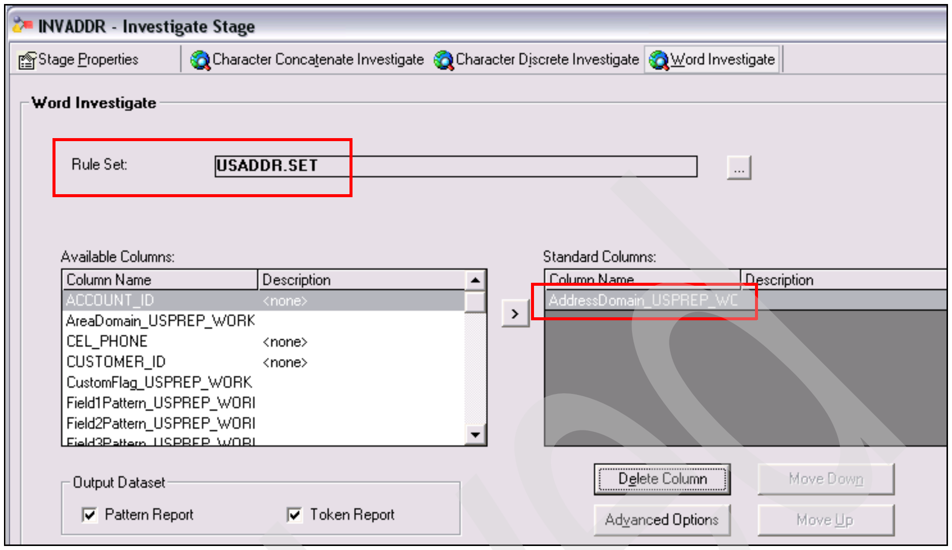

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j05c_INVW_CONTACT_INFO_WORK_XXPREP *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NAB_CONTACT_INFO_WORK_XXPREP_ADDRESS_TOKEN_REPORT
NAB_CONTACT_INFO_WORK_XXPREP_ADDRESS_PATTERN_REPORT
AT
AP
INVADDR
ADDRESS
NAB_CONTACT_INFO_WORK_XXPREP
IN
COPY
AREA
INVAREA
RP
RT
NAB_CONTACT_INFO_WORK_XXPREP_AREA_TOKEN_REPORT
NAB_CONTACT_INFO_WORK_XXPREP_AREA_PATTERN_REPORT

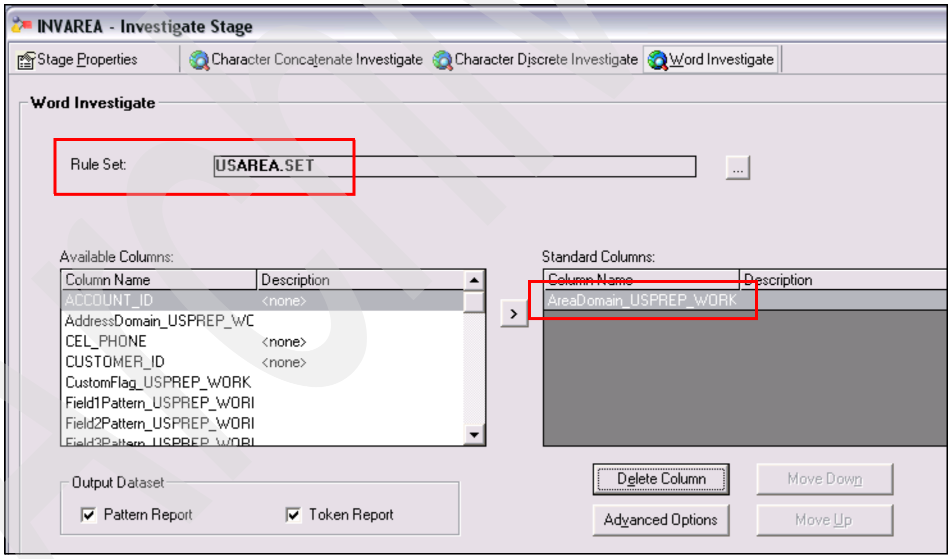
Figure 2-98 Create j05c_INVW_CONTACT_INFO_WORK_XXPREP 1/9



Column Name	Description
ACCOUNT_ID	<none>
AreaDomain_USPREP_WORK	
CEL_PHONE	<none>
CUSTOMER_ID	<none>
CustomFlag_USPREP_WORK	
Field1Pattern_USPREP_WORI	
Field2Pattern_USPREP_WORI	

Column Name	Description
AddressDomain_USPREP_WC	

Figure 2-99 Create j05c_INVW_CONTACT_INFO_WORK_XXPREP 2/9



Column Name	Description
ACCOUNT_ID	<none>
AddressDomain_USPREP_WC	
CEL_PHONE	<none>
CUSTOMER_ID	<none>
CustomFlag_USPREP_WORK	
Field1Pattern_USPREP_WORI	
Field2Pattern_USPREP_WORI	

Column Name	Description
AreaDomain_USPREP_WORK	

Figure 2-100 Create j05c_INVW_CONTACT_INFO_WORK_XXPREP 3/9

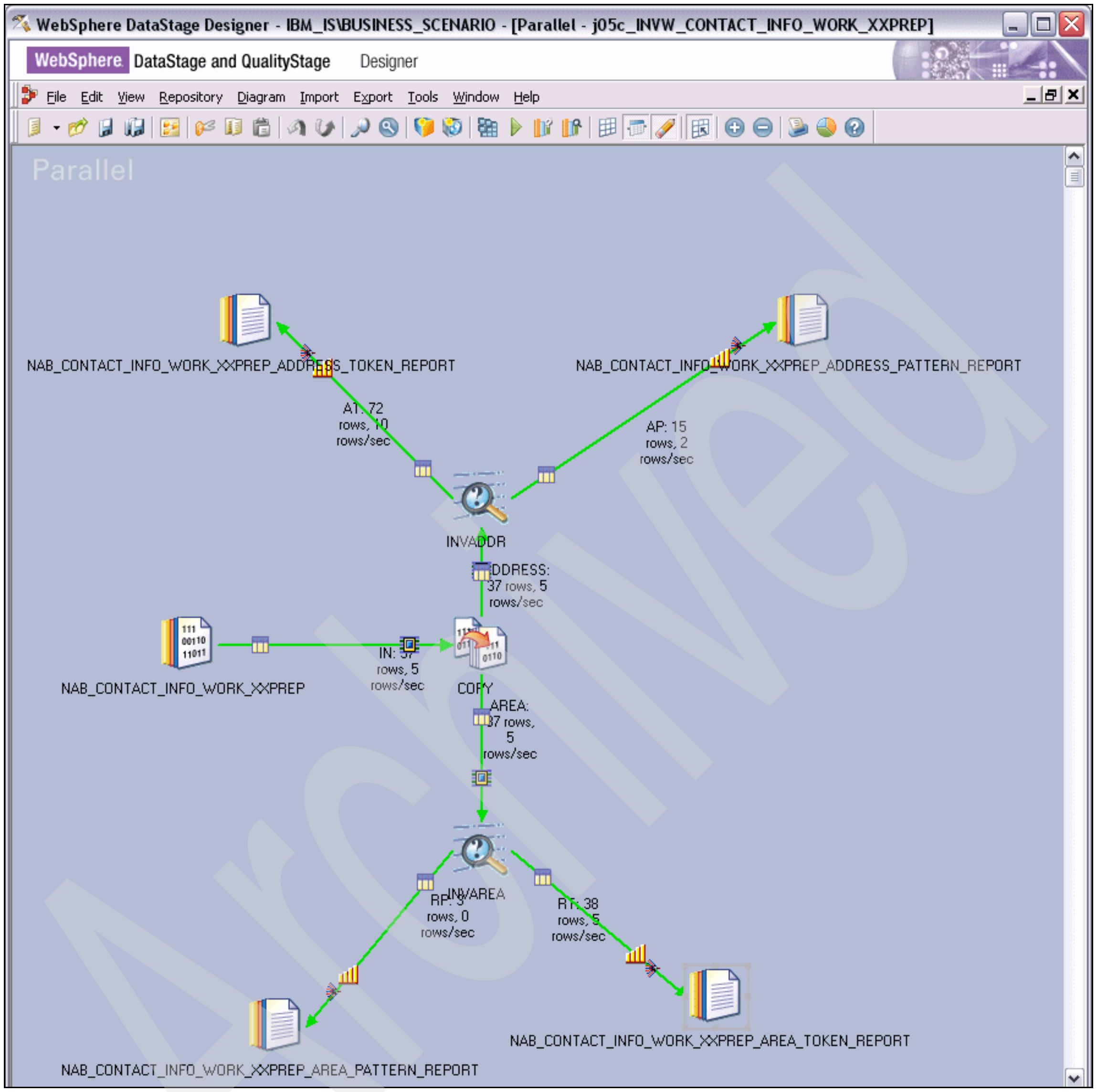
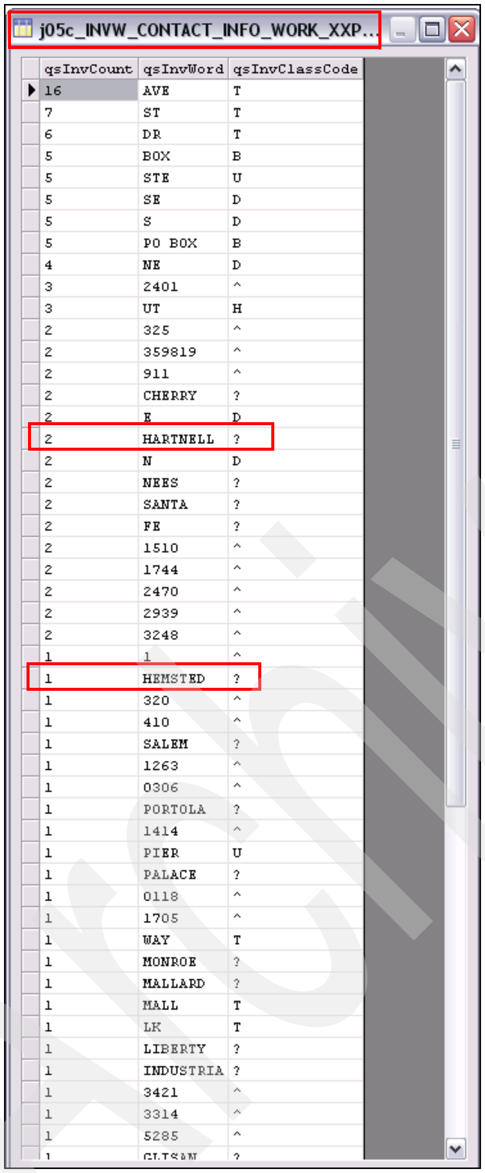

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j05c_INVW_CONTACT_INFO_WORK_XXPREP]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NAB_CONTACT_INFO_WORK_XXPREP_ADDRESS_TOKEN_REPORT
NAB_CONTACT_INFO_WORK_XXPREP_ADDRESS_PATTERN_REPORT
AT: 72 rows, 10 rows/sec
AP: 15 rows, 2 rows/sec
INVADDR
ADDRESS: 37 rows, 5 rows/sec
NAB_CONTACT_INFO_WORK_XXPREP
IN: 37 rows, 5 rows/sec
COPY
AREA: 37 rows, 5 rows/sec
INVAREA
RP: 3 rows, 0 rows/sec
RT: 38 rows, 5 rows/sec
NAB_CONTACT_INFO_WORK_XXPREP_AREA_TOKEN_REPORT
NAB_CONTACT_INFO_WORK_XXPREP_AREA_PATTERN_REPORT

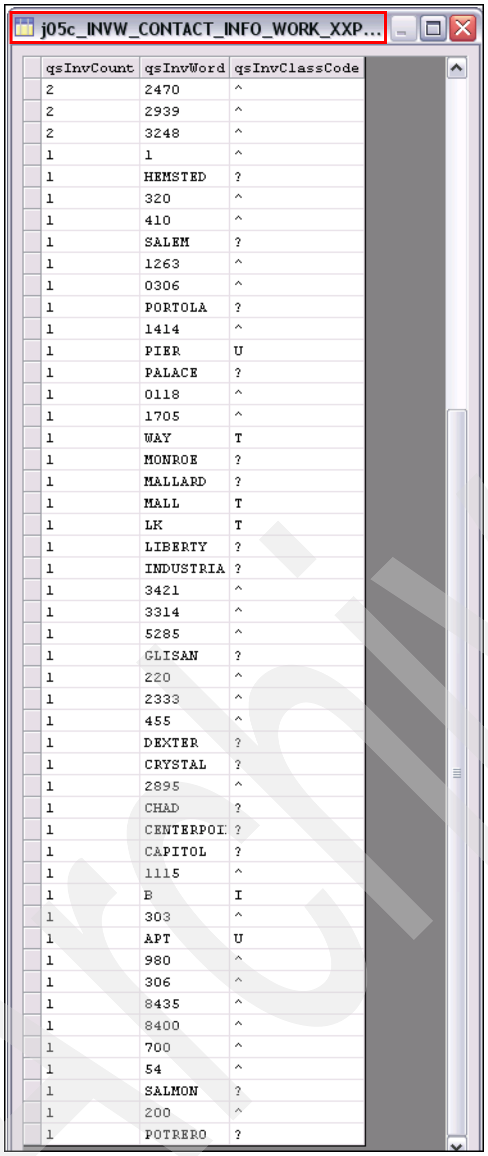
Figure 2-101 Create j05c_INVW_CONTACT_INFO_WORK_XXPREP 4/9



j05c_INVW_CONTACT_INFO_WORK_XXP...

qsInvCount	qsInvWord	qsInvClassCode
16	AVE	T
7	ST	T
6	DR	T
5	BOX	B
5	STE	U
5	SE	D
5	S	D
5	PO BOX	B
4	NE	D
3	2401	^
3	UT	H
2	325	^
2	359819	^
2	911	^
2	CHERRY	?
2	E	D
2	HARTNELL	?
2	N	D
2	NEES	?
2	SANTA	?
2	FE	?
2	1510	^
2	1744	^
2	2470	^
2	2939	^
2	3248	^
1	1	^
1	HEMSTED	?
1	320	^
1	410	^
1	SALEM	?
1	1263	^
1	0306	^
1	PORTOLA	?
1	1414	^
1	PIER	U
1	PALACE	?
1	0118	^
1	1705	^
1	WAY	T
1	MONROE	?
1	MALLARD	?
1	MALL	T
1	LK	T
1	LIBERTY	?
1	INDUSTRIA	?
1	3421	^
1	3314	^
1	5285	^
1	GLISAN	?

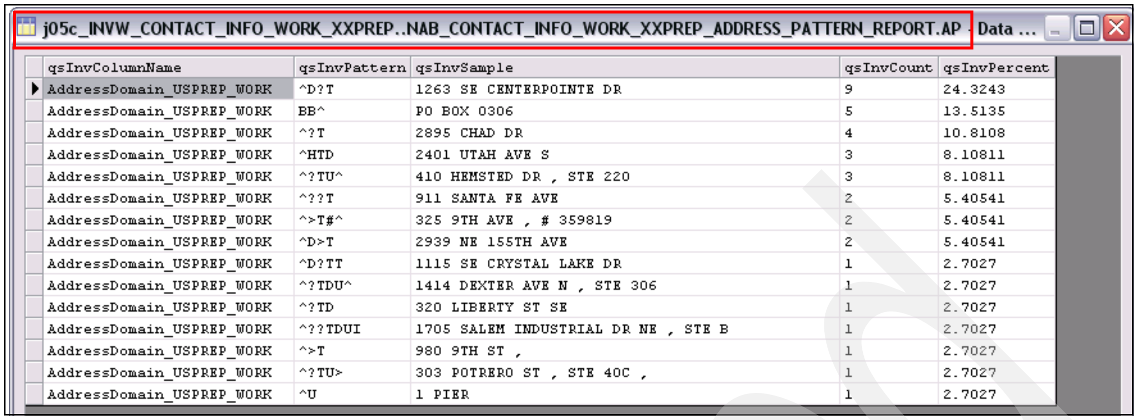
Figure 2-102 Create j05c_INVW_CONTACT_INFO_WORK_XXPREP 5/9



j05c_INVW_CONTACT_INFO_WORK_XXP...

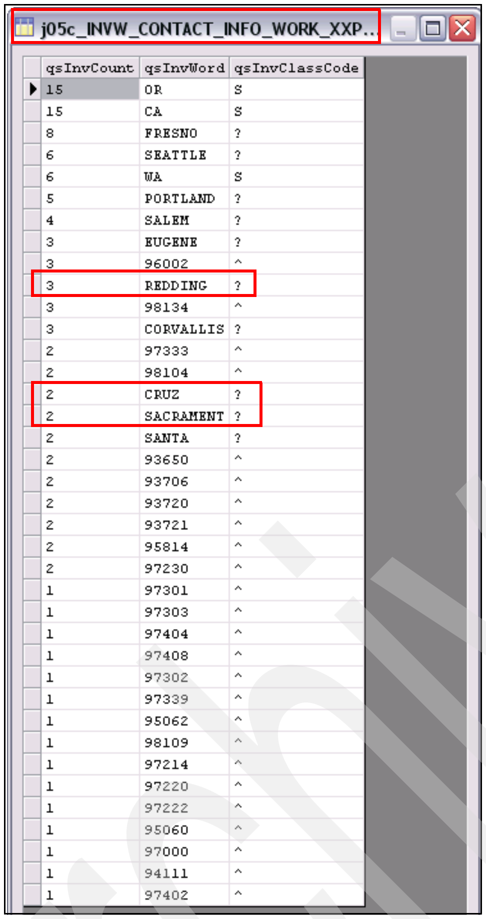
qsInvCount	qsInvWord	qsInvClassCode
2	2470	^
2	2939	^
2	3248	^
1	1	^
1	HEMSTED	?
1	320	^
1	410	^
1	SALEM	?
1	1263	^
1	0306	^
1	PORTOLA	?
1	1414	^
1	PIER	U
1	PALACE	?
1	0118	^
1	1705	^
1	WAY	T
1	MONROE	?
1	MALLARD	?
1	MALL	T
1	LK	T
1	LIBERTY	?
1	INDUSTRIA	?
1	3421	^
1	3314	^
1	5285	^
1	GLISAN	?
1	220	^
1	2333	^
1	455	^
1	DEXTER	?
1	CRYSTAL	?
1	2895	^
1	CHAD	?
1	CENTERPOI	?
1	CAPITOL	?
1	1115	^
1	B	I
1	303	^
1	APT	U
1	980	^
1	306	^
1	8435	^
1	8400	^
1	700	^
1	54	^
1	SALMON	?
1	200	^
1	POTRERO	?

Figure 2-103 Create j05c_INVW_CONTACT_INFO_WORK_XXPREP 6/9


j05c_INVW_CONTACT_INFO_WORK_XXPREP..NAB_CONTACT_INFO_WORK_XXPREP_ADDRESS_PATTERN_REPORT.AP - Data ...

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
AddressDomain_USPREP_WORK	^D?T	1263 SE CENTERPOINTE DR	9	24.3243
AddressDomain_USPREP_WORK	BB^	PO BOX 0306	5	13.5135
AddressDomain_USPREP_WORK	^?T	2895 CHAD DR	4	10.8108
AddressDomain_USPREP_WORK	^HTD	2401 UTAH AVE S	3	8.10811
AddressDomain_USPREP_WORK	^?TU^	410 HEMSTED DR , STE 220	3	8.10811
AddressDomain_USPREP_WORK	^??T	911 SANTA FE AVE	2	5.40541
AddressDomain_USPREP_WORK	^>T#^	325 9TH AVE , # 359819	2	5.40541
AddressDomain_USPREP_WORK	^D>T	2939 NE 155TH AVE	2	5.40541
AddressDomain_USPREP_WORK	^D?TT	1115 SE CRYSTAL LAKE DR	1	2.7027
AddressDomain_USPREP_WORK	^?TDU^	1414 DEXTER AVE N , STE 306	1	2.7027
AddressDomain_USPREP_WORK	^?TD	320 LIBERTY ST SE	1	2.7027
AddressDomain_USPREP_WORK	^??TDUI	1705 SALEM INDUSTRIAL DR NE , STE B	1	2.7027
AddressDomain_USPREP_WORK	^>T	980 9TH ST ,	1	2.7027
AddressDomain_USPREP_WORK	^?TU>	303 POTRERO ST , STE 40C ,	1	2.7027
AddressDomain_USPREP_WORK	^U	1 PIER	1	2.7027

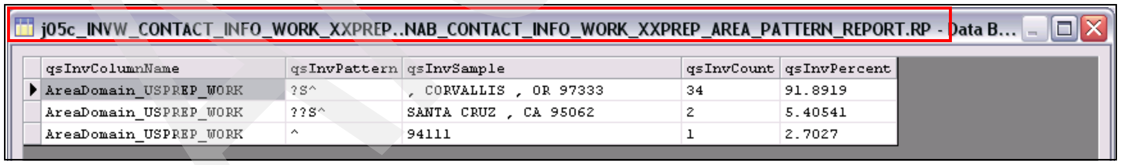
Figure 2-104 Create j05c_INVW_CONTACT_INFO_WORK_XXPREP 7/9



j05c_INVW_CONTACT_INFO_WORK_XXP...

qsInvCount	qsInvWord	qsInvClassCode
15	OR	S
15	CA	S
8	FRESNO	?
6	SEATTLE	?
6	WA	S
5	PORTLAND	?
4	SALEM	?
3	EUGENE	?
3	96002	^
3	REDDING	?
3	98134	^
3	CORVALLIS	?
2	97333	^
2	98104	^
2	CRUZ	?
2	SACRAMENT	?
2	SANTA	?
2	93650	^
2	93706	^
2	93720	^
2	93721	^
2	95814	^
2	97230	^
1	97301	^
1	97303	^
1	97404	^
1	97408	^
1	97302	^
1	97339	^
1	95062	^
1	98109	^
1	97214	^
1	97220	^
1	97222	^
1	95060	^
1	97000	^
1	94111	^
1	97402	^

Figure 2-105   Create j05c_INVW_CONTACT_INFO_WORK_XXPREP 8/9



j05c_INVW_CONTACT_INFO_WORK_XXPREP..NAB_CONTACT_INFO_WORK_XXPREP_AREA_PATTERN_REPORT.RP - Data B...

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
AreaDomain_USPREP_WORK	?S^	, CORVALLIS , OR 97333	34	91.8919
AreaDomain_USPREP_WORK	??S^	SANTA CRUZ , CA 95062	2	5.40541
AreaDomain_USPREP_WORK	^	94111	1	2.7027

Figure 2-106   Create j05c_INVW_CONTACT_INFO_WORK_XXPREP 9/9

j05d_INVW_DRIVER_XXPREP

In this step, we analyze the DRIVER file with the U.S. addresses containing the name, address, and area buckets that were generated by the “j03_STAN_XXPREP_NAB” on page 530 job by the Investigate stage using word investigate and the domain-specific USNAME, USADDR, and USAREA rule sets to determine the degree of success that is achieved by the rule sets in identifying the tokens correctly.

Because a single Investigate stage can only have a single rule set associated with it, we split the DRIVER file with the U.S. addresses (using a Copy stage) and process the files by three independent Investigate stages, with each stage using a particular domain-specific rule set. Both the pattern and token reports are generated in each Investigate stage.

Figure 2-107 on page 594 shows the various stages that are used in this job, including the data set that was created in “j03_STAN_XXPREP_NAB” on page 530, a Copy stage, three Investigate stages that each use a different domain-specific rule set, and two sequential file stages (one each for the token report and pattern report) for each Investigate stage. We modified the names of the stages as shown.

Because the configuration of this job is very similar to that described in “J04_INVW_USPREP” on page 203, we do not repeat it here. However, some of the configurations of interest are as follows:

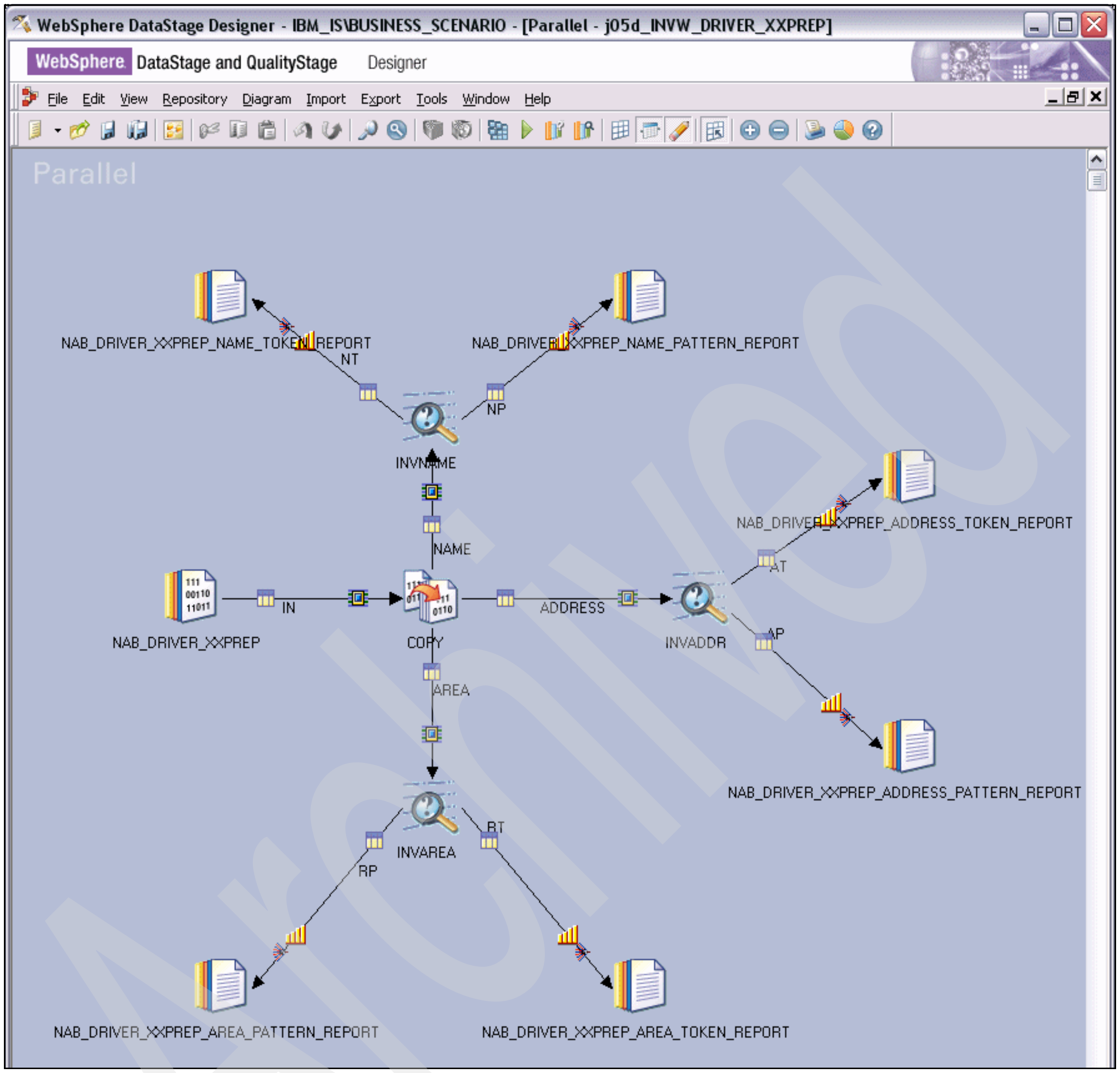
- Figure 2-108 on page 595 shows the INVNAME - Investigate Stage window using the USNAME rule set with the Word Investigate option. The selected column is NameDomain_USPREP.

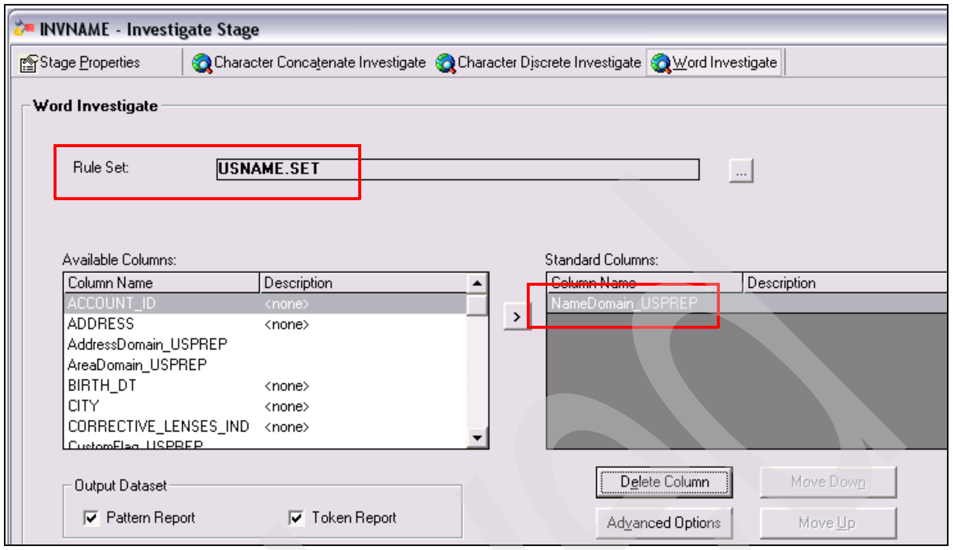
  The INVADDR - Investigate Stage and INVAREA - Investigate Stage windows are similar to those described in Figure 2-91 on page 580 and Figure 2-92 on page 580. We do not repeat them here.

- After saving, compiling, and running this job (Figure 2-109 on page 596), the contents of the output of the Investigate stage are written to the sequential files shown in Figure 2-110 on page 597 through Figure 2-116 on page 602.

  The reports show a number of tokens in the various reports not being recognized with proper classifications (code ?).

Proceed now to “j05e_INVW_CUSTOMER_XXPREP” on page 603.


WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j05d_INVW_DRIVER_XXPREP]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NAB_DRIVER_XXPREP_NAME_TOKEN_REPORT
NT
NAB_DRIVER_XXPREP_NAME_PATTERN_REPORT
NP
INVNAME
NAME
NAB_DRIVER_XXPREP_ADDRESS_TOKEN_REPORT
AT
NAB_DRIVER_XXPREP
IN
COPY
ADDRESS
INVADDR
AP
AREA
NAB_DRIVER_XXPREP_ADDRESS_PATTERN_REPORT
INVAREA
RT
RP
NAB_DRIVER_XXPREP_AREA_PATTERN_REPORT
NAB_DRIVER_XXPREP_AREA_TOKEN_REPORT
Figure 2-107   Create j05d_INVW_DRIVER_XXPREP 1/10



INVNAME - Investigate Stage

Stage Properties | Character Concatenate Investigate | Character Discrete Investigate | Word Investigate

Word Investigate

Rule Set: USNAME.SET

Available Columns:

Column Name	Description
ACCOUNT_ID	<none>
ADDRESS	<none>
AddressDomain_USPREP	
AreaDomain_USPREP	
BIRTH_DT	<none>
CITY	<none>
CORRECTIVE_LENSES_IND	<none>
CustomFlag_USPREP	

Standard Columns:

Column Name	Description
NameDomain_USPREP	

Output Dataset

☑ Pattern Report ☑ Token Report

Figure 2-108 Create j05d_INVW_DRIVER_XXPREP 2/10

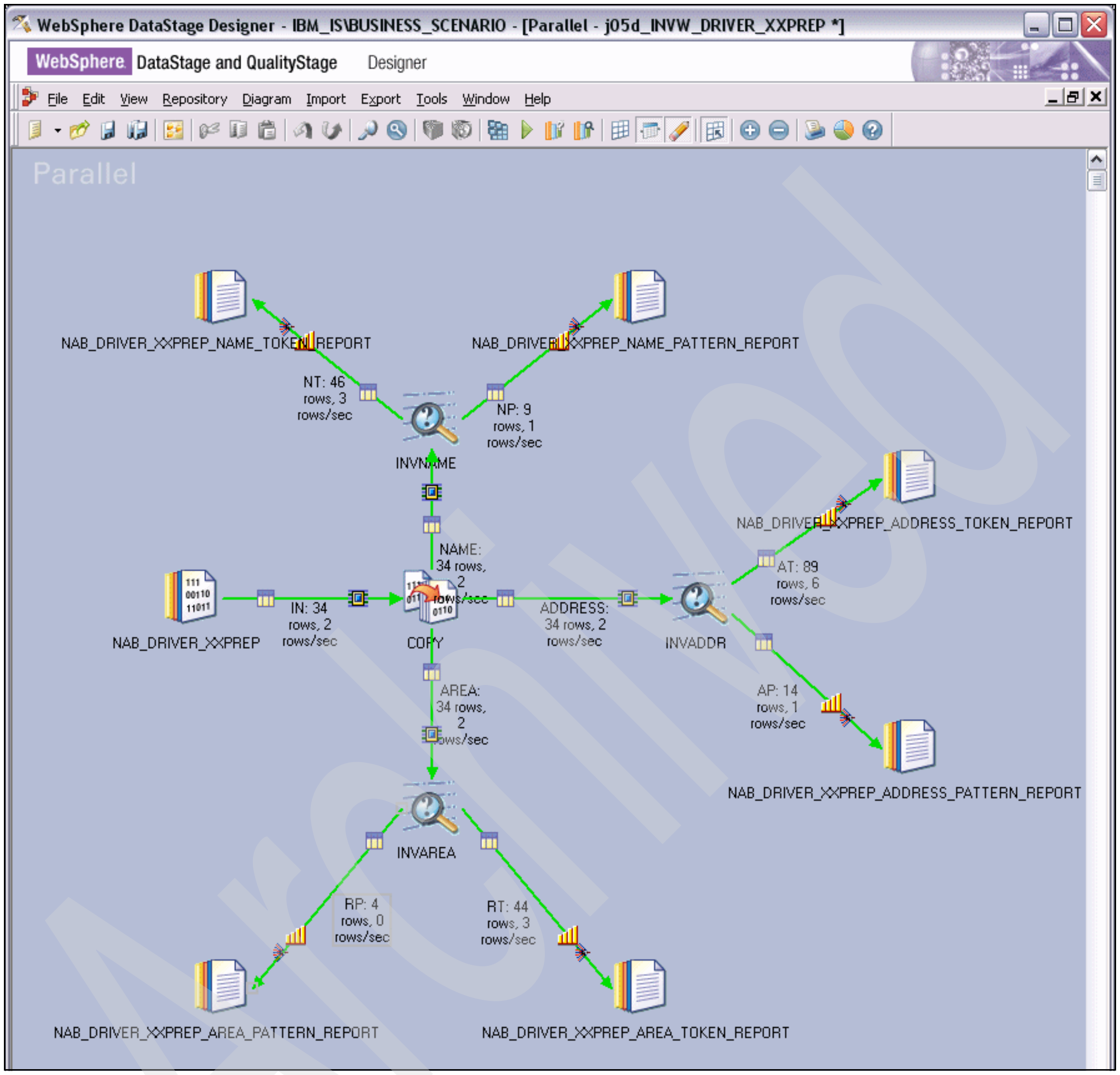
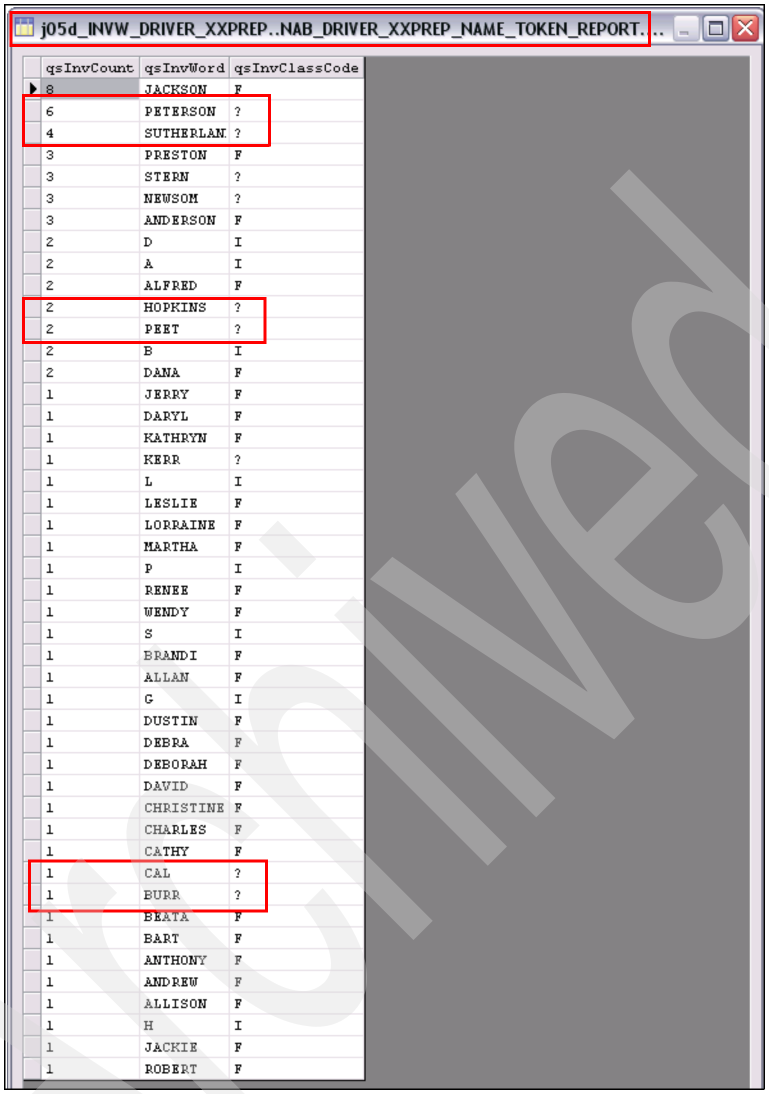
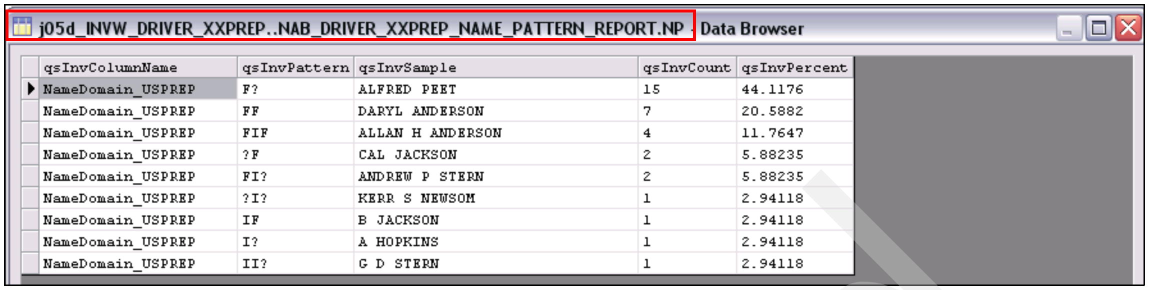

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j05d_INVW_DRIVER_XXPREP *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NAB_DRIVER_XXPREP_NAME_TOKEN_REPORT
NAB_DRIVER_XXPREP_NAME_PATTERN_REPORT
NT: 46 rows, 3 rows/sec
NP: 9 rows, 1 rows/sec
INVNAME
NAME: 34 rows, 2 rows/sec
NAB_DRIVER_XXPREP_ADDRESS_TOKEN_REPORT
AT: 89 rows, 6 rows/sec
NAB_DRIVER_XXPREP
IN: 34 rows, 2 rows/sec
COPY
ADDRESS: 34 rows, 2 rows/sec
INVADDR
AP: 14 rows, 1 rows/sec
AREA: 34 rows, 2 rows/sec
NAB_DRIVER_XXPREP_ADDRESS_PATTERN_REPORT
INVAREA
RP: 4 rows, 0 rows/sec
RT: 44 rows, 3 rows/sec
NAB_DRIVER_XXPREP_AREA_PATTERN_REPORT
NAB_DRIVER_XXPREP_AREA_TOKEN_REPORT

Figure 2-109 Create j05d_INVW_DRIVER_XXPREP 3/10


j05d_INVW_DRIVER_XXPREP..NAB_DRIVER_XXPREP_NAME_TOKEN_REPORT...

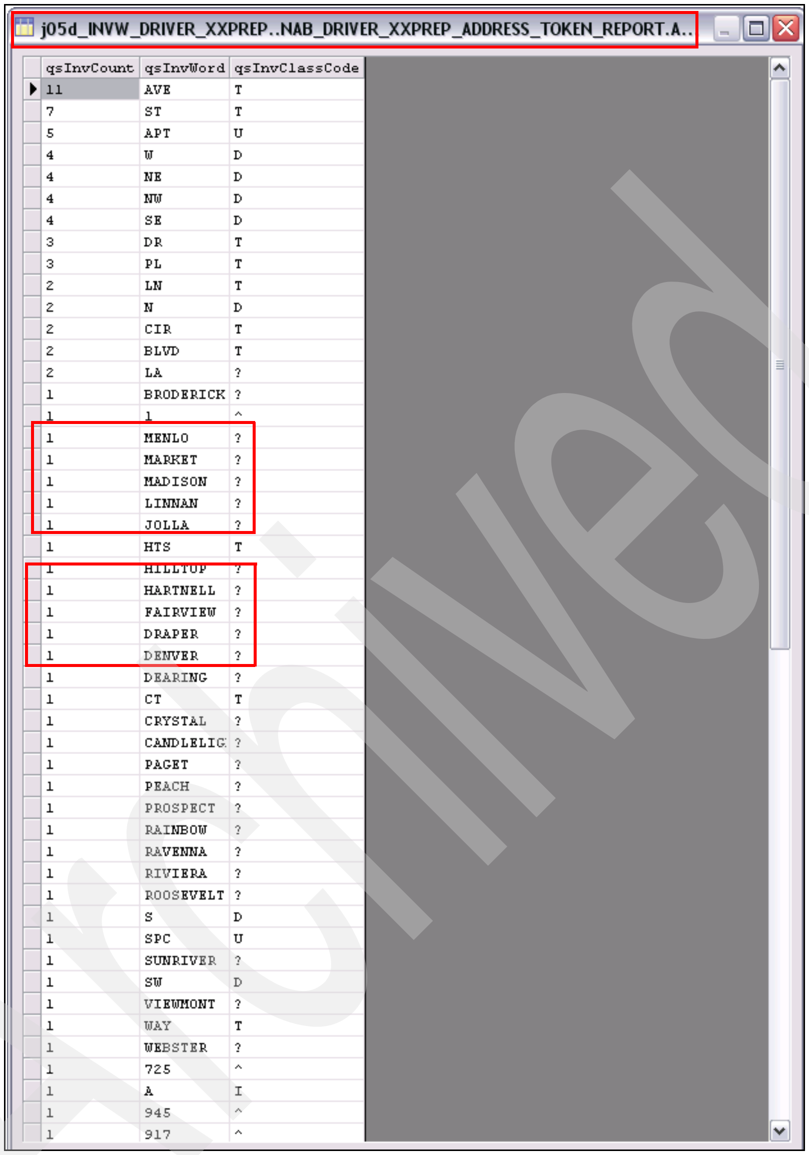
qsInvCount	qsInvWord	qsInvClassCode
8	JACKSON	F
6	PETERSON	?
4	SUTHERLAN	?
3	PRESTON	F
3	STERN	?
3	NEWSOM	?
3	ANDERSON	F
2	D	I
2	A	I
2	ALFRED	F
2	HOPKINS	?
2	PEET	?
2	B	I
2	DANA	F
1	JERRY	F
1	DARYL	F
1	KATHRYN	F
1	KERR	?
1	L	I
1	LESLIE	F
1	LORRAINE	F
1	MARTHA	F
1	P	I
1	RENEE	F
1	WENDY	F
1	S	I
1	BRANDI	F
1	ALLAN	F
1	G	I
1	DUSTIN	F
1	DEBRA	F
1	DEBORAH	F
1	DAVID	F
1	CHRISTINE	F
1	CHARLES	F
1	CATHY	F
1	CAL	?
1	BURR	?
1	BEATA	F
1	BART	F
1	ANTHONY	F
1	ANDREW	F
1	ALLISON	F
1	H	I
1	JACKIE	F
1	ROBERT	F

Figure 2-110   Create j05d_INVW_DRIVER_XXPREP 4/10


j05d_INVW_DRIVER_XXPREP..NAB_DRIVER_XXPREP_NAME_PATTERN_REPORT.NP - Data Browser

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
NameDomain_USPREP	F?	ALFRED PEET	15	44.1176
NameDomain_USPREP	FF	DARYL ANDERSON	7	20.5882
NameDomain_USPREP	FIF	ALLAN H ANDERSON	4	11.7647
NameDomain_USPREP	?F	CAL JACKSON	2	5.88235
NameDomain_USPREP	FI?	ANDREW P STERN	2	5.88235
NameDomain_USPREP	?I?	KERR S NEWSOM	1	2.94118
NameDomain_USPREP	IF	B JACKSON	1	2.94118
NameDomain_USPREP	I?	A HOPKINS	1	2.94118
NameDomain_USPREP	II?	G D STERN	1	2.94118

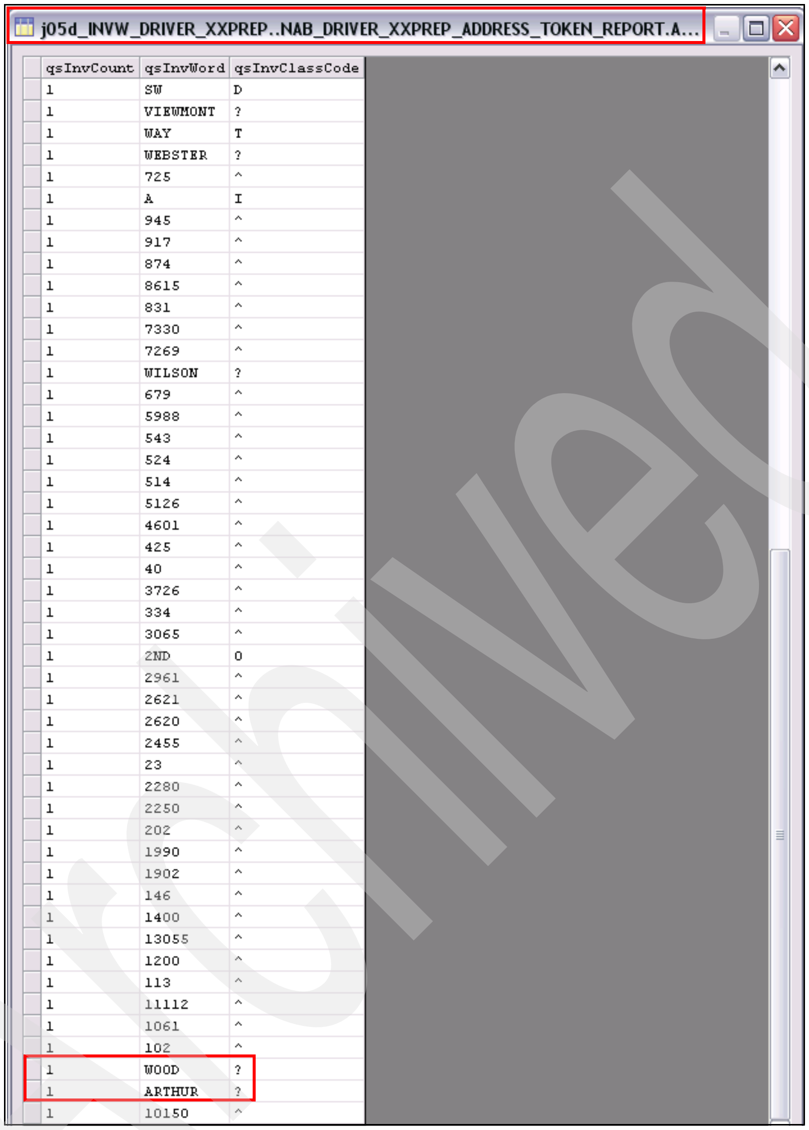
Figure 2-111 Create j05d_INVW_DRIVER_XXPREP 5/10



j05d_INVW_DRIVER_XXPREP..NAB_DRIVER_XXPREP_ADDRESS_TOKEN_REPORT.A..

qsInvCount	qsInvWord	qsInvClassCode
11	AVE	T
7	ST	T
5	APT	U
4	W	D
4	NE	D
4	NW	D
4	SE	D
3	DR	T
3	PL	T
2	LN	T
2	N	D
2	CIR	T
2	BLVD	T
2	LA	?
1	BRODERICK	?
1	1	^
1	MENLO	?
1	MARKET	?
1	MADISON	?
1	LINNAN	?
1	JOLLA	?
1	HTS	T
1	HILLTOP	?
1	HARTNELL	?
1	FAIRVIEW	?
1	DRAPER	?
1	DENVER	?
1	DEARING	?
1	CT	T
1	CRYSTAL	?
1	CANDLELIG	?
1	PAGET	?
1	PEACH	?
1	PROSPECT	?
1	RAINBOW	?
1	RAVENNA	?
1	RIVIERA	?
1	ROOSEVELT	?
1	S	D
1	SPC	U
1	SUNRIVER	?
1	SW	D
1	VIEWMONT	?
1	WAY	T
1	WEBSTER	?
1	725	^
1	A	I
1	945	^
1	917	^

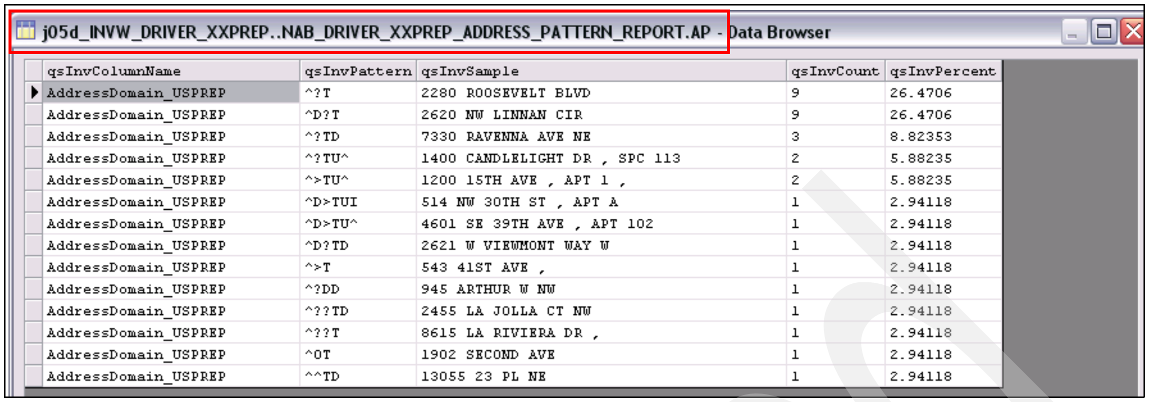
Figure 2-112 Create j05d_INVW_DRIVER_XXPREP 6/10



j05d_INVW_DRIVER_XXPREP..NAB_DRIVER_XXPREP_ADDRESS_TOKEN_REPORT.A...

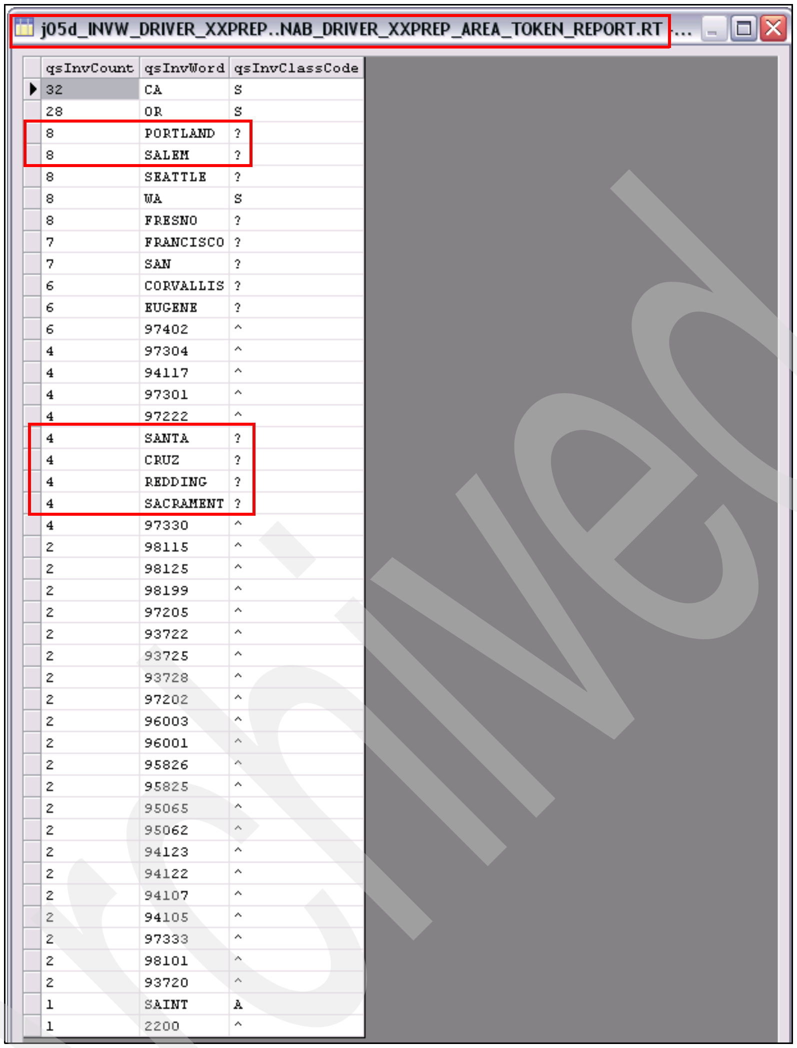
qsInvCount	qsInvWord	qsInvClassCode
1	SW	D
1	VIEWMONT	?
1	WAY	T
1	WEBSTER	?
1	725	^
1	A	I
1	945	^
1	917	^
1	874	^
1	8615	^
1	831	^
1	7330	^
1	7269	^
1	WILSON	?
1	679	^
1	5988	^
1	543	^
1	524	^
1	514	^
1	5126	^
1	4601	^
1	425	^
1	40	^
1	3726	^
1	334	^
1	3065	^
1	2ND	0
1	2961	^
1	2621	^
1	2620	^
1	2455	^
1	23	^
1	2280	^
1	2250	^
1	202	^
1	1990	^
1	1902	^
1	146	^
1	1400	^
1	13055	^
1	1200	^
1	113	^
1	11112	^
1	1061	^
1	102	^
1	WOOD	?
1	ARTHUR	?
1	10150	^

Figure 2-113 Create j05d_INVW_DRIVER_XXPREP 7/10


j05d_INVW_DRIVER_XXPREP..NAB_DRIVER_XXPREP_ADDRESS_PATTERN_REPORT.AP - Data Browser

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
AddressDomain_USPREP	^?T	2280 ROOSEVELT BLVD	9	26.4706
AddressDomain_USPREP	^D?T	2620 NW LINNAN CIR	9	26.4706
AddressDomain_USPREP	^?TD	7330 RAVENNA AVE NE	3	8.82353
AddressDomain_USPREP	^?TU^	1400 CANDLELIGHT DR , SPC 113	2	5.88235
AddressDomain_USPREP	^>TU^	1200 15TH AVE , APT 1 ,	2	5.88235
AddressDomain_USPREP	^D>TUI	514 NW 30TH ST , APT A	1	2.94118
AddressDomain_USPREP	^D>TU^	4601 SE 39TH AVE , APT 102	1	2.94118
AddressDomain_USPREP	^D?TD	2621 W VIEWMONT WAY W	1	2.94118
AddressDomain_USPREP	^>T	543 41ST AVE ,	1	2.94118
AddressDomain_USPREP	^?DD	945 ARTHUR W NW	1	2.94118
AddressDomain_USPREP	^??TD	2455 LA JOLLA CT NW	1	2.94118
AddressDomain_USPREP	^??T	8615 LA RIVIERA DR ,	1	2.94118
AddressDomain_USPREP	^0T	1902 SECOND AVE	1	2.94118
AddressDomain_USPREP	^^TD	13055 23 PL NE	1	2.94118

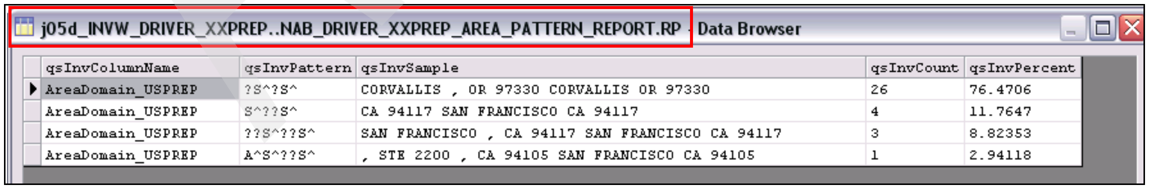
Figure 2-114 Create j05d_INVW_DRIVER_XXPREP 8/10



j05d_INVW_DRIVER_XXPREP..NAB_DRIVER_XXPREP_AREA_TOKEN_REPORT.RT ...

qsInvCount	qsInvWord	qsInvClassCode
32	CA	S
28	OR	S
8	PORTLAND	?
8	SALEM	?
8	SEATTLE	?
8	WA	S
8	FRESNO	?
7	FRANCISCO	?
7	SAN	?
6	CORVALLIS	?
6	EUGENE	?
6	97402	^
4	97304	^
4	94117	^
4	97301	^
4	97222	^
4	SANTA	?
4	CRUZ	?
4	REDDING	?
4	SACRAMENT	?
4	97330	^
2	98115	^
2	98125	^
2	98199	^
2	97205	^
2	93722	^
2	93725	^
2	93728	^
2	97202	^
2	96003	^
2	96001	^
2	95826	^
2	95825	^
2	95065	^
2	95062	^
2	94123	^
2	94122	^
2	94107	^
2	94105	^
2	97333	^
2	98101	^
2	93720	^
1	SAINT	A
1	2200	^

Figure 2-115 Create j05d_INVW_DRIVER_XXPREP 9/10



j05d_INVW_DRIVER_XXPREP..NAB_DRIVER_XXPREP_AREA_PATTERN_REPORT.RP - Data Browser

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
AreaDomain_USPREP	?S^?S^	CORVALLIS , OR 97330 CORVALLIS OR 97330	26	76.4706
AreaDomain_USPREP	S^??S^	CA 94117 SAN FRANCISCO CA 94117	4	11.7647
AreaDomain_USPREP	??S^??S^	SAN FRANCISCO , CA 94117 SAN FRANCISCO CA 94117	3	8.82353
AreaDomain_USPREP	A^S^??S^	, STE 2200 , CA 94105 SAN FRANCISCO CA 94105	1	2.94118

Figure 2-116 Create j05d_INVW_DRIVER_XXPREP 10/10

j05e_INVW_CUSTOMER_XXPREP

In this step, we analyze the CUSTOMER file with the U.S. addresses containing the name, address, and area buckets that were generated by the "j03_STAN_XXPREP_NAB" on page 530 job by the Investigate stage using word investigate and the domain-specific USNAME rule set to determine the degree of success achieved by the rule set in identifying the tokens correctly.

Figure 2-117 on page 603 shows the various stages in this job (as well as the statistics collected after compiling and running the job). It includes the data set that was created in "j03_STAN_XXPREP_NAB" on page 530 and one Investigate stage that uses the USNAME domain-specific rule set (because it only has name information and no address information), and two sequential file stages (one each for the token report and pattern report). We modified the names of the stages as shown.

Because the configuration of this job is very similar to that described in "J04_INVW_USPREP" on page 203, we do not repeat it here. However, some of the configurations of interest are as follows:

- The outputs of the Investigate stage written to the sequential files are shown in Figure 2-118 on page 604 through Figure 2-119 on page 605.

  The reports show a few tokens in the various reports that are not recognized with proper classifications (code ?).

Proceed now to "j06_XXPREP_CASS" on page 605.

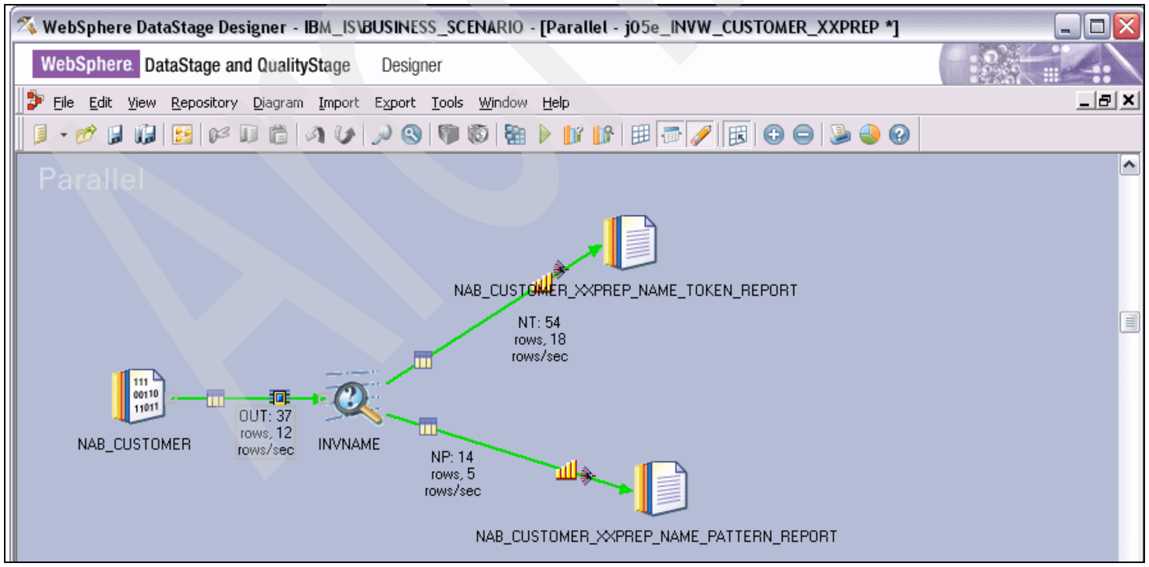
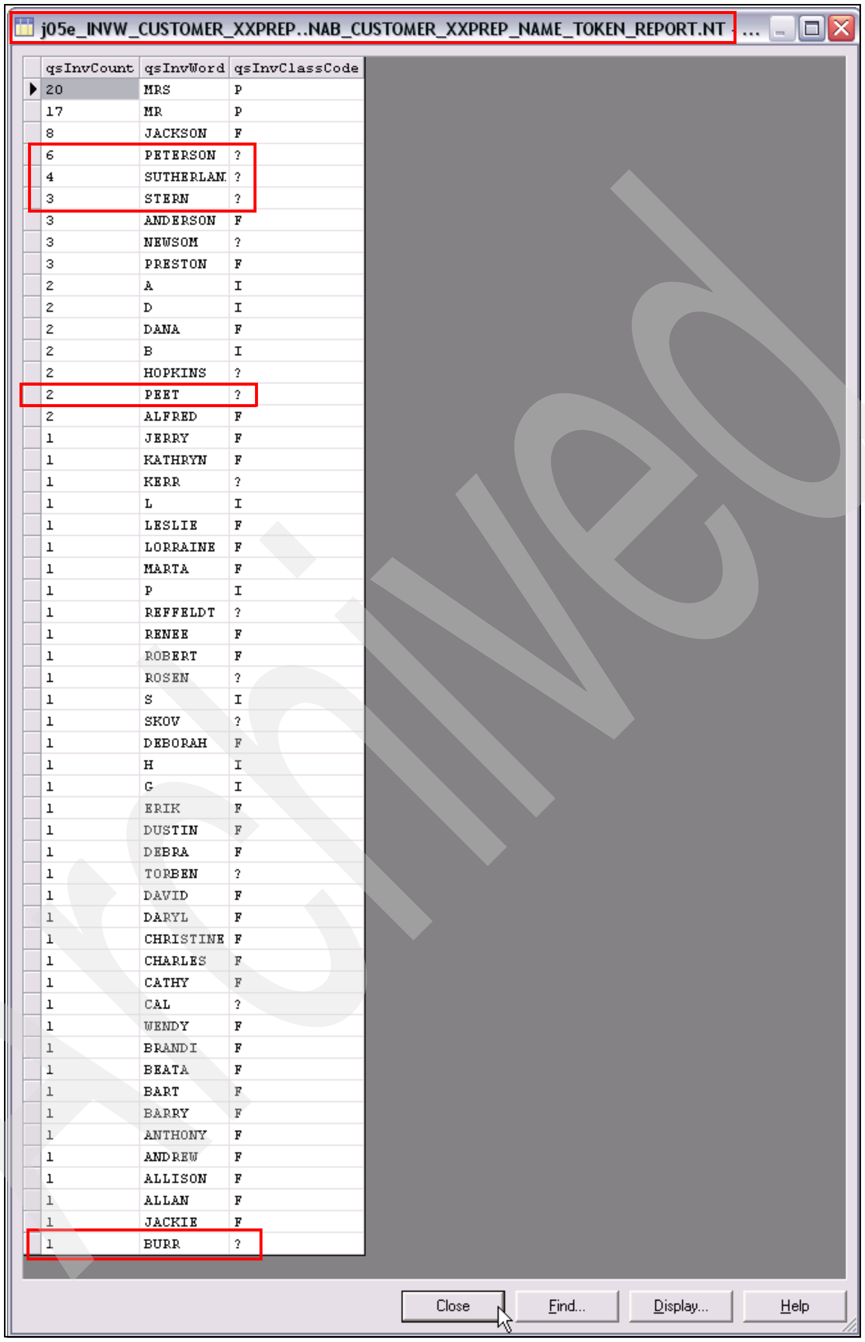
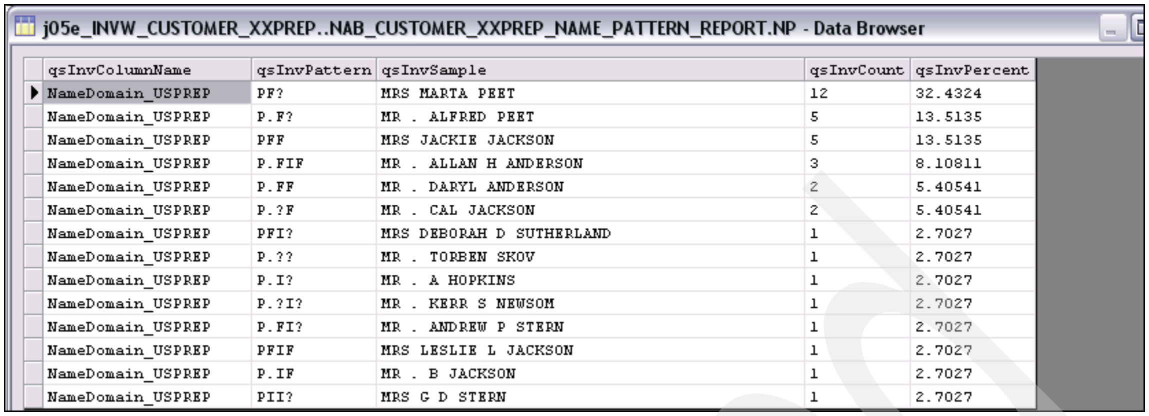

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j05e_INVW_CUSTOMER_XXPREP *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NAB_CUSTOMER
OUT: 37 rows, 12 rows/sec
INVNAME
NT: 54 rows, 18 rows/sec
NAB_CUSTOMER_XXPREP_NAME_TOKEN_REPORT
NP: 14 rows, 5 rows/sec
NAB_CUSTOMER_XXPREP_NAME_PATTERN_REPORT

Figure 2-117 Create j05e_INVW_CUSTOMER_XXPREP 1/3


j05e_INVW_CUSTOMER_XXPREP..NAB_CUSTOMER_XXPREP_NAME_TOKEN_REPORT.NT ...

qsInvCount	qsInvWord	qsInvClassCode
20	MRS	P
17	MR	P
8	JACKSON	F
6	PETERSON	?
4	SUTHERLAN	?
3	STERN	?
3	ANDERSON	F
3	NEWSOM	?
3	PRESTON	F
2	A	I
2	D	I
2	DANA	F
2	B	I
2	HOPKINS	?
2	PEET	?
2	ALFRED	F
1	JERRY	F
1	KATHRYN	F
1	KERR	?
1	L	I
1	LESLIE	F
1	LORRAINE	F
1	MARTA	F
1	P	I
1	REFFELDT	?
1	RENEE	F
1	ROBERT	F
1	ROSEN	?
1	S	I
1	SKOV	?
1	DEBORAH	F
1	H	I
1	G	I
1	ERIK	F
1	DUSTIN	F
1	DEBRA	F
1	TORBEN	?
1	DAVID	F
1	DARYL	F
1	CHRISTINE	F
1	CHARLES	F
1	CATHY	F
1	CAL	?
1	WENDY	F
1	BRANDI	F
1	BEATA	F
1	BART	F
1	BARRY	F
1	ANTHONY	F
1	ANDREW	F
1	ALLISON	F
1	ALLAN	F
1	JACKIE	F
1	BURR	?

Figure 2-118 Create j05e_INVW_CUSTOMER_XXPREP 2/3



j05e_INVW_CUSTOMER_XXPREP..NAB_CUSTOMER_XXPREP_NAME_PATTERN_REPORT.NP - Data Browser

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
NameDomain_USPREP	PF?	MRS MARTA PEET	12	32.4324
NameDomain_USPREP	P.F?	MR . ALFRED PEET	5	13.5135
NameDomain_USPREP	PFF	MRS JACKIE JACKSON	5	13.5135
NameDomain_USPREP	P.FIF	MR . ALLAN H ANDERSON	3	8.10811
NameDomain_USPREP	P.FF	MR . DARYL ANDERSON	2	5.40541
NameDomain_USPREP	P.?F	MR . CAL JACKSON	2	5.40541
NameDomain_USPREP	PFI?	MRS DEBORAH D SUTHERLAND	1	2.7027
NameDomain_USPREP	P.??	MR . TORBEN SKOV	1	2.7027
NameDomain_USPREP	P.I?	MR . A HOPKINS	1	2.7027
NameDomain_USPREP	P.?I?	MR . KERR S NEWSOM	1	2.7027
NameDomain_USPREP	P.FI?	MR . ANDREW P STERN	1	2.7027
NameDomain_USPREP	PFIF	MRS LESLIE L JACKSON	1	2.7027
NameDomain_USPREP	P.IF	MR . B JACKSON	1	2.7027
NameDomain_USPREP	PII?	MRS G D STERN	1	2.7027

Figure 2-119 Create j05e_INVW_CUSTOMER_XXPREP 3/3

j06_XXPREP_CASS

For the records with U.S. home and work addresses, we use the CASS stage to validate, correct, and standardize the U.S. addresses in the Address domain. We include a Transformer stage to add a second address line column to customer file because CASS requires two address lines as input for its processing.

Figure 2-120 on page 606 shows the various stages that are used in this job, including the data sets that were created in "j03_STAN_XXPREP_NAB" on page 530, and a Transformer stage, a CASS stage, and a Data Set stage for each input data set. We modified the names of the stages as shown.

Because the configuration of this job is very similar to that described in "J05_CASS_USPREP" on page 219, we do not repeat it here. However, some of the configurations of interest are as follows:

- Figure 2-121 on page 607 shows the ADD_SPACE_FIELD - Transformer Stage window that describes the addition of the Space_Field of character length 1 to serve as the second address line required by the CASS stage.
- Figure 2-122 on page 607, Figure 2-123 on page 608, and Figure 2-124 on page 608 show the CASS stage configurations for each of the stages, with details such as the assignment of columns to Address Line 1 and Address Line 2 and the location of the output file.
- After saving, compiling, and running this job (Figure 2-125 on page 609), review the contents of the output of the CASS stage as shown in Figure 2-126 on page 610 through Figure 2-131 on page 615. It shows validation errors with some addresses as described in "J05_CASS_USPREP" on page 219.

Note: Generally, you need to present any data error for review by appropriate personnel as early as possible in the process. An `A1` should be investigated and appropriate action taken.

Proceed now to "j07_INVCC_CASS" on page 615.

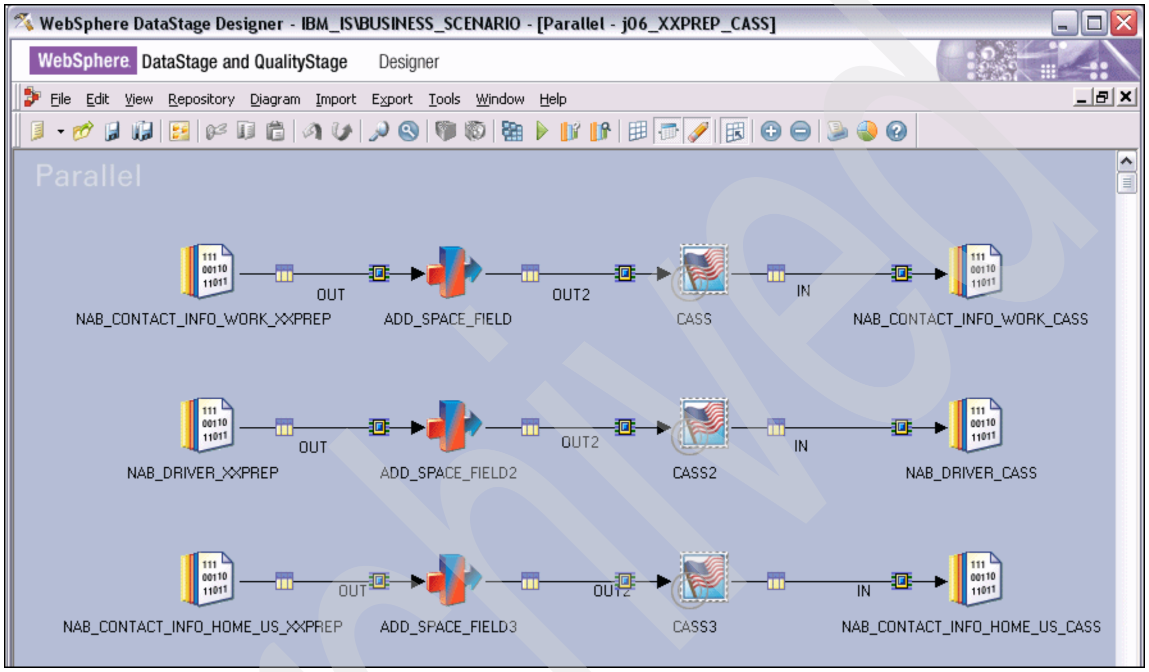
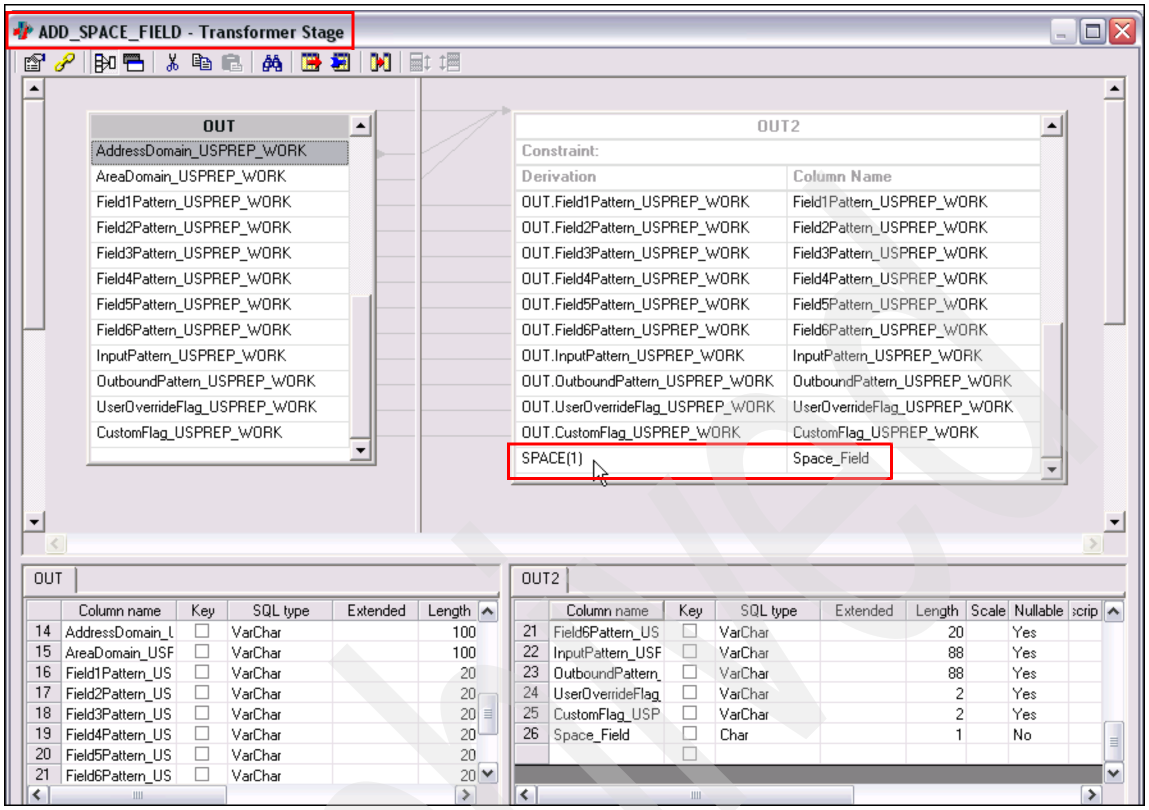

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j06_XXPREP_CASS]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NAB_CONTACT_INFO_WORK_XXPREP
OUT
ADD_SPACE_FIELD
OUT2
CASS
IN
NAB_CONTACT_INFO_WORK_CASS
NAB_DRIVER_XXPREP
OUT
ADD_SPACE_FIELD2
OUT2
CASS2
IN
NAB_DRIVER_CASS
NAB_CONTACT_INFO_HOME_US_XXPREP
OUT
ADD_SPACE_FIELD3
OUT2
CASS3
IN
NAB_CONTACT_INFO_HOME_US_CASS

Figure 2-120 Create j06_XXPREP_CASS 1/12



ADD_SPACE_FIELD - Transformer Stage

OUT
AddressDomain_USPREP_WORK
AreaDomain_USPREP_WORK
Field1Pattern_USPREP_WORK
Field2Pattern_USPREP_WORK
Field3Pattern_USPREP_WORK
Field4Pattern_USPREP_WORK
Field5Pattern_USPREP_WORK
Field6Pattern_USPREP_WORK
InputPattern_USPREP_WORK
OutboundPattern_USPREP_WORK
UserOverrideFlag_USPREP_WORK
CustomFlag_USPREP_WORK

OUT2

Constraint:

Derivation	Column Name
OUT.Field1Pattern_USPREP_WORK	Field1Pattern_USPREP_WORK
OUT.Field2Pattern_USPREP_WORK	Field2Pattern_USPREP_WORK
OUT.Field3Pattern_USPREP_WORK	Field3Pattern_USPREP_WORK
OUT.Field4Pattern_USPREP_WORK	Field4Pattern_USPREP_WORK
OUT.Field5Pattern_USPREP_WORK	Field5Pattern_USPREP_WORK
OUT.Field6Pattern_USPREP_WORK	Field6Pattern_USPREP_WORK
OUT.InputPattern_USPREP_WORK	InputPattern_USPREP_WORK
OUT.OutboundPattern_USPREP_WORK	OutboundPattern_USPREP_WORK
OUT.UserOverrideFlag_USPREP_WORK	UserOverrideFlag_USPREP_WORK
OUT.CustomFlag_USPREP_WORK	CustomFlag_USPREP_WORK
SPACE(1)	Space_Field

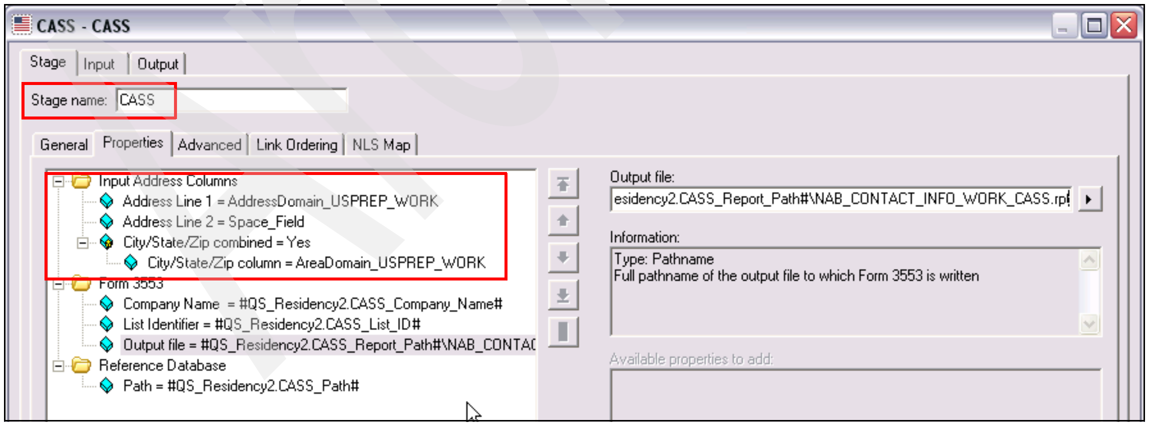
OUT

	Column name	Key	SQL type	Extended	Length
14	AddressDomain_L	☐	VarChar		100
15	AreaDomain_USF	☐	VarChar		100
16	Field1Pattern_US	☐	VarChar		20
17	Field2Pattern_US	☐	VarChar		20
18	Field3Pattern_US	☐	VarChar		20
19	Field4Pattern_US	☐	VarChar		20
20	Field5Pattern_US	☐	VarChar		20
21	Field6Pattern_US	☐	VarChar		20

OUT2

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	:crip
21	Field6Pattern_US	☐	VarChar		20		Yes	
22	InputPattern_USF	☐	VarChar		88		Yes	
23	OutboundPattern_	☐	VarChar		88		Yes	
24	UserOverrideFlag	☐	VarChar		2		Yes	
25	CustomFlag_USP	☐	VarChar		2		Yes	
26	Space_Field	☐	Char		1		No	

Figure 2-121   Create j06_XXPREP_CASS 2/12



CASS - CASS

Stage | Input | Output

Stage name: CASS

General | Properties | Advanced | Link Ordering | NLS Map

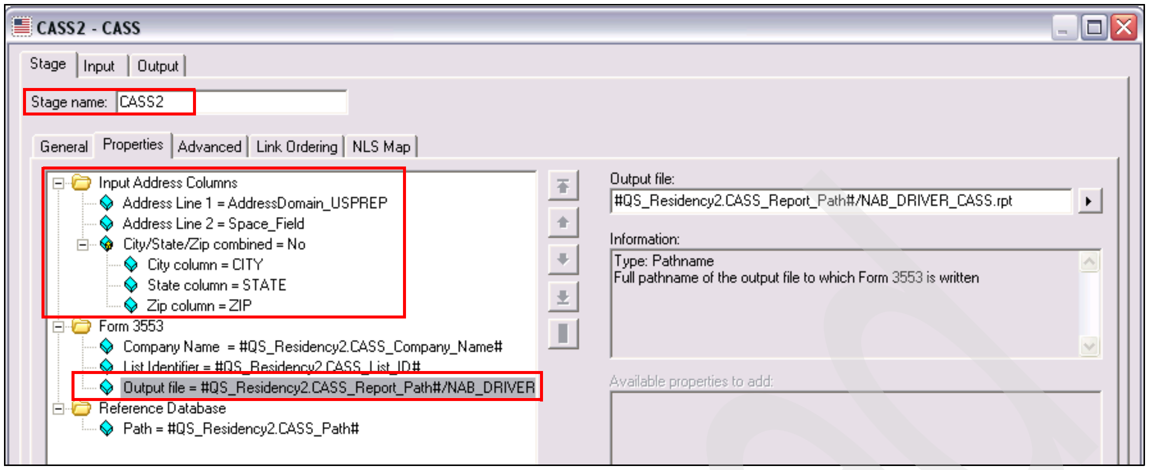
- Input Address Columns
  - Address Line 1 = AddressDomain_USPREP_WORK
  - Address Line 2 = Space_Field
  - City/State/Zip combined = Yes
    - City/State/Zip column = AreaDomain_USPREP_WORK
- Form 3553
  - Company Name = #QS_Residency2.CASS_Company_Name#
  - List Identifier = #QS_Residency2.CASS_List_ID#
  - Output file = #QS_Residency2.CASS_Report_Path#\NAB_CONTAC
- Reference Database
  - Path = #QS_Residency2.CASS_Path#

Output file:
esidency2.CASS_Report_Path#\NAB_CONTACT_INFO_WORK_CASS.rpt

Information:
Type: Pathname
Full pathname of the output file to which Form 3553 is written

Available properties to add:

Figure 2-122   Create j06_XXPREP_CASS 3/12



CASS2 - CASS

Stage | Input | Output

Stage name: CASS2

General | Properties | Advanced | Link Ordering | NLS Map

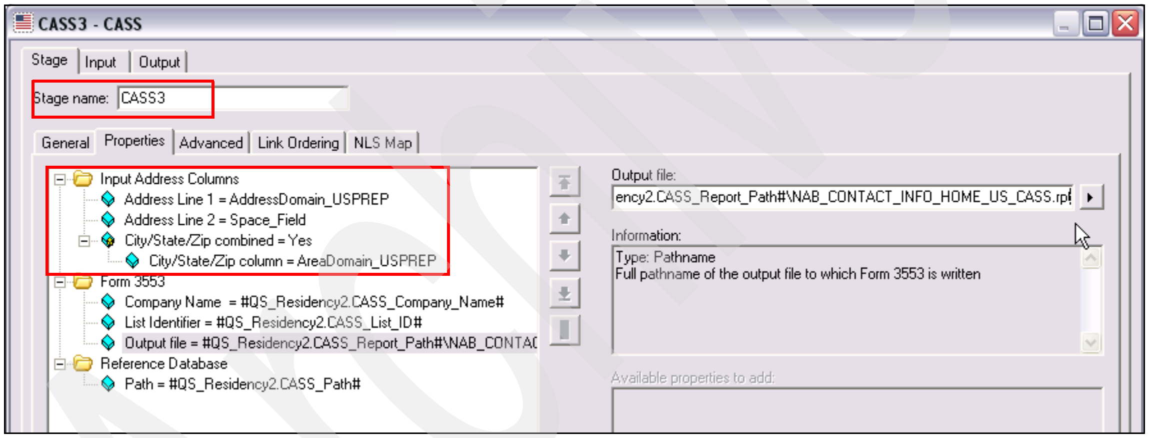
```
Input Address Columns
    Address Line 1 = AddressDomain_USPREP
    Address Line 2 = Space_Field
    City/State/Zip combined = No
        City column = CITY
        State column = STATE
        Zip column = ZIP
Form 3553
    Company Name  = #QS_Residency2.CASS_Company_Name#
    List Identifier = #QS_Residency2.CASS_List_ID#
    Output file = #QS_Residency2.CASS_Report_Path#/NAB_DRIVER
Reference Database
    Path = #QS_Residency2.CASS_Path#
```

Output file: #QS_Residency2.CASS_Report_Path#/NAB_DRIVER_CASS.rpt

Information:
Type: Pathname
Full pathname of the output file to which Form 3553 is written

Available properties to add:

Figure 2-123   Create j06_XXPREP_CASS 4/12



CASS3 - CASS

Stage | Input | Output

Stage name: CASS3

General | Properties | Advanced | Link Ordering | NLS Map

```
Input Address Columns
    Address Line 1 = AddressDomain_USPREP
    Address Line 2 = Space_Field
    City/State/Zip combined = Yes
        City/State/Zip column = AreaDomain_USPREP
Form 3553
    Company Name  = #QS_Residency2.CASS_Company_Name#
    List Identifier = #QS_Residency2.CASS_List_ID#
    Output file = #QS_Residency2.CASS_Report_Path#\NAB_CONTAC
Reference Database
    Path = #QS_Residency2.CASS_Path#
```

Output file: ency2.CASS_Report_Path#\NAB_CONTACT_INFO_HOME_US_CASS.rpt

Information:
Type: Pathname
Full pathname of the output file to which Form 3553 is written

Available properties to add:

Figure 2-124   Create j06_XXPREP_CASS 5/12

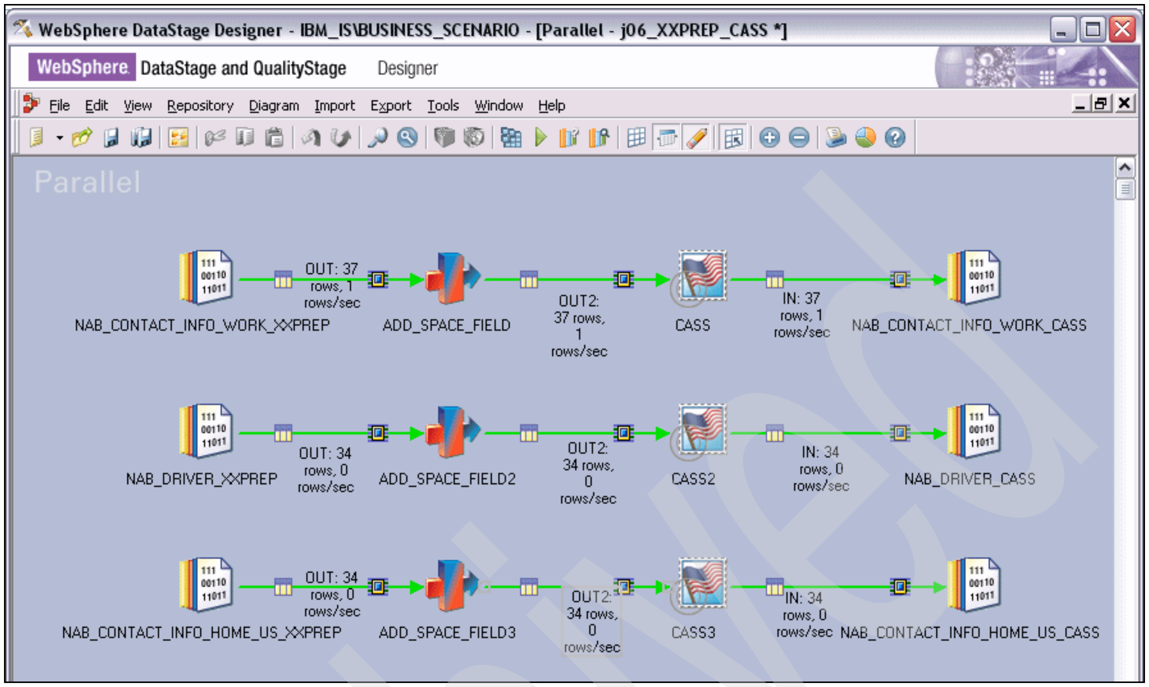
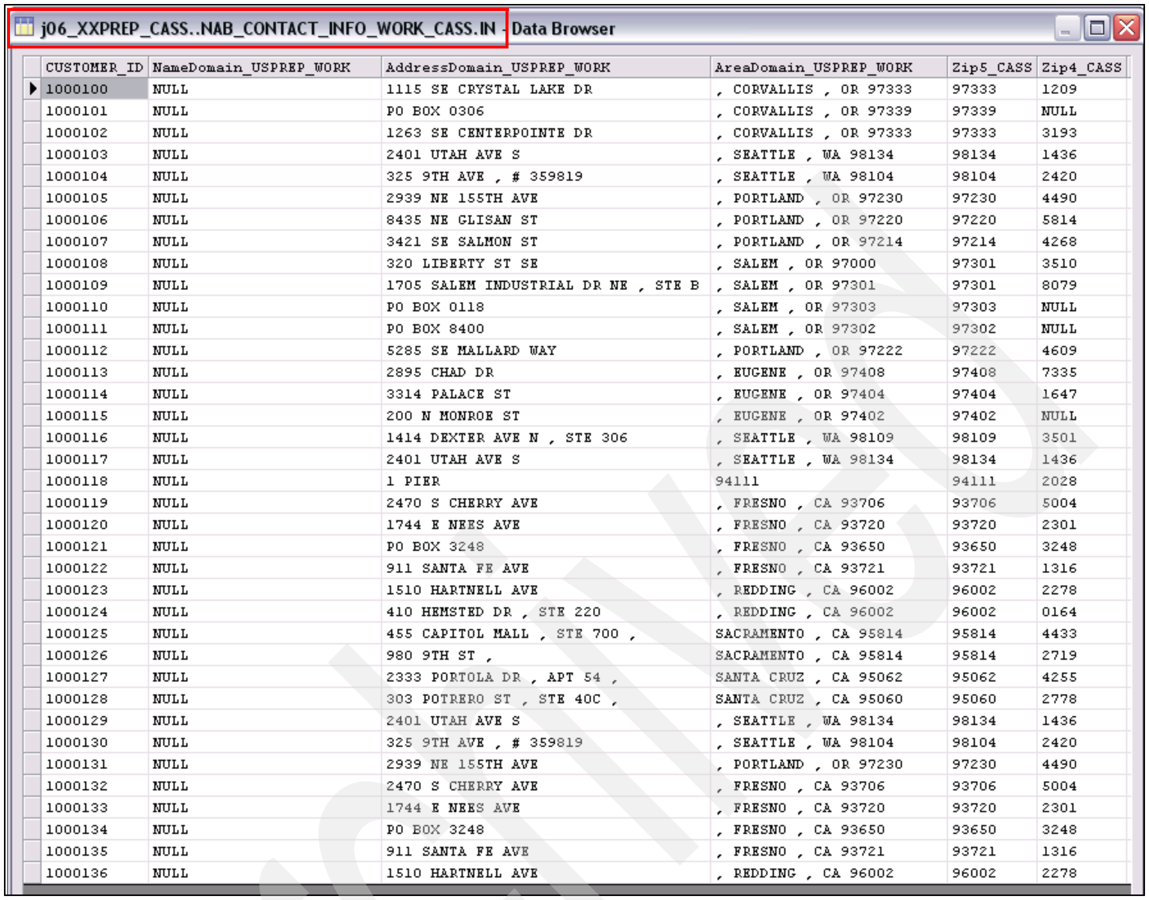
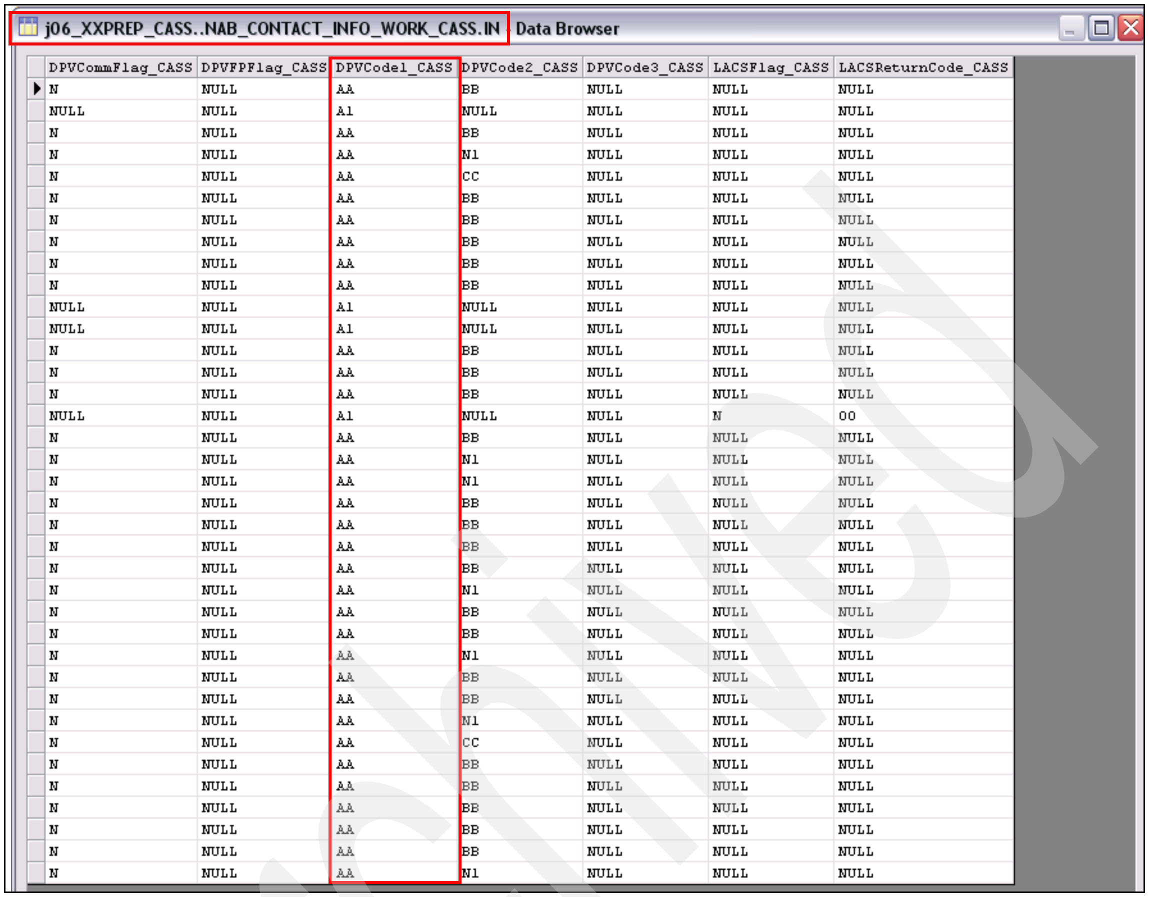

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j06_XXPREP_CASS *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NAB_CONTACT_INFO_WORK_XXPREP
OUT: 37 rows, 1 rows/sec
ADD_SPACE_FIELD
OUT2: 37 rows, 1 rows/sec
CASS
IN: 37 rows, 1 rows/sec
NAB_CONTACT_INFO_WORK_CASS
NAB_DRIVER_XXPREP
OUT: 34 rows, 0 rows/sec
ADD_SPACE_FIELD2
OUT2: 34 rows, 0 rows/sec
CASS2
IN: 34 rows, 0 rows/sec
NAB_DRIVER_CASS
NAB_CONTACT_INFO_HOME_US_XXPREP
OUT: 34 rows, 0 rows/sec
ADD_SPACE_FIELD3
OUT2: 34 rows, 0 rows/sec
CASS3
IN: 34 rows, 0 rows/sec
NAB_CONTACT_INFO_HOME_US_CASS

Figure 2-125 Create j06_XXPREP_CASS 6/12


j06_XXPREP_CASS..NAB_CONTACT_INFO_WORK_CASS.IN - Data Browser

CUSTOMER_ID	NameDomain_USPREP_WORK	AddressDomain_USPREP_WORK	AreaDomain_USPREP_WORK	Zip5_CASS	Zip4_CASS
1000100	NULL	1115 SE CRYSTAL LAKE DR	, CORVALLIS , OR 97333	97333	1209
1000101	NULL	PO BOX 0306	, CORVALLIS , OR 97339	97339	NULL
1000102	NULL	1263 SE CENTERPOINTE DR	, CORVALLIS , OR 97333	97333	3193
1000103	NULL	2401 UTAH AVE S	, SEATTLE , WA 98134	98134	1436
1000104	NULL	325 9TH AVE , # 359819	, SEATTLE , WA 98104	98104	2420
1000105	NULL	2939 NE 155TH AVE	, PORTLAND , OR 97230	97230	4490
1000106	NULL	8435 NE GLISAN ST	, PORTLAND , OR 97220	97220	5814
1000107	NULL	3421 SE SALMON ST	, PORTLAND , OR 97214	97214	4268
1000108	NULL	320 LIBERTY ST SE	, SALEM , OR 97000	97301	3510
1000109	NULL	1705 SALEM INDUSTRIAL DR NE , STE B	, SALEM , OR 97301	97301	8079
1000110	NULL	PO BOX 0118	, SALEM , OR 97303	97303	NULL
1000111	NULL	PO BOX 8400	, SALEM , OR 97302	97302	NULL
1000112	NULL	5285 SE MALLARD WAY	, PORTLAND , OR 97222	97222	4609
1000113	NULL	2895 CHAD DR	, EUGENE , OR 97408	97408	7335
1000114	NULL	3314 PALACE ST	, EUGENE , OR 97404	97404	1647
1000115	NULL	200 N MONROE ST	, EUGENE , OR 97402	97402	NULL
1000116	NULL	1414 DEXTER AVE N , STE 306	, SEATTLE , WA 98109	98109	3501
1000117	NULL	2401 UTAH AVE S	, SEATTLE , WA 98134	98134	1436
1000118	NULL	1 PIER	94111	94111	2028
1000119	NULL	2470 S CHERRY AVE	, FRESNO , CA 93706	93706	5004
1000120	NULL	1744 E NEES AVE	, FRESNO , CA 93720	93720	2301
1000121	NULL	PO BOX 3248	, FRESNO , CA 93650	93650	3248
1000122	NULL	911 SANTA FE AVE	, FRESNO , CA 93721	93721	1316
1000123	NULL	1510 HARTNELL AVE	, REDDING , CA 96002	96002	2278
1000124	NULL	410 HEMSTED DR , STE 220	, REDDING , CA 96002	96002	0164
1000125	NULL	455 CAPITOL MALL , STE 700 ,	SACRAMENTO , CA 95814	95814	4433
1000126	NULL	980 9TH ST ,	SACRAMENTO , CA 95814	95814	2719
1000127	NULL	2333 PORTOLA DR , APT 54 ,	SANTA CRUZ , CA 95062	95062	4255
1000128	NULL	303 POTRERO ST , STE 40C ,	SANTA CRUZ , CA 95060	95060	2778
1000129	NULL	2401 UTAH AVE S	, SEATTLE , WA 98134	98134	1436
1000130	NULL	325 9TH AVE , # 359819	, SEATTLE , WA 98104	98104	2420
1000131	NULL	2939 NE 155TH AVE	, PORTLAND , OR 97230	97230	4490
1000132	NULL	2470 S CHERRY AVE	, FRESNO , CA 93706	93706	5004
1000133	NULL	1744 E NEES AVE	, FRESNO , CA 93720	93720	2301
1000134	NULL	PO BOX 3248	, FRESNO , CA 93650	93650	3248
1000135	NULL	911 SANTA FE AVE	, FRESNO , CA 93721	93721	1316
1000136	NULL	1510 HARTNELL AVE	, REDDING , CA 96002	96002	2278

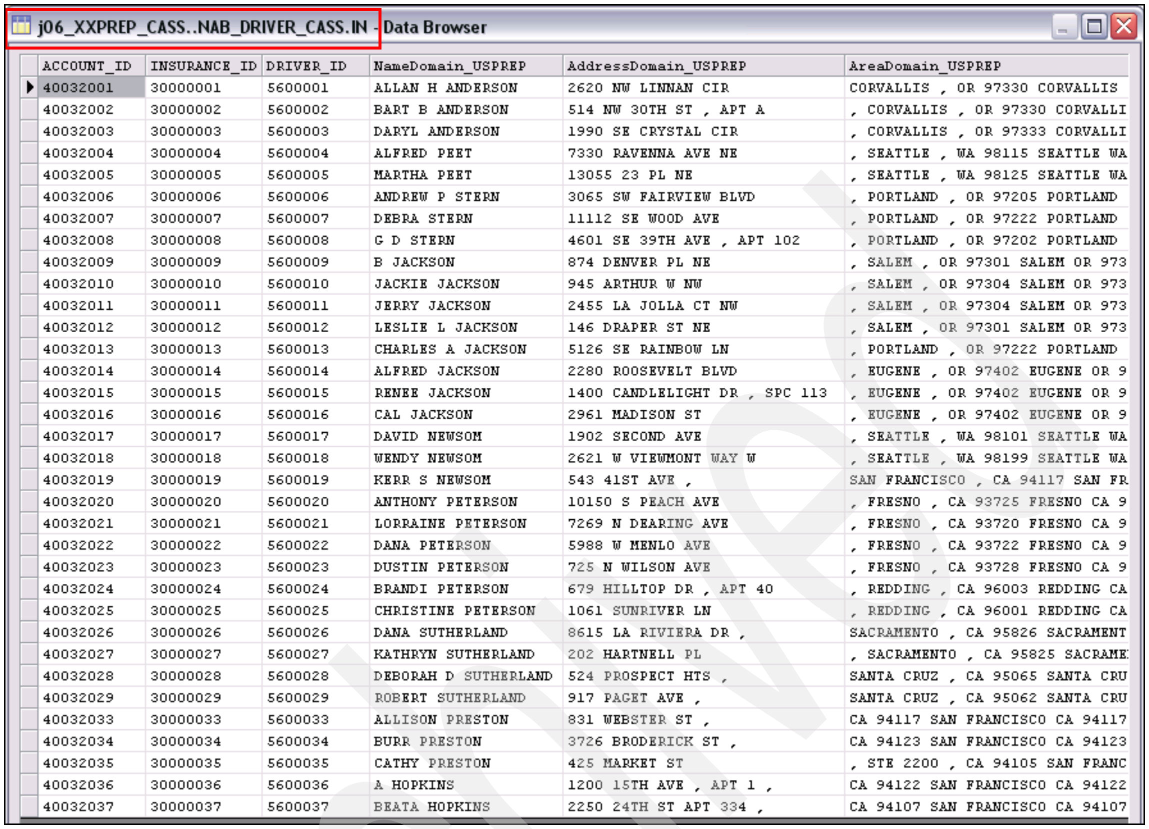
Figure 2-126 Create j06_XXPREP_CASS 7/12



j06_XXPREP_CASS..NAB_CONTACT_INFO_WORK_CASS.IN - Data Browser

DPVCommFlag_CASS	DPVFPFlag_CASS	DPVCode1_CASS	DPVCode2_CASS	DPVCode3_CASS	LACSFlag_CASS	LACSReturnCode_CASS
N	NULL	AA	BB	NULL	NULL	NULL
NULL	NULL	A1	NULL	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	N1	NULL	NULL	NULL
N	NULL	AA	CC	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
NULL	NULL	A1	NULL	NULL	NULL	NULL
NULL	NULL	A1	NULL	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
NULL	NULL	A1	NULL	NULL	N	00
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	N1	NULL	NULL	NULL
N	NULL	AA	N1	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	N1	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	N1	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	N1	NULL	NULL	NULL
N	NULL	AA	CC	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	N1	NULL	NULL	NULL

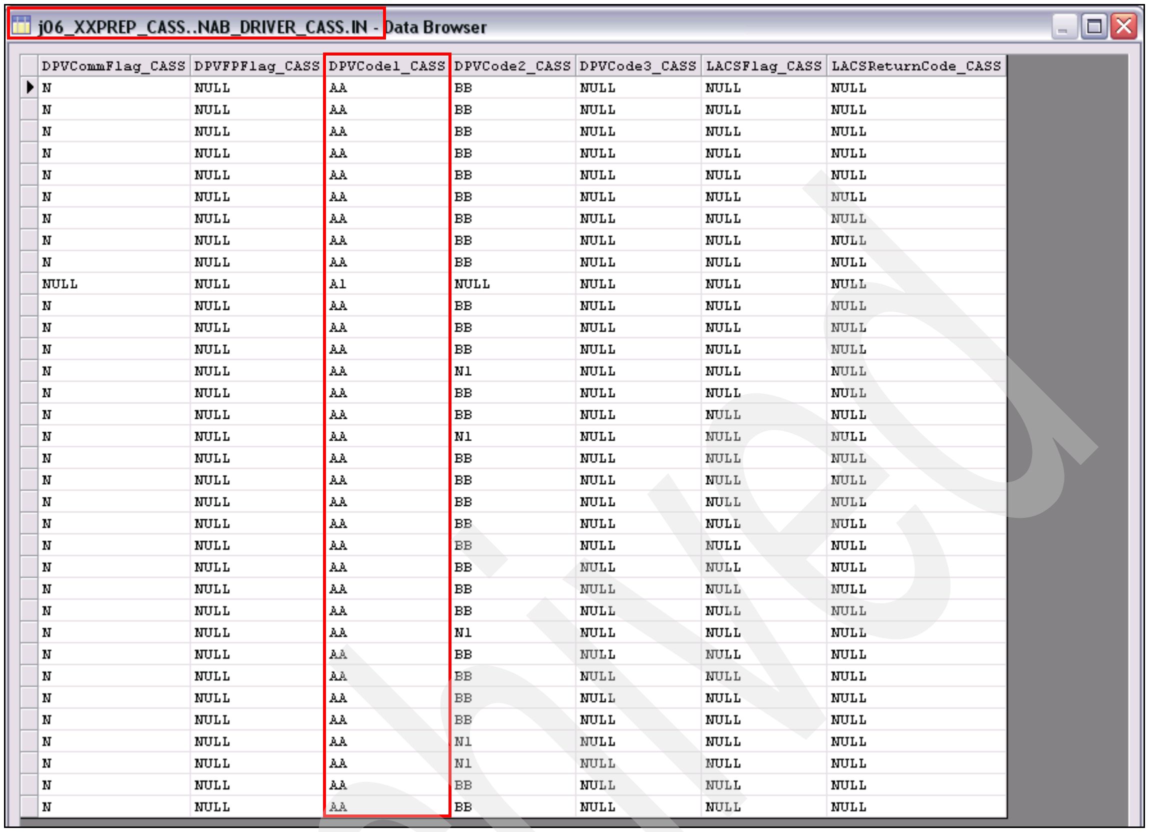
Figure 2-127 Create j06_XXPREP_CASS 8/12



j06_XXPREP_CASS..NAB_DRIVER_CASS.IN - Data Browser

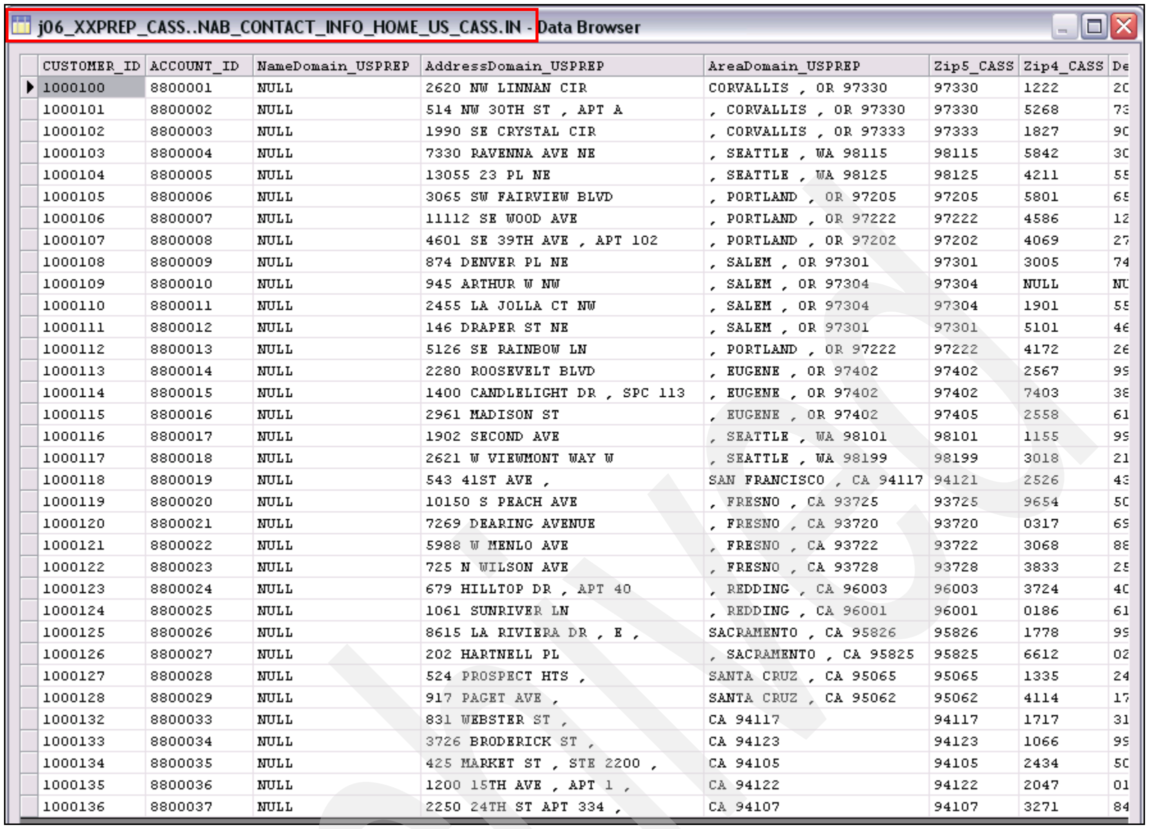
ACCOUNT_ID	INSURANCE_ID	DRIVER_ID	NameDomain_USPREP	AddressDomain_USPREP	AreaDomain_USPREP
40032001	30000001	5600001	ALLAN H ANDERSON	2620 NW LINNAN CIR	CORVALLIS , OR 97330 CORVALLIS
40032002	30000002	5600002	BART B ANDERSON	514 NW 30TH ST , APT A	, CORVALLIS , OR 97330 CORVALLI
40032003	30000003	5600003	DARYL ANDERSON	1990 SE CRYSTAL CIR	, CORVALLIS , OR 97333 CORVALLI
40032004	30000004	5600004	ALFRED PEET	7330 RAVENNA AVE NE	, SEATTLE , WA 98115 SEATTLE WA
40032005	30000005	5600005	MARTHA PEET	13055 23 PL NE	, SEATTLE , WA 98125 SEATTLE WA
40032006	30000006	5600006	ANDREW P STERN	3065 SW FAIRVIEW BLVD	, PORTLAND , OR 97205 PORTLAND
40032007	30000007	5600007	DEBRA STERN	11112 SE WOOD AVE	, PORTLAND , OR 97222 PORTLAND
40032008	30000008	5600008	G D STERN	4601 SE 39TH AVE , APT 102	, PORTLAND , OR 97202 PORTLAND
40032009	30000009	5600009	B JACKSON	874 DENVER PL NE	, SALEM , OR 97301 SALEM OR 973
40032010	30000010	5600010	JACKIE JACKSON	945 ARTHUR W NW	, SALEM , OR 97304 SALEM OR 973
40032011	30000011	5600011	JERRY JACKSON	2455 LA JOLLA CT NW	, SALEM , OR 97304 SALEM OR 973
40032012	30000012	5600012	LESLIE L JACKSON	146 DRAPER ST NE	, SALEM , OR 97301 SALEM OR 973
40032013	30000013	5600013	CHARLES A JACKSON	5126 SE RAINBOW LN	, PORTLAND , OR 97222 PORTLAND
40032014	30000014	5600014	ALFRED JACKSON	2280 ROOSEVELT BLVD	, EUGENE , OR 97402 EUGENE OR 9
40032015	30000015	5600015	RENEE JACKSON	1400 CANDLELIGHT DR , SPC 113	, EUGENE , OR 97402 EUGENE OR 9
40032016	30000016	5600016	CAL JACKSON	2961 MADISON ST	, EUGENE , OR 97402 EUGENE OR 9
40032017	30000017	5600017	DAVID NEWSOM	1902 SECOND AVE	, SEATTLE , WA 98101 SEATTLE WA
40032018	30000018	5600018	WENDY NEWSOM	2621 W VIEWMONT WAY W	, SEATTLE , WA 98199 SEATTLE WA
40032019	30000019	5600019	KERR S NEWSOM	543 41ST AVE ,	SAN FRANCISCO , CA 94117 SAN FR
40032020	30000020	5600020	ANTHONY PETERSON	10150 S PEACH AVE	, FRESNO , CA 93725 FRESNO CA 9
40032021	30000021	5600021	LORRAINE PETERSON	7269 N DEARING AVE	, FRESNO , CA 93720 FRESNO CA 9
40032022	30000022	5600022	DANA PETERSON	5988 W MENLO AVE	, FRESNO , CA 93722 FRESNO CA 9
40032023	30000023	5600023	DUSTIN PETERSON	725 N WILSON AVE	, FRESNO , CA 93728 FRESNO CA 9
40032024	30000024	5600024	BRANDI PETERSON	679 HILLTOP DR , APT 40	, REDDING , CA 96003 REDDING CA
40032025	30000025	5600025	CHRISTINE PETERSON	1061 SUNRIVER LN	, REDDING , CA 96001 REDDING CA
40032026	30000026	5600026	DANA SUTHERLAND	8615 LA RIVIERA DR ,	SACRAMENTO , CA 95826 SACRAMENT
40032027	30000027	5600027	KATHRYN SUTHERLAND	202 HARTNELL PL	, SACRAMENTO , CA 95825 SACRAME
40032028	30000028	5600028	DEBORAH D SUTHERLAND	524 PROSPECT HTS ,	SANTA CRUZ , CA 95065 SANTA CRU
40032029	30000029	5600029	ROBERT SUTHERLAND	917 PAGET AVE ,	SANTA CRUZ , CA 95062 SANTA CRU
40032033	30000033	5600033	ALLISON PRESTON	831 WEBSTER ST ,	CA 94117 SAN FRANCISCO CA 94117
40032034	30000034	5600034	BURR PRESTON	3726 BRODERICK ST ,	CA 94123 SAN FRANCISCO CA 94123
40032035	30000035	5600035	CATHY PRESTON	425 MARKET ST	, STE 2200 , CA 94105 SAN FRANC
40032036	30000036	5600036	A HOPKINS	1200 15TH AVE , APT 1 ,	CA 94122 SAN FRANCISCO CA 94122
40032037	30000037	5600037	BEATA HOPKINS	2250 24TH ST APT 334 ,	CA 94107 SAN FRANCISCO CA 94107

Figure 2-128   Create j06_XXPREP_CASS 9/12


j06_XXPREP_CASS..NAB_DRIVER_CASS.IN - Data Browser

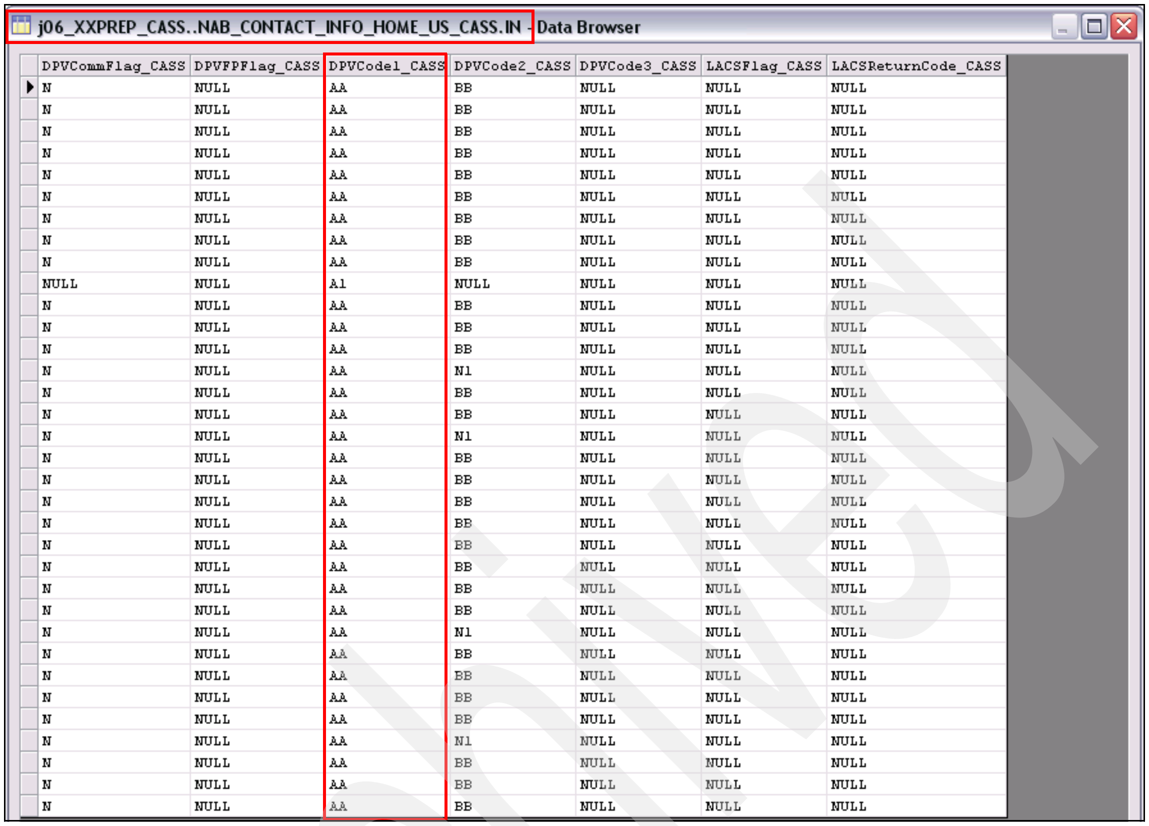
DPVCommFlag_CASS	DPVFPFlag_CASS	DPVCode1_CASS	DPVCode2_CASS	DPVCode3_CASS	LACSFlag_CASS	LACSReturnCode_CASS
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
NULL	NULL	A1	NULL	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	N1	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	N1	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	N1	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	N1	NULL	NULL	NULL
N	NULL	AA	N1	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL

Figure 2-129 Create j06_XXPREP_CASS 10/12


j06_XXPREP_CASS..NAB_CONTACT_INFO_HOME_US_CASS.IN - Data Browser

	CUSTOMER_ID	ACCOUNT_ID	NameDomain_USPREP	AddressDomain_USPREP	AreaDomain_USPREP	Zip5_CASS	Zip4_CASS	De
▶	1000100	8800001	NULL	2620 NW LINNAN CIR	CORVALLIS , OR 97330	97330	1222	20
	1000101	8800002	NULL	514 NW 30TH ST , APT A	, CORVALLIS , OR 97330	97330	5268	73
	1000102	8800003	NULL	1990 SE CRYSTAL CIR	, CORVALLIS , OR 97333	97333	1827	90
	1000103	8800004	NULL	7330 RAVENNA AVE NE	, SEATTLE , WA 98115	98115	5842	30
	1000104	8800005	NULL	13055 23 PL NE	, SEATTLE , WA 98125	98125	4211	55
	1000105	8800006	NULL	3065 SW FAIRVIEW BLVD	, PORTLAND , OR 97205	97205	5801	65
	1000106	8800007	NULL	11112 SE WOOD AVE	, PORTLAND , OR 97222	97222	4586	12
	1000107	8800008	NULL	4601 SE 39TH AVE , APT 102	, PORTLAND , OR 97202	97202	4069	27
	1000108	8800009	NULL	874 DENVER PL NE	, SALEM , OR 97301	97301	3005	74
	1000109	8800010	NULL	945 ARTHUR W NW	, SALEM , OR 97304	97304	NULL	NU
	1000110	8800011	NULL	2455 LA JOLLA CT NW	, SALEM , OR 97304	97304	1901	55
	1000111	8800012	NULL	146 DRAPER ST NE	, SALEM , OR 97301	97301	5101	46
	1000112	8800013	NULL	5126 SE RAINBOW LN	, PORTLAND , OR 97222	97222	4172	26
	1000113	8800014	NULL	2280 ROOSEVELT BLVD	, EUGENE , OR 97402	97402	2567	99
	1000114	8800015	NULL	1400 CANDLELIGHT DR , SPC 113	, EUGENE , OR 97402	97402	7403	38
	1000115	8800016	NULL	2961 MADISON ST	, EUGENE , OR 97402	97405	2558	61
	1000116	8800017	NULL	1902 SECOND AVE	, SEATTLE , WA 98101	98101	1155	99
	1000117	8800018	NULL	2621 W VIEWMONT WAY W	, SEATTLE , WA 98199	98199	3018	21
	1000118	8800019	NULL	543 41ST AVE ,	SAN FRANCISCO , CA 94117	94121	2526	43
	1000119	8800020	NULL	10150 S PEACH AVE	, FRESNO , CA 93725	93725	9654	50
	1000120	8800021	NULL	7269 DEARING AVENUE	, FRESNO , CA 93720	93720	0317	69
	1000121	8800022	NULL	5988 W MENLO AVE	, FRESNO , CA 93722	93722	3068	88
	1000122	8800023	NULL	725 N WILSON AVE	, FRESNO , CA 93728	93728	3833	25
	1000123	8800024	NULL	679 HILLTOP DR , APT 40	, REDDING , CA 96003	96003	3724	40
	1000124	8800025	NULL	1061 SUNRIVER LN	, REDDING , CA 96001	96001	0186	61
	1000125	8800026	NULL	8615 LA RIVIERA DR , E ,	SACRAMENTO , CA 95826	95826	1778	99
	1000126	8800027	NULL	202 HARTNELL PL	, SACRAMENTO , CA 95825	95825	6612	02
	1000127	8800028	NULL	524 PROSPECT HTS ,	SANTA CRUZ , CA 95065	95065	1335	24
	1000128	8800029	NULL	917 PAGET AVE ,	SANTA CRUZ , CA 95062	95062	4114	17
	1000132	8800033	NULL	831 WEBSTER ST ,	CA 94117	94117	1717	31
	1000133	8800034	NULL	3726 BRODERICK ST ,	CA 94123	94123	1066	99
	1000134	8800035	NULL	425 MARKET ST , STE 2200 ,	CA 94105	94105	2434	50
	1000135	8800036	NULL	1200 15TH AVE , APT 1 ,	CA 94122	94122	2047	01
	1000136	8800037	NULL	2250 24TH ST APT 334 ,	CA 94107	94107	3271	84

Figure 2-130 Create j06_XXPREP_CASS 11/12



j06_XXPREP_CASS..NAB_CONTACT_INFO_HOME_US_CASS.IN - Data Browser

DPVCommFlag_CASS	DPVFPFlag_CASS	DPVCode1_CASS	DPVCode2_CASS	DPVCode3_CASS	LACSFlag_CASS	LACSReturnCode_CASS
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
NULL	NULL	A1	NULL	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	N1	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	N1	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	N1	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	N1	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL

Figure 2-131 Create j06_XXPREP_CASS 12/12

j07_INVCC_CASS

We then run the Investigate stage with character concatenate option on the (home address related columns in CONTACT_INFO and DRIVER) results of job "j06_XXPREP_CASS" on page 605 step to determine addresses that are not recognized by CASS (delivery point verification or DPV).

> **Note:** Due to a bug with handling nulls, we introduced a Transformer stage to convert nulls to a space using column derivation.

We investigate using character concatenate (on CASS generated columns DPVMATCHFLAG_CASS and DPVCODE1_CASS) using a C mask, and X masks on the DeliveryAddressLine1_CASS, City_CASS, State_CASS, and Zip5_CASS columns. A value of `A1` in the DPVCODE1_CASS field indicates a potential problem.

Figure 2-132 on page 617 shows the various stages that are used in this job, including the data sets that were created in “j06_XXPREP_CASS” on page 605, a Transformer stage for handling nulls, an Investigate stage, and an output Sequential File stage for each input data set. We modified the names of the stages as shown.

Because the configuration of this job is very similar to that described in “J06_INVCC_CASS” on page 228, we do not repeat it here. However, some of the configurations of interest are as follows:

- Figure 2-133 on page 617 shows the configuration of the INV_DPV_INFO Investigate stage window with columns DPVMatchFlag_CASS and DPVCode1_CASS selected with the *C* masks for each character. The *X* masks are used for columns Delivery_AddressLine1_CASS,City_CASS, and Zip5_CASS.
- After saving, compiling, and running this job (Figure 2-134 on page 618), the output of the Investigate stages are shown in Figure 2-135 on page 618 through Figure 2-139 on page 621. It shows a a few records with values `A1` in the DPVMatchFlag_CASS and DPVCode1_CASS character concatenated columns, indicating a potential problem that requires further investigation.

Proceed now to “j08_USPREP_CASS” on page 619.

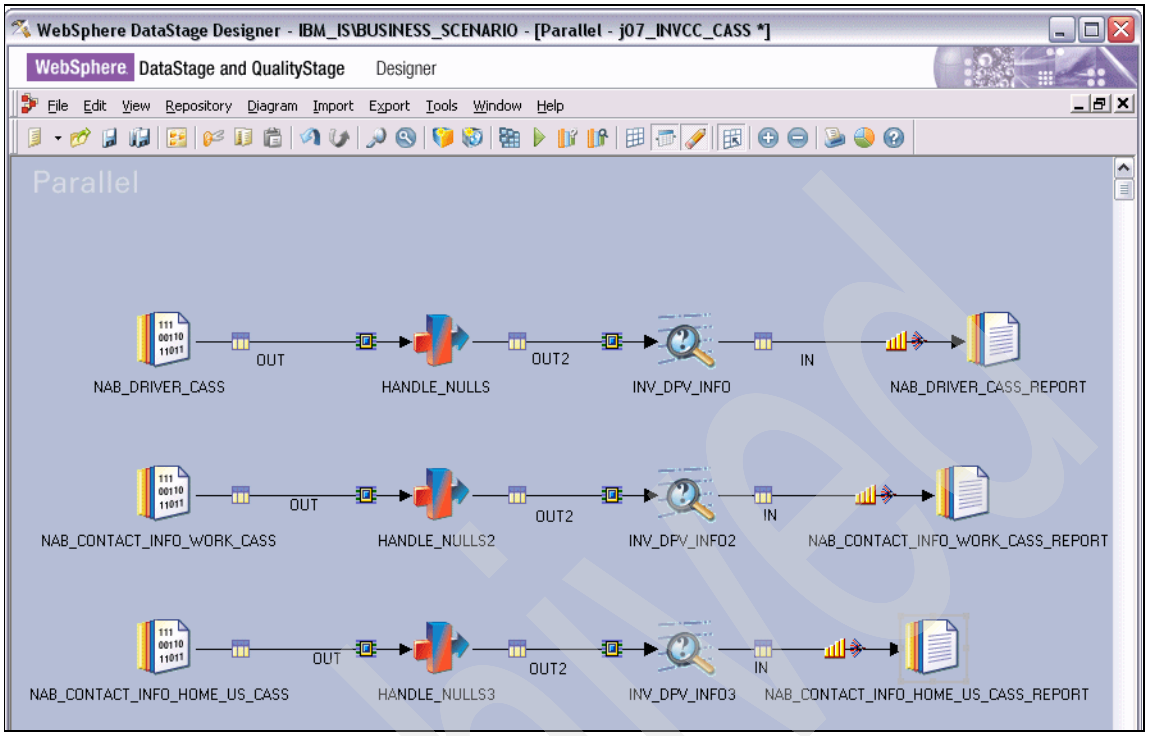
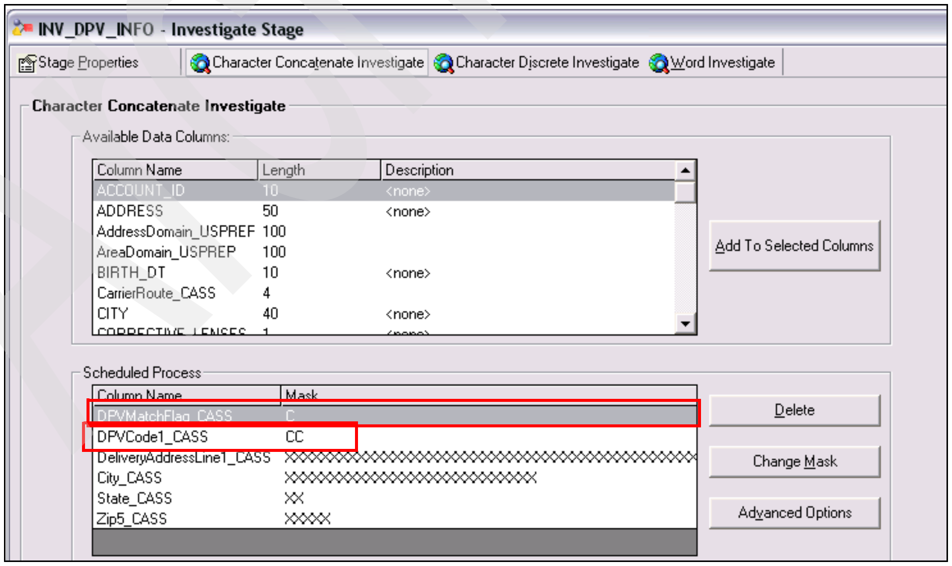

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j07_INVCC_CASS *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NAB_DRIVER_CASS
OUT
HANDLE_NULLS
OUT2
INV_DPV_INFO
IN
NAB_DRIVER_CASS_REPORT
NAB_CONTACT_INFO_WORK_CASS
OUT
HANDLE_NULLS2
OUT2
INV_DPV_INFO2
IN
NAB_CONTACT_INFO_WORK_CASS_REPORT
NAB_CONTACT_INFO_HOME_US_CASS
OUT
HANDLE_NULLS3
OUT2
INV_DPV_INFO3
IN
NAB_CONTACT_INFO_HOME_US_CASS_REPORT

Figure 2-132 Create j07_INVCC_CASS 1/7


INV_DPV_INFO - Investigate Stage
Stage Properties
Character Concatenate Investigate
Character Discrete Investigate
Word Investigate
Add To Selected Columns
Delete
Change Mask
Advanced Options

Character Concatenate Investigate

Available Data Columns:

Column Name	Length	Description
ACCOUNT_ID	10	<none>
ADDRESS	50	<none>
AddressDomain_USPREF	100	
AreaDomain_USPREP	100	
BIRTH_DT	10	<none>
CarrierRoute_CASS	4	
CITY	40	<none>
CORRECTIVE_LENSES	1	<none>

Scheduled Process

Column Name	Mask
DPVMatchFlag_CASS	C
DPVCode1_CASS	CC
DeliveryAddressLine1_CASS	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
City_CASS	XXXXXXXXXXXXXXXXXXXXXXXXXXXX
State_CASS	XX
Zip5_CASS	XXXXX

Figure 2-133 Create j07_INVCC_CASS 2/7

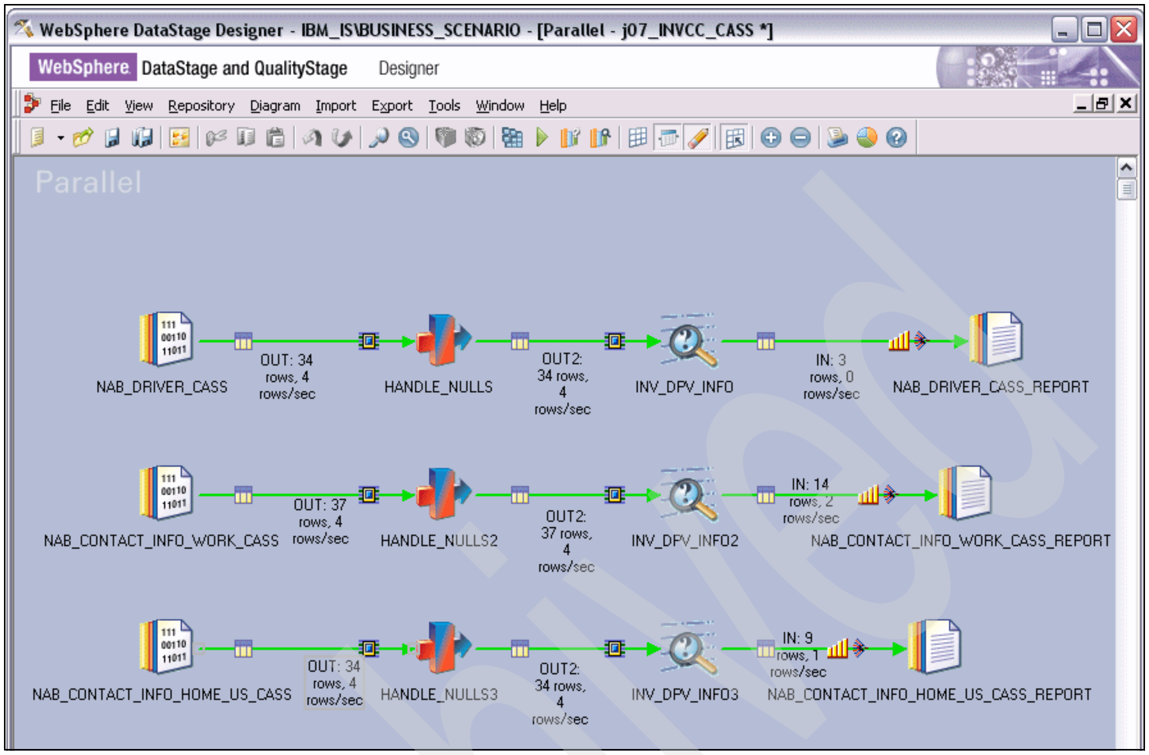
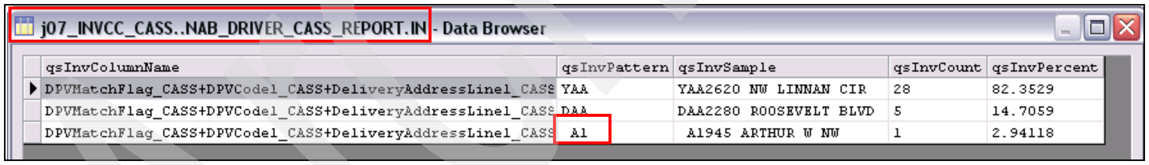
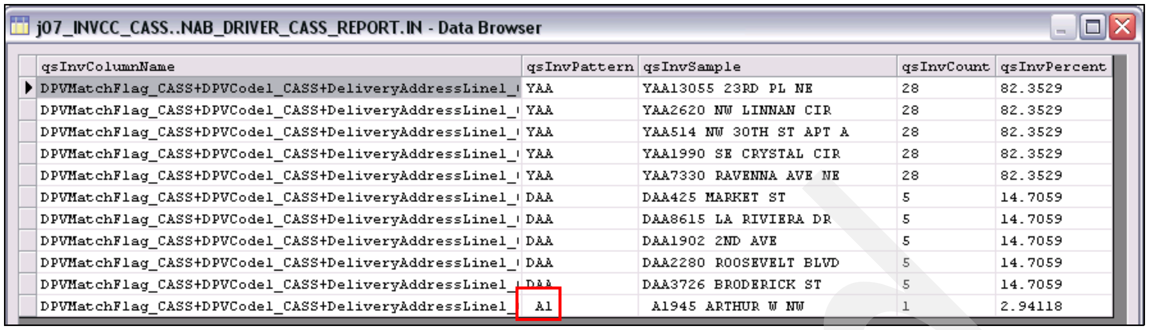

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j07_INVCC_CASS *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NAB_DRIVER_CASS
OUT: 34 rows, 4 rows/sec
HANDLE_NULLS
OUT2: 34 rows, 4 rows/sec
INV_DPV_INFO
IN: 3 rows, 0 rows/sec
NAB_DRIVER_CASS_REPORT
NAB_CONTACT_INFO_WORK_CASS
OUT: 37 rows, 4 rows/sec
HANDLE_NULLS2
OUT2: 37 rows, 4 rows/sec
INV_DPV_INFO2
IN: 14 rows, 2 rows/sec
NAB_CONTACT_INFO_WORK_CASS_REPORT
NAB_CONTACT_INFO_HOME_US_CASS
OUT: 34 rows, 4 rows/sec
HANDLE_NULLS3
OUT2: 34 rows, 4 rows/sec
INV_DPV_INFO3
IN: 9 rows, 1 rows/sec
NAB_CONTACT_INFO_HOME_US_CASS_REPORT

Figure 2-134 Create j07_INVCC_CASS 3/7


j07_INVCC_CASS..NAB_DRIVER_CASS_REPORT.IN - Data Browser

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_CASS	YAA	YAA2620 NW LINNAN CIR	28	82.3529
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_CASS	DAA	DAA2280 ROOSEVELT BLVD	5	14.7059
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_CASS	A1	A1945 ARTHUR W NW	1	2.94118

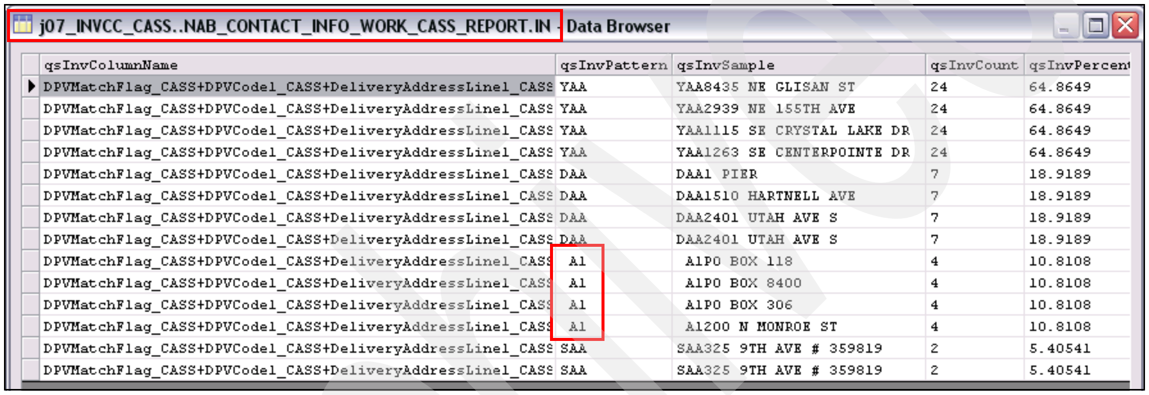
Figure 2-135 Create j07_INVCC_CASS 4/7



j07_INVCC_CASS..NAB_DRIVER_CASS_REPORT.IN - Data Browser

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_	YAA	YAA13055 23RD PL NE	28	82.3529
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_	YAA	YAA2620 NW LINNAN CIR	28	82.3529
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_	YAA	YAA514 NW 30TH ST APT A	28	82.3529
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_	YAA	YAA1990 SE CRYSTAL CIR	28	82.3529
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_	YAA	YAA7330 RAVENNA AVE NE	28	82.3529
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_	DAA	DAA425 MARKET ST	5	14.7059
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_	DAA	DAA8615 LA RIVIERA DR	5	14.7059
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_	DAA	DAA1902 2ND AVE	5	14.7059
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_	DAA	DAA2280 ROOSEVELT BLVD	5	14.7059
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_	DAA	DAA3726 BRODERICK ST	5	14.7059
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_	A1	A1945 ARTHUR W NW	1	2.94118

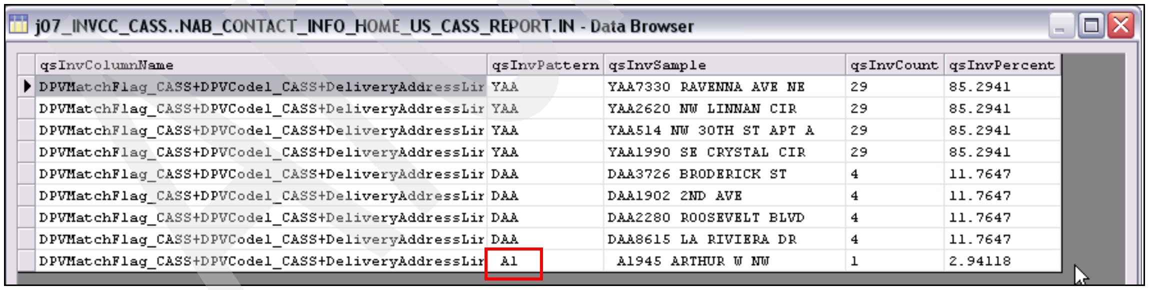
Figure 2-136 Create j07_INVCC_CASS 5/7



j07_INVCC_CASS..NAB_CONTACT_INFO_WORK_CASS_REPORT.IN - Data Browser

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercen
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_CASS	YAA	YAA8435 NE GLISAN ST	24	64.8649
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_CASS	YAA	YAA2939 NE 155TH AVE	24	64.8649
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_CASS	YAA	YAA1115 SE CRYSTAL LAKE DR	24	64.8649
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_CASS	YAA	YAA1263 SE CENTERPOINTE DR	24	64.8649
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_CASS	DAA	DAA1 PIER	7	18.9189
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_CASS	DAA	DAA1510 HARTNELL AVE	7	18.9189
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_CASS	DAA	DAA2401 UTAH AVE S	7	18.9189
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_CASS	DAA	DAA2401 UTAH AVE S	7	18.9189
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_CASS	A1	A1PO BOX 118	4	10.8108
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_CASS	A1	A1PO BOX 8400	4	10.8108
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_CASS	A1	A1PO BOX 306	4	10.8108
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_CASS	A1	A1200 N MONROE ST	4	10.8108
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_CASS	SAA	SAA325 9TH AVE # 359819	2	5.40541
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLine1_CASS	SAA	SAA325 9TH AVE # 359819	2	5.40541

Figure 2-137 Create j07_INVCC_CASS 6/7



j07_INVCC_CASS..NAB_CONTACT_INFO_HOME_US_CASS_REPORT.IN - Data Browser

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLir	YAA	YAA7330 RAVENNA AVE NE	29	85.2941
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLir	YAA	YAA2620 NW LINNAN CIR	29	85.2941
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLir	YAA	YAA514 NW 30TH ST APT A	29	85.2941
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLir	YAA	YAA1990 SE CRYSTAL CIR	29	85.2941
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLir	DAA	DAA3726 BRODERICK ST	4	11.7647
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLir	DAA	DAA1902 2ND AVE	4	11.7647
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLir	DAA	DAA2280 ROOSEVELT BLVD	4	11.7647
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLir	DAA	DAA8615 LA RIVIERA DR	4	11.7647
DPVMatchFlag_CASS+DPVCode1_CASS+DeliveryAddressLir	A1	A1945 ARTHUR W NW	1	2.94118

Figure 2-138 Create j07_INVCC_CASS 7/7

j08_USPREP_CASS

In this step, we standardize the name and address contents of the output of job "j06_XXPREP_CASS" on page 605 using the domain-preprocessor USPREP rule set. This parses and moves name, address, and area content from selected input columns to the NameDomain_USPREP, AddressDomain_USPREP, and AreaDomain_USPREP columns. We also add a column to the output that only had the first three characters of the ZIP code using a Transformer stage. This new column (ZIP3) is used as a blocking variable in an upcoming matching stage.

After CASS validates and corrects the address fields (but not the name fields), the Standardize stage is run again with the domain-preprocessor rule set USPREP on the name fields and CASS corrects address fields and the literals ZQPUTRZQ and ZQPUTAZQ.[6]

Figure 2-139 on page 621 shows the various stages that are used in this job, including the data sets that were created in "j06_XXPREP_CASS" on page 605, a Standardize stage, a Transformer stage to add a column, and an output Data Set stage for each of the input data sets. We modified the names of the stages as shown.

Because the configuration of this job is very similar to that described in "J07_STAN_CUSTOMER_Domain_Preprocessor" on page 234, we do not repeat it here. However, some of the configurations of interest are as follows:

- The Standardize Rule Process windows in Figure 2-140 on page 622 and Figure 2-141 on page 622 show the domain preprocessor USPREP rule set with literals and selected columns as follows for the U.S. home addresses of the CONTACT_INFO file:

  ```
  ZQPUTAZQ DeliveryAddressLine1_CASS DeliveryAddressLine2_CASS
  ZQPUTRZQ City_CASS State_CASS Zip5_CASS
  ```

- The ADD_ZIP3a - Transformer Stage windows in Figure 2-142 on page 623 and Figure 2-143 on page 624 show the addition of a ZIP3 column to the output that only contains the first three digits of the 5-digit ZIP5_CASS field for the U.S. home addresses of the CONTACT_INFO file.
- Because the configuration is somewhat similar for the U.S. work addresses of the CONTACT_INFO file and the U.S. addresses of the DRIVER file, we do not repeat it here.

  Note: The DRIVER file has the additional NAME field and, therefore, the Standardize Rule Process window has the following configuration:

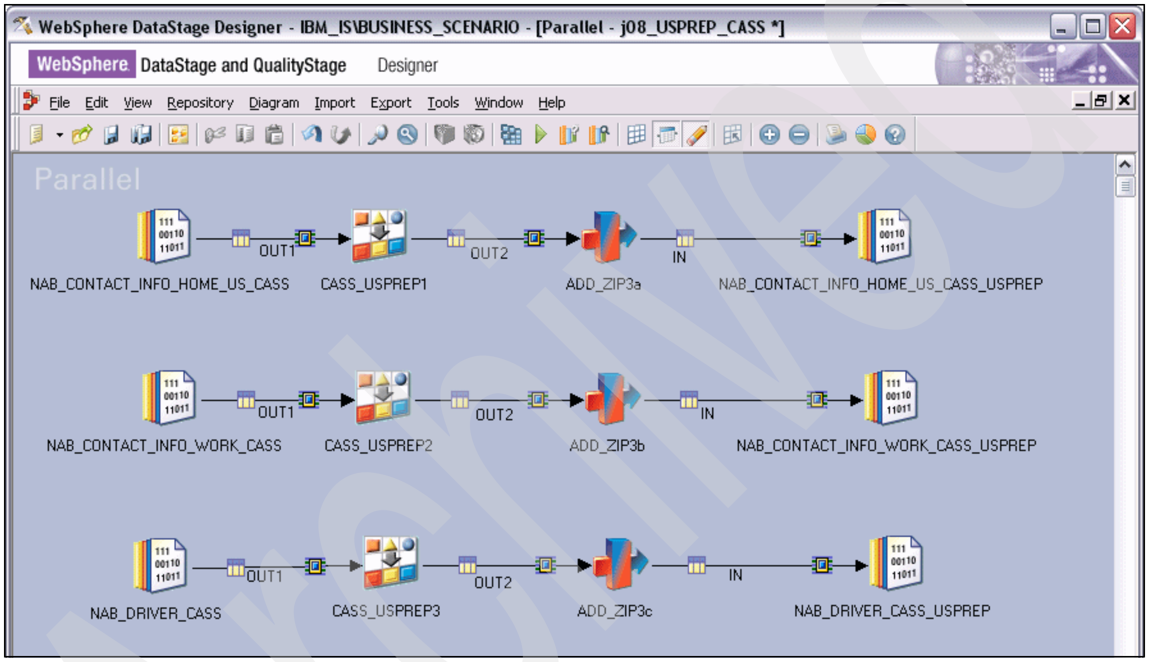
  ```
  ZQNAMEZQ NAME ZQPUTAZQ DeliveryAddressLine1_CASS
  DeliveryAddressLine2_CASS ZQPUTRZQ City_CASS State_CASS Zip5_CASS
  ```

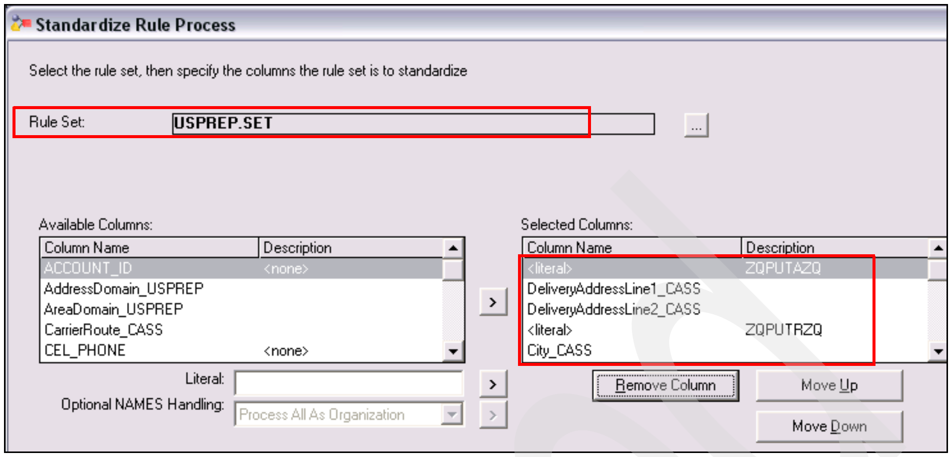
[6] The literal ZQPUTAZQ defaults the entire field to the Address Domain automatically, while ZQPUTRZQ defaults the entire field to the Area Domain automatically. Example 1-1 on page 33 has a brief overview of the available literals.

- After saving, compiling, and running this job (Figure 2-144 on page 625), you can view the content of output data set as shown in Figure 2-145 on page 626 through Figure 2-150 on page 631.

  This report shows the columns NameDomain_USPREP (contains prefix, first name, last name, suffix tokens), AddressDomain_USPREP (contains apartment, street name and street type tokens), and AreaDomain_USPREP (contains state, ZIP code tokens) that were parsed from the input columns.

Proceed now to “j09_STAN_CASS” on page 631.


WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j08_USPREP_CASS *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NAB_CONTACT_INFO_HOME_US_CASS
OUT1
CASS_USPREP1
OUT2
ADD_ZIP3a
IN
NAB_CONTACT_INFO_HOME_US_CASS_USPREP
NAB_CONTACT_INFO_WORK_CASS
OUT1
CASS_USPREP2
OUT2
ADD_ZIP3b
IN
NAB_CONTACT_INFO_WORK_CASS_USPREP
NAB_DRIVER_CASS
OUT1
CASS_USPREP3
OUT2
ADD_ZIP3c
IN
NAB_DRIVER_CASS_USPREP
Figure 2-139 Create j08_USPREP_CASS 1/12


Standardize Rule Process

Select the rule set, then specify the columns the rule set is to standardize

Rule Set: USPREP.SET

Available Columns:

Column Name	Description
ACCOUNT_ID	<none>
AddressDomain_USPREP	
AreaDomain_USPREP	
CarrierRoute_CASS	
CEL_PHONE	<none>

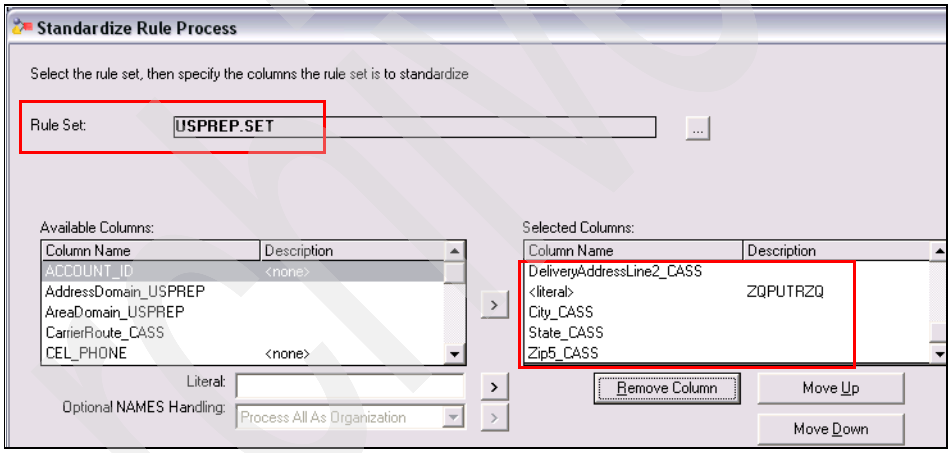
Selected Columns:

Column Name	Description
<literal>	ZQPUTAZQ
DeliveryAddressLine1_CASS	
DeliveryAddressLine2_CASS	
<literal>	ZQPUTRZQ
City_CASS	

Literal:

Optional NAMES Handling: Process All As Organization

Figure 2-140 Create j08_USPREP_CASS 2/12


Standardize Rule Process

Select the rule set, then specify the columns the rule set is to standardize

Rule Set: USPREP.SET

Available Columns:

Column Name	Description
ACCOUNT_ID	<none>
AddressDomain_USPREP	
AreaDomain_USPREP	
CarrierRoute_CASS	
CEL_PHONE	<none>

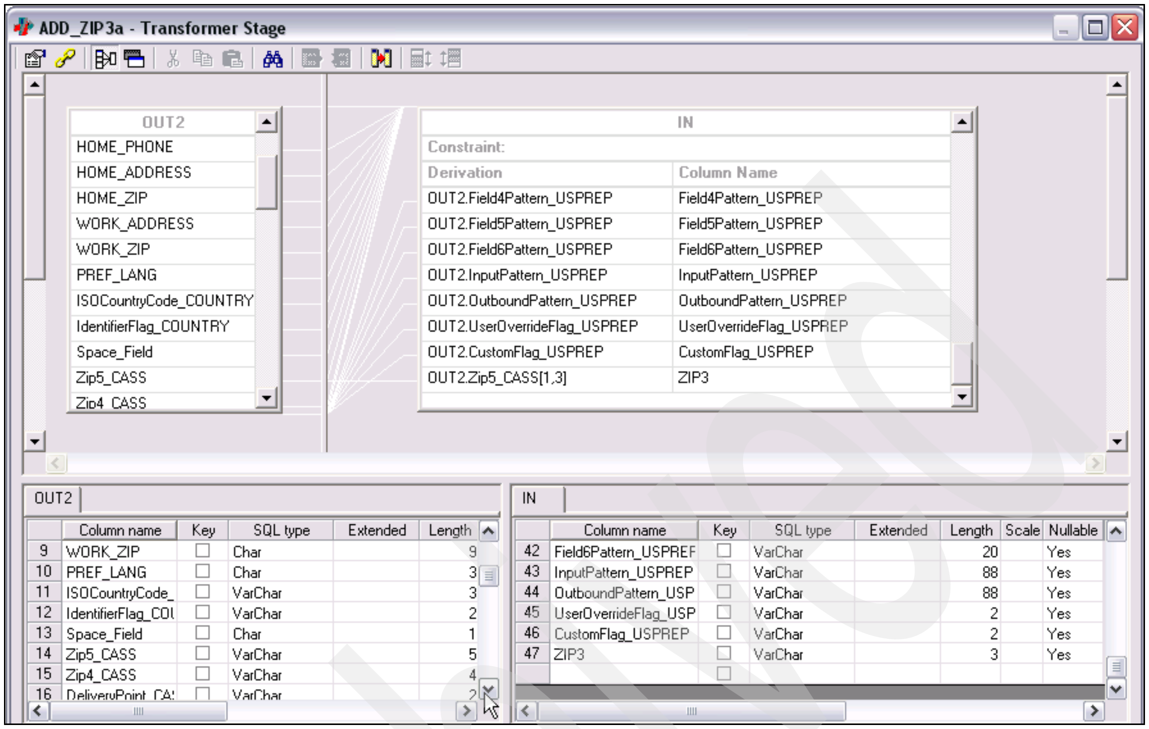
Selected Columns:

Column Name	Description
DeliveryAddressLine2_CASS	
<literal>	ZQPUTRZQ
City_CASS	
State_CASS	
Zip5_CASS	

Literal:

Optional NAMES Handling: Process All As Organization

Figure 2-141 Create j08_USPREP_CASS 3/12



OUT2
HOME_PHONE
HOME_ADDRESS
HOME_ZIP
WORK_ADDRESS
WORK_ZIP
PREF_LANG
ISOCountryCode_COUNTRY
IdentifierFlag_COUNTRY
Space_Field
Zip5_CASS
Zip4_CASS

IN

Constraint:

Derivation	Column Name
OUT2.Field4Pattern_USPREP	Field4Pattern_USPREP
OUT2.Field5Pattern_USPREP	Field5Pattern_USPREP
OUT2.Field6Pattern_USPREP	Field6Pattern_USPREP
OUT2.InputPattern_USPREP	InputPattern_USPREP
OUT2.OutboundPattern_USPREP	OutboundPattern_USPREP
OUT2.UserOverrideFlag_USPREP	UserOverrideFlag_USPREP
OUT2.CustomFlag_USPREP	CustomFlag_USPREP
OUT2.Zip5_CASS[1,3]	ZIP3

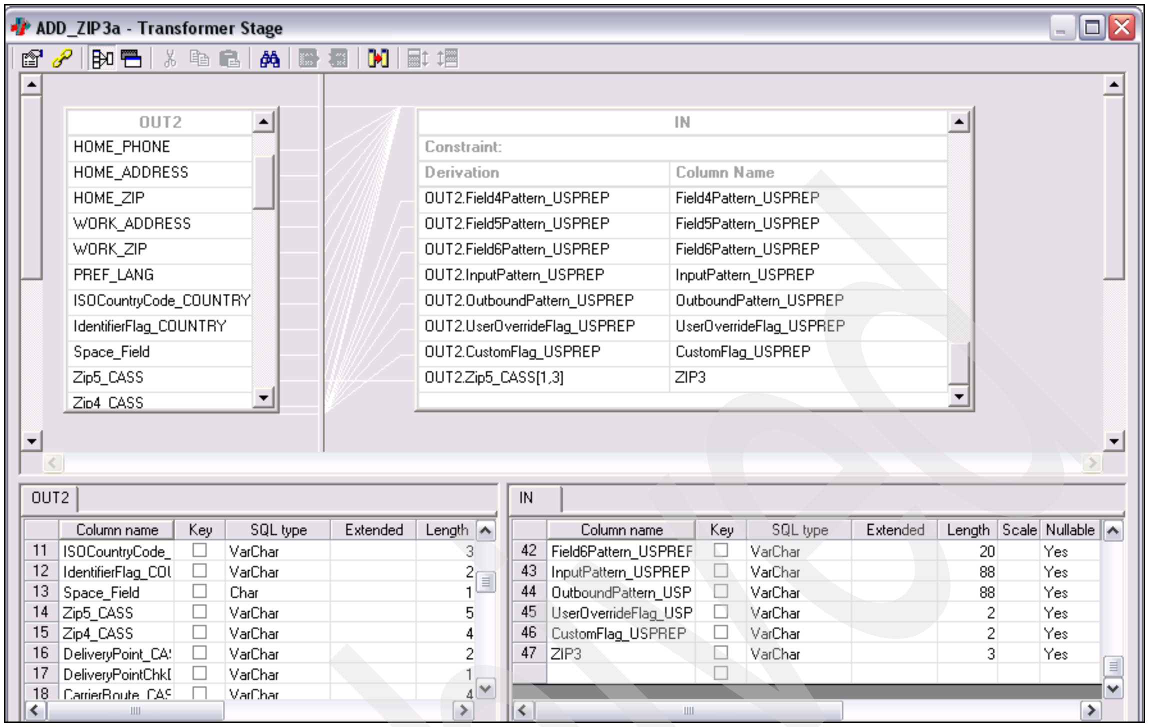
OUT2

	Column name	Key	SQL type	Extended	Length
9	WORK_ZIP	☐	Char		9
10	PREF_LANG	☐	Char		3
11	ISOCountryCode_	☐	VarChar		3
12	IdentifierFlag_COU	☐	VarChar		2
13	Space_Field	☐	Char		1
14	Zip5_CASS	☐	VarChar		5
15	Zip4_CASS	☐	VarChar		4
16	DeliveryPoint_CA:	☐	VarChar		2

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable
42	Field6Pattern_USPREF	☐	VarChar		20		Yes
43	InputPattern_USPREP	☐	VarChar		88		Yes
44	OutboundPattern_USP	☐	VarChar		88		Yes
45	UserOverrideFlag_USP	☐	VarChar		2		Yes
46	CustomFlag_USPREP	☐	VarChar		2		Yes
47	ZIP3	☐	VarChar		3		Yes
		☐					

Figure 2-142 Create j08_USPREP_CASS 4/12


ADD_ZIP3a - Transformer Stage

OUT2
HOME_PHONE
HOME_ADDRESS
HOME_ZIP
WORK_ADDRESS
WORK_ZIP
PREF_LANG
ISOCountryCode_COUNTRY
IdentifierFlag_COUNTRY
Space_Field
Zip5_CASS
Zip4_CASS

IN

Constraint:

Derivation	Column Name
OUT2.Field4Pattern_USPREP	Field4Pattern_USPREP
OUT2.Field5Pattern_USPREP	Field5Pattern_USPREP
OUT2.Field6Pattern_USPREP	Field6Pattern_USPREP
OUT2.InputPattern_USPREP	InputPattern_USPREP
OUT2.OutboundPattern_USPREP	OutboundPattern_USPREP
OUT2.UserOverrideFlag_USPREP	UserOverrideFlag_USPREP
OUT2.CustomFlag_USPREP	CustomFlag_USPREP
OUT2.Zip5_CASS[1,3]	ZIP3

OUT2

	Column name	Key	SQL type	Extended	Length
11	ISOCountryCode_	☐	VarChar		3
12	IdentifierFlag_COI	☐	VarChar		2
13	Space_Field	☐	Char		1
14	Zip5_CASS	☐	VarChar		5
15	Zip4_CASS	☐	VarChar		4
16	DeliveryPoint_CA:	☐	VarChar		2
17	DeliveryPointChkI	☐	VarChar		1
18	CarrierRoute_CAS	☐	VarChar		4

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable
42	Field6Pattern_USPREF	☐	VarChar		20		Yes
43	InputPattern_USPREP	☐	VarChar		88		Yes
44	OutboundPattern_USP	☐	VarChar		88		Yes
45	UserOverrideFlag_USP	☐	VarChar		2		Yes
46	CustomFlag_USPREP	☐	VarChar		2		Yes
47	ZIP3	☐	VarChar		3		Yes
		☐					

Figure 2-143 Create j08_USPREP_CASS 5/12

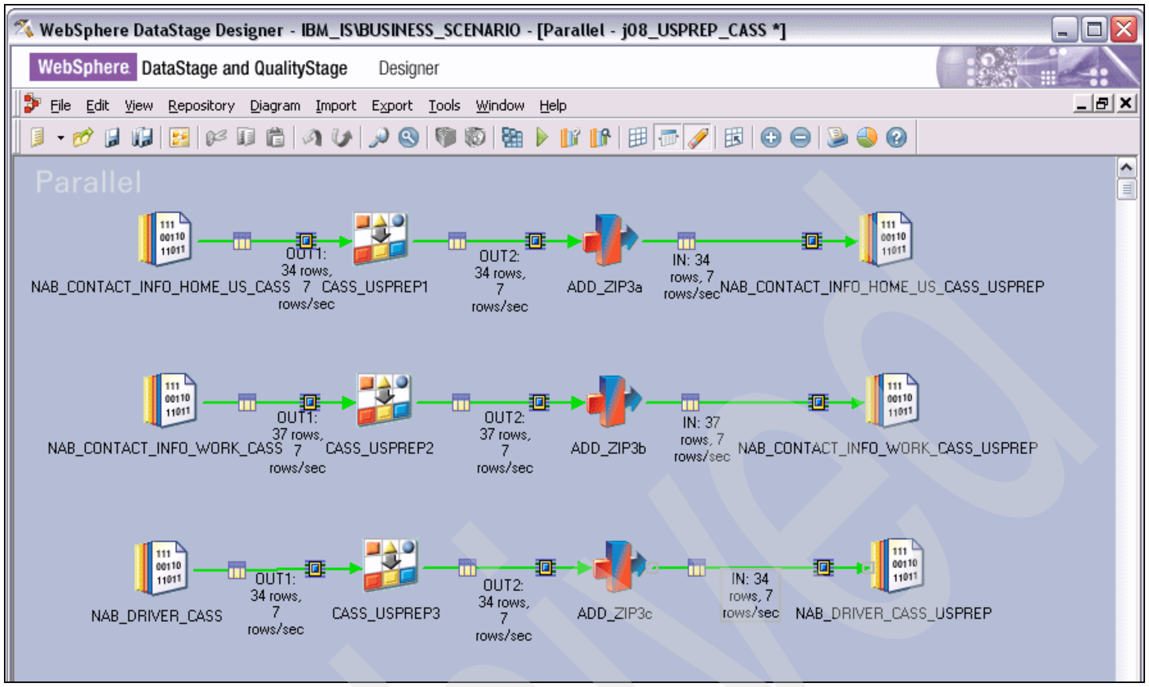
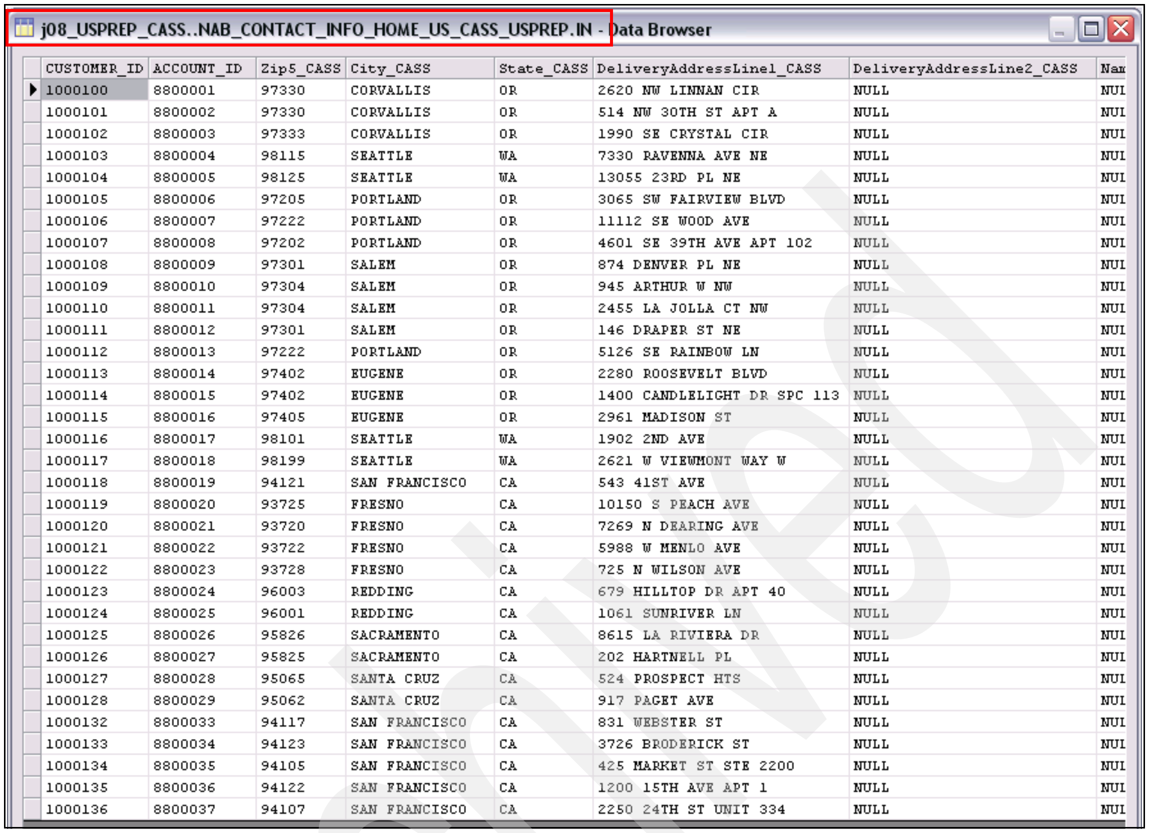

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j08_USPREP_CASS *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NAB_CONTACT_INFO_HOME_US_CASS
OUT1: 34 rows, 7 rows/sec
CASS_USPREP1
OUT2: 34 rows, 7 rows/sec
ADD_ZIP3a
IN: 34 rows, 7 rows/sec
NAB_CONTACT_INFO_HOME_US_CASS_USPREP
NAB_CONTACT_INFO_WORK_CASS
OUT1: 37 rows, 7 rows/sec
CASS_USPREP2
OUT2: 37 rows, 7 rows/sec
ADD_ZIP3b
IN: 37 rows, 7 rows/sec
NAB_CONTACT_INFO_WORK_CASS_USPREP
NAB_DRIVER_CASS
OUT1: 34 rows, 7 rows/sec
CASS_USPREP3
OUT2: 34 rows, 7 rows/sec
ADD_ZIP3c
IN: 34 rows, 7 rows/sec
NAB_DRIVER_CASS_USPREP

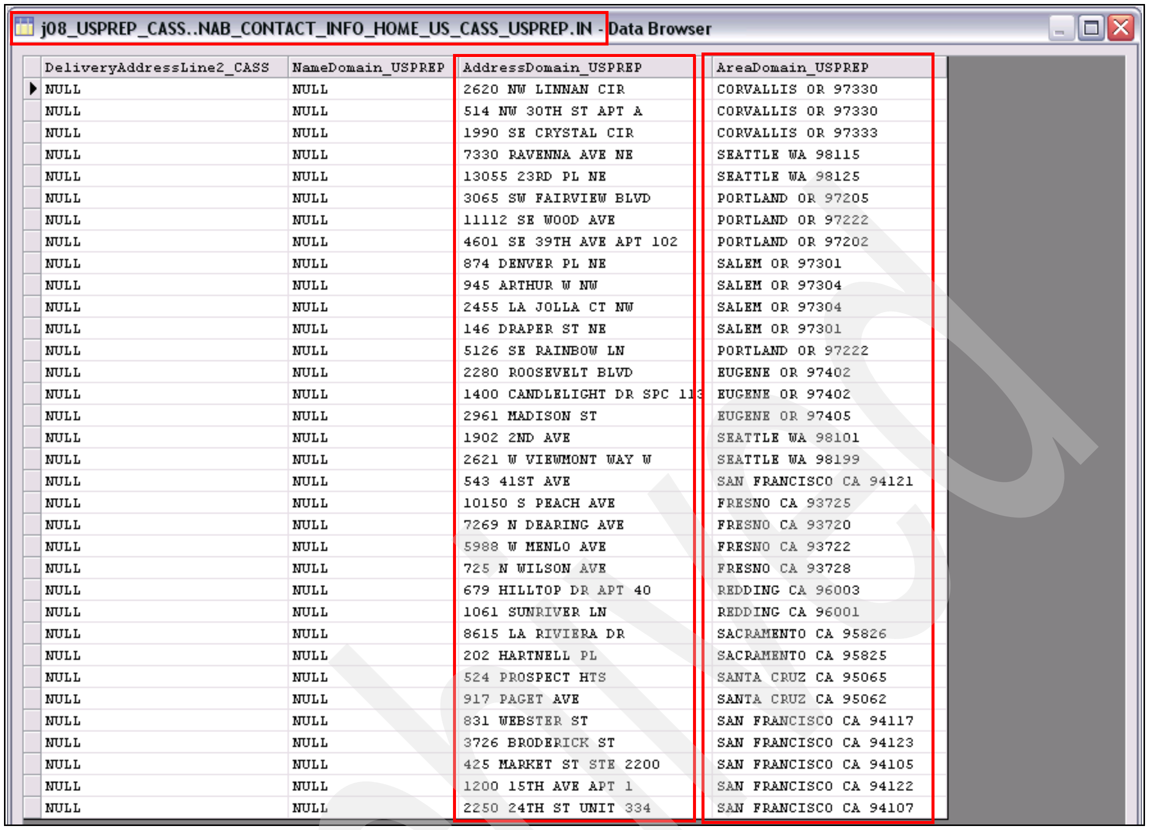
Figure 2-144 Create j08_USPREP_CASS 6/12



j08_USPREP_CASS..NAB_CONTACT_INFO_HOME_US_CASS_USPREP.IN - Data Browser

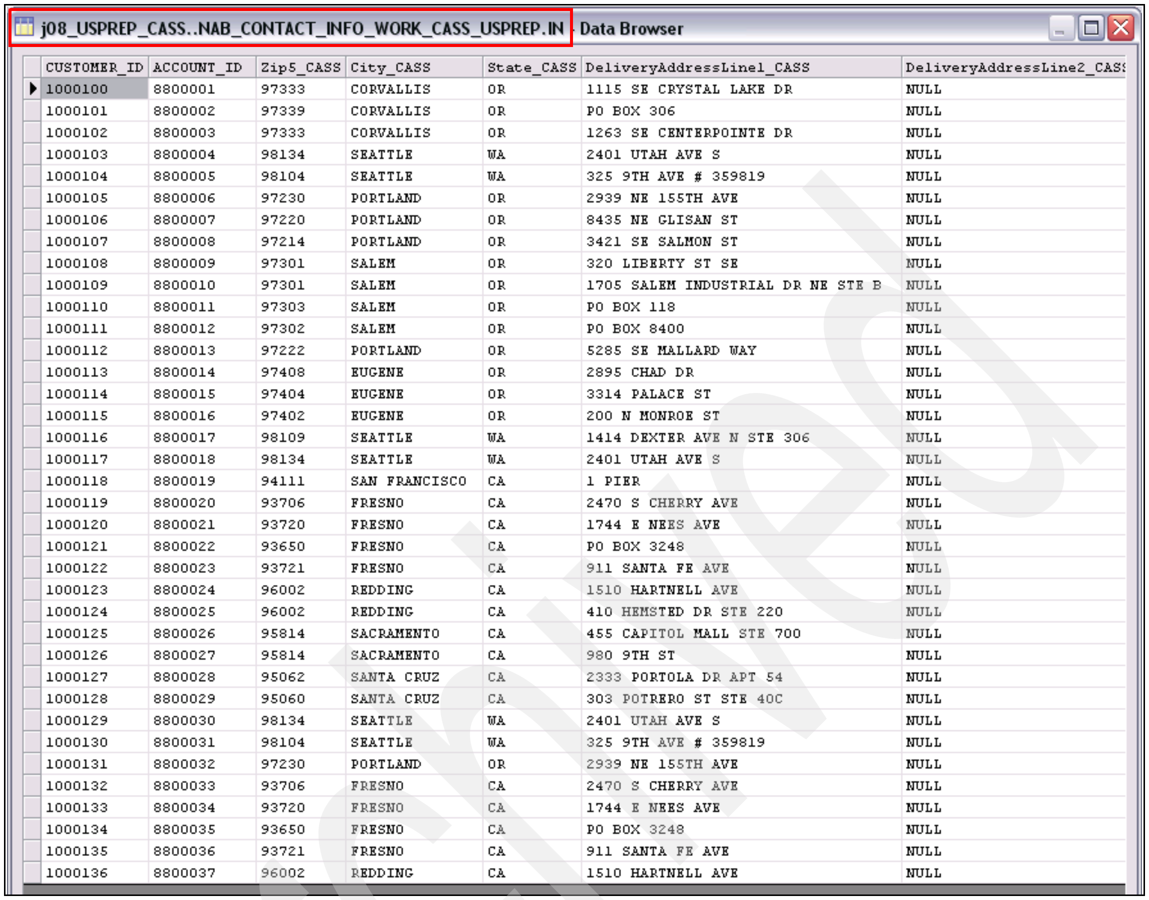
CUSTOMER_ID	ACCOUNT_ID	Zip5_CASS	City_CASS	State_CASS	DeliveryAddressLine1_CASS	DeliveryAddressLine2_CASS	Nam
1000100	8800001	97330	CORVALLIS	OR	2620 NW LINNAN CIR	NULL	NUI
1000101	8800002	97330	CORVALLIS	OR	514 NW 30TH ST APT A	NULL	NUI
1000102	8800003	97333	CORVALLIS	OR	1990 SE CRYSTAL CIR	NULL	NUI
1000103	8800004	98115	SEATTLE	WA	7330 RAVENNA AVE NE	NULL	NUI
1000104	8800005	98125	SEATTLE	WA	13055 23RD PL NE	NULL	NUI
1000105	8800006	97205	PORTLAND	OR	3065 SW FAIRVIEW BLVD	NULL	NUI
1000106	8800007	97222	PORTLAND	OR	11112 SE WOOD AVE	NULL	NUI
1000107	8800008	97202	PORTLAND	OR	4601 SE 39TH AVE APT 102	NULL	NUI
1000108	8800009	97301	SALEM	OR	874 DENVER PL NE	NULL	NUI
1000109	8800010	97304	SALEM	OR	945 ARTHUR W NW	NULL	NUI
1000110	8800011	97304	SALEM	OR	2455 LA JOLLA CT NW	NULL	NUI
1000111	8800012	97301	SALEM	OR	146 DRAPER ST NE	NULL	NUI
1000112	8800013	97222	PORTLAND	OR	5126 SE RAINBOW LN	NULL	NUI
1000113	8800014	97402	EUGENE	OR	2280 ROOSEVELT BLVD	NULL	NUI
1000114	8800015	97402	EUGENE	OR	1400 CANDLELIGHT DR SPC 113	NULL	NUI
1000115	8800016	97405	EUGENE	OR	2961 MADISON ST	NULL	NUI
1000116	8800017	98101	SEATTLE	WA	1902 2ND AVE	NULL	NUI
1000117	8800018	98199	SEATTLE	WA	2621 W VIEWMONT WAY W	NULL	NUI
1000118	8800019	94121	SAN FRANCISCO	CA	543 41ST AVE	NULL	NUI
1000119	8800020	93725	FRESNO	CA	10150 S PEACH AVE	NULL	NUI
1000120	8800021	93720	FRESNO	CA	7269 N DEARING AVE	NULL	NUI
1000121	8800022	93722	FRESNO	CA	5988 W MENLO AVE	NULL	NUI
1000122	8800023	93728	FRESNO	CA	725 N WILSON AVE	NULL	NUI
1000123	8800024	96003	REDDING	CA	679 HILLTOP DR APT 40	NULL	NUI
1000124	8800025	96001	REDDING	CA	1061 SUNRIVER LN	NULL	NUI
1000125	8800026	95826	SACRAMENTO	CA	8615 LA RIVIERA DR	NULL	NUI
1000126	8800027	95825	SACRAMENTO	CA	202 HARTNELL PL	NULL	NUI
1000127	8800028	95065	SANTA CRUZ	CA	524 PROSPECT HTS	NULL	NUI
1000128	8800029	95062	SANTA CRUZ	CA	917 PAGET AVE	NULL	NUI
1000132	8800033	94117	SAN FRANCISCO	CA	831 WEBSTER ST	NULL	NUI
1000133	8800034	94123	SAN FRANCISCO	CA	3726 BRODERICK ST	NULL	NUI
1000134	8800035	94105	SAN FRANCISCO	CA	425 MARKET ST STE 2200	NULL	NUI
1000135	8800036	94122	SAN FRANCISCO	CA	1200 15TH AVE APT 1	NULL	NUI
1000136	8800037	94107	SAN FRANCISCO	CA	2250 24TH ST UNIT 334	NULL	NUI

Figure 2-145 Create j08_USPREP_CASS 7/12


j08_USPREP_CASS..NAB_CONTACT_INFO_HOME_US_CASS_USPREP.IN - Data Browser

DeliveryAddressLine2_CASS	NameDomain_USPREP	AddressDomain_USPREP	AreaDomain_USPREP
NULL	NULL	2620 NW LINNAN CIR	CORVALLIS OR 97330
NULL	NULL	514 NW 30TH ST APT A	CORVALLIS OR 97330
NULL	NULL	1990 SE CRYSTAL CIR	CORVALLIS OR 97333
NULL	NULL	7330 RAVENNA AVE NE	SEATTLE WA 98115
NULL	NULL	13055 23RD PL NE	SEATTLE WA 98125
NULL	NULL	3065 SW FAIRVIEW BLVD	PORTLAND OR 97205
NULL	NULL	11112 SE WOOD AVE	PORTLAND OR 97222
NULL	NULL	4601 SE 39TH AVE APT 102	PORTLAND OR 97202
NULL	NULL	874 DENVER PL NE	SALEM OR 97301
NULL	NULL	945 ARTHUR W NW	SALEM OR 97304
NULL	NULL	2455 LA JOLLA CT NW	SALEM OR 97304
NULL	NULL	146 DRAPER ST NE	SALEM OR 97301
NULL	NULL	5126 SE RAINBOW LN	PORTLAND OR 97222
NULL	NULL	2280 ROOSEVELT BLVD	EUGENE OR 97402
NULL	NULL	1400 CANDLELIGHT DR SPC 113	EUGENE OR 97402
NULL	NULL	2961 MADISON ST	EUGENE OR 97405
NULL	NULL	1902 2ND AVE	SEATTLE WA 98101
NULL	NULL	2621 W VIEWMONT WAY W	SEATTLE WA 98199
NULL	NULL	543 41ST AVE	SAN FRANCISCO CA 94121
NULL	NULL	10150 S PEACH AVE	FRESNO CA 93725
NULL	NULL	7269 N DEARING AVE	FRESNO CA 93720
NULL	NULL	5988 W MENLO AVE	FRESNO CA 93722
NULL	NULL	725 N WILSON AVE	FRESNO CA 93728
NULL	NULL	679 HILLTOP DR APT 40	REDDING CA 96003
NULL	NULL	1061 SUNRIVER LN	REDDING CA 96001
NULL	NULL	8615 LA RIVIERA DR	SACRAMENTO CA 95826
NULL	NULL	202 HARTNELL PL	SACRAMENTO CA 95825
NULL	NULL	524 PROSPECT HTS	SANTA CRUZ CA 95065
NULL	NULL	917 PAGET AVE	SANTA CRUZ CA 95062
NULL	NULL	831 WEBSTER ST	SAN FRANCISCO CA 94117
NULL	NULL	3726 BRODERICK ST	SAN FRANCISCO CA 94123
NULL	NULL	425 MARKET ST STE 2200	SAN FRANCISCO CA 94105
NULL	NULL	1200 15TH AVE APT 1	SAN FRANCISCO CA 94122
NULL	NULL	2250 24TH ST UNIT 334	SAN FRANCISCO CA 94107

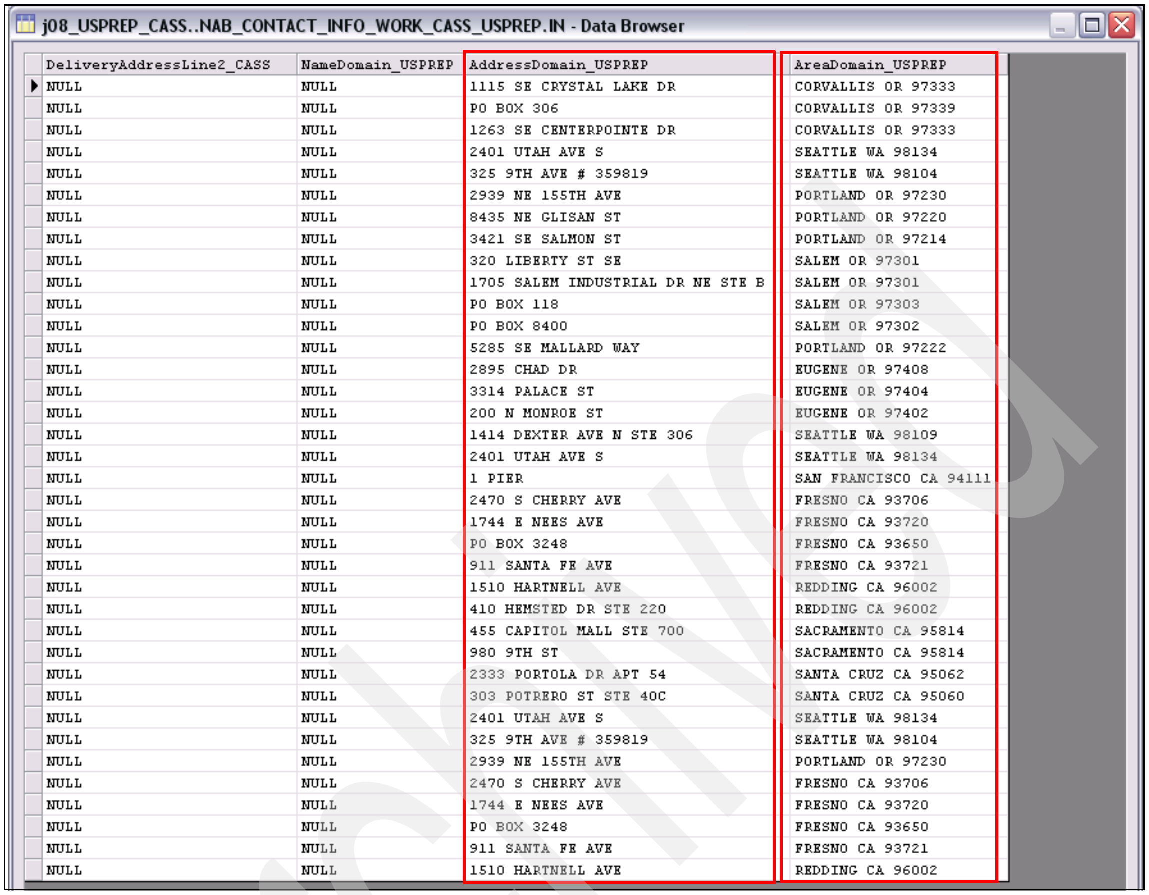
Figure 2-146   Create j08_USPREP_CASS 8/12



j08_USPREP_CASS..NAB_CONTACT_INFO_WORK_CASS_USPREP.IN - Data Browser

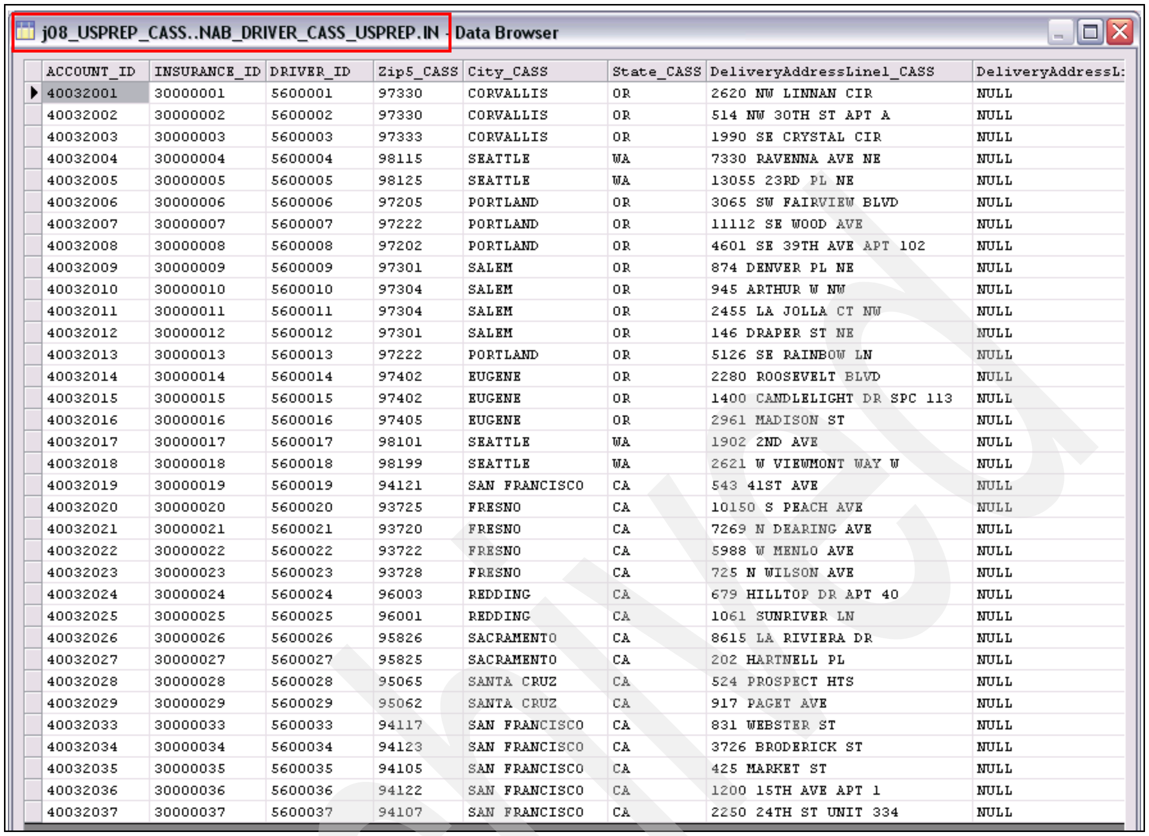
CUSTOMER_ID	ACCOUNT_ID	Zip5_CASS	City_CASS	State_CASS	DeliveryAddressLine1_CASS	DeliveryAddressLine2_CAS
1000100	8800001	97333	CORVALLIS	OR	1115 SE CRYSTAL LAKE DR	NULL
1000101	8800002	97339	CORVALLIS	OR	PO BOX 306	NULL
1000102	8800003	97333	CORVALLIS	OR	1263 SE CENTERPOINTE DR	NULL
1000103	8800004	98134	SEATTLE	WA	2401 UTAH AVE S	NULL
1000104	8800005	98104	SEATTLE	WA	325 9TH AVE # 359819	NULL
1000105	8800006	97230	PORTLAND	OR	2939 NE 155TH AVE	NULL
1000106	8800007	97220	PORTLAND	OR	8435 NE GLISAN ST	NULL
1000107	8800008	97214	PORTLAND	OR	3421 SE SALMON ST	NULL
1000108	8800009	97301	SALEM	OR	320 LIBERTY ST SE	NULL
1000109	8800010	97301	SALEM	OR	1705 SALEM INDUSTRIAL DR NE STE B	NULL
1000110	8800011	97303	SALEM	OR	PO BOX 118	NULL
1000111	8800012	97302	SALEM	OR	PO BOX 8400	NULL
1000112	8800013	97222	PORTLAND	OR	5285 SE MALLARD WAY	NULL
1000113	8800014	97408	EUGENE	OR	2895 CHAD DR	NULL
1000114	8800015	97404	EUGENE	OR	3314 PALACE ST	NULL
1000115	8800016	97402	EUGENE	OR	200 N MONROE ST	NULL
1000116	8800017	98109	SEATTLE	WA	1414 DEXTER AVE N STE 306	NULL
1000117	8800018	98134	SEATTLE	WA	2401 UTAH AVE S	NULL
1000118	8800019	94111	SAN FRANCISCO	CA	1 PIER	NULL
1000119	8800020	93706	FRESNO	CA	2470 S CHERRY AVE	NULL
1000120	8800021	93720	FRESNO	CA	1744 E NEES AVE	NULL
1000121	8800022	93650	FRESNO	CA	PO BOX 3248	NULL
1000122	8800023	93721	FRESNO	CA	911 SANTA FE AVE	NULL
1000123	8800024	96002	REDDING	CA	1510 HARTNELL AVE	NULL
1000124	8800025	96002	REDDING	CA	410 HEMSTED DR STE 220	NULL
1000125	8800026	95814	SACRAMENTO	CA	455 CAPITOL MALL STE 700	NULL
1000126	8800027	95814	SACRAMENTO	CA	980 9TH ST	NULL
1000127	8800028	95062	SANTA CRUZ	CA	2333 PORTOLA DR APT 54	NULL
1000128	8800029	95060	SANTA CRUZ	CA	303 POTRERO ST STE 40C	NULL
1000129	8800030	98134	SEATTLE	WA	2401 UTAH AVE S	NULL
1000130	8800031	98104	SEATTLE	WA	325 9TH AVE # 359819	NULL
1000131	8800032	97230	PORTLAND	OR	2939 NE 155TH AVE	NULL
1000132	8800033	93706	FRESNO	CA	2470 S CHERRY AVE	NULL
1000133	8800034	93720	FRESNO	CA	1744 E NEES AVE	NULL
1000134	8800035	93650	FRESNO	CA	PO BOX 3248	NULL
1000135	8800036	93721	FRESNO	CA	911 SANTA FE AVE	NULL
1000136	8800037	96002	REDDING	CA	1510 HARTNELL AVE	NULL

Figure 2-147 Create j08_USPREP_CASS 9/12


j08_USPREP_CASS..NAB_CONTACT_INFO_WORK_CASS_USPREP.IN - Data Browser

DeliveryAddressLine2_CASS	NameDomain_USPREP	AddressDomain_USPREP	AreaDomain_USPREP
NULL	NULL	1115 SE CRYSTAL LAKE DR	CORVALLIS OR 97333
NULL	NULL	PO BOX 306	CORVALLIS OR 97339
NULL	NULL	1263 SE CENTERPOINTE DR	CORVALLIS OR 97333
NULL	NULL	2401 UTAH AVE S	SEATTLE WA 98134
NULL	NULL	325 9TH AVE # 359819	SEATTLE WA 98104
NULL	NULL	2939 NE 155TH AVE	PORTLAND OR 97230
NULL	NULL	8435 NE GLISAN ST	PORTLAND OR 97220
NULL	NULL	3421 SE SALMON ST	PORTLAND OR 97214
NULL	NULL	320 LIBERTY ST SE	SALEM OR 97301
NULL	NULL	1705 SALEM INDUSTRIAL DR NE STE B	SALEM OR 97301
NULL	NULL	PO BOX 118	SALEM OR 97303
NULL	NULL	PO BOX 8400	SALEM OR 97302
NULL	NULL	5285 SE MALLARD WAY	PORTLAND OR 97222
NULL	NULL	2895 CHAD DR	EUGENE OR 97408
NULL	NULL	3314 PALACE ST	EUGENE OR 97404
NULL	NULL	200 N MONROE ST	EUGENE OR 97402
NULL	NULL	1414 DEXTER AVE N STE 306	SEATTLE WA 98109
NULL	NULL	2401 UTAH AVE S	SEATTLE WA 98134
NULL	NULL	1 PIER	SAN FRANCISCO CA 94111
NULL	NULL	2470 S CHERRY AVE	FRESNO CA 93706
NULL	NULL	1744 E NEES AVE	FRESNO CA 93720
NULL	NULL	PO BOX 3248	FRESNO CA 93650
NULL	NULL	911 SANTA FE AVE	FRESNO CA 93721
NULL	NULL	1510 HARTNELL AVE	REDDING CA 96002
NULL	NULL	410 HEMSTED DR STE 220	REDDING CA 96002
NULL	NULL	455 CAPITOL MALL STE 700	SACRAMENTO CA 95814
NULL	NULL	980 9TH ST	SACRAMENTO CA 95814
NULL	NULL	2333 PORTOLA DR APT 54	SANTA CRUZ CA 95062
NULL	NULL	303 POTRERO ST STE 40C	SANTA CRUZ CA 95060
NULL	NULL	2401 UTAH AVE S	SEATTLE WA 98134
NULL	NULL	325 9TH AVE # 359819	SEATTLE WA 98104
NULL	NULL	2939 NE 155TH AVE	PORTLAND OR 97230
NULL	NULL	2470 S CHERRY AVE	FRESNO CA 93706
NULL	NULL	1744 E NEES AVE	FRESNO CA 93720
NULL	NULL	PO BOX 3248	FRESNO CA 93650
NULL	NULL	911 SANTA FE AVE	FRESNO CA 93721
NULL	NULL	1510 HARTNELL AVE	REDDING CA 96002

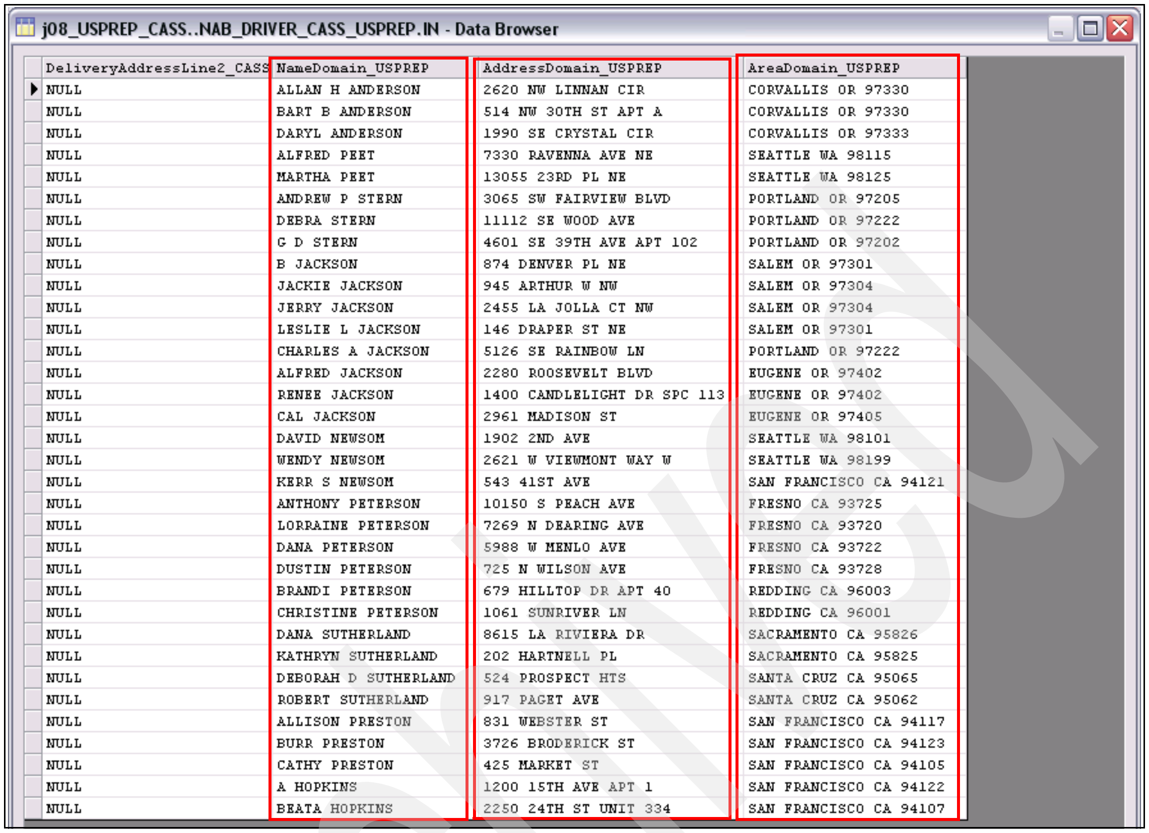
Figure 2-148 Create j08_USPREP_CASS 10/12



j08_USPREP_CASS..NAB_DRIVER_CASS_USPREP.IN - Data Browser

ACCOUNT_ID	INSURANCE_ID	DRIVER_ID	Zip5_CASS	City_CASS	State_CASS	DeliveryAddressLine1_CASS	DeliveryAddressL:
40032001	30000001	5600001	97330	CORVALLIS	OR	2620 NW LINNAN CIR	NULL
40032002	30000002	5600002	97330	CORVALLIS	OR	514 NW 30TH ST APT A	NULL
40032003	30000003	5600003	97333	CORVALLIS	OR	1990 SE CRYSTAL CIR	NULL
40032004	30000004	5600004	98115	SEATTLE	WA	7330 RAVENNA AVE NE	NULL
40032005	30000005	5600005	98125	SEATTLE	WA	13055 23RD PL NE	NULL
40032006	30000006	5600006	97205	PORTLAND	OR	3065 SW FAIRVIEW BLVD	NULL
40032007	30000007	5600007	97222	PORTLAND	OR	11112 SE WOOD AVE	NULL
40032008	30000008	5600008	97202	PORTLAND	OR	4601 SE 39TH AVE APT 102	NULL
40032009	30000009	5600009	97301	SALEM	OR	874 DENVER PL NE	NULL
40032010	30000010	5600010	97304	SALEM	OR	945 ARTHUR W NW	NULL
40032011	30000011	5600011	97304	SALEM	OR	2455 LA JOLLA CT NW	NULL
40032012	30000012	5600012	97301	SALEM	OR	146 DRAPER ST NE	NULL
40032013	30000013	5600013	97222	PORTLAND	OR	5126 SE RAINBOW LN	NULL
40032014	30000014	5600014	97402	EUGENE	OR	2280 ROOSEVELT BLVD	NULL
40032015	30000015	5600015	97402	EUGENE	OR	1400 CANDLELIGHT DR SPC 113	NULL
40032016	30000016	5600016	97405	EUGENE	OR	2961 MADISON ST	NULL
40032017	30000017	5600017	98101	SEATTLE	WA	1902 2ND AVE	NULL
40032018	30000018	5600018	98199	SEATTLE	WA	2621 W VIEWMONT WAY W	NULL
40032019	30000019	5600019	94121	SAN FRANCISCO	CA	543 41ST AVE	NULL
40032020	30000020	5600020	93725	FRESNO	CA	10150 S PEACH AVE	NULL
40032021	30000021	5600021	93720	FRESNO	CA	7269 N DEARING AVE	NULL
40032022	30000022	5600022	93722	FRESNO	CA	5988 W MENLO AVE	NULL
40032023	30000023	5600023	93728	FRESNO	CA	725 N WILSON AVE	NULL
40032024	30000024	5600024	96003	REDDING	CA	679 HILLTOP DR APT 40	NULL
40032025	30000025	5600025	96001	REDDING	CA	1061 SUNRIVER LN	NULL
40032026	30000026	5600026	95826	SACRAMENTO	CA	8615 LA RIVIERA DR	NULL
40032027	30000027	5600027	95825	SACRAMENTO	CA	202 HARTNELL PL	NULL
40032028	30000028	5600028	95065	SANTA CRUZ	CA	524 PROSPECT HTS	NULL
40032029	30000029	5600029	95062	SANTA CRUZ	CA	917 PAGET AVE	NULL
40032033	30000033	5600033	94117	SAN FRANCISCO	CA	831 WEBSTER ST	NULL
40032034	30000034	5600034	94123	SAN FRANCISCO	CA	3726 BRODERICK ST	NULL
40032035	30000035	5600035	94105	SAN FRANCISCO	CA	425 MARKET ST	NULL
40032036	30000036	5600036	94122	SAN FRANCISCO	CA	1200 15TH AVE APT 1	NULL
40032037	30000037	5600037	94107	SAN FRANCISCO	CA	2250 24TH ST UNIT 334	NULL

Figure 2-149   Create j08_USPREP_CASS 11/12


j08_USPREP_CASS..NAB_DRIVER_CASS_USPREP.IN - Data Browser

DeliveryAddressLine2_CASS	NameDomain_USPREP	AddressDomain_USPREP	AreaDomain_USPREP
NULL	ALLAN H ANDERSON	2620 NW LINNAN CIR	CORVALLIS OR 97330
NULL	BART B ANDERSON	514 NW 30TH ST APT A	CORVALLIS OR 97330
NULL	DARYL ANDERSON	1990 SE CRYSTAL CIR	CORVALLIS OR 97333
NULL	ALFRED PEET	7330 RAVENNA AVE NE	SEATTLE WA 98115
NULL	MARTHA PEET	13055 23RD PL NE	SEATTLE WA 98125
NULL	ANDREW P STERN	3065 SW FAIRVIEW BLVD	PORTLAND OR 97205
NULL	DEBRA STERN	11112 SE WOOD AVE	PORTLAND OR 97222
NULL	G D STERN	4601 SE 39TH AVE APT 102	PORTLAND OR 97202
NULL	B JACKSON	874 DENVER PL NE	SALEM OR 97301
NULL	JACKIE JACKSON	945 ARTHUR W NW	SALEM OR 97304
NULL	JERRY JACKSON	2455 LA JOLLA CT NW	SALEM OR 97304
NULL	LESLIE L JACKSON	146 DRAPER ST NE	SALEM OR 97301
NULL	CHARLES A JACKSON	5126 SE RAINBOW LN	PORTLAND OR 97222
NULL	ALFRED JACKSON	2280 ROOSEVELT BLVD	EUGENE OR 97402
NULL	RENEE JACKSON	1400 CANDLELIGHT DR SPC 113	EUGENE OR 97402
NULL	CAL JACKSON	2961 MADISON ST	EUGENE OR 97405
NULL	DAVID NEWSOM	1902 2ND AVE	SEATTLE WA 98101
NULL	WENDY NEWSOM	2621 W VIEWMONT WAY W	SEATTLE WA 98199
NULL	KERR S NEWSOM	543 41ST AVE	SAN FRANCISCO CA 94121
NULL	ANTHONY PETERSON	10150 S PEACH AVE	FRESNO CA 93725
NULL	LORRAINE PETERSON	7269 N DEARING AVE	FRESNO CA 93720
NULL	DANA PETERSON	5988 W MENLO AVE	FRESNO CA 93722
NULL	DUSTIN PETERSON	725 N WILSON AVE	FRESNO CA 93728
NULL	BRANDI PETERSON	679 HILLTOP DR APT 40	REDDING CA 96003
NULL	CHRISTINE PETERSON	1061 SUNRIVER LN	REDDING CA 96001
NULL	DANA SUTHERLAND	8615 LA RIVIERA DR	SACRAMENTO CA 95826
NULL	KATHRYN SUTHERLAND	202 HARTNELL PL	SACRAMENTO CA 95825
NULL	DEBORAH D SUTHERLAND	524 PROSPECT HTS	SANTA CRUZ CA 95065
NULL	ROBERT SUTHERLAND	917 PAGET AVE	SANTA CRUZ CA 95062
NULL	ALLISON PRESTON	831 WEBSTER ST	SAN FRANCISCO CA 94117
NULL	BURR PRESTON	3726 BRODERICK ST	SAN FRANCISCO CA 94123
NULL	CATHY PRESTON	425 MARKET ST	SAN FRANCISCO CA 94105
NULL	A HOPKINS	1200 15TH AVE APT 1	SAN FRANCISCO CA 94122
NULL	BEATA HOPKINS	2250 24TH ST UNIT 334	SAN FRANCISCO CA 94107

Figure 2-150 Create j08_USPREP_CASS 12/12

j09_STAN_CASS

In this step, we standardize the name and address contents of the output of jobs “j03_STAN_XXPREP_NAB” on page 530 and “j08_USPREP_CASS” on page 619 using the domain-specific rule sets USNAME (with column NameDomain_USPREP), USADDR (with column AddressDomain_USPREP), USAREA (with column AreaDomain_USPREP), CAADDR (with column AddressDomain_CAPREP), CAAREA (with column AreaDomain_CAPREP). Separate processes are defined—one for each address type, such as U.S. home addresses and Canadian home addresses.

Note: Here again, we had to introduce a null handling Transformer stage to circumvent a bug.

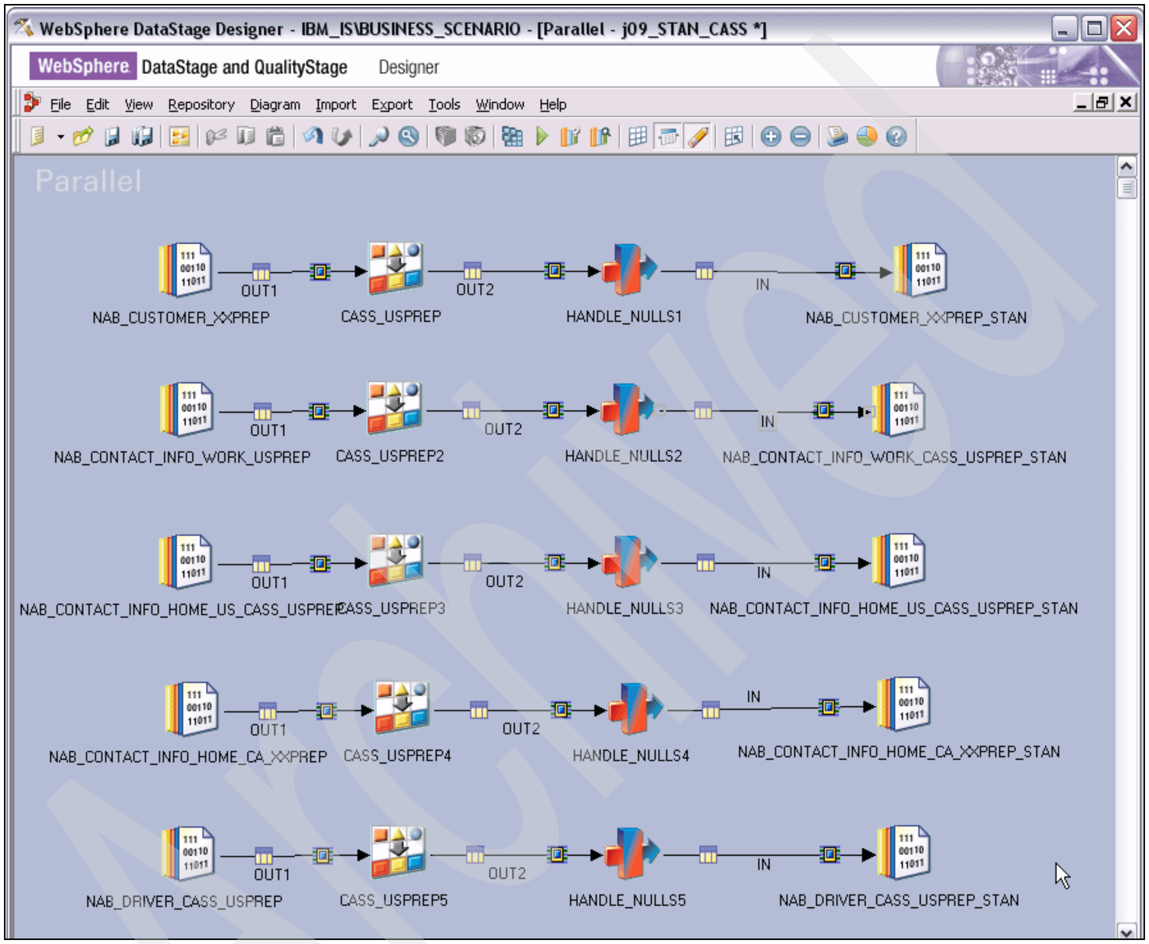
Figure 2-151 on page 633 shows the various stages that are used in this job. It includes the data sets that were created in “j03_STAN_XXPREP_NAB” on page 530 and “j08_USPREP_CASS” on page 619, a Standardize stage, a Transformer stage to handle nulls, and an output Data Set stage for each of the input data sets. We modified the names of the stages as shown.

Because the configuration of this job is very similar to that described in “J08_STAN_CUSTOMER_Domain_Specific” on page 241, we do not repeat it here. However, some of the configurations of interest are as follows:

- Figure 2-152 on page 634 through Figure 2-156 on page 636 show the Standardize Stage window for each of the input data sets as follows:
  - USNAME rule set for the NameDomain_USPREP column for the CUSTOMER file as shown in Figure 2-152 on page 634.
  - USADDR and USAREA rule sets for the AddressDomain_USPREP and AreaDomain_USPREP columns respectively for the U.S. work addresses in the CONTACT_INFO file as shown in Figure 2-153 on page 634.
  - USADDR and USAREA rule sets for the AddressDomain_USPREP and AreaDomain_USPREP columns respectively for the U.S. home addresses in the CONTACT_INFO file as shown in Figure 2-154 on page 635.
  - CAADDR and CAAREA rule sets for the AddressDomain_CAPREP and AreaDomain_CAPREP columns respectively for the Canadian home addresses in the CONTACT_INFO file as shown in Figure 2-155 on page 635.
  - USNAME, USADDR and USAREA rule sets for the NameDomain_USPREP, AddressDomain_USPREP and AreaDomain_USPREP columns respectively for the U.S. addresses in the DRIVER file as shown in Figure 2-156 on page 636.
- After saving, compiling, and running this job (Figure 2-157 on page 637), the standardized output is written to new columns such as:
  - FirstName_USNAME and PrimaryName_USNAME
  - InputPattern_USNAME
  - UnhandledPattern_USNAME
  - UnhandledData_USNAME
  - InputPattern_USNAME
  - UnhandledPattern_USADDR
  - UnhandledData_USADDR
  - InputPattern_USADDR
  - UnhandledPattern_USAREA
  - UnhandledData_USAREA

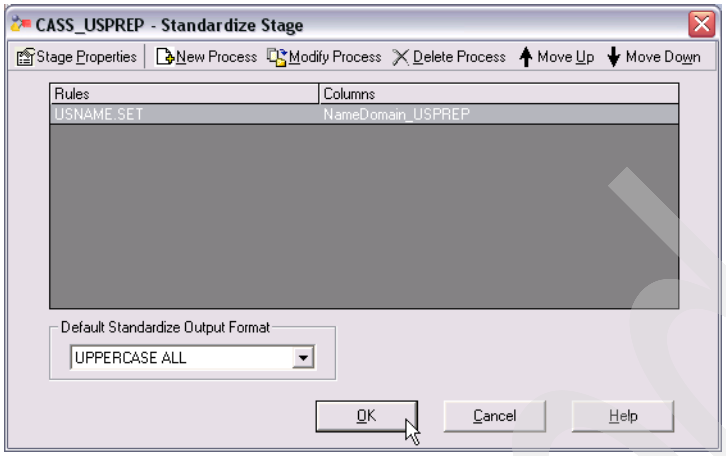
  Similar columns are added for the Canadian home addresses. We do not show these here.

Proceed now to identify any unhandled patterns and classifications by performing an investigate as described in “j10a_INVCC_CUSTOMER_XXPREP_STAN” on page 637 through “j10e_INVCC_DRIVER_CASS_USPREP_STAN” on page 652.



WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j09_STAN_CASS *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NAB_CUSTOMER_XXPREP
OUT1
CASS_USPREP
OUT2
HANDLE_NULLS1
IN
NAB_CUSTOMER_XXPREP_STAN
NAB_CONTACT_INFO_WORK_USPREP
OUT1
CASS_USPREP2
OUT2
HANDLE_NULLS2
IN
NAB_CONTACT_INFO_WORK_CASS_USPREP_STAN
NAB_CONTACT_INFO_HOME_US_CASS_USPREP
OUT1
CASS_USPREP3
OUT2
HANDLE_NULLS3
IN
NAB_CONTACT_INFO_HOME_US_CASS_USPREP_STAN
NAB_CONTACT_INFO_HOME_CA_XXPREP
OUT1
CASS_USPREP4
OUT2
HANDLE_NULLS4
IN
NAB_CONTACT_INFO_HOME_CA_XXPREP_STAN
NAB_DRIVER_CASS_USPREP
OUT1
CASS_USPREP5
OUT2
HANDLE_NULLS5
IN
NAB_DRIVER_CASS_USPREP_STAN

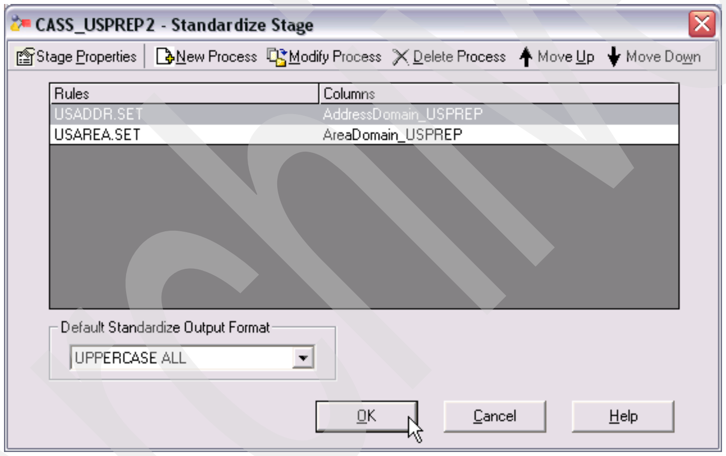
Figure 2-151 Create j09_STAN_CASS 1/7


CASS_USPREP - Standardize Stage

Rules	Columns
USNAME.SET	NameDomain_USPREP

Default Standardize Output Format: UPPERCASE ALL

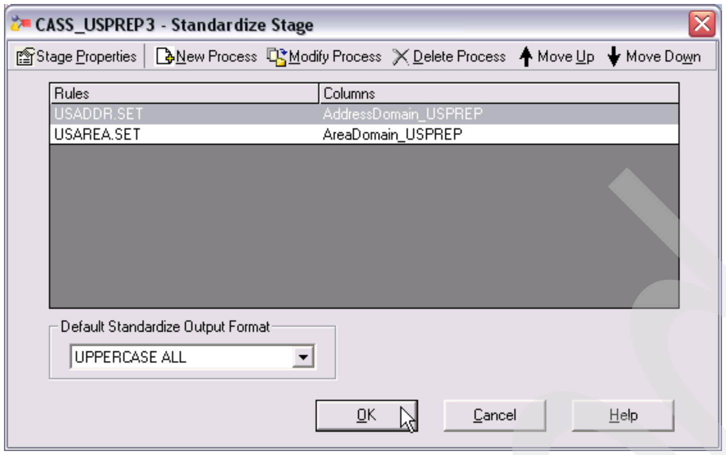
Figure 2-152 Create j09_STAN_CASS 2/7


CASS_USPREP2 - Standardize Stage

Rules	Columns
USADDR.SET	AddressDomain_USPREP
USAREA.SET	AreaDomain_USPREP

Default Standardize Output Format: UPPERCASE ALL

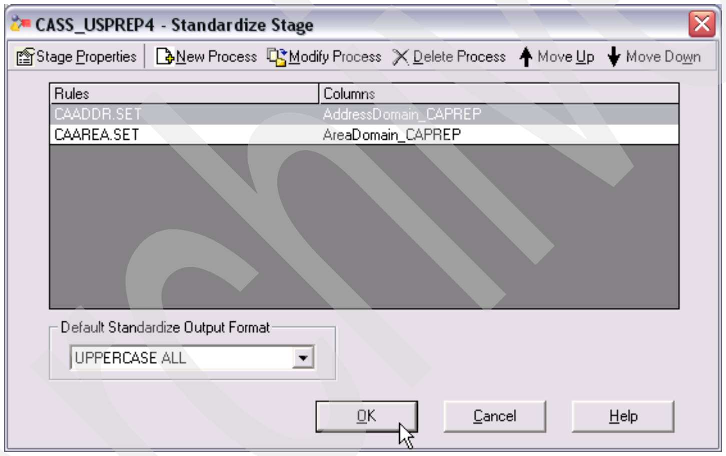
Figure 2-153 Create j09_STAN_CASS 3/7


CASS_USPREP3 - Standardize Stage

Rules	Columns
USADDR.SET	AddressDomain_USPREP
USAREA.SET	AreaDomain_USPREP

Default Standardize Output Format: UPPERCASE ALL

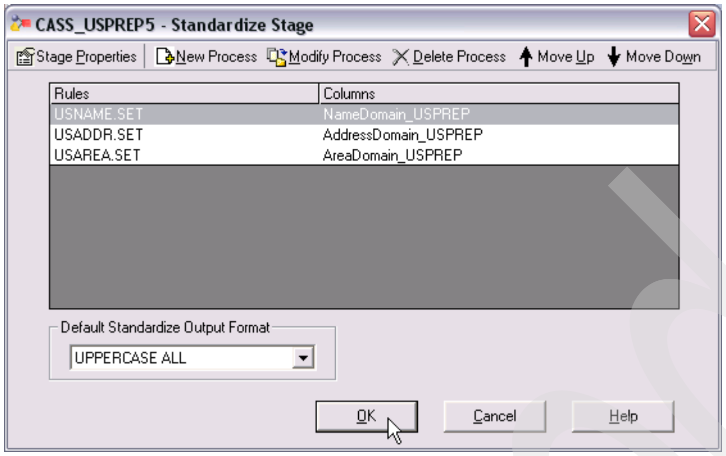
Figure 2-154 Create j09_STAN_CASS 4/7


CASS_USPREP4 - Standardize Stage

Rules	Columns
CAADDR.SET	AddressDomain_CAPREP
CAAREA.SET	AreaDomain_CAPREP

Default Standardize Output Format: UPPERCASE ALL

Figure 2-155 Create j09_STAN_CASS 5/7



Rules	Columns
USNAME.SET	NameDomain_USPREP
USADDR.SET	AddressDomain_USPREP
USAREA.SET	AreaDomain_USPREP

Default Standardize Output Format: UPPERCASE ALL

Figure 2-156 Create j09_STAN_CASS 6/7

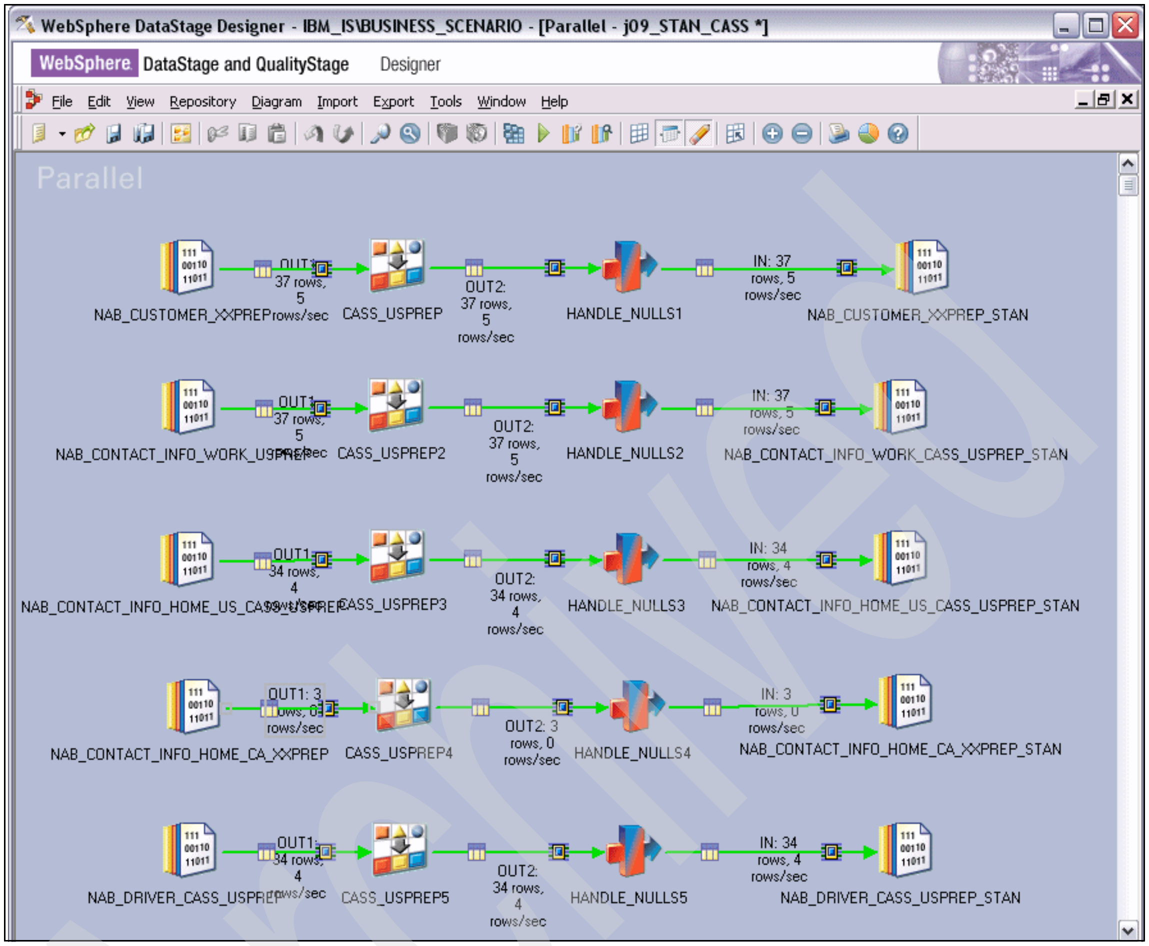

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j09_STAN_CASS *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NAB_CUSTOMER_XXPREP
OUT1: 37 rows, 5 rows/sec
CASS_USPREP
OUT2: 37 rows, 5 rows/sec
HANDLE_NULLS1
IN: 37 rows, 5 rows/sec
NAB_CUSTOMER_XXPREP_STAN
NAB_CONTACT_INFO_WORK_USPREP
OUT1: 37 rows, 5 rows/sec
CASS_USPREP2
OUT2: 37 rows, 5 rows/sec
HANDLE_NULLS2
IN: 37 rows, 5 rows/sec
NAB_CONTACT_INFO_WORK_CASS_USPREP_STAN
NAB_CONTACT_INFO_HOME_US_CASS_USPREP
OUT1: 34 rows, 4 rows/sec
CASS_USPREP3
OUT2: 34 rows, 4 rows/sec
HANDLE_NULLS3
IN: 34 rows, 4 rows/sec
NAB_CONTACT_INFO_HOME_US_CASS_USPREP_STAN
NAB_CONTACT_INFO_HOME_CA_XXPREP
OUT1: 3 rows, 0 rows/sec
CASS_USPREP4
OUT2: 3 rows, 0 rows/sec
HANDLE_NULLS4
IN: 3 rows, 0 rows/sec
NAB_CONTACT_INFO_HOME_CA_XXPREP_STAN
NAB_DRIVER_CASS_USPREP
OUT1: 34 rows, 4 rows/sec
CASS_USPREP5
OUT2: 34 rows, 4 rows/sec
HANDLE_NULLS5
IN: 34 rows, 4 rows/sec
NAB_DRIVER_CASS_USPREP_STAN

Figure 2-157 Create j09_STAN_CASS 7/7

j10a_INVCC_CUSTOMER_XXPREP_STAN

In this step, we identify unhandled patterns and classifications in "j09_STAN_CASS" on page 631. We run a series of Investigate stage with the character concatenate option using the *C* mask on the unhandled pattern column and *X* masks on additional columns. The columns investigated corresponded to the name, address, and area domains. We review unhandled patterns and generate classification and input pattern overrides to rectify the problem. The Standardize stage is then rerun with the overrides in place and verified using Investigate that all the unhandled patterns are resolved.

The following jobs perform these steps:

- “j10a_INVCC_CUSTOMER_XXPREP_STAN” on page 637
- “j10b_INVCC_CONTACT_INFO_WORK_CASS_USPREP_STAN” on page 640
- “j10c_INVCC_CONTACT_INFO_HOME_US_CASS_USPREP_STAN” on page 645
- “j10d_INVCC_CONTACT_INFO_HOME_CA_XXPREP_STAN” on page 647
- “j10e_INVCC_DRIVER_CASS_USPREP_STAN” on page 652

Figure 2-158 on page 639 shows the various stages that are used in this job. It includes the data set that was created in “j09_STAN_CASS” on page 631, an Investigate stage with the character concatenate option, and a sequential file for the column frequency report. We modified the names of the stages as shown.

Because the configuration of this job is very similar to that described in “J09_INVCC_STAN_CUSTOMER” on page 253, we do not repeat it here. However, some of the configurations of interest are as follows:

- Figure 2-159 on page 639 shows the INV1 - Investigate Stage window with the USNAME rule set, and the columns selected for Character Concatenate Investigate as follows:
  - UnhandledPattern_USNAME with *C* mask
  - UnhandledData_USNAME with *X* mask
  - InputPattern_USNAME with *X* mask
  - ExceptionData_USNAME with *X* mask
- After saving, compiling, and running this job (Figure 2-160 on page 640), the contents of the output of the investigate stage shows no unhandled patterns in Figure 2-161 on page 640.

Proceed now to unhandled patterns in the address domain in “j10b_INVCC_CONTACT_INFO_WORK_CASS_USPREP_STAN” on page 640.

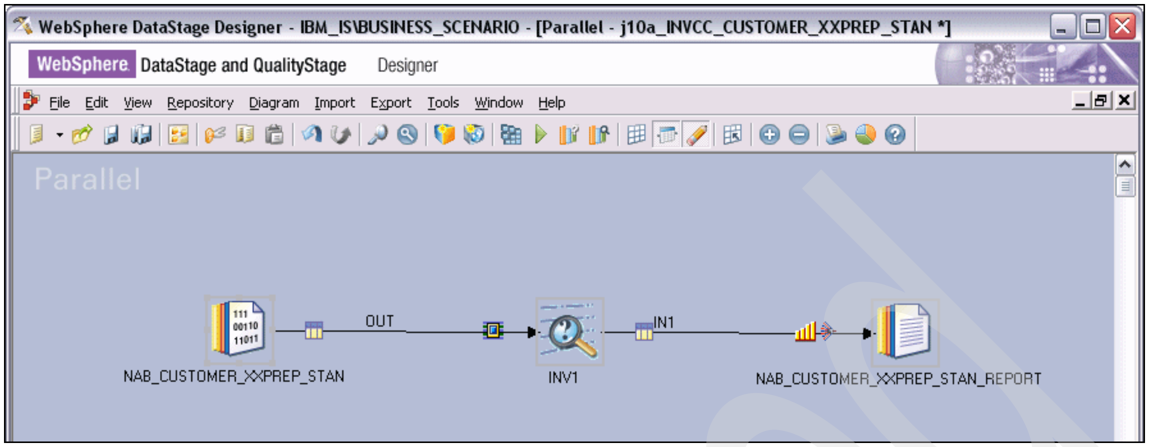
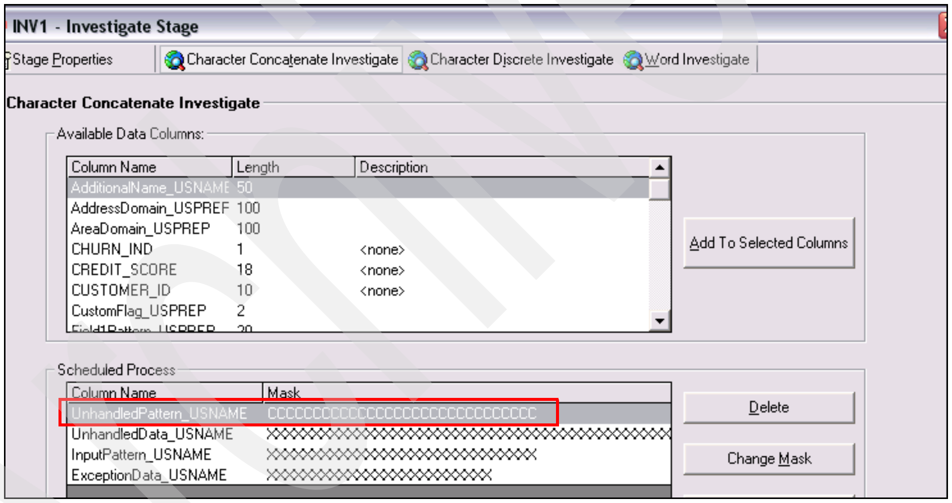

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j10a_INVCC_CUSTOMER_XXPREP_STAN *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
OUT
IN1
NAB_CUSTOMER_XXPREP_STAN
INV1
NAB_CUSTOMER_XXPREP_STAN_REPORT

Figure 2-158 Create j10a_INVCC_CUSTOMER_XXPREP_STAN 1/4



Column Name	Length	Description
AdditionalName_USNAME	50	
AddressDomain_USPREP	100	
AreaDomain_USPREP	100	
CHURN_IND	1	<none>
CREDIT_SCORE	18	<none>
CUSTOMER_ID	10	<none>
CustomFlag_USPREP	2	

Column Name	Mask
UnhandledPattern_USNAME	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
UnhandledData_USNAME	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
InputPattern_USNAME	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
ExceptionData_USNAME	XXXXXXXXXXXXXXXXXXXXXXXXX

Figure 2-159 Create j10a_INVCC_CUSTOMER_XXPREP_STAN 2/4

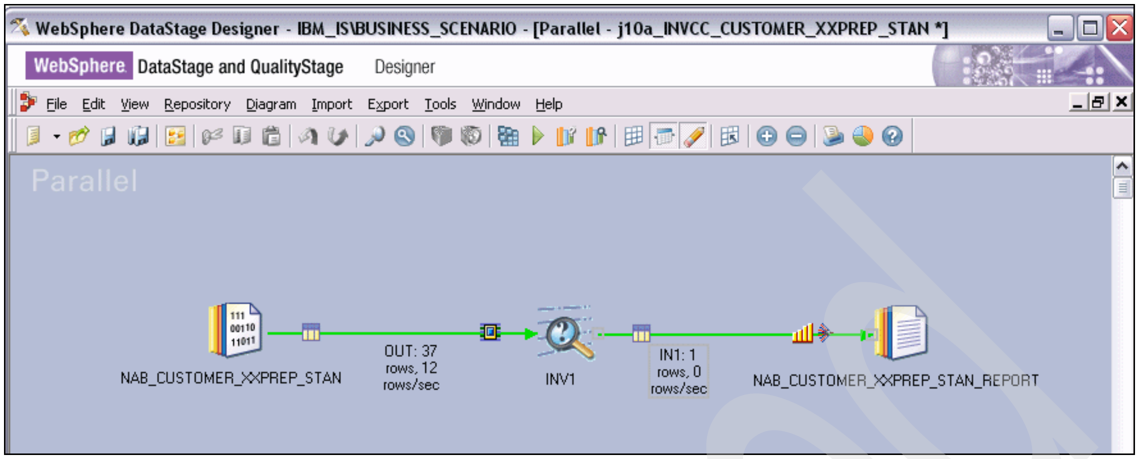
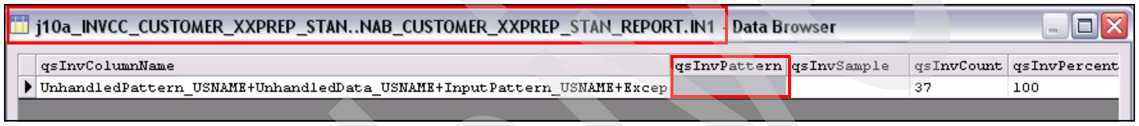

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j10a_INVCC_CUSTOMER_XXPREP_STAN *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NAB_CUSTOMER_XXPREP_STAN
OUT: 37 rows, 12 rows/sec
INV1
IN1: 1 rows, 0 rows/sec
NAB_CUSTOMER_XXPREP_STAN_REPORT

Figure 2-160 Create j10a_INVCC_CUSTOMER_XXPREP_STAN 3/4



j10a_INVCC_CUSTOMER_XXPREP_STAN..NAB_CUSTOMER_XXPREP_STAN_REPORT.IN1 - Data Browser

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
UnhandledPattern_USNAME+UnhandledData_USNAME+InputPattern_USNAME+Excep			37	100

Figure 2-161 Create j10a_INVCC_CUSTOMER_XXPREP_STAN 4/4

j10b_INVCC_CONTACT_INFO_WORK_CASS_USPREP_STAN

In this step, we investigate the unhandled patterns of the U.S. work address CONTACT_INFO file that was created in “j09_STAN_CASS” on page 631.

Because a single Investigate stage can only have a single rule set associated with it, we copy the output data set of “j09_STAN_CASS” on page 631 (using a Copy stage) and process the file by two independent Investigate stages, with each stage using a particular domain-specific rule set. The column frequency report is generated in each Investigate stage.

Figure 2-162 on page 642 shows the various stages that are used in this job. It includes the data set that was created in “j09_STAN_CASS” on page 631, a Copy stage, two Investigate stages that each use a different domain-specific rule set, and a sequential file for each Investigate stage. We modified the names of the stages as shown.

Because the configuration of this job is very similar to that described in "J09_INVCC_STAN_CUSTOMER" on page 253, we do not repeat it here. However, some of the configurations of interest are as follows:

- Figure 2-163 on page 643 shows the INV1 - Investigate Stage window with the USADDR rule set, and the columns selected for Character Concatenate Investigate as follows:
  - UnhandledPattern_USADDR with *C* mask
  - UnhandledData_USADDR with *X* mask
  - InputPattern_USADDR with *X* mask
  - ExceptionData_USADDR with *X* mask
- Figure 2-164 on page 643 shows the INV2 - Investigate Stage window with the USAREA rule set, and the columns selected for Character Concatenate Investigate as follows:
  - UnhandledPattern_USAREA with *C* mask
  - UnhandledData_USAREA with *X* mask
  - InputPattern_USAREA with *X* mask
  - ExceptionData_USAREA with *X* mask
- After saving, compiling, and running this job (Figure 2-165 on page 644), the contents of the output of the investigate stages show no unhandled patterns in Figure 2-166 on page 644 and Figure 2-167 on page 644.

Proceed now to unhandled patterns in the address domain in "j10c_INVCC_CONTACT_INFO_HOME_US_CASS_USPREP_STAN" on page 645.

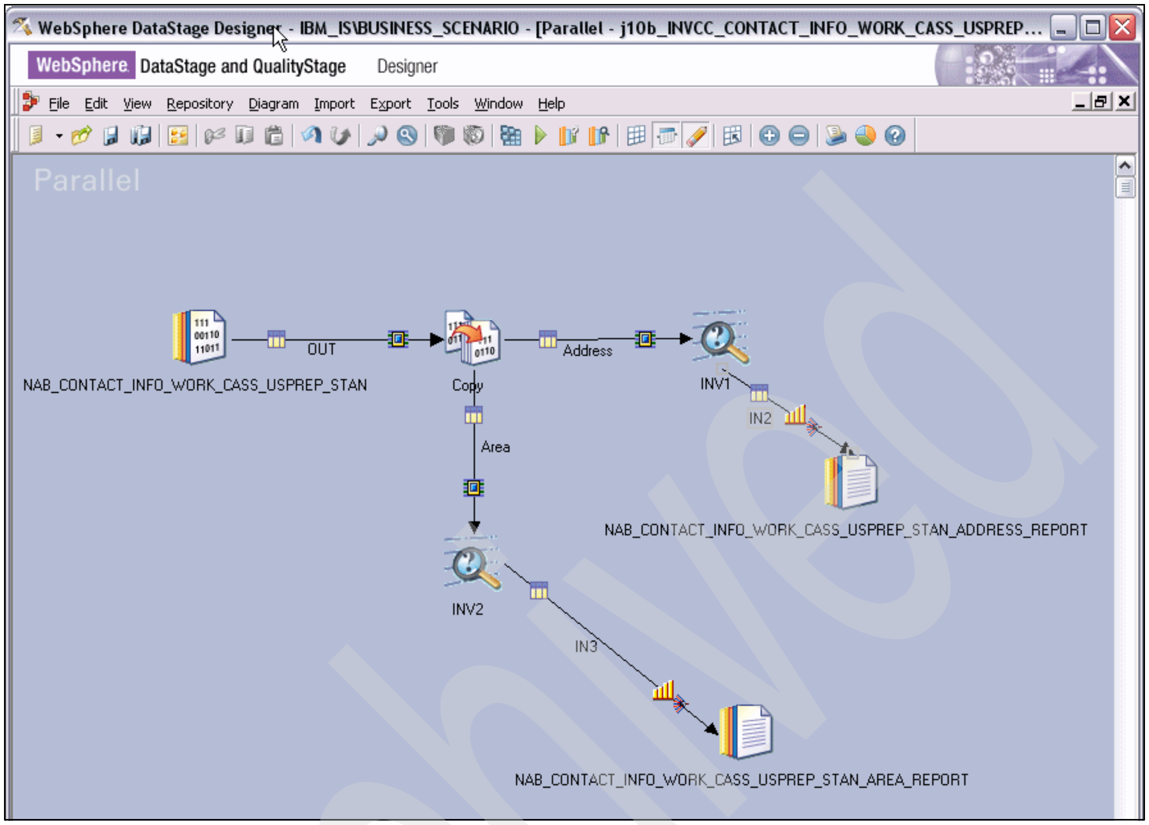
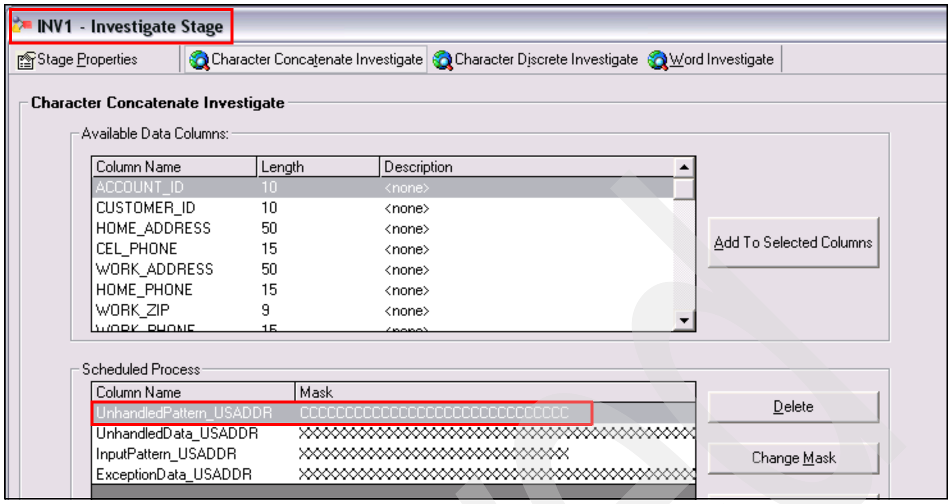

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j10b_INVCC_CONTACT_INFO_WORK_CASS_USPREP...
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NAB_CONTACT_INFO_WORK_CASS_USPREP_STAN
OUT
Copy
Address
INV1
IN2
Area
NAB_CONTACT_INFO_WORK_CASS_USPREP_STAN_ADDRESS_REPORT
INV2
IN3
NAB_CONTACT_INFO_WORK_CASS_USPREP_STAN_AREA_REPORT

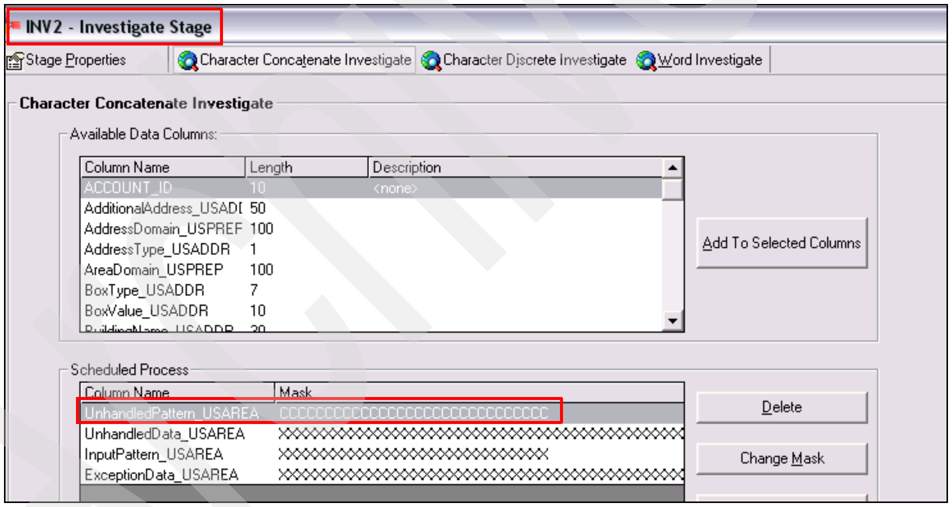
Figure 2-162 Create j10b_INVCC_CONTACT_INFO_WORK_CASS_USPREP_STAN 1/6



Column Name	Length	Description
ACCOUNT_ID	10	<none>
CUSTOMER_ID	10	<none>
HOME_ADDRESS	50	<none>
CEL_PHONE	15	<none>
WORK_ADDRESS	50	<none>
HOME_PHONE	15	<none>
WORK_ZIP	9	<none>
WORK_PHONE	15	<none>

Column Name	Mask
UnhandledPattern_USADDR	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
UnhandledData_USADDR	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
InputPattern_USADDR	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
ExceptionData_USADDR	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Figure 2-163 Create j10b_INVCC_CONTACT_INFO_WORK_CASS_USPREP_STAN 2/6



Column Name	Length	Description
ACCOUNT_ID	10	<none>
AdditionalAddress_USADI	50	
AddressDomain_USPREF	100	
AddressType_USADDR	1	
AreaDomain_USPREP	100	
BoxType_USADDR	7	
BoxValue_USADDR	10	
BuildingName_USADDR	30	

Column Name	Mask
UnhandledPattern_USAREA	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
UnhandledData_USAREA	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
InputPattern_USAREA	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
ExceptionData_USAREA	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Figure 2-164 Create j10b_INVCC_CONTACT_INFO_WORK_CASS_USPREP_STAN 3/6

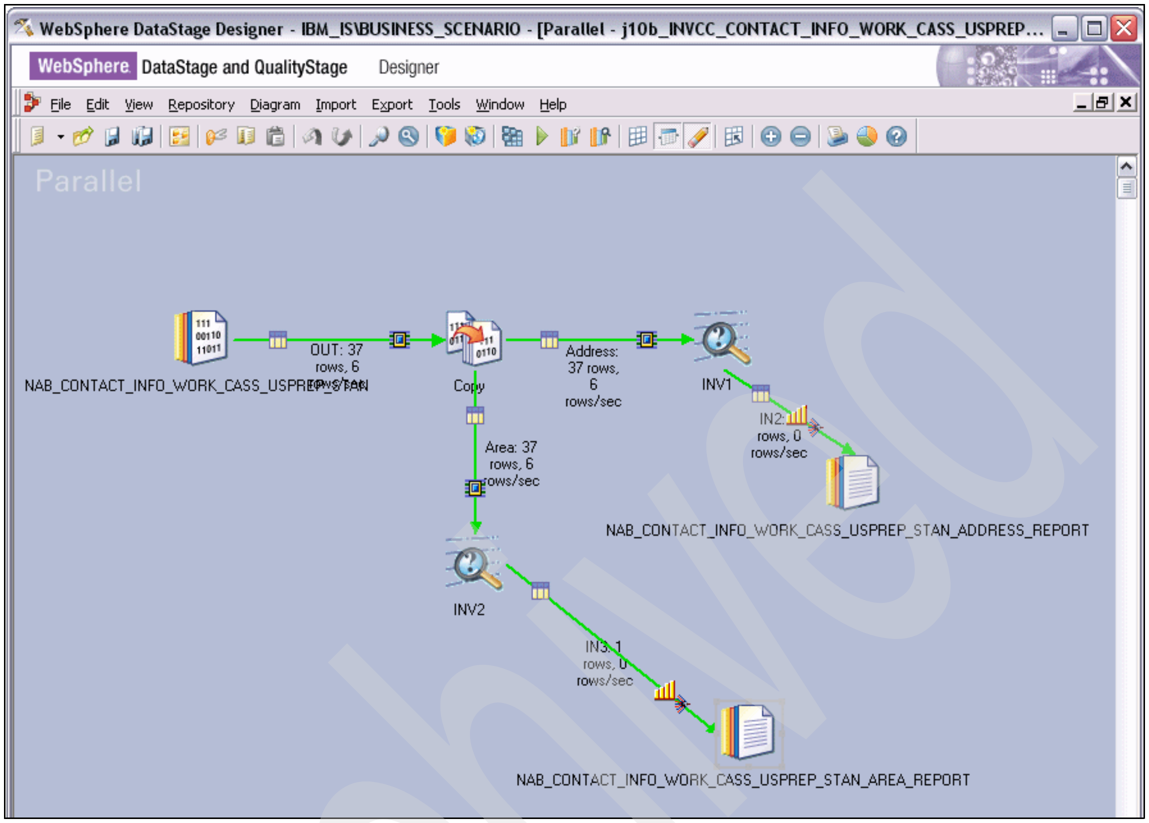
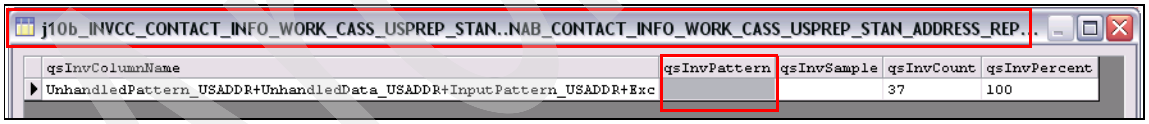
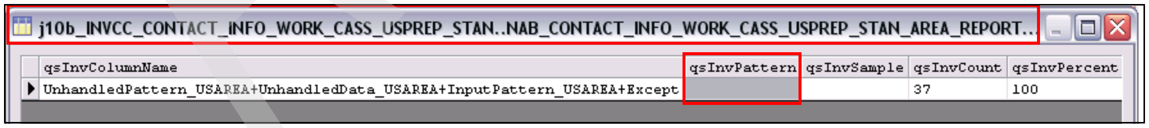

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j10b_INVCC_CONTACT_INFO_WORK_CASS_USPREP...
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NAB_CONTACT_INFO_WORK_CASS_USPREP_STAN
OUT: 37 rows, 6 rows/sec
Copy
Address: 37 rows, 6 rows/sec
INV1
IN2: 1 rows, 0 rows/sec
NAB_CONTACT_INFO_WORK_CASS_USPREP_STAN_ADDRESS_REPORT
Area: 37 rows, 6 rows/sec
INV2
IN3: 1 rows, 0 rows/sec
NAB_CONTACT_INFO_WORK_CASS_USPREP_STAN_AREA_REPORT

Figure 2-165 Create j10b_INVCC_CONTACT_INFO_WORK_CASS_USPREP_STAN 4/6


j10b_INVCC_CONTACT_INFO_WORK_CASS_USPREP_STAN..NAB_CONTACT_INFO_WORK_CASS_USPREP_STAN_ADDRESS_REP...

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
UnhandledPattern_USADDR+UnhandledData_USADDR+InputPattern_USADDR+Exc			37	100

Figure 2-166 Create j10b_INVCC_CONTACT_INFO_WORK_CASS_USPREP_STAN 5/6


j10b_INVCC_CONTACT_INFO_WORK_CASS_USPREP_STAN..NAB_CONTACT_INFO_WORK_CASS_USPREP_STAN_AREA_REPORT...

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
UnhandledPattern_USAREA+UnhandledData_USAREA+InputPattern_USAREA+Except			37	100

Figure 2-167 Create j10b_INVCC_CONTACT_INFO_WORK_CASS_USPREP_STAN 6/6

j10c_INVCC_CONTACT_INFO_HOME_US_CASS_USPREP_STAN

In this step, we investigate the unhandled patterns of the U.S. home address file created in the “j09_STAN_CASS” on page 631 step.

Because a single Investigate stage can only have a single rule set associated with it, we copy the output data set of the “j09_STAN_CASS” on page 631 job (using a Copy stage) and process it by two independent Investigate stages, with each stage using a particular domain-specific rule set. The column frequency report is generated in each Investigate stage.

Figure 2-168 on page 646 shows the various stages that are used in this job, including the data set that was created in “j09_STAN_CASS” on page 631, a Copy stage, two Investigate stages that each use a different domain-specific rule set, and a sequential file for each Investigate stage. We modified the names of the stages as shown.

Because the configuration of this job is very similar to that described in “J09_INVCC_STAN_CUSTOMER” on page 253, we do not repeat it here. The configuration of the Investigate stages are similar to that described in Figure 2-163 on page 643 and Figure 2-164 on page 643 and also not repeated here.

After saving, compiling, and running this job (Figure 2-168 on page 646), one unhandled pattern (^+DD) is identified as shown in Figure 2-169 on page 646.

An Input Pattern override was generated for the unhandled pattern as shown in Figure 2-170 on page 647. We then reran the “j09_STAN_CASS” on page 631 and “j10c_INVCC_CONTACT_INFO_HOME_US_CASS_USPREP_STAN” on page 645 jobs. Figure 2-171 on page 647 and Figure 2-172 on page 647 show no more unhandled patterns.

Proceed now to unhandled patterns in the address domain that must be managed in “j10d_INVCC_CONTACT_INFO_HOME_CA_XXPREP_STAN” on page 647.

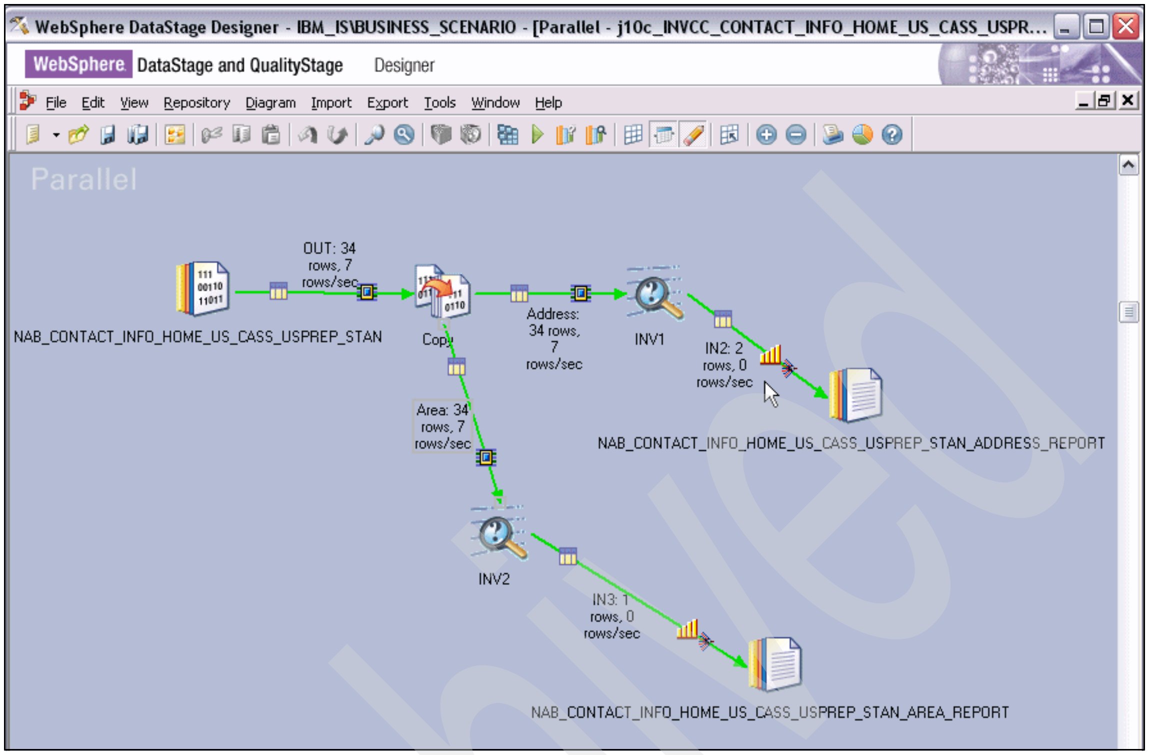
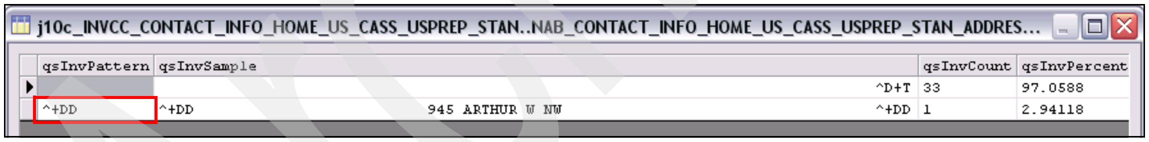
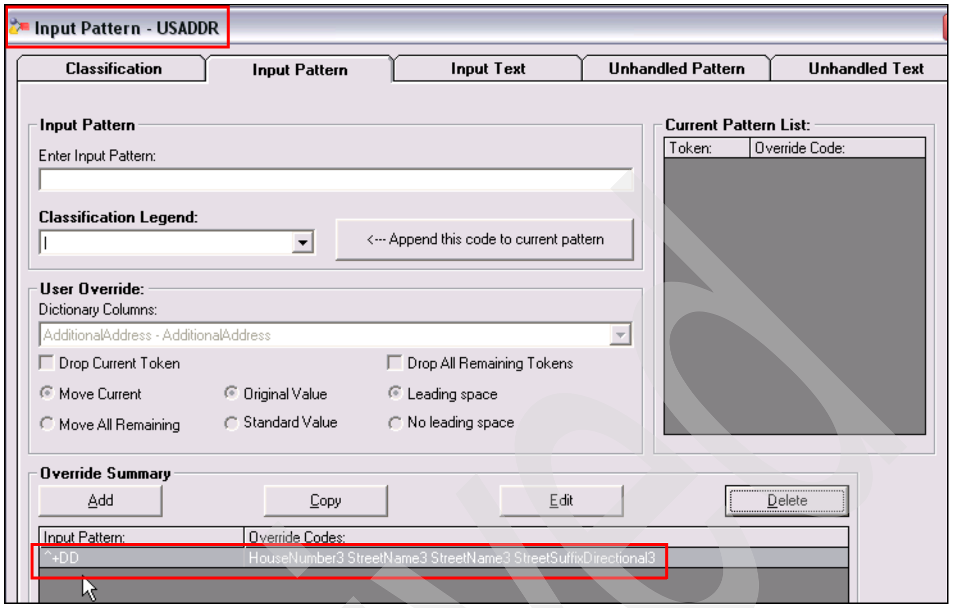

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j10c_INVCC_CONTACT_INFO_HOME_US_CASS_USPR...
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NAB_CONTACT_INFO_HOME_US_CASS_USPREP_STAN
OUT: 34 rows, 7 rows/sec
Copy
Address: 34 rows, 7 rows/sec
INV1
IN2: 2 rows, 0 rows/sec
NAB_CONTACT_INFO_HOME_US_CASS_USPREP_STAN_ADDRESS_REPORT
Area: 34 rows, 7 rows/sec
INV2
IN3: 1 rows, 0 rows/sec
NAB_CONTACT_INFO_HOME_US_CASS_USPREP_STAN_AREA_REPORT

Figure 2-168 Create j10c_INVCC_CONTACT_INFO_HOME_US_CASS_USPREP_STAN 1/5


j10c_INVCC_CONTACT_INFO_HOME_US_CASS_USPREP_STAN..NAB_CONTACT_INFO_HOME_US_CASS_USPREP_STAN_ADDRES...

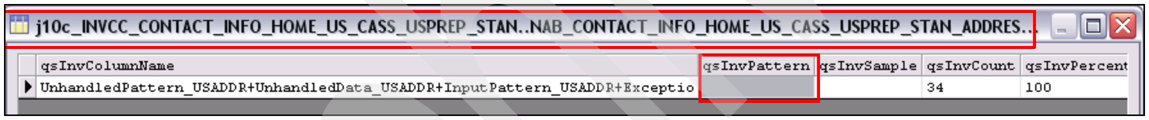
qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
	^D+T	33	97.0588
^+DD	^+DD 945 ARTHUR W NW ^+DD	1	2.94118

Figure 2-169 Create j10c_INVCC_CONTACT_INFO_HOME_US_CASS_USPREP_STAN 2/5



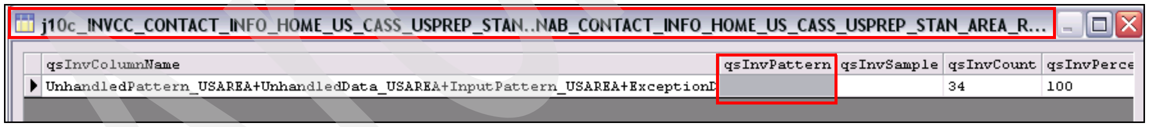
Input Pattern:	Override Codes:
^+DD	HouseNumber3 StreetName3 StreetName3 StreetSuffixDirectional3

Figure 2-170 Create j10c_INVCC_CONTACT_INFO_HOME_US_CASS_USPREP_STAN 3/5


j10c_INVCC_CONTACT_INFO_HOME_US_CASS_USPREP_STAN..NAB_CONTACT_INFO_HOME_US_CASS_USPREP_STAN_ADDRES...

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
UnhandledPattern_USADDR+UnhandledData_USADDR+InputPattern_USADDR+Exceptio			34	100

Figure 2-171 Create j10c_INVCC_CONTACT_INFO_HOME_US_CASS_USPREP_STAN 4/5


j10c_INVCC_CONTACT_INFO_HOME_US_CASS_USPREP_STAN..NAB_CONTACT_INFO_HOME_US_CASS_USPREP_STAN_AREA_R...

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPerce
UnhandledPattern_USAREA+UnhandledData_USAREA+InputPattern_USAREA+ExceptionD			34	100

Figure 2-172 Create j10c_INVCC_CONTACT_INFO_HOME_US_CASS_USPREP_STAN 5/5

j10d_INVCC_CONTACT_INFO_HOME_CA_XXPREP_STAN

In this step, we investigate the unhandled patterns of the Canadian home address file created in the “j09_STAN_CASS” on page 631 step.

Because a single Investigate stage can only have a single rule set associated with it, we copy the output data set of the “j09_STAN_CASS” on page 631 job (using a Copy stage) and process it by two independent Investigate stages, with each stage using a particular domain-specific rule set. The column frequency report is generated in each Investigate stage.

Figure 2-173 on page 649 shows the various stages that are used in this job, including the data set that was created in “j09_STAN_CASS” on page 631, a Copy stage, two Investigate stages that each use a different domain-specific rule set, and a sequential file for each Investigate stage. We modified the names of the stages as shown.

Because the configuration of this job is very similar to that described in “J09_INVCC_STAN_CUSTOMER” on page 253, we do not repeat it here. However, some of the configurations of interest are as follows:

- Figure 2-174 on page 649 shows the INV1 - Investigate Stage window with the CAADDR rule set, and the columns selected for Character Concatenate Investigate as follows:
  - UnhandledPattern_CAADDR with *C* mask
  - UnhandledData_CAADDR with *X* mask
  - InputPattern_CAADDR with *X* mask
  - AddressDomain_CAPREP with *X* mask
- Figure 2-175 on page 650 shows the INV2 - Investigate Stage window with the CAAREA rule set, and the columns selected for Character Concatenate Investigate as follows:
  - UnhandledPattern_CAADDR with *C* mask
  - UnhandledData_CAADDR with *X* mask
  - InputPattern_CAADDR with *X* mask
  - AreaDomain_CAPREP with *X* mask
- After saving, compiling, and running this job (Figure 2-176 on page 650), one unhandled pattern (^D^) is identified as shown in Figure 2-177 on page 651.

  An Input Pattern override is generated for the unhandled pattern as shown in Figure 2-179 on page 651. We then rerun the “j09_STAN_CASS” on page 631 and “j10d_INVCC_CONTACT_INFO_HOME_CA_XXPREP_STAN” on page 647 jobs. Figure 2-180 on page 651 shows no more unhandled patterns.

Proceed now to unhandled patterns in the name and address domains in “j10e_INVCC_DRIVER_CASS_USPREP_STAN” on page 652.

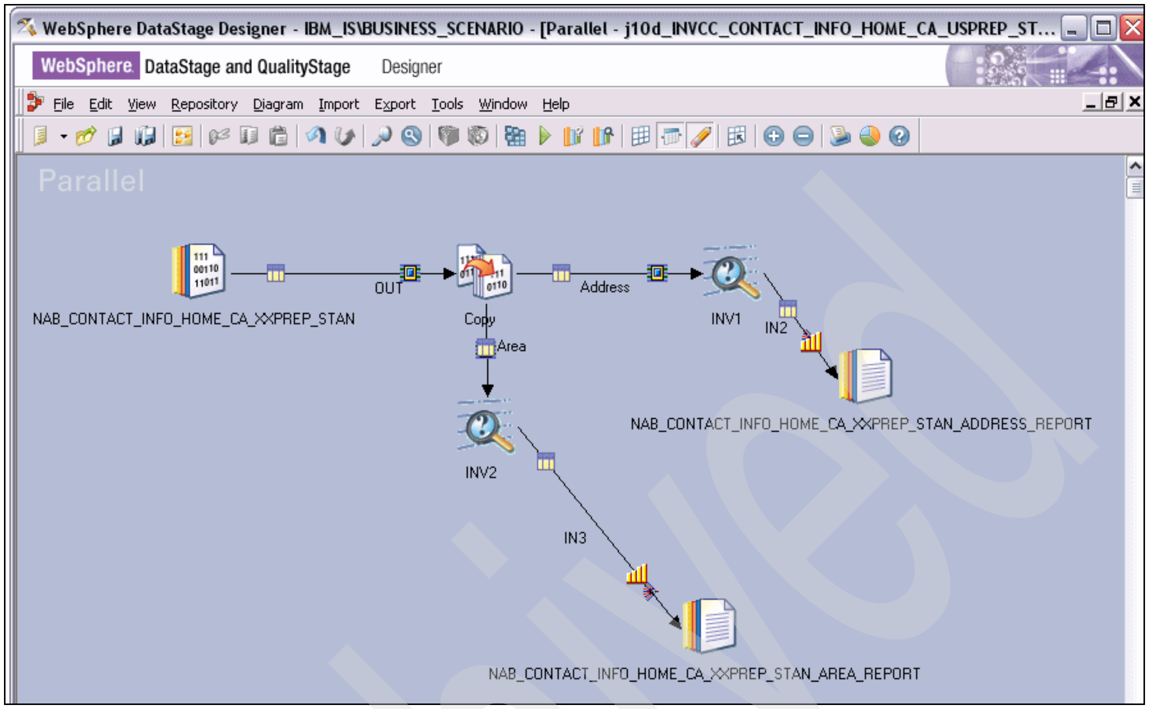
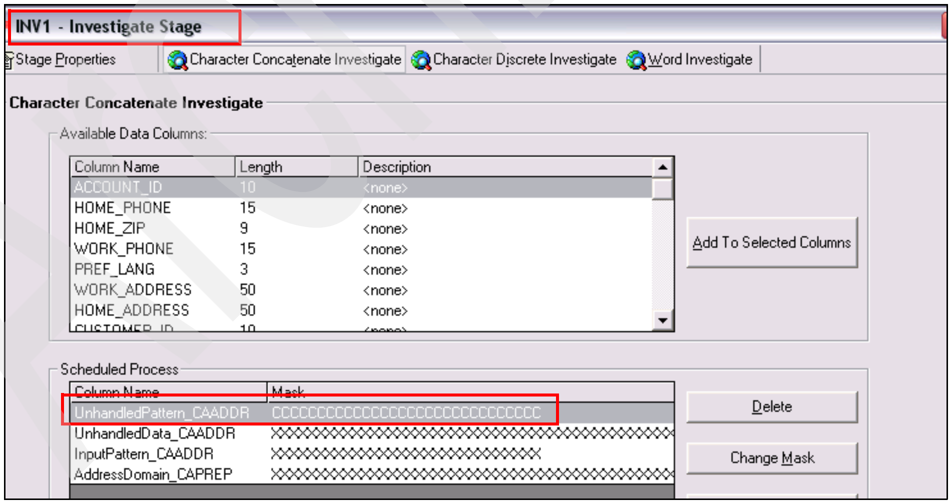

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j10d_INVCC_CONTACT_INFO_HOME_CA_USPREP_ST...
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NAB_CONTACT_INFO_HOME_CA_XXPREP_STAN
OUT
Copy
Address
INV1
IN2
Area
NAB_CONTACT_INFO_HOME_CA_XXPREP_STAN_ADDRESS_REPORT
INV2
IN3
NAB_CONTACT_INFO_HOME_CA_XXPREP_STAN_AREA_REPORT

Figure 2-173 Create j10d_INVCC_CONTACT_INFO_HOME_CA_XXPREP_STAN 1/8



Character Concatenate Investigate

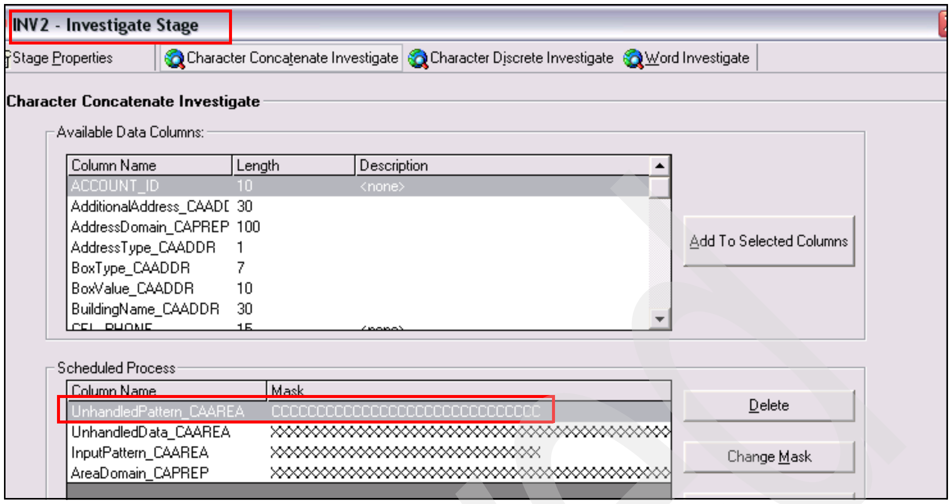
Available Data Columns:

Column Name	Length	Description
ACCOUNT_ID	10	<none>
HOME_PHONE	15	<none>
HOME_ZIP	9	<none>
WORK_PHONE	15	<none>
PREF_LANG	3	<none>
WORK_ADDRESS	50	<none>
HOME_ADDRESS	50	<none>
CUSTOMER_ID	10	<none>

Scheduled Process

Column Name	Mask
UnhandledPattern_CAADDR	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
UnhandledData_CAADDR	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
InputPattern_CAADDR	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
AddressDomain_CAPREP	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Figure 2-174 Create j10d_INVCC_CONTACT_INFO_HOME_CA_XXPREP_STAN 2/8


INV2 - Investigate Stage

Stage Properties | Character Concatenate Investigate | Character Discrete Investigate | Word Investigate

Character Concatenate Investigate

Available Data Columns:

Column Name	Length	Description
ACCOUNT_ID	10	<none>
AdditionalAddress_CAADI	30	
AddressDomain_CAPREP	100	
AddressType_CAADDR	1	
BoxType_CAADDR	7	
BoxValue_CAADDR	10	
BuildingName_CAADDR	30	
CEL_PHONE	15	<none>

Add To Selected Columns

Scheduled Process

Column Name	Mask
UnhandledPattern_CAAREA	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
UnhandledData_CAAREA	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
InputPattern_CAAREA	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
AreaDomain_CAPREP	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Delete

Change Mask

Figure 2-175 Create j10d_INVCC_CONTACT_INFO_HOME_CA_XXPREP_STAN 3/8

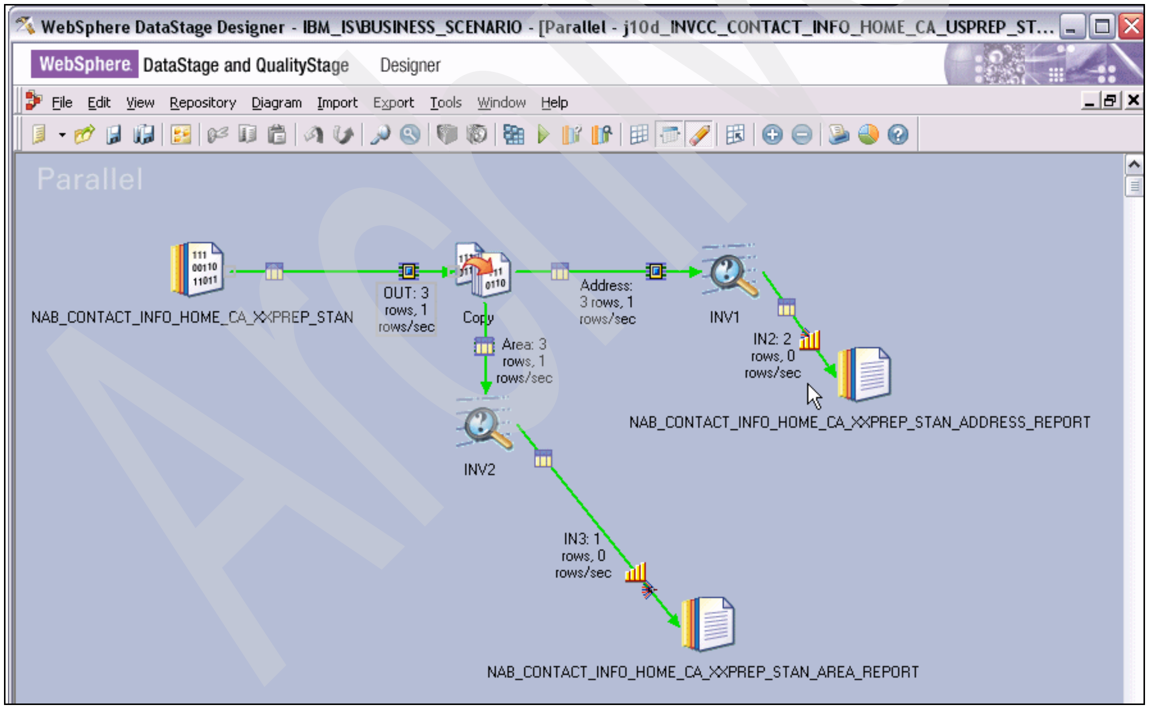
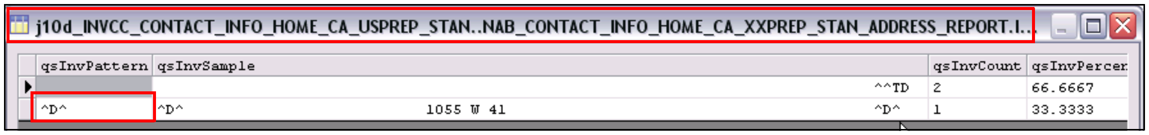

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j10d_INVCC_CONTACT_INFO_HOME_CA_USPREP_ST...
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NAB_CONTACT_INFO_HOME_CA_XXPREP_STAN
OUT: 3 rows, 1 rows/sec
Copy
Address: 3 rows, 1 rows/sec
INV1
IN2: 2 rows, 0 rows/sec
NAB_CONTACT_INFO_HOME_CA_XXPREP_STAN_ADDRESS_REPORT
Area: 3 rows, 1 rows/sec
INV2
IN3: 1 rows, 0 rows/sec
NAB_CONTACT_INFO_HOME_CA_XXPREP_STAN_AREA_REPORT

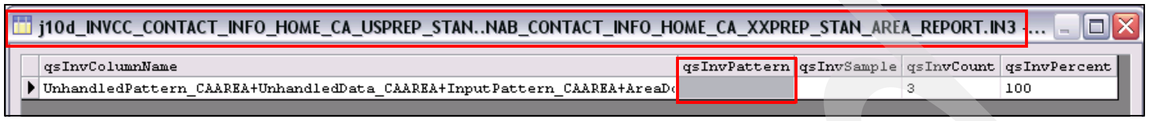
Figure 2-176 Create j10d_INVCC_CONTACT_INFO_HOME_CA_XXPREP_STAN 4/8



j10d_INVCC_CONTACT_INFO_HOME_CA_USPREP_STAN..NAB_CONTACT_INFO_HOME_CA_XXPREP_STAN_ADDRESS_REPORT.I...

qsInvPattern	qsInvSample	qsInvCount	qsInvPercen
	^^TD	2	66.6667
^D^	^D^ 1055 W 41 ^D^	1	33.3333

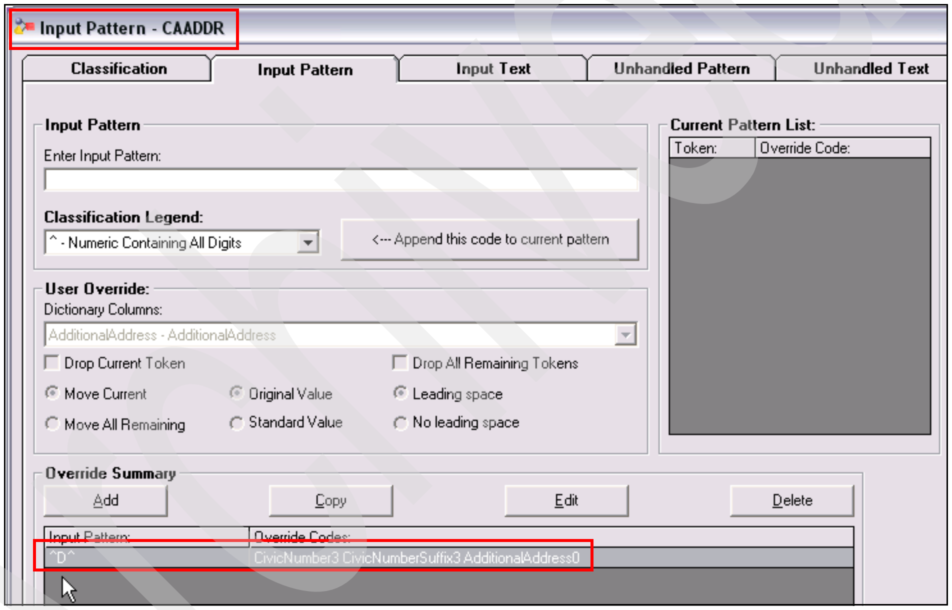
Figure 2-177 Create j10d_INVCC_CONTACT_INFO_HOME_CA_XXPREP_STAN 5/8



j10d_INVCC_CONTACT_INFO_HOME_CA_USPREP_STAN..NAB_CONTACT_INFO_HOME_CA_XXPREP_STAN_AREA_REPORT.IN3 ...

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
UnhandledPattern_CAAREA+UnhandledData_CAAREA+InputPattern_CAAREA+AreaD(			3	100

Figure 2-178 Create j10d_INVCC_CONTACT_INFO_HOME_CA_XXPREP_STAN 6/8



Input Pattern - CAADDR

Input Pattern

Enter Input Pattern:

Classification Legend: ^ - Numeric Containing All Digits

Current Pattern List:

Token:	Override Code:

User Override:

Dictionary Columns: AdditionalAddress - AdditionalAddress

Drop Current Token [ ] Drop All Remaining Tokens [ ]

[✓ Move Current] Move All Remaining

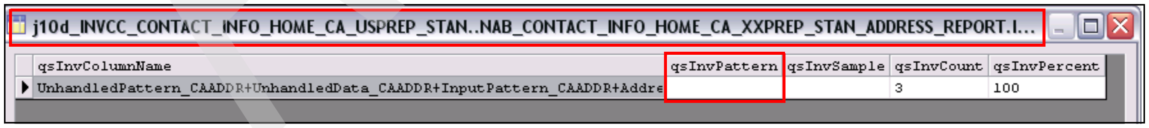
[✓ Original Value] Standard Value

[✓ Leading space] No leading space

Override Summary

Input Pattern:	Override Codes:
^D^	CivicNumber3 CivicNumberSuffix3 AdditionalAddress0

Figure 2-179 Create j10d_INVCC_CONTACT_INFO_HOME_CA_XXPREP_STAN 7/8



j10d_INVCC_CONTACT_INFO_HOME_CA_USPREP_STAN..NAB_CONTACT_INFO_HOME_CA_XXPREP_STAN_ADDRESS_REPORT.I...

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
UnhandledPattern_CAADDR+UnhandledData_CAADDR+InputPattern_CAADDR+Addre			3	100

Figure 2-180 Create j10d_INVCC_CONTACT_INFO_HOME_CA_XXPREP_STAN 8/8

j10e_INVCC_DRIVER_CASS_USPREP_STAN

In this step, we investigate the unhandled patterns of the name and U.S. addresses file that was created in the "j09_STAN_CASS" on page 631 step.

Because a single Investigate stage can only have a single rule set associated with it, we copy the output data set of "j09_STAN_CASS" on page 631 (using a Copy stage) and process the file by three independent Investigate stages, with each stage using a particular domain-specific rule set. The column frequency report is generated in each Investigate stage.

Figure 2-181 on page 653 shows the various stages that are used in this job, including the data set that was created in "j09_STAN_CASS" on page 631, a Copy stage, three Investigate stages that each use a different domain-specific rule set, and a sequential file for each Investigate stage. We modified the names of the stages as shown.

Because the configuration of this job is very similar to that described in "J09_INVCC_STAN_CUSTOMER" on page 253, we do not repeat it here. However, some of the configurations of interest are as follows:

- Figure 2-182 on page 654 shows the INV1 - Investigate Stage window with the USNAME rule set and the columns selected for Character Concatenate Investigate as follows:
  - UnhandledPattern_USNAME with *C* mask
  - UnhandledData_USNAME with *X* mask
  - InputPattern_USNAME with *X* mask
  - NameDomain_USPREP with *X* mask
- Figure 2-183 on page 654 shows the INV2 - Investigate Stage window with the USAREA rule set and the columns selected for Character Concatenate Investigate as follows:
  - UnhandledPattern_USAREA with *C* mask
  - UnhandledData_USAREA with *X* mask
  - InputPattern_USAREA with *X* mask
  - AreaDomain_USPREP with *X* mask
- Figure 2-184 on page 655 shows the INV3 - Investigate Stage window with the USADDR rule set and the columns selected for Character Concatenate Investigate as follows:
  - UnhandledPattern_USADDR with *C* mask
  - UnhandledData_USADDR with *X* mask
  - InputPattern_USADDR with *X* mask
  - AddressDomain_USPREP with *X* mask

- After saving, compiling, and running this job (Figure 2-185 on page 655), three unhandled patterns are identified as shown in Figure 2-186 on page 656.

  Input Classification and Input Pattern overrides are generated for the unhandled patterns as shown in Figure 2-187 on page 656 and Figure 2-188 on page 657. We then rerun the "j09_STAN_CASS" on page 631 and "j10e_INVCC_DRIVER_CASS_USPREP_STAN" on page 652 jobs. Figure 2-189 on page 657 shows no more unhandled patterns.

Proceed now to "j11_JOIN_NAB_NAME_AND_ADDR_DATA" on page 657.

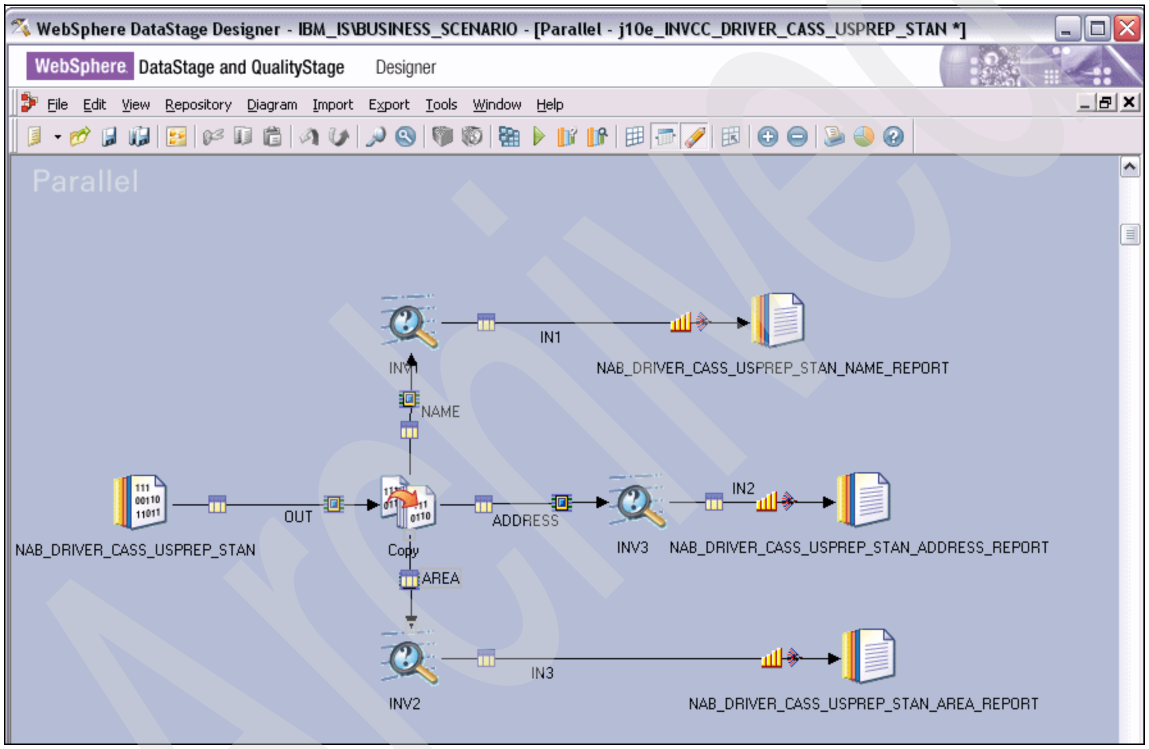
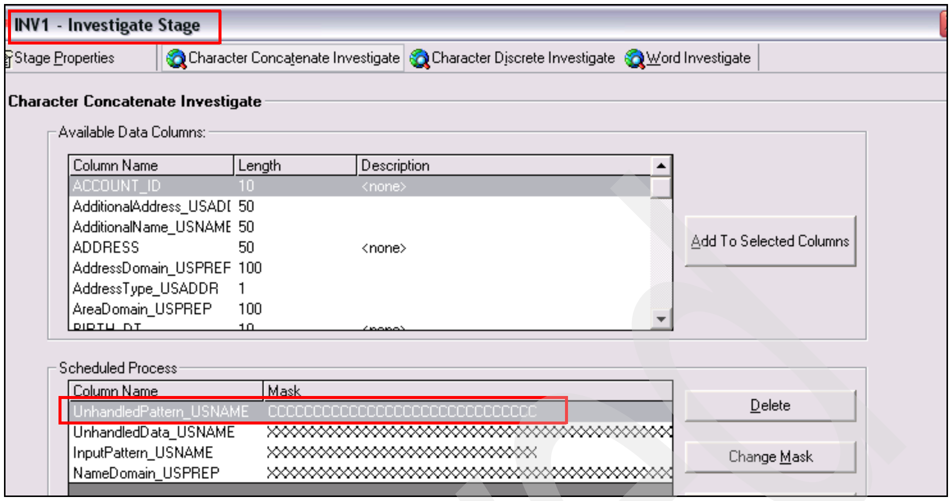

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j10e_INVCC_DRIVER_CASS_USPREP_STAN *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
IN1
NAB_DRIVER_CASS_USPREP_STAN_NAME_REPORT
INV1
NAME
NAB_DRIVER_CASS_USPREP_STAN
OUT
Copy
ADDRESS
INV3
IN2
NAB_DRIVER_CASS_USPREP_STAN_ADDRESS_REPORT
AREA
INV2
IN3
NAB_DRIVER_CASS_USPREP_STAN_AREA_REPORT

Figure 2-181 Create j10e_INVCC_DRIVER_CASS_USPREP_STAN 1/9



INV1 - Investigate Stage

Stage Properties | Character Concatenate Investigate | Character Discrete Investigate | Word Investigate

Character Concatenate Investigate

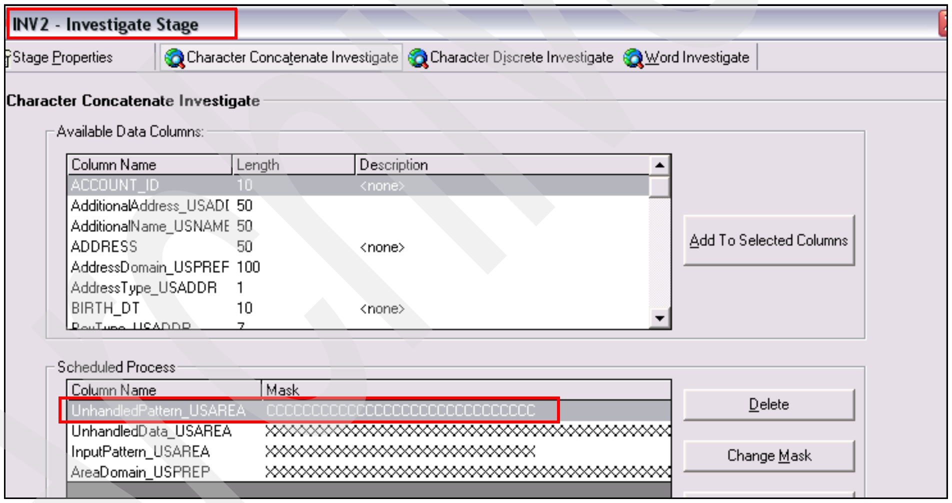
Available Data Columns:

Column Name	Length	Description
ACCOUNT_ID	10	<none>
AdditionalAddress_USADI	50	
AdditionalName_USNAME	50	
ADDRESS	50	<none>
AddressDomain_USPREF	100	
AddressType_USADDR	1	
AreaDomain_USPREP	100	

Scheduled Process

Column Name	Mask
UnhandledPattern_USNAME	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
UnhandledData_USNAME	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
InputPattern_USNAME	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
NameDomain_USPREP	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Figure 2-182 Create j10e_INVCC_DRIVER_CASS_USPREP_STAN 2/9



INV2 - Investigate Stage

Stage Properties | Character Concatenate Investigate | Character Discrete Investigate | Word Investigate

Character Concatenate Investigate

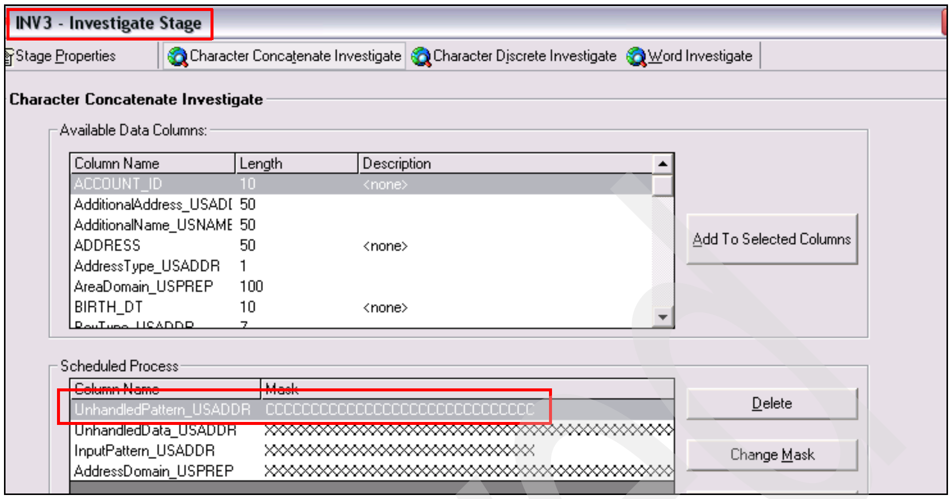
Available Data Columns:

Column Name	Length	Description
ACCOUNT_ID	10	<none>
AdditionalAddress_USADI	50	
AdditionalName_USNAME	50	
ADDRESS	50	<none>
AddressDomain_USPREF	100	
AddressType_USADDR	1	
BIRTH_DT	10	<none>

Scheduled Process

Column Name	Mask
UnhandledPattern_USAREA	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
UnhandledData_USAREA	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
InputPattern_USAREA	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
AreaDomain_USPREP	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Figure 2-183 Create j10e_INVCC_DRIVER_CASS_USPREP_STAN 3/9


INV3 - Investigate Stage

Stage Properties | Character Concatenate Investigate | Character Discrete Investigate | Word Investigate

Character Concatenate Investigate

Available Data Columns:

Column Name	Length	Description
ACCOUNT_ID	10	<none>
AdditionalAddress_USADI	50	
AdditionalName_USNAME	50	
ADDRESS	50	<none>
AddressType_USADDR	1	
AreaDomain_USPREP	100	
BIRTH_DT	10	<none>
BoxType_USADDR	7	

Add To Selected Columns

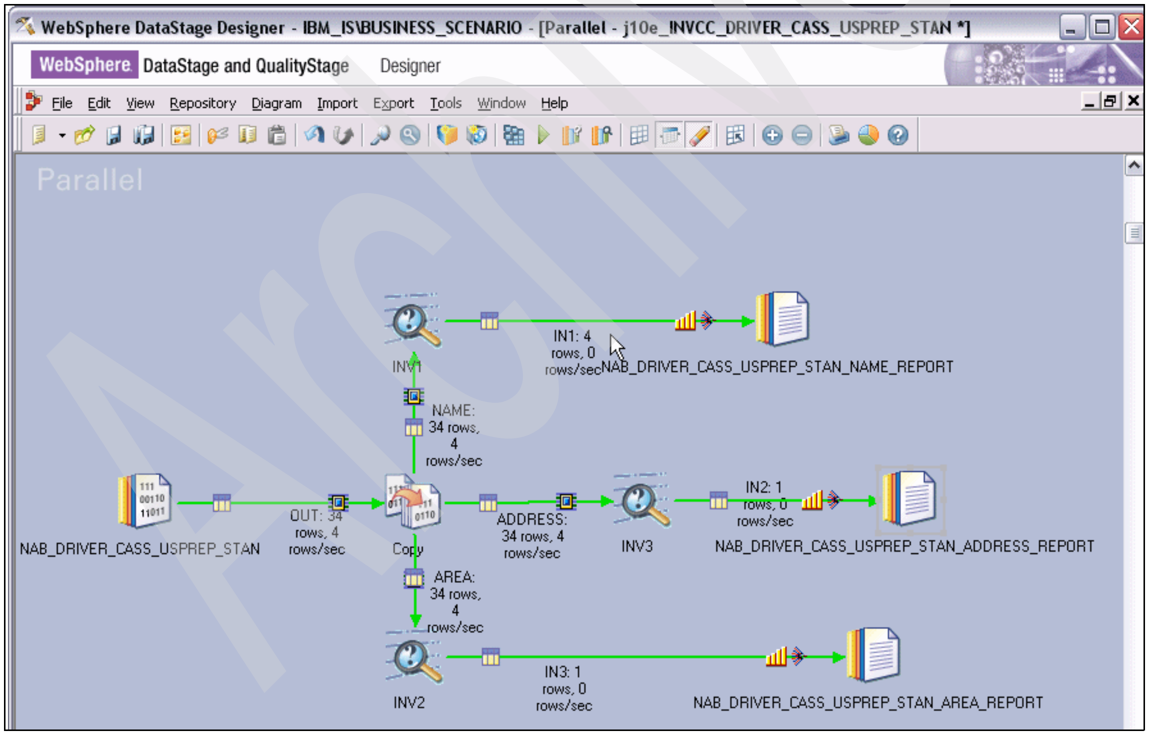
Scheduled Process

Column Name	Mask
UnhandledPattern_USADDR	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
UnhandledData_USADDR	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
InputPattern_USADDR	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
AddressDomain_USPREP	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Delete

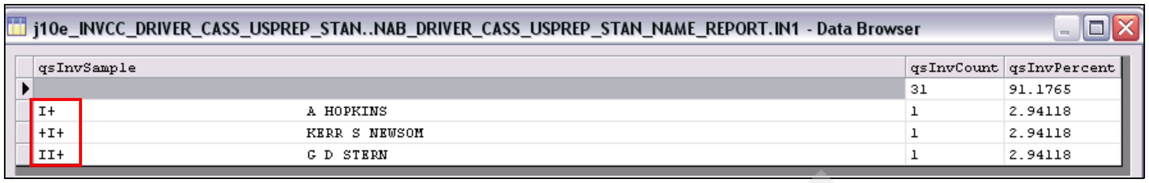
Change Mask

Figure 2-184 Create j10e_INVCC_DRIVER_CASS_USPREP_STAN 4/9


WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j10e_INVCC_DRIVER_CASS_USPREP_STAN *]

WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
INV1
IN1: 4 rows, 0 rows/sec
NAB_DRIVER_CASS_USPREP_STAN_NAME_REPORT
NAME: 34 rows, 4 rows/sec
NAB_DRIVER_CASS_USPREP_STAN
OUT: 34 rows, 4 rows/sec
Copy
ADDRESS: 34 rows, 4 rows/sec
INV3
IN2: 1 rows, 0 rows/sec
NAB_DRIVER_CASS_USPREP_STAN_ADDRESS_REPORT
AREA: 34 rows, 4 rows/sec
INV2
IN3: 1 rows, 0 rows/sec
NAB_DRIVER_CASS_USPREP_STAN_AREA_REPORT

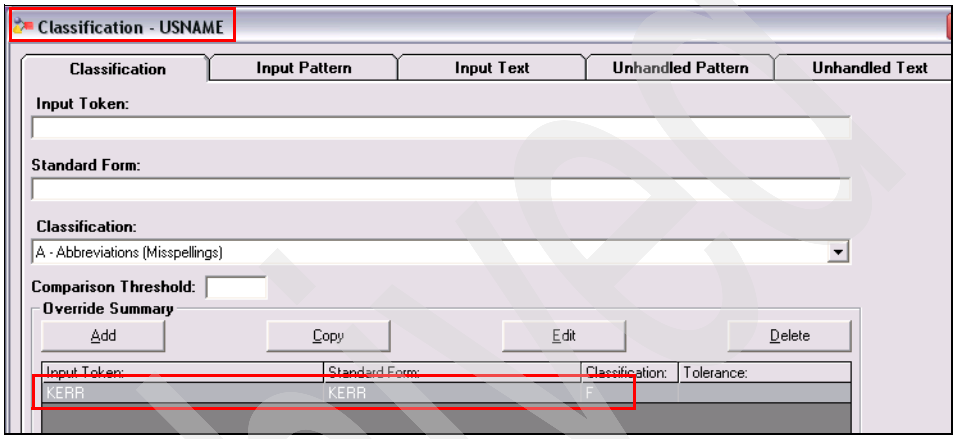
Figure 2-185 Create j10e_INVCC_DRIVER_CASS_USPREP_STAN 5/9



j10e_INVCC_DRIVER_CASS_USPREP_STAN..NAB_DRIVER_CASS_USPREP_STAN_NAME_REPORT.IN1 - Data Browser

qsInvSample		qsInvCount	qsInvPercent
		31	91.1765
I+	A HOPKINS	1	2.94118
+I+	KERR S NEWSOM	1	2.94118
II+	G D STERN	1	2.94118

Figure 2-186 Create j10e_INVCC_DRIVER_CASS_USPREP_STAN 6/9



Classification - USNAME

Input Token:

Standard Form:

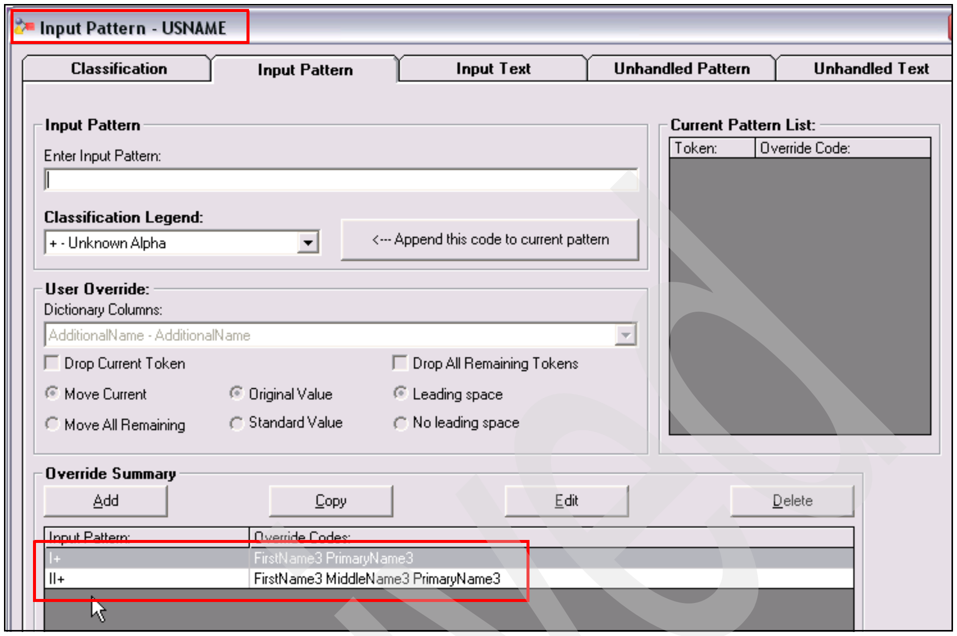
Classification: A - Abbreviations (Misspellings)

Comparison Threshold:

Override Summary

Input Token:	Standard Form:	Classification:	Tolerance:
KERR	KERR	F	

Figure 2-187 Create j10e_INVCC_DRIVER_CASS_USPREP_STAN 7/9


Input Pattern - USNAME

Input Pattern

Enter Input Pattern:

Classification Legend: + - Unknown Alpha

User Override:

Dictionary Columns: AdditionalName - AdditionalName

☐ Drop Current Token ☐ Drop All Remaining Tokens

[✓ Move Current] Move All Remaining

[✓ Original Value] Standard Value

[✓ Leading space] No leading space

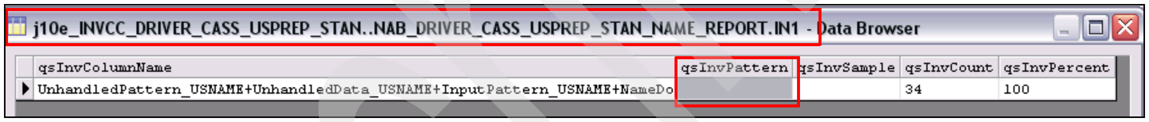
Current Pattern List:

Token:	Override Code:

Override Summary

Input Pattern:	Override Codes:
I+	FirstName3 PrimaryName3
II+	FirstName3 MiddleName3 PrimaryName3

Figure 2-188 Create j10e_INVCC_DRIVER_CASS_USPREP_STAN 8/9


j10e_INVCC_DRIVER_CASS_USPREP_STAN..NAB_DRIVER_CASS_USPREP_STAN_NAME_REPORT.IN1 - Data Browser

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
UnhandledPattern_USNAME+UnhandledData_USNAME+InputPattern_USNAME+NameDo			34	100

Figure 2-189 Create j10e_INVCC_DRIVER_CASS_USPREP_STAN 9/9

j11_JOIN_NAB_NAME_AND_ADDR_DATA

After the unhandled patterns are handled, we reconstruct two files as follows:

- Standardized name (CUSTOMER file) and standardized U.S. home address information (CONTACT_INFO) from the standardized output of the “j09_STAN_CASS” on page 631 step by joining them on the CUSTOMER_ID.
- Standardized name (CUSTOMER file) and standardized Canadian home address information (CONTACT_INFO) from the standardized output of the “j09_STAN_CASS” on page 631 step by joining them on the CUSTOMER_ID.

These reconstructed files are required to survive the “best” data from the North American Bank’s core and non-core systems into the CRM system.

We also rename certain columns in the standardized work address file to avoid column name conflicts in the upcoming “j12_JOIN_NAB_WORK_AND_HOME” on page 674 job.

Figure 2-190 on page 659 shows the various stages that are used in this job. It includes the data sets that were created in “j09_STAN_CASS” on page 631, a Join stage, and an output Data Set stage for each Join stage. It also includes a Transformer stage to rename columns in the U.S. work address CONTACT_INFO file created in the “j09_STAN_CASS” on page 631 step. A rename of columns was required for this file so that there is no column name conflict when it is joined with another standardized file in the “j12_JOIN_NAB_WORK_AND_HOME” on page 674 step. We modified the names of the stages as shown.

Figure 2-191 on page 660 shows the Join1 - Join window, which indicates an inner join of the standardized name CUSTOMER file with and standardized U.S. home addresses customer CONTACT_INFO file on the CUSTOMER_ID column. With an inner join, records from input data sets whose key columns contain equal values to the output data set are transferred to the output. Records whose key columns do not contain equal values are dropped. The resulting columns from the mapping of the two input sources are shown in Figure 2-192 on page 661 through Figure 2-194 on page 663.

Figure 2-195 on page 664 through Figure 2-198 on page 667 corresponds the inner join of the standardized name CUSTOMER file with and standardized Canadian home addresses customer CONTACT_INFO file on the CUSTOMER_ID column.

Figure 2-199 on page 668 through Figure 2-201 on page 670 show the RENAME_COLUMNS - Transformer Stage window, which renames some of the Standardize stage generated columns (such as HouseNumber_USADDR to HouseNumber_USADDR_WORK) to avoid name conflicts when this file is joined to the standardized customer name and U.S. home addresses customer CONTACT_INFO file. Figure 2-202 on page 671 through Figure 2-204 on page 673 show the metadata definitions of the output Data Set with the modified column names.

Figure 2-205 on page 674 shows the results after compiling and running this job.

Proceed now to “j12_JOIN_NAB_WORK_AND_HOME” on page 674.

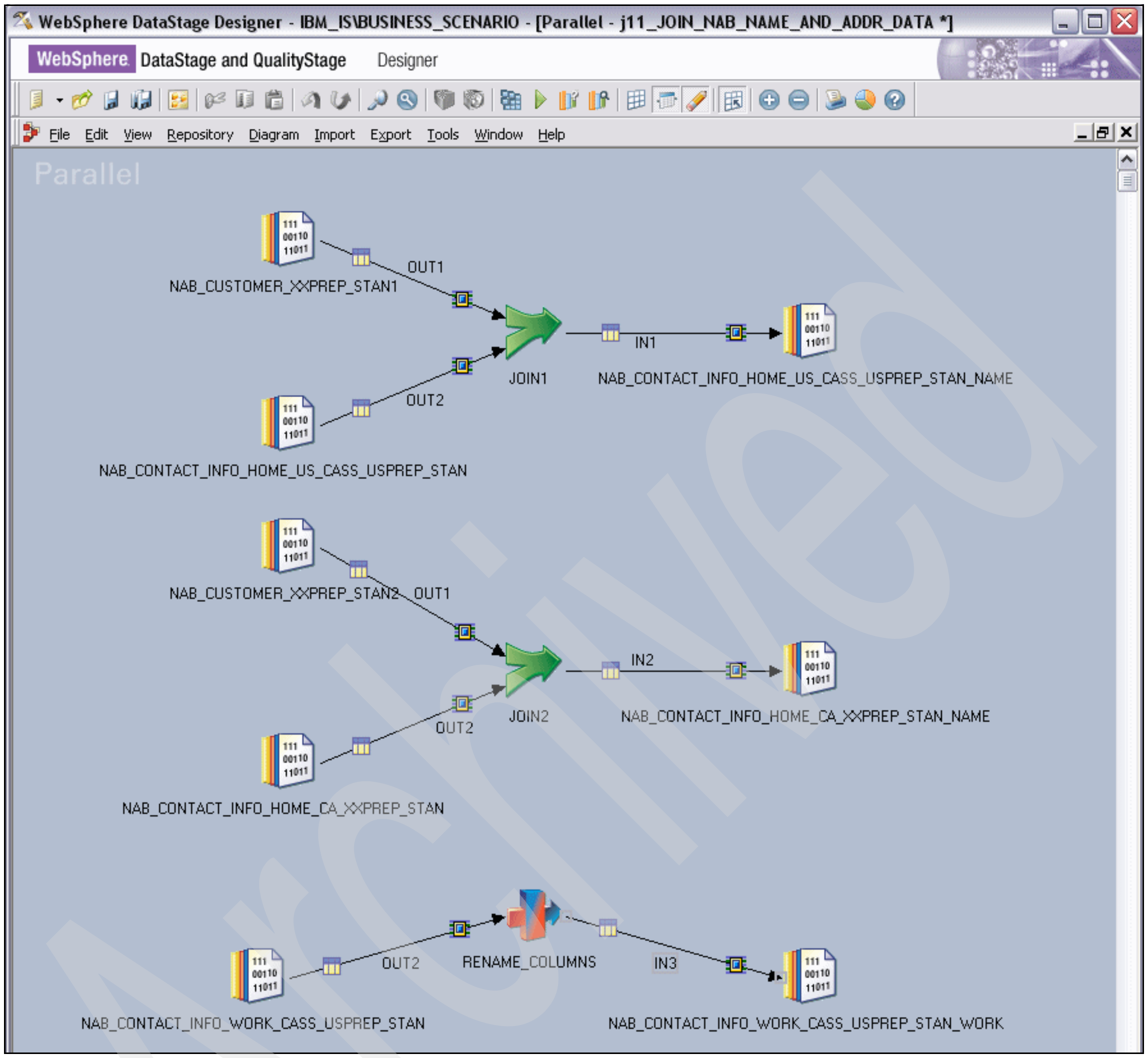
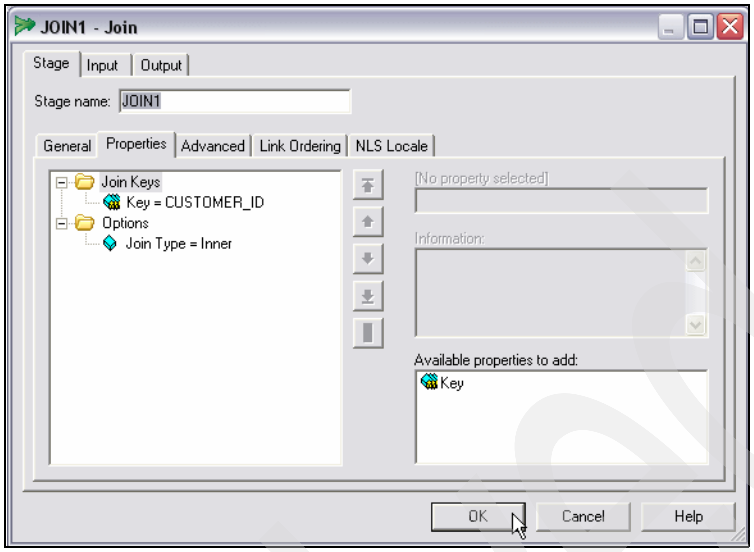

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j11_JOIN_NAB_NAME_AND_ADDR_DATA *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NAB_CUSTOMER_XXPREP_STAN1
OUT1
JOIN1
IN1
NAB_CONTACT_INFO_HOME_US_CASS_USPREP_STAN_NAME
OUT2
NAB_CONTACT_INFO_HOME_US_CASS_USPREP_STAN
NAB_CUSTOMER_XXPREP_STAN2
OUT1
JOIN2
IN2
NAB_CONTACT_INFO_HOME_CA_XXPREP_STAN_NAME
OUT2
NAB_CONTACT_INFO_HOME_CA_XXPREP_STAN
RENAME_COLUMNS
OUT2
IN3
NAB_CONTACT_INFO_WORK_CASS_USPREP_STAN
NAB_CONTACT_INFO_WORK_CASS_USPREP_STAN_WORK

Figure 2-190 Create j11_JOIN_NAB_NAME_AND_ADDR_DATA 1/16



JOIN1 - Join

Stage name: JOIN1

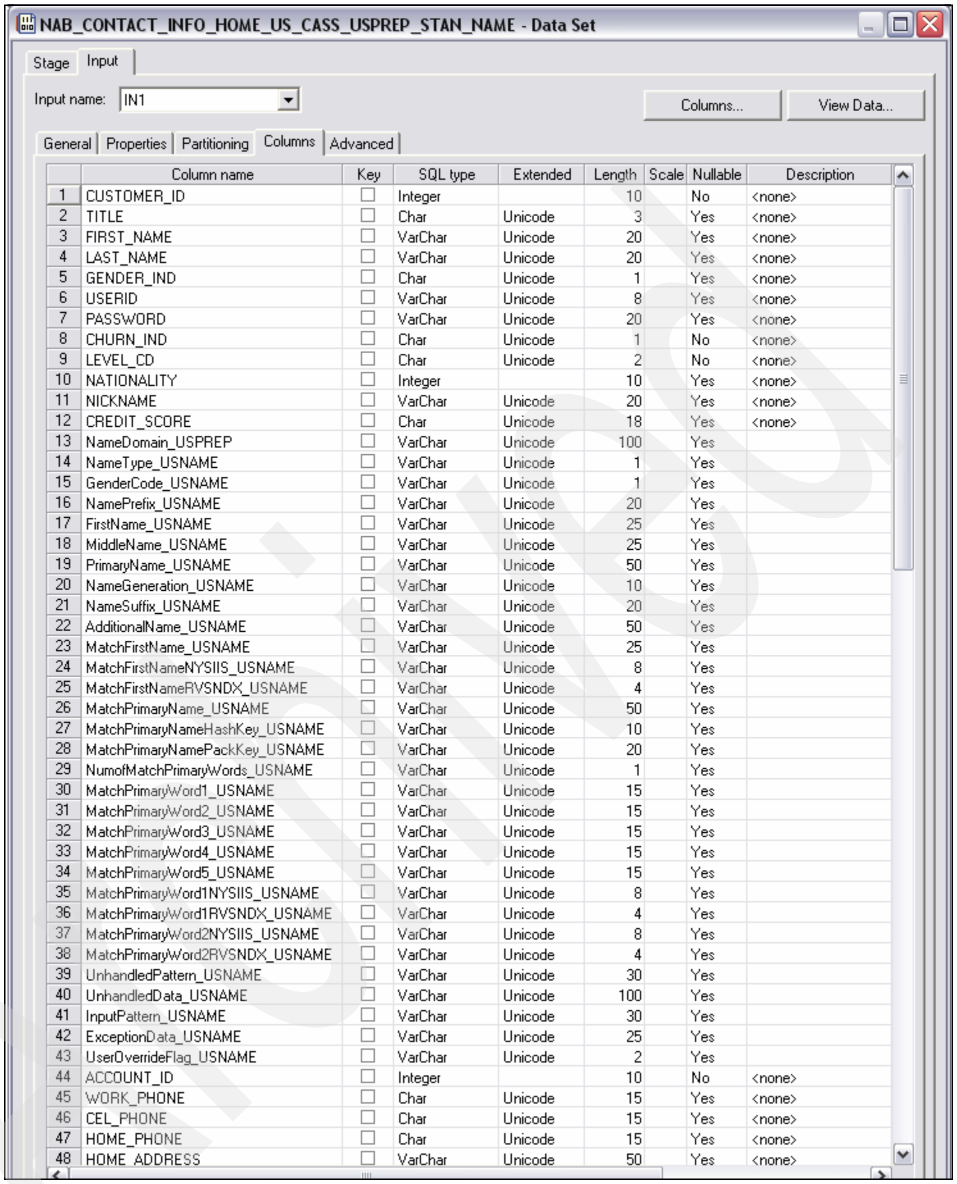
Join Keys
Key = CUSTOMER_ID
Options
Join Type = Inner

[No property selected]

Information:

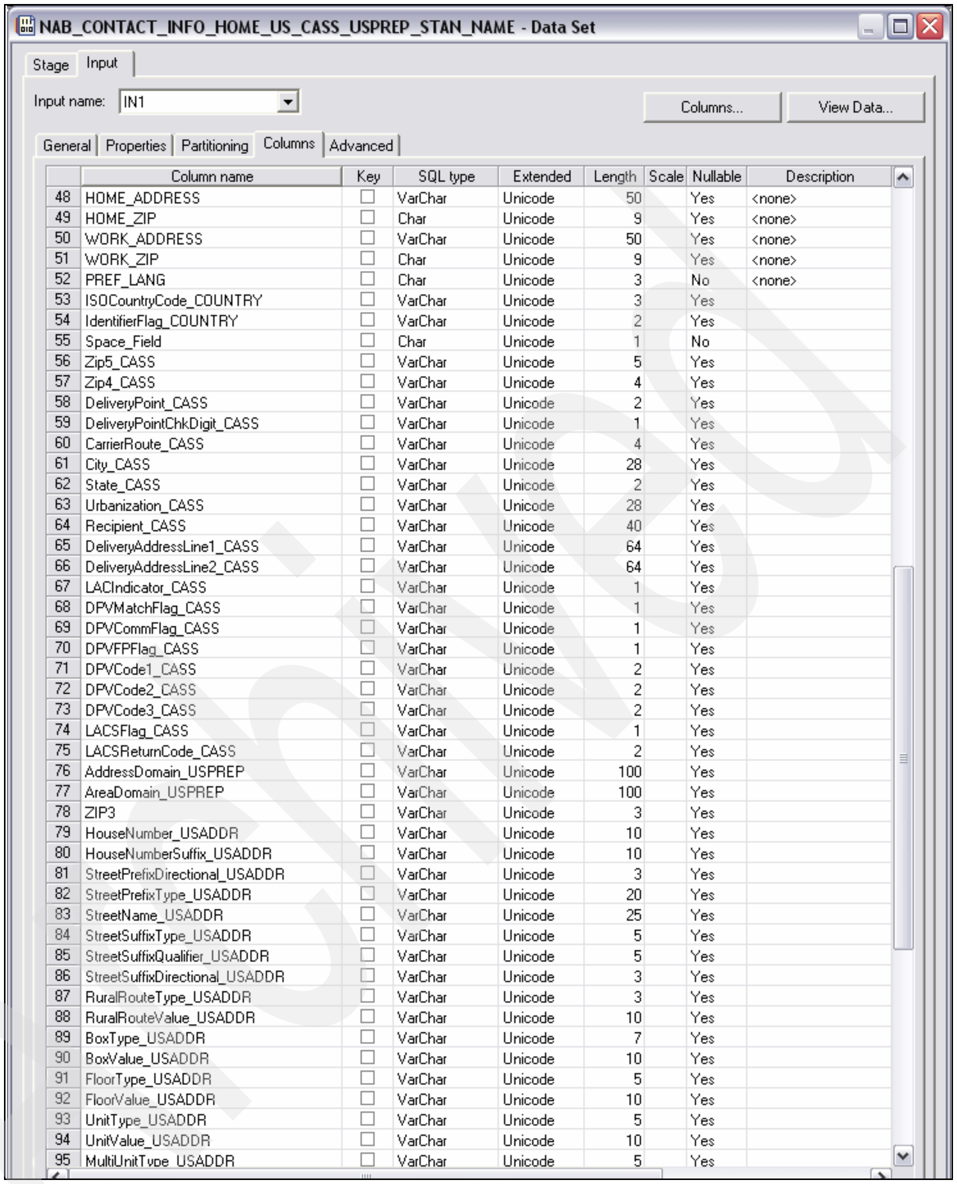
Available properties to add:
Key

Figure 2-191 Create j11_JOIN_NAB_NAME_AND_ADDR_DATA 2/16



	Column name	Key	SQL type	Extended	Length	Scale	Nullable	Description
1	CUSTOMER_ID	☐	Integer		10		No	<none>
2	TITLE	☐	Char	Unicode	3		Yes	<none>
3	FIRST_NAME	☐	VarChar	Unicode	20		Yes	<none>
4	LAST_NAME	☐	VarChar	Unicode	20		Yes	<none>
5	GENDER_IND	☐	Char	Unicode	1		Yes	<none>
6	USERID	☐	VarChar	Unicode	8		Yes	<none>
7	PASSWORD	☐	VarChar	Unicode	20		Yes	<none>
8	CHURN_IND	☐	Char	Unicode	1		No	<none>
9	LEVEL_CD	☐	Char	Unicode	2		No	<none>
10	NATIONALITY	☐	Integer		10		Yes	<none>
11	NICKNAME	☐	VarChar	Unicode	20		Yes	<none>
12	CREDIT_SCORE	☐	Char	Unicode	18		Yes	<none>
13	NameDomain_USPREP	☐	VarChar	Unicode	100		Yes	
14	NameType_USNAME	☐	VarChar	Unicode	1		Yes	
15	GenderCode_USNAME	☐	VarChar	Unicode	1		Yes	
16	NamePrefix_USNAME	☐	VarChar	Unicode	20		Yes	
17	FirstName_USNAME	☐	VarChar	Unicode	25		Yes	
18	MiddleName_USNAME	☐	VarChar	Unicode	25		Yes	
19	PrimaryName_USNAME	☐	VarChar	Unicode	50		Yes	
20	NameGeneration_USNAME	☐	VarChar	Unicode	10		Yes	
21	NameSuffix_USNAME	☐	VarChar	Unicode	20		Yes	
22	AdditionalName_USNAME	☐	VarChar	Unicode	50		Yes	
23	MatchFirstName_USNAME	☐	VarChar	Unicode	25		Yes	
24	MatchFirstNameNYSIIS_USNAME	☐	VarChar	Unicode	8		Yes	
25	MatchFirstNameRVSNDX_USNAME	☐	VarChar	Unicode	4		Yes	
26	MatchPrimaryName_USNAME	☐	VarChar	Unicode	50		Yes	
27	MatchPrimaryNameHashKey_USNAME	☐	VarChar	Unicode	10		Yes	
28	MatchPrimaryNamePackKey_USNAME	☐	VarChar	Unicode	20		Yes	
29	NumofMatchPrimaryWords_USNAME	☐	VarChar	Unicode	1		Yes	
30	MatchPrimaryWord1_USNAME	☐	VarChar	Unicode	15		Yes	
31	MatchPrimaryWord2_USNAME	☐	VarChar	Unicode	15		Yes	
32	MatchPrimaryWord3_USNAME	☐	VarChar	Unicode	15		Yes	
33	MatchPrimaryWord4_USNAME	☐	VarChar	Unicode	15		Yes	
34	MatchPrimaryWord5_USNAME	☐	VarChar	Unicode	15		Yes	
35	MatchPrimaryWord1NYSIIS_USNAME	☐	VarChar	Unicode	8		Yes	
36	MatchPrimaryWord1RVSNDX_USNAME	☐	VarChar	Unicode	4		Yes	
37	MatchPrimaryWord2NYSIIS_USNAME	☐	VarChar	Unicode	8		Yes	
38	MatchPrimaryWord2RVSNDX_USNAME	☐	VarChar	Unicode	4		Yes	
39	UnhandledPattern_USNAME	☐	VarChar	Unicode	30		Yes	
40	UnhandledData_USNAME	☐	VarChar	Unicode	100		Yes	
41	InputPattern_USNAME	☐	VarChar	Unicode	30		Yes	
42	ExceptionData_USNAME	☐	VarChar	Unicode	25		Yes	
43	UserOverrideFlag_USNAME	☐	VarChar	Unicode	2		Yes	
44	ACCOUNT_ID	☐	Integer		10		No	<none>
45	WORK_PHONE	☐	Char	Unicode	15		Yes	<none>
46	CEL_PHONE	☐	Char	Unicode	15		Yes	<none>
47	HOME_PHONE	☐	Char	Unicode	15		Yes	<none>
48	HOME_ADDRESS	☐	VarChar	Unicode	50		Yes	<none>

Figure 2-192 Create j11_JOIN_NAB_NAME_AND_ADDR_DATA 3/16



NAB_CONTACT_INFO_HOME_US_CASS_USPREP_STAN_NAME - Data Set

Stage | Input

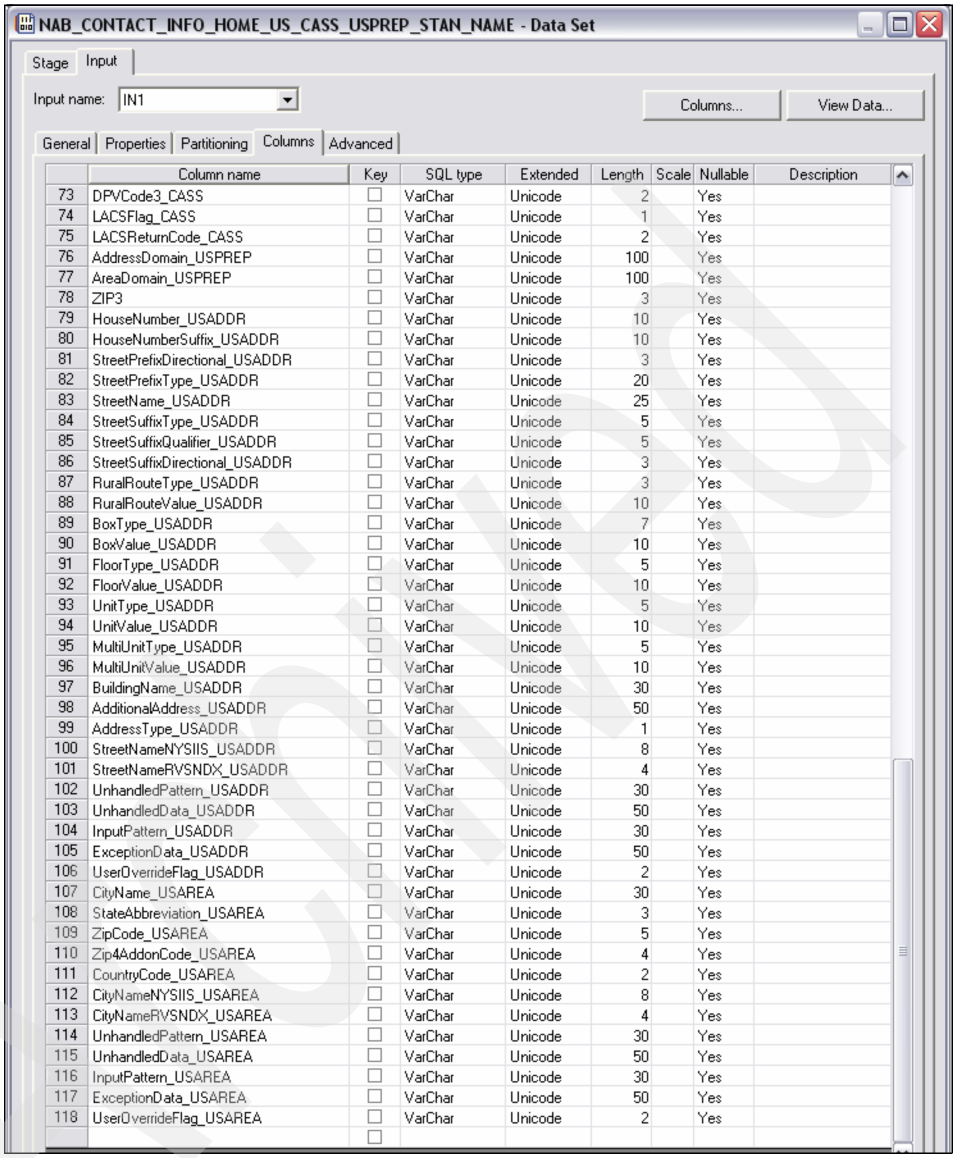
Input name: IN1

Columns... | View Data...

General | Properties | Partitioning | Columns | Advanced

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	Description
48	HOME_ADDRESS	☐	VarChar	Unicode	50		Yes	<none>
49	HOME_ZIP	☐	Char	Unicode	9		Yes	<none>
50	WORK_ADDRESS	☐	VarChar	Unicode	50		Yes	<none>
51	WORK_ZIP	☐	Char	Unicode	9		Yes	<none>
52	PREF_LANG	☐	Char	Unicode	3		No	<none>
53	ISOCountryCode_COUNTRY	☐	VarChar	Unicode	3		Yes	
54	IdentifierFlag_COUNTRY	☐	VarChar	Unicode	2		Yes	
55	Space_Field	☐	Char	Unicode	1		No	
56	Zip5_CASS	☐	VarChar	Unicode	5		Yes	
57	Zip4_CASS	☐	VarChar	Unicode	4		Yes	
58	DeliveryPoint_CASS	☐	VarChar	Unicode	2		Yes	
59	DeliveryPointChkDigit_CASS	☐	VarChar	Unicode	1		Yes	
60	CarrierRoute_CASS	☐	VarChar	Unicode	4		Yes	
61	City_CASS	☐	VarChar	Unicode	28		Yes	
62	State_CASS	☐	VarChar	Unicode	2		Yes	
63	Urbanization_CASS	☐	VarChar	Unicode	28		Yes	
64	Recipient_CASS	☐	VarChar	Unicode	40		Yes	
65	DeliveryAddressLine1_CASS	☐	VarChar	Unicode	64		Yes	
66	DeliveryAddressLine2_CASS	☐	VarChar	Unicode	64		Yes	
67	LACIndicator_CASS	☐	VarChar	Unicode	1		Yes	
68	DPVMatchFlag_CASS	☐	VarChar	Unicode	1		Yes	
69	DPVCommFlag_CASS	☐	VarChar	Unicode	1		Yes	
70	DPVFPFlag_CASS	☐	VarChar	Unicode	1		Yes	
71	DPVCode1_CASS	☐	VarChar	Unicode	2		Yes	
72	DPVCode2_CASS	☐	VarChar	Unicode	2		Yes	
73	DPVCode3_CASS	☐	VarChar	Unicode	2		Yes	
74	LACSFlag_CASS	☐	VarChar	Unicode	1		Yes	
75	LACSReturnCode_CASS	☐	VarChar	Unicode	2		Yes	
76	AddressDomain_USPREP	☐	VarChar	Unicode	100		Yes	
77	AreaDomain_USPREP	☐	VarChar	Unicode	100		Yes	
78	ZIP3	☐	VarChar	Unicode	3		Yes	
79	HouseNumber_USADDR	☐	VarChar	Unicode	10		Yes	
80	HouseNumberSuffix_USADDR	☐	VarChar	Unicode	10		Yes	
81	StreetPrefixDirectional_USADDR	☐	VarChar	Unicode	3		Yes	
82	StreetPrefixType_USADDR	☐	VarChar	Unicode	20		Yes	
83	StreetName_USADDR	☐	VarChar	Unicode	25		Yes	
84	StreetSuffixType_USADDR	☐	VarChar	Unicode	5		Yes	
85	StreetSuffixQualifier_USADDR	☐	VarChar	Unicode	5		Yes	
86	StreetSuffixDirectional_USADDR	☐	VarChar	Unicode	3		Yes	
87	RuralRouteType_USADDR	☐	VarChar	Unicode	3		Yes	
88	RuralRouteValue_USADDR	☐	VarChar	Unicode	10		Yes	
89	BoxType_USADDR	☐	VarChar	Unicode	7		Yes	
90	BoxValue_USADDR	☐	VarChar	Unicode	10		Yes	
91	FloorType_USADDR	☐	VarChar	Unicode	5		Yes	
92	FloorValue_USADDR	☐	VarChar	Unicode	10		Yes	
93	UnitType_USADDR	☐	VarChar	Unicode	5		Yes	
94	UnitValue_USADDR	☐	VarChar	Unicode	10		Yes	
95	MultiUnitType_USADDR	☐	VarChar	Unicode	5		Yes	

Figure 2-193 Create j11_JOIN_NAB_NAME_AND_ADDR_DATA 4/16



NAB_CONTACT_INFO_HOME_US_CASS_USPREP_STAN_NAME - Data Set

Stage | Input

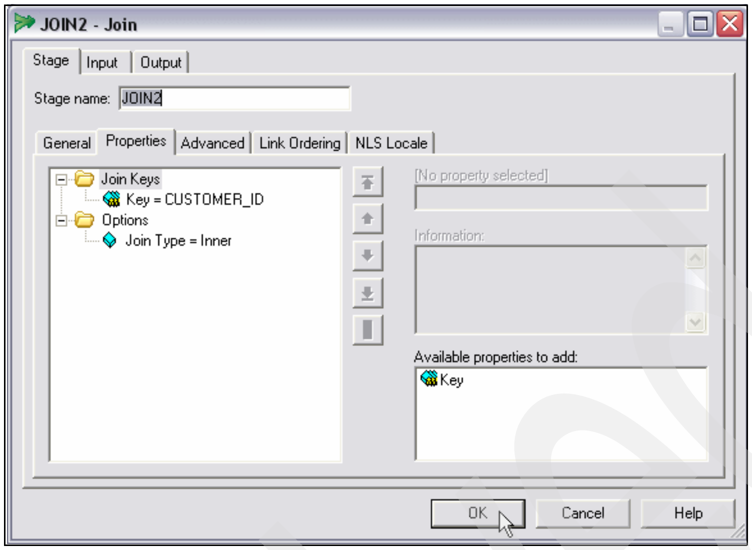
Input name: IN1

Columns... | View Data...

General | Properties | Partitioning | Columns | Advanced

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	Description
73	DPVCode3_CASS	☐	VarChar	Unicode	2		Yes	
74	LACSFlag_CASS	☐	VarChar	Unicode	1		Yes	
75	LACSReturnCode_CASS	☐	VarChar	Unicode	2		Yes	
76	AddressDomain_USPREP	☐	VarChar	Unicode	100		Yes	
77	AreaDomain_USPREP	☐	VarChar	Unicode	100		Yes	
78	ZIP3	☐	VarChar	Unicode	3		Yes	
79	HouseNumber_USADDR	☐	VarChar	Unicode	10		Yes	
80	HouseNumberSuffix_USADDR	☐	VarChar	Unicode	10		Yes	
81	StreetPrefixDirectional_USADDR	☐	VarChar	Unicode	3		Yes	
82	StreetPrefixType_USADDR	☐	VarChar	Unicode	20		Yes	
83	StreetName_USADDR	☐	VarChar	Unicode	25		Yes	
84	StreetSuffixType_USADDR	☐	VarChar	Unicode	5		Yes	
85	StreetSuffixQualifier_USADDR	☐	VarChar	Unicode	5		Yes	
86	StreetSuffixDirectional_USADDR	☐	VarChar	Unicode	3		Yes	
87	RuralRouteType_USADDR	☐	VarChar	Unicode	3		Yes	
88	RuralRouteValue_USADDR	☐	VarChar	Unicode	10		Yes	
89	BoxType_USADDR	☐	VarChar	Unicode	7		Yes	
90	BoxValue_USADDR	☐	VarChar	Unicode	10		Yes	
91	FloorType_USADDR	☐	VarChar	Unicode	5		Yes	
92	FloorValue_USADDR	☐	VarChar	Unicode	10		Yes	
93	UnitType_USADDR	☐	VarChar	Unicode	5		Yes	
94	UnitValue_USADDR	☐	VarChar	Unicode	10		Yes	
95	MultiUnitType_USADDR	☐	VarChar	Unicode	5		Yes	
96	MultiUnitValue_USADDR	☐	VarChar	Unicode	10		Yes	
97	BuildingName_USADDR	☐	VarChar	Unicode	30		Yes	
98	AdditionalAddress_USADDR	☐	VarChar	Unicode	50		Yes	
99	AddressType_USADDR	☐	VarChar	Unicode	1		Yes	
100	StreetNameNYSIIS_USADDR	☐	VarChar	Unicode	8		Yes	
101	StreetNameRVSNDX_USADDR	☐	VarChar	Unicode	4		Yes	
102	UnhandledPattern_USADDR	☐	VarChar	Unicode	30		Yes	
103	UnhandledData_USADDR	☐	VarChar	Unicode	50		Yes	
104	InputPattern_USADDR	☐	VarChar	Unicode	30		Yes	
105	ExceptionData_USADDR	☐	VarChar	Unicode	50		Yes	
106	UserOverrideFlag_USADDR	☐	VarChar	Unicode	2		Yes	
107	CityName_USAREA	☐	VarChar	Unicode	30		Yes	
108	StateAbbreviation_USAREA	☐	VarChar	Unicode	3		Yes	
109	ZipCode_USAREA	☐	VarChar	Unicode	5		Yes	
110	Zip4AddonCode_USAREA	☐	VarChar	Unicode	4		Yes	
111	CountryCode_USAREA	☐	VarChar	Unicode	2		Yes	
112	CityNameNYSIIS_USAREA	☐	VarChar	Unicode	8		Yes	
113	CityNameRVSNDX_USAREA	☐	VarChar	Unicode	4		Yes	
114	UnhandledPattern_USAREA	☐	VarChar	Unicode	30		Yes	
115	UnhandledData_USAREA	☐	VarChar	Unicode	50		Yes	
116	InputPattern_USAREA	☐	VarChar	Unicode	30		Yes	
117	ExceptionData_USAREA	☐	VarChar	Unicode	50		Yes	
118	UserOverrideFlag_USAREA	☐	VarChar	Unicode	2		Yes	

Figure 2-194 Create j11_JOIN_NAB_NAME_AND_ADDR_DATA 5/16

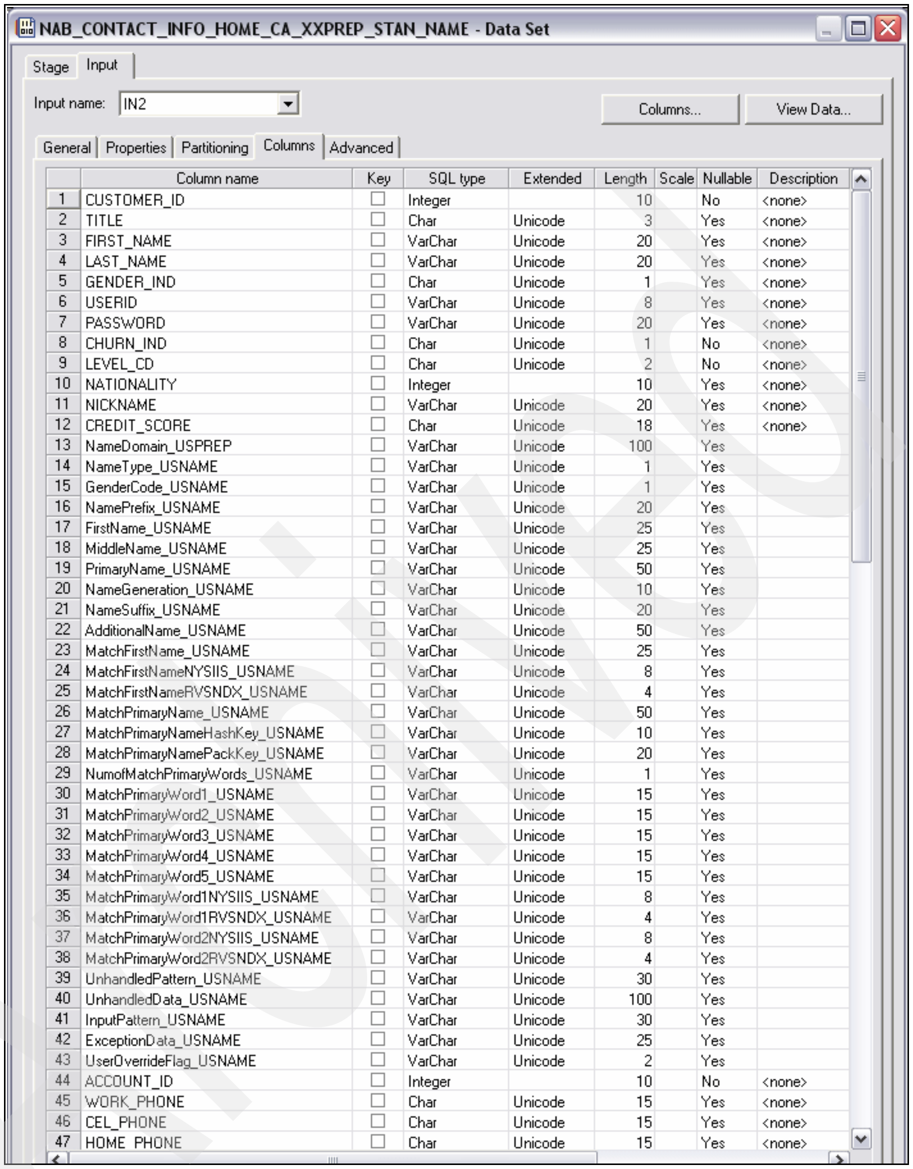


Stage name: JOIN2

Join Keys
Key = CUSTOMER_ID
Options
Join Type = Inner

Available properties to add:
Key

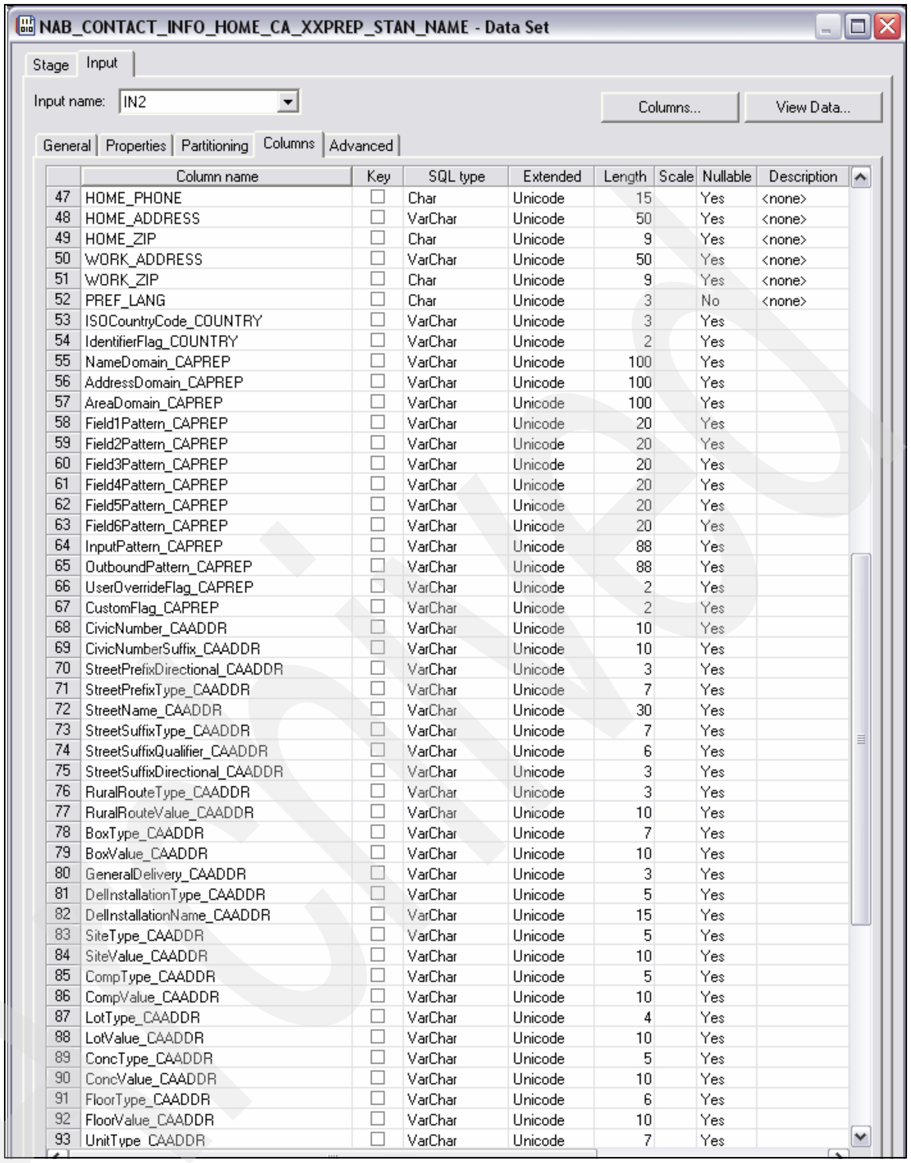
Figure 2-195 Create j11_JOIN_NAB_NAME_AND_ADDR_DATA 6/16


NAB_CONTACT_INFO_HOME_CA_XXPREP_STAN_NAME - Data Set

Input name: IN2

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	Description
1	CUSTOMER_ID	☐	Integer		10		No	<none>
2	TITLE	☐	Char	Unicode	3		Yes	<none>
3	FIRST_NAME	☐	VarChar	Unicode	20		Yes	<none>
4	LAST_NAME	☐	VarChar	Unicode	20		Yes	<none>
5	GENDER_IND	☐	Char	Unicode	1		Yes	<none>
6	USERID	☐	VarChar	Unicode	8		Yes	<none>
7	PASSWORD	☐	VarChar	Unicode	20		Yes	<none>
8	CHURN_IND	☐	Char	Unicode	1		No	<none>
9	LEVEL_CD	☐	Char	Unicode	2		No	<none>
10	NATIONALITY	☐	Integer		10		Yes	<none>
11	NICKNAME	☐	VarChar	Unicode	20		Yes	<none>
12	CREDIT_SCORE	☐	Char	Unicode	18		Yes	<none>
13	NameDomain_USPREP	☐	VarChar	Unicode	100		Yes	
14	NameType_USNAME	☐	VarChar	Unicode	1		Yes	
15	GenderCode_USNAME	☐	VarChar	Unicode	1		Yes	
16	NamePrefix_USNAME	☐	VarChar	Unicode	20		Yes	
17	FirstName_USNAME	☐	VarChar	Unicode	25		Yes	
18	MiddleName_USNAME	☐	VarChar	Unicode	25		Yes	
19	PrimaryName_USNAME	☐	VarChar	Unicode	50		Yes	
20	NameGeneration_USNAME	☐	VarChar	Unicode	10		Yes	
21	NameSuffix_USNAME	☐	VarChar	Unicode	20		Yes	
22	AdditionalName_USNAME	☐	VarChar	Unicode	50		Yes	
23	MatchFirstName_USNAME	☐	VarChar	Unicode	25		Yes	
24	MatchFirstNameNYSIIS_USNAME	☐	VarChar	Unicode	8		Yes	
25	MatchFirstNameRVSNDX_USNAME	☐	VarChar	Unicode	4		Yes	
26	MatchPrimaryName_USNAME	☐	VarChar	Unicode	50		Yes	
27	MatchPrimaryNameHashKey_USNAME	☐	VarChar	Unicode	10		Yes	
28	MatchPrimaryNamePackKey_USNAME	☐	VarChar	Unicode	20		Yes	
29	NumofMatchPrimaryWords_USNAME	☐	VarChar	Unicode	1		Yes	
30	MatchPrimaryWord1_USNAME	☐	VarChar	Unicode	15		Yes	
31	MatchPrimaryWord2_USNAME	☐	VarChar	Unicode	15		Yes	
32	MatchPrimaryWord3_USNAME	☐	VarChar	Unicode	15		Yes	
33	MatchPrimaryWord4_USNAME	☐	VarChar	Unicode	15		Yes	
34	MatchPrimaryWord5_USNAME	☐	VarChar	Unicode	15		Yes	
35	MatchPrimaryWord1NYSIIS_USNAME	☐	VarChar	Unicode	8		Yes	
36	MatchPrimaryWord1RVSNDX_USNAME	☐	VarChar	Unicode	4		Yes	
37	MatchPrimaryWord2NYSIIS_USNAME	☐	VarChar	Unicode	8		Yes	
38	MatchPrimaryWord2RVSNDX_USNAME	☐	VarChar	Unicode	4		Yes	
39	UnhandledPattern_USNAME	☐	VarChar	Unicode	30		Yes	
40	UnhandledData_USNAME	☐	VarChar	Unicode	100		Yes	
41	InputPattern_USNAME	☐	VarChar	Unicode	30		Yes	
42	ExceptionData_USNAME	☐	VarChar	Unicode	25		Yes	
43	UserOverrideFlag_USNAME	☐	VarChar	Unicode	2		Yes	
44	ACCOUNT_ID	☐	Integer		10		No	<none>
45	WORK_PHONE	☐	Char	Unicode	15		Yes	<none>
46	CEL_PHONE	☐	Char	Unicode	15		Yes	<none>
47	HOME_PHONE	☐	Char	Unicode	15		Yes	<none>

Figure 2-196 Create j11_JOIN_NAB_NAME_AND_ADDR_DATA 7/16



NAB_CONTACT_INFO_HOME_CA_XXPREP_STAN_NAME - Data Set

Stage | Input

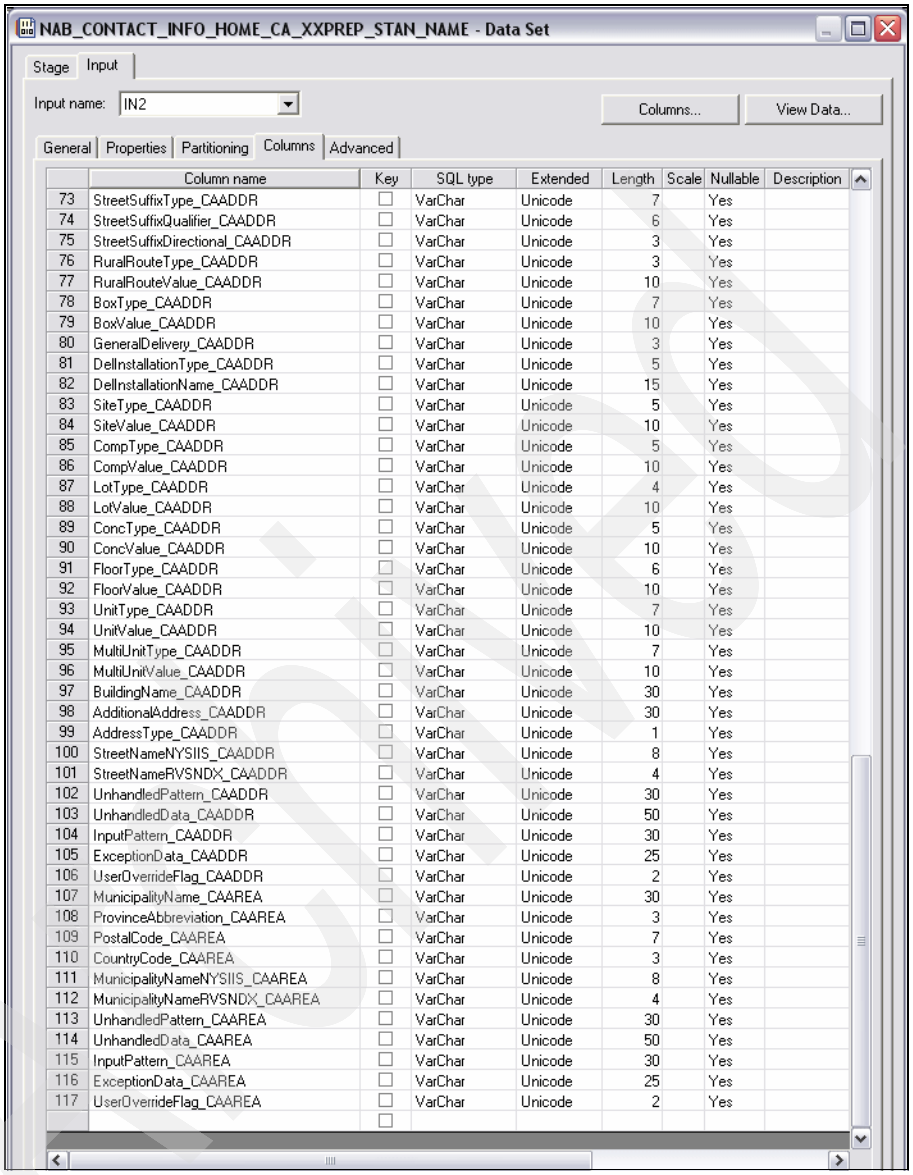
Input name: IN2

Columns... | View Data...

General | Properties | Partitioning | Columns | Advanced

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	Description
47	HOME_PHONE	☐	Char	Unicode	15		Yes	<none>
48	HOME_ADDRESS	☐	VarChar	Unicode	50		Yes	<none>
49	HOME_ZIP	☐	Char	Unicode	9		Yes	<none>
50	WORK_ADDRESS	☐	VarChar	Unicode	50		Yes	<none>
51	WORK_ZIP	☐	Char	Unicode	9		Yes	<none>
52	PREF_LANG	☐	Char	Unicode	3		No	<none>
53	ISOCountryCode_COUNTRY	☐	VarChar	Unicode	3		Yes	
54	IdentifierFlag_COUNTRY	☐	VarChar	Unicode	2		Yes	
55	NameDomain_CAPREP	☐	VarChar	Unicode	100		Yes	
56	AddressDomain_CAPREP	☐	VarChar	Unicode	100		Yes	
57	AreaDomain_CAPREP	☐	VarChar	Unicode	100		Yes	
58	Field1Pattern_CAPREP	☐	VarChar	Unicode	20		Yes	
59	Field2Pattern_CAPREP	☐	VarChar	Unicode	20		Yes	
60	Field3Pattern_CAPREP	☐	VarChar	Unicode	20		Yes	
61	Field4Pattern_CAPREP	☐	VarChar	Unicode	20		Yes	
62	Field5Pattern_CAPREP	☐	VarChar	Unicode	20		Yes	
63	Field6Pattern_CAPREP	☐	VarChar	Unicode	20		Yes	
64	InputPattern_CAPREP	☐	VarChar	Unicode	88		Yes	
65	OutboundPattern_CAPREP	☐	VarChar	Unicode	88		Yes	
66	UserOverrideFlag_CAPREP	☐	VarChar	Unicode	2		Yes	
67	CustomFlag_CAPREP	☐	VarChar	Unicode	2		Yes	
68	CivicNumber_CAADDR	☐	VarChar	Unicode	10		Yes	
69	CivicNumberSuffix_CAADDR	☐	VarChar	Unicode	10		Yes	
70	StreetPrefixDirectional_CAADDR	☐	VarChar	Unicode	3		Yes	
71	StreetPrefixType_CAADDR	☐	VarChar	Unicode	7		Yes	
72	StreetName_CAADDR	☐	VarChar	Unicode	30		Yes	
73	StreetSuffixType_CAADDR	☐	VarChar	Unicode	7		Yes	
74	StreetSuffixQualifier_CAADDR	☐	VarChar	Unicode	6		Yes	
75	StreetSuffixDirectional_CAADDR	☐	VarChar	Unicode	3		Yes	
76	RuralRouteType_CAADDR	☐	VarChar	Unicode	3		Yes	
77	RuralRouteValue_CAADDR	☐	VarChar	Unicode	10		Yes	
78	BoxType_CAADDR	☐	VarChar	Unicode	7		Yes	
79	BoxValue_CAADDR	☐	VarChar	Unicode	10		Yes	
80	GeneralDelivery_CAADDR	☐	VarChar	Unicode	3		Yes	
81	DelInstallationType_CAADDR	☐	VarChar	Unicode	5		Yes	
82	DelInstallationName_CAADDR	☐	VarChar	Unicode	15		Yes	
83	SiteType_CAADDR	☐	VarChar	Unicode	5		Yes	
84	SiteValue_CAADDR	☐	VarChar	Unicode	10		Yes	
85	CompType_CAADDR	☐	VarChar	Unicode	5		Yes	
86	CompValue_CAADDR	☐	VarChar	Unicode	10		Yes	
87	LotType_CAADDR	☐	VarChar	Unicode	4		Yes	
88	LotValue_CAADDR	☐	VarChar	Unicode	10		Yes	
89	ConcType_CAADDR	☐	VarChar	Unicode	5		Yes	
90	ConcValue_CAADDR	☐	VarChar	Unicode	10		Yes	
91	FloorType_CAADDR	☐	VarChar	Unicode	6		Yes	
92	FloorValue_CAADDR	☐	VarChar	Unicode	10		Yes	
93	UnitType_CAADDR	☐	VarChar	Unicode	7		Yes	

Figure 2-197 Create j11_JOIN_NAB_NAME_AND_ADDR_DATA 8/16

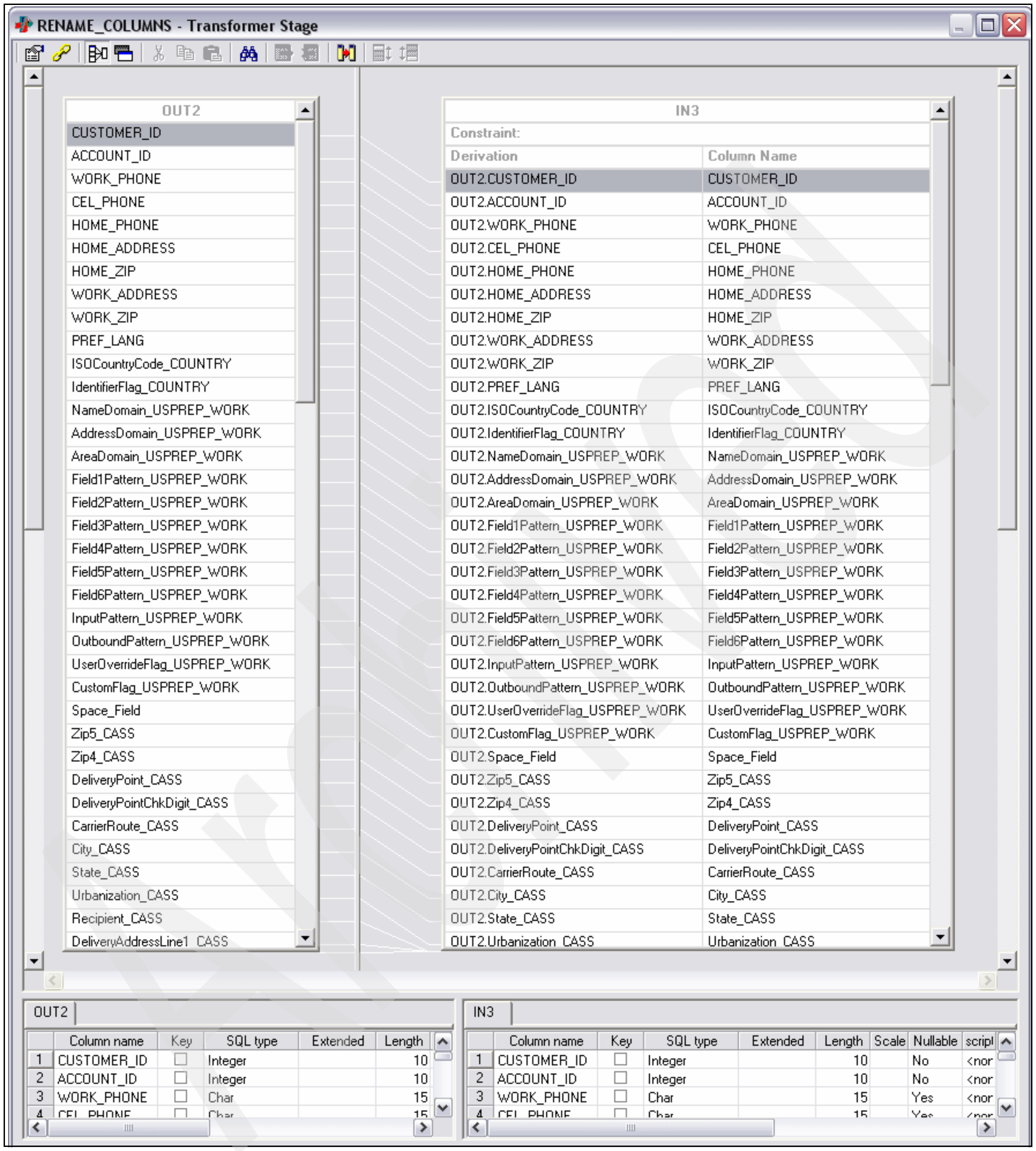


NAB_CONTACT_INFO_HOME_CA_XXPREP_STAN_NAME - Data Set

Input name: IN2

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	Description
73	StreetSuffixType_CAADDR	☐	VarChar	Unicode	7		Yes	
74	StreetSuffixQualifier_CAADDR	☐	VarChar	Unicode	6		Yes	
75	StreetSuffixDirectional_CAADDR	☐	VarChar	Unicode	3		Yes	
76	RuralRouteType_CAADDR	☐	VarChar	Unicode	3		Yes	
77	RuralRouteValue_CAADDR	☐	VarChar	Unicode	10		Yes	
78	BoxType_CAADDR	☐	VarChar	Unicode	7		Yes	
79	BoxValue_CAADDR	☐	VarChar	Unicode	10		Yes	
80	GeneralDelivery_CAADDR	☐	VarChar	Unicode	3		Yes	
81	DelInstallationType_CAADDR	☐	VarChar	Unicode	5		Yes	
82	DelInstallationName_CAADDR	☐	VarChar	Unicode	15		Yes	
83	SiteType_CAADDR	☐	VarChar	Unicode	5		Yes	
84	SiteValue_CAADDR	☐	VarChar	Unicode	10		Yes	
85	CompType_CAADDR	☐	VarChar	Unicode	5		Yes	
86	CompValue_CAADDR	☐	VarChar	Unicode	10		Yes	
87	LotType_CAADDR	☐	VarChar	Unicode	4		Yes	
88	LotValue_CAADDR	☐	VarChar	Unicode	10		Yes	
89	ConcType_CAADDR	☐	VarChar	Unicode	5		Yes	
90	ConcValue_CAADDR	☐	VarChar	Unicode	10		Yes	
91	FloorType_CAADDR	☐	VarChar	Unicode	6		Yes	
92	FloorValue_CAADDR	☐	VarChar	Unicode	10		Yes	
93	UnitType_CAADDR	☐	VarChar	Unicode	7		Yes	
94	UnitValue_CAADDR	☐	VarChar	Unicode	10		Yes	
95	MultiUnitType_CAADDR	☐	VarChar	Unicode	7		Yes	
96	MultiUnitValue_CAADDR	☐	VarChar	Unicode	10		Yes	
97	BuildingName_CAADDR	☐	VarChar	Unicode	30		Yes	
98	AdditionalAddress_CAADDR	☐	VarChar	Unicode	30		Yes	
99	AddressType_CAADDR	☐	VarChar	Unicode	1		Yes	
100	StreetNameNYSIIS_CAADDR	☐	VarChar	Unicode	8		Yes	
101	StreetNameRVSNDX_CAADDR	☐	VarChar	Unicode	4		Yes	
102	UnhandledPattern_CAADDR	☐	VarChar	Unicode	30		Yes	
103	UnhandledData_CAADDR	☐	VarChar	Unicode	50		Yes	
104	InputPattern_CAADDR	☐	VarChar	Unicode	30		Yes	
105	ExceptionData_CAADDR	☐	VarChar	Unicode	25		Yes	
106	UserOverrideFlag_CAADDR	☐	VarChar	Unicode	2		Yes	
107	MunicipalityName_CAAREA	☐	VarChar	Unicode	30		Yes	
108	ProvinceAbbreviation_CAAREA	☐	VarChar	Unicode	3		Yes	
109	PostalCode_CAAREA	☐	VarChar	Unicode	7		Yes	
110	CountryCode_CAAREA	☐	VarChar	Unicode	3		Yes	
111	MunicipalityNameNYSIIS_CAAREA	☐	VarChar	Unicode	8		Yes	
112	MunicipalityNameRVSNDX_CAAREA	☐	VarChar	Unicode	4		Yes	
113	UnhandledPattern_CAAREA	☐	VarChar	Unicode	30		Yes	
114	UnhandledData_CAAREA	☐	VarChar	Unicode	50		Yes	
115	InputPattern_CAAREA	☐	VarChar	Unicode	30		Yes	
116	ExceptionData_CAAREA	☐	VarChar	Unicode	25		Yes	
117	UserOverrideFlag_CAAREA	☐	VarChar	Unicode	2		Yes	
		☐						

Figure 2-198 Create j11_JOIN_NAB_NAME_AND_ADDR_DATA 9/16


RENAME_COLUMNS - Transformer Stage

OUT2
CUSTOMER_ID
ACCOUNT_ID
WORK_PHONE
CEL_PHONE
HOME_PHONE
HOME_ADDRESS
HOME_ZIP
WORK_ADDRESS
WORK_ZIP
PREF_LANG
ISOCountryCode_COUNTRY
IdentifierFlag_COUNTRY
NameDomain_USPREP_WORK
AddressDomain_USPREP_WORK
AreaDomain_USPREP_WORK
Field1Pattern_USPREP_WORK
Field2Pattern_USPREP_WORK
Field3Pattern_USPREP_WORK
Field4Pattern_USPREP_WORK
Field5Pattern_USPREP_WORK
Field6Pattern_USPREP_WORK
InputPattern_USPREP_WORK
OutboundPattern_USPREP_WORK
UserOverrideFlag_USPREP_WORK
CustomFlag_USPREP_WORK
Space_Field
Zip5_CASS
Zip4_CASS
DeliveryPoint_CASS
DeliveryPointChkDigit_CASS
CarrierRoute_CASS
City_CASS
State_CASS
Urbanization_CASS
Recipient_CASS
DeliveryAddressLine1_CASS

IN3

Constraint:

Derivation	Column Name
OUT2.CUSTOMER_ID	CUSTOMER_ID
OUT2.ACCOUNT_ID	ACCOUNT_ID
OUT2.WORK_PHONE	WORK_PHONE
OUT2.CEL_PHONE	CEL_PHONE
OUT2.HOME_PHONE	HOME_PHONE
OUT2.HOME_ADDRESS	HOME_ADDRESS
OUT2.HOME_ZIP	HOME_ZIP
OUT2.WORK_ADDRESS	WORK_ADDRESS
OUT2.WORK_ZIP	WORK_ZIP
OUT2.PREF_LANG	PREF_LANG
OUT2.ISOCountryCode_COUNTRY	ISOCountryCode_COUNTRY
OUT2.IdentifierFlag_COUNTRY	IdentifierFlag_COUNTRY
OUT2.NameDomain_USPREP_WORK	NameDomain_USPREP_WORK
OUT2.AddressDomain_USPREP_WORK	AddressDomain_USPREP_WORK
OUT2.AreaDomain_USPREP_WORK	AreaDomain_USPREP_WORK
OUT2.Field1Pattern_USPREP_WORK	Field1Pattern_USPREP_WORK
OUT2.Field2Pattern_USPREP_WORK	Field2Pattern_USPREP_WORK
OUT2.Field3Pattern_USPREP_WORK	Field3Pattern_USPREP_WORK
OUT2.Field4Pattern_USPREP_WORK	Field4Pattern_USPREP_WORK
OUT2.Field5Pattern_USPREP_WORK	Field5Pattern_USPREP_WORK
OUT2.Field6Pattern_USPREP_WORK	Field6Pattern_USPREP_WORK
OUT2.InputPattern_USPREP_WORK	InputPattern_USPREP_WORK
OUT2.OutboundPattern_USPREP_WORK	OutboundPattern_USPREP_WORK
OUT2.UserOverrideFlag_USPREP_WORK	UserOverrideFlag_USPREP_WORK
OUT2.CustomFlag_USPREP_WORK	CustomFlag_USPREP_WORK
OUT2.Space_Field	Space_Field
OUT2.Zip5_CASS	Zip5_CASS
OUT2.Zip4_CASS	Zip4_CASS
OUT2.DeliveryPoint_CASS	DeliveryPoint_CASS
OUT2.DeliveryPointChkDigit_CASS	DeliveryPointChkDigit_CASS
OUT2.CarrierRoute_CASS	CarrierRoute_CASS
OUT2.City_CASS	City_CASS
OUT2.State_CASS	State_CASS
OUT2.Urbanization_CASS	Urbanization_CASS

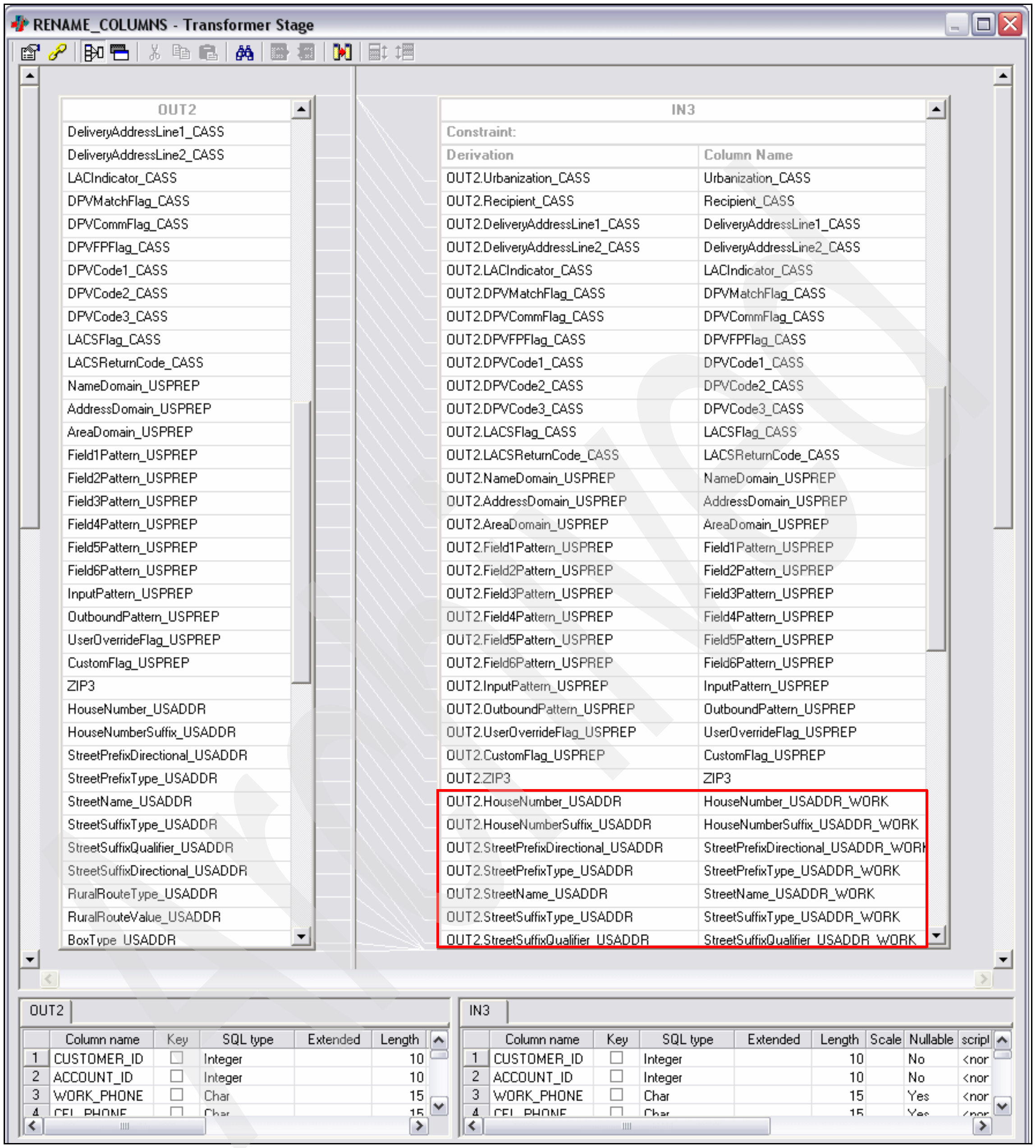
OUT2

	Column name	Key	SQL type	Extended	Length
1	CUSTOMER_ID	☐	Integer		10
2	ACCOUNT_ID	☐	Integer		10
3	WORK_PHONE	☐	Char		15
4	CEL_PHONE	☐	Char		15

IN3

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripl
1	CUSTOMER_ID	☐	Integer		10		No	<nor
2	ACCOUNT_ID	☐	Integer		10		No	<nor
3	WORK_PHONE	☐	Char		15		Yes	<nor
4	CEL_PHONE	☐	Char		15		Yes	<nor

Figure 2-199 Create j11_JOIN_NAB_NAME_AND_ADDR_DATA 10/16



RENAME_COLUMNS - Transformer Stage

OUT2
DeliveryAddressLine1_CASS
DeliveryAddressLine2_CASS
LACIndicator_CASS
DPVMatchFlag_CASS
DPVCommFlag_CASS
DPVFPFlag_CASS
DPVCode1_CASS
DPVCode2_CASS
DPVCode3_CASS
LACSFlag_CASS
LACSReturnCode_CASS
NameDomain_USPREP
AddressDomain_USPREP
AreaDomain_USPREP
Field1Pattern_USPREP
Field2Pattern_USPREP
Field3Pattern_USPREP
Field4Pattern_USPREP
Field5Pattern_USPREP
Field6Pattern_USPREP
InputPattern_USPREP
OutboundPattern_USPREP
UserOverrideFlag_USPREP
CustomFlag_USPREP
ZIP3
HouseNumber_USADDR
HouseNumberSuffix_USADDR
StreetPrefixDirectional_USADDR
StreetPrefixType_USADDR
StreetName_USADDR
StreetSuffixType_USADDR
StreetSuffixQualifier_USADDR
StreetSuffixDirectional_USADDR
RuralRouteType_USADDR
RuralRouteValue_USADDR
BoxType_USADDR

IN3

Constraint:

Derivation	Column Name
OUT2.Urbanization_CASS	Urbanization_CASS
OUT2.Recipient_CASS	Recipient_CASS
OUT2.DeliveryAddressLine1_CASS	DeliveryAddressLine1_CASS
OUT2.DeliveryAddressLine2_CASS	DeliveryAddressLine2_CASS
OUT2.LACIndicator_CASS	LACIndicator_CASS
OUT2.DPVMatchFlag_CASS	DPVMatchFlag_CASS
OUT2.DPVCommFlag_CASS	DPVCommFlag_CASS
OUT2.DPVFPFlag_CASS	DPVFPFlag_CASS
OUT2.DPVCode1_CASS	DPVCode1_CASS
OUT2.DPVCode2_CASS	DPVCode2_CASS
OUT2.DPVCode3_CASS	DPVCode3_CASS
OUT2.LACSFlag_CASS	LACSFlag_CASS
OUT2.LACSReturnCode_CASS	LACSReturnCode_CASS
OUT2.NameDomain_USPREP	NameDomain_USPREP
OUT2.AddressDomain_USPREP	AddressDomain_USPREP
OUT2.AreaDomain_USPREP	AreaDomain_USPREP
OUT2.Field1Pattern_USPREP	Field1Pattern_USPREP
OUT2.Field2Pattern_USPREP	Field2Pattern_USPREP
OUT2.Field3Pattern_USPREP	Field3Pattern_USPREP
OUT2.Field4Pattern_USPREP	Field4Pattern_USPREP
OUT2.Field5Pattern_USPREP	Field5Pattern_USPREP
OUT2.Field6Pattern_USPREP	Field6Pattern_USPREP
OUT2.InputPattern_USPREP	InputPattern_USPREP
OUT2.OutboundPattern_USPREP	OutboundPattern_USPREP
OUT2.UserOverrideFlag_USPREP	UserOverrideFlag_USPREP
OUT2.CustomFlag_USPREP	CustomFlag_USPREP
OUT2.ZIP3	ZIP3
OUT2.HouseNumber_USADDR	HouseNumber_USADDR_WORK
OUT2.HouseNumberSuffix_USADDR	HouseNumberSuffix_USADDR_WORK
OUT2.StreetPrefixDirectional_USADDR	StreetPrefixDirectional_USADDR_WORK
OUT2.StreetPrefixType_USADDR	StreetPrefixType_USADDR_WORK
OUT2.StreetName_USADDR	StreetName_USADDR_WORK
OUT2.StreetSuffixType_USADDR	StreetSuffixType_USADDR_WORK
OUT2.StreetSuffixQualifier_USADDR	StreetSuffixQualifier_USADDR_WORK

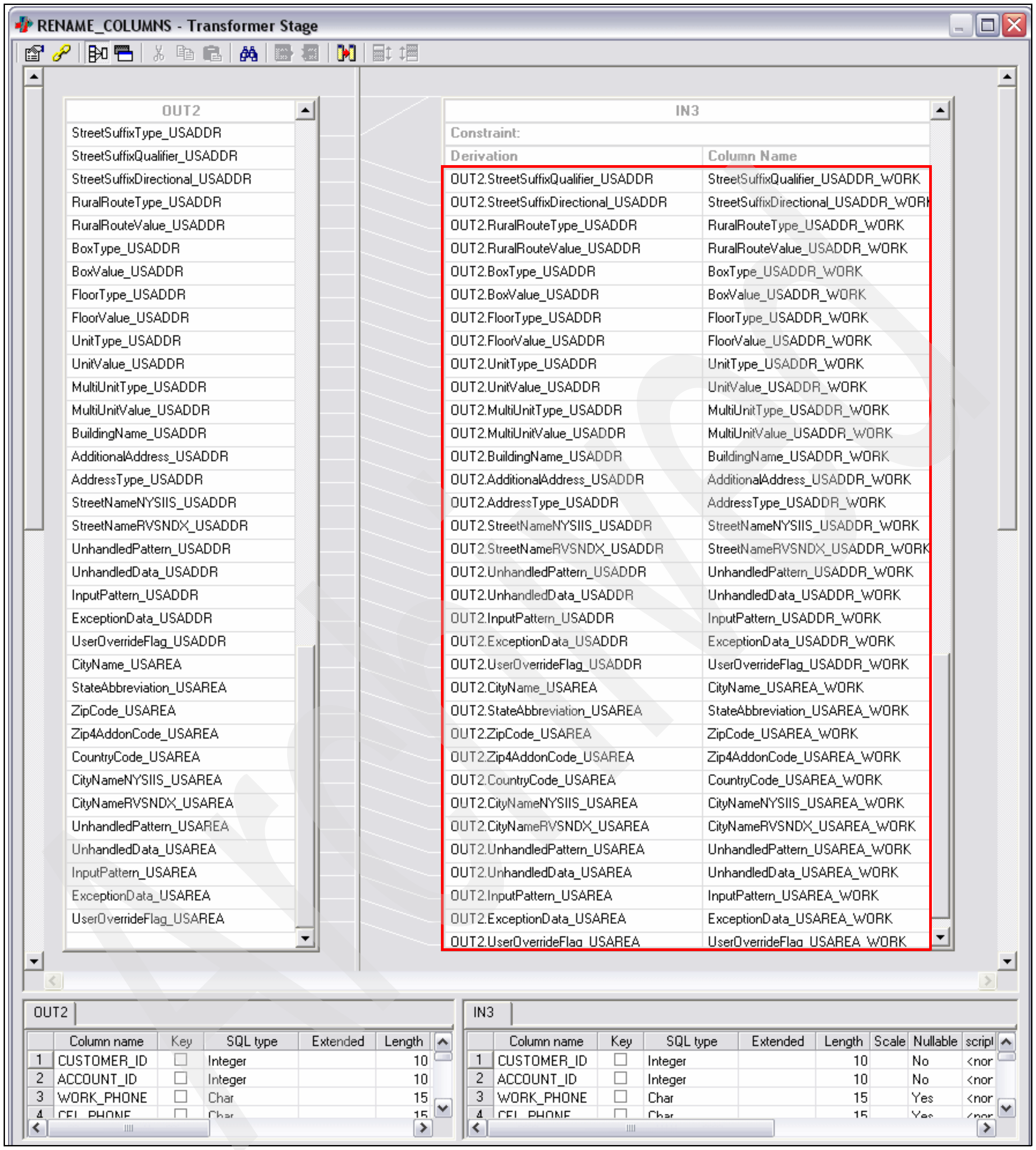
OUT2

	Column name	Key	SQL type	Extended	Length
1	CUSTOMER_ID	☐	Integer		10
2	ACCOUNT_ID	☐	Integer		10
3	WORK_PHONE	☐	Char		15
4	CEL_PHONE	☐	Char		15

IN3

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripl
1	CUSTOMER_ID	☐	Integer		10		No	<nor
2	ACCOUNT_ID	☐	Integer		10		No	<nor
3	WORK_PHONE	☐	Char		15		Yes	<nor
4	CEL_PHONE	☐	Char		15		Yes	<nor

Figure 2-200 Create j11_JOIN_NAB_NAME_AND_ADDR_DATA 11/16



RENAME_COLUMNS - Transformer Stage

OUT2
StreetSuffixType_USADDR
StreetSuffixQualifier_USADDR
StreetSuffixDirectional_USADDR
RuralRouteType_USADDR
RuralRouteValue_USADDR
BoxType_USADDR
BoxValue_USADDR
FloorType_USADDR
FloorValue_USADDR
UnitType_USADDR
UnitValue_USADDR
MultiUnitType_USADDR
MultiUnitValue_USADDR
BuildingName_USADDR
AdditionalAddress_USADDR
AddressType_USADDR
StreetNameNYSIIS_USADDR
StreetNameRVSNDX_USADDR
UnhandledPattern_USADDR
UnhandledData_USADDR
InputPattern_USADDR
ExceptionData_USADDR
UserOverrideFlag_USADDR
CityName_USAREA
StateAbbreviation_USAREA
ZipCode_USAREA
Zip4AddonCode_USAREA
CountryCode_USAREA
CityNameNYSIIS_USAREA
CityNameRVSNDX_USAREA
UnhandledPattern_USAREA
UnhandledData_USAREA
InputPattern_USAREA
ExceptionData_USAREA
UserOverrideFlag_USAREA

IN3

Constraint:

Derivation	Column Name
OUT2.StreetSuffixQualifier_USADDR	StreetSuffixQualifier_USADDR_WORK
OUT2.StreetSuffixDirectional_USADDR	StreetSuffixDirectional_USADDR_WORK
OUT2.RuralRouteType_USADDR	RuralRouteType_USADDR_WORK
OUT2.RuralRouteValue_USADDR	RuralRouteValue_USADDR_WORK
OUT2.BoxType_USADDR	BoxType_USADDR_WORK
OUT2.BoxValue_USADDR	BoxValue_USADDR_WORK
OUT2.FloorType_USADDR	FloorType_USADDR_WORK
OUT2.FloorValue_USADDR	FloorValue_USADDR_WORK
OUT2.UnitType_USADDR	UnitType_USADDR_WORK
OUT2.UnitValue_USADDR	UnitValue_USADDR_WORK
OUT2.MultiUnitType_USADDR	MultiUnitType_USADDR_WORK
OUT2.MultiUnitValue_USADDR	MultiUnitValue_USADDR_WORK
OUT2.BuildingName_USADDR	BuildingName_USADDR_WORK
OUT2.AdditionalAddress_USADDR	AdditionalAddress_USADDR_WORK
OUT2.AddressType_USADDR	AddressType_USADDR_WORK
OUT2.StreetNameNYSIIS_USADDR	StreetNameNYSIIS_USADDR_WORK
OUT2.StreetNameRVSNDX_USADDR	StreetNameRVSNDX_USADDR_WORK
OUT2.UnhandledPattern_USADDR	UnhandledPattern_USADDR_WORK
OUT2.UnhandledData_USADDR	UnhandledData_USADDR_WORK
OUT2.InputPattern_USADDR	InputPattern_USADDR_WORK
OUT2.ExceptionData_USADDR	ExceptionData_USADDR_WORK
OUT2.UserOverrideFlag_USADDR	UserOverrideFlag_USADDR_WORK
OUT2.CityName_USAREA	CityName_USAREA_WORK
OUT2.StateAbbreviation_USAREA	StateAbbreviation_USAREA_WORK
OUT2.ZipCode_USAREA	ZipCode_USAREA_WORK
OUT2.Zip4AddonCode_USAREA	Zip4AddonCode_USAREA_WORK
OUT2.CountryCode_USAREA	CountryCode_USAREA_WORK
OUT2.CityNameNYSIIS_USAREA	CityNameNYSIIS_USAREA_WORK
OUT2.CityNameRVSNDX_USAREA	CityNameRVSNDX_USAREA_WORK
OUT2.UnhandledPattern_USAREA	UnhandledPattern_USAREA_WORK
OUT2.UnhandledData_USAREA	UnhandledData_USAREA_WORK
OUT2.InputPattern_USAREA	InputPattern_USAREA_WORK
OUT2.ExceptionData_USAREA	ExceptionData_USAREA_WORK
OUT2.UserOverrideFlag_USAREA	UserOverrideFlag_USAREA_WORK

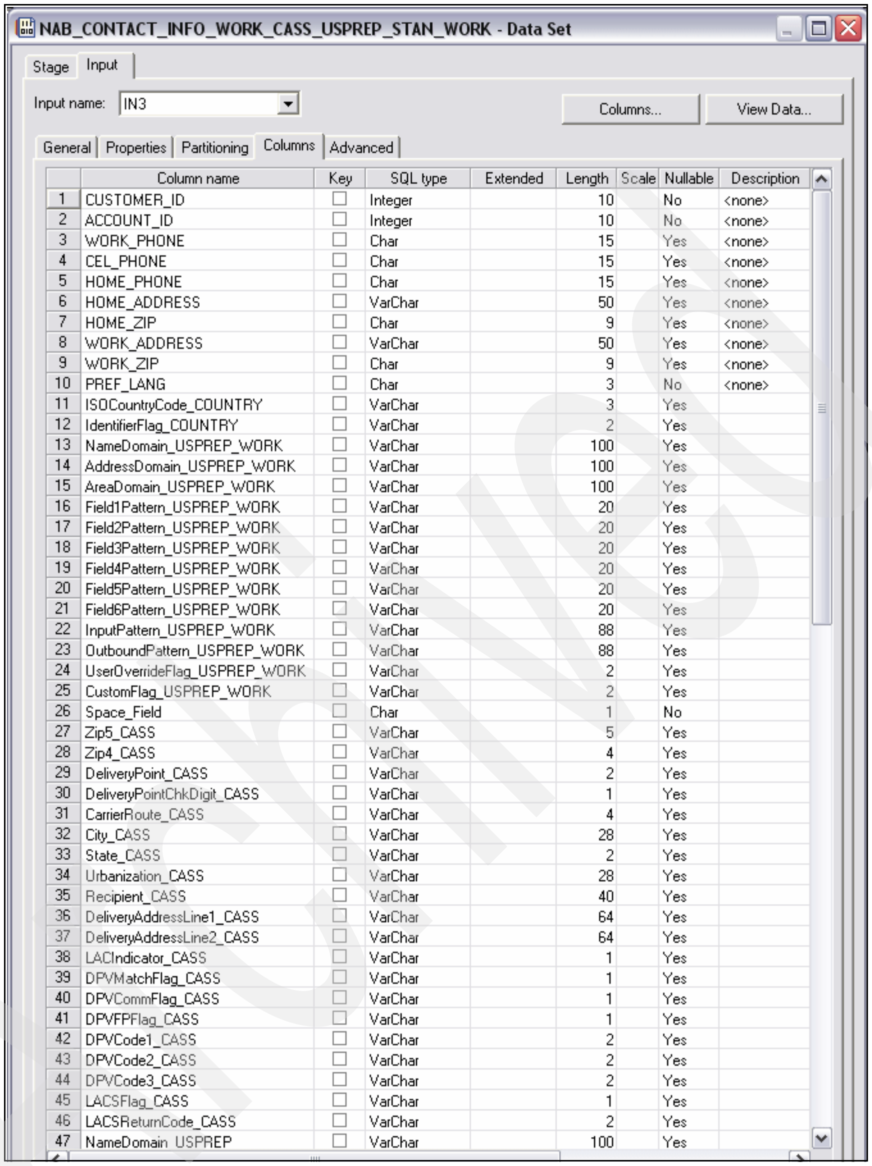
OUT2

	Column name	Key	SQL type	Extended	Length
1	CUSTOMER_ID	☐	Integer		10
2	ACCOUNT_ID	☐	Integer		10
3	WORK_PHONE	☐	Char		15
4	CEL_PHONE	☐	Char		15

IN3

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripl
1	CUSTOMER_ID	☐	Integer		10		No	<nor
2	ACCOUNT_ID	☐	Integer		10		No	<nor
3	WORK_PHONE	☐	Char		15		Yes	<nor
4	CEL_PHONE	☐	Char		15		Yes	<nor

Figure 2-201 Create j11_JOIN_NAB_NAME_AND_ADDR_DATA 12/16



NAB_CONTACT_INFO_WORK_CASS_USPREP_STAN_WORK - Data Set

Stage | Input

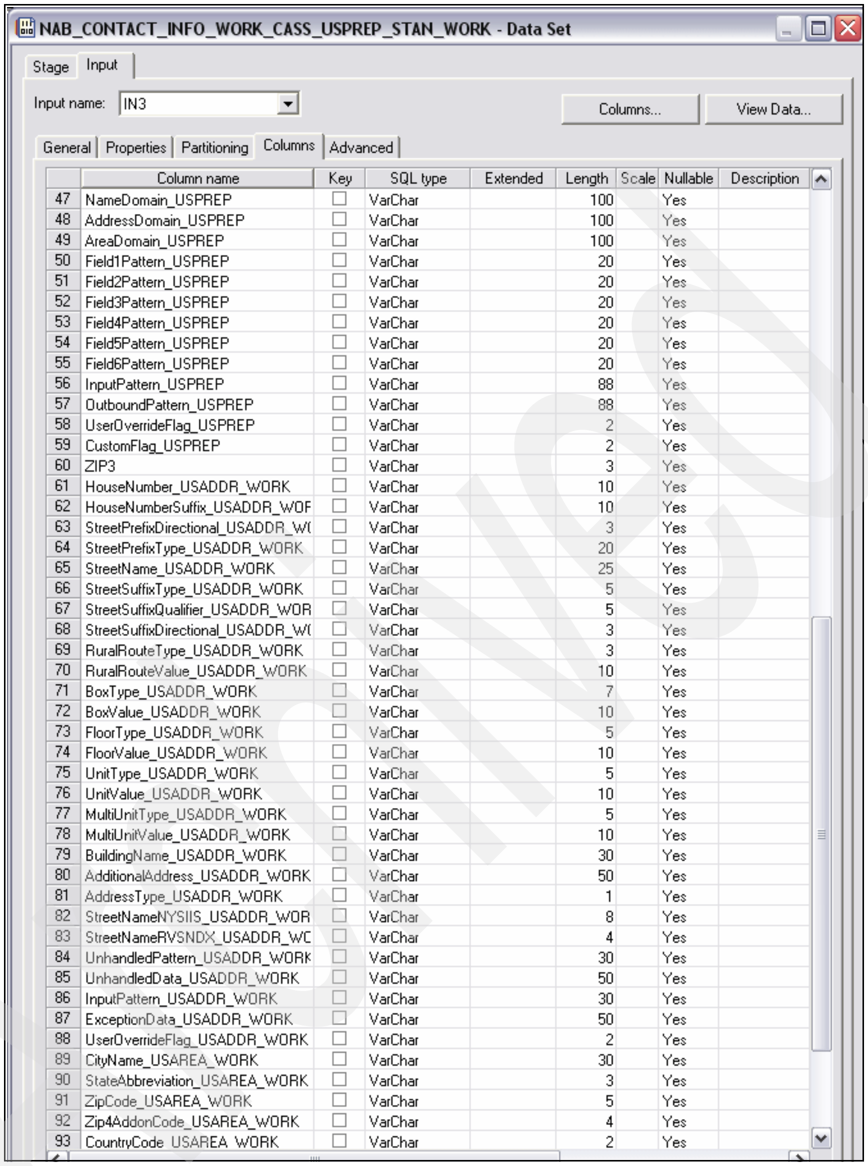
Input name: IN3

Columns... | View Data...

General | Properties | Partitioning | Columns | Advanced

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	Description
1	CUSTOMER_ID	☐	Integer		10		No	<none>
2	ACCOUNT_ID	☐	Integer		10		No	<none>
3	WORK_PHONE	☐	Char		15		Yes	<none>
4	CEL_PHONE	☐	Char		15		Yes	<none>
5	HOME_PHONE	☐	Char		15		Yes	<none>
6	HOME_ADDRESS	☐	VarChar		50		Yes	<none>
7	HOME_ZIP	☐	Char		9		Yes	<none>
8	WORK_ADDRESS	☐	VarChar		50		Yes	<none>
9	WORK_ZIP	☐	Char		9		Yes	<none>
10	PREF_LANG	☐	Char		3		No	<none>
11	ISOCountryCode_COUNTRY	☐	VarChar		3		Yes	
12	IdentifierFlag_COUNTRY	☐	VarChar		2		Yes	
13	NameDomain_USPREP_WORK	☐	VarChar		100		Yes	
14	AddressDomain_USPREP_WORK	☐	VarChar		100		Yes	
15	AreaDomain_USPREP_WORK	☐	VarChar		100		Yes	
16	Field1Pattern_USPREP_WORK	☐	VarChar		20		Yes	
17	Field2Pattern_USPREP_WORK	☐	VarChar		20		Yes	
18	Field3Pattern_USPREP_WORK	☐	VarChar		20		Yes	
19	Field4Pattern_USPREP_WORK	☐	VarChar		20		Yes	
20	Field5Pattern_USPREP_WORK	☐	VarChar		20		Yes	
21	Field6Pattern_USPREP_WORK	☐	VarChar		20		Yes	
22	InputPattern_USPREP_WORK	☐	VarChar		88		Yes	
23	OutboundPattern_USPREP_WORK	☐	VarChar		88		Yes	
24	UserOverrideFlag_USPREP_WORK	☐	VarChar		2		Yes	
25	CustomFlag_USPREP_WORK	☐	VarChar		2		Yes	
26	Space_Field	☐	Char		1		No	
27	Zip5_CASS	☐	VarChar		5		Yes	
28	Zip4_CASS	☐	VarChar		4		Yes	
29	DeliveryPoint_CASS	☐	VarChar		2		Yes	
30	DeliveryPointChkDigit_CASS	☐	VarChar		1		Yes	
31	CarrierRoute_CASS	☐	VarChar		4		Yes	
32	City_CASS	☐	VarChar		28		Yes	
33	State_CASS	☐	VarChar		2		Yes	
34	Urbanization_CASS	☐	VarChar		28		Yes	
35	Recipient_CASS	☐	VarChar		40		Yes	
36	DeliveryAddressLine1_CASS	☐	VarChar		64		Yes	
37	DeliveryAddressLine2_CASS	☐	VarChar		64		Yes	
38	LACIndicator_CASS	☐	VarChar		1		Yes	
39	DPVMatchFlag_CASS	☐	VarChar		1		Yes	
40	DPVCommFlag_CASS	☐	VarChar		1		Yes	
41	DPVFPFlag_CASS	☐	VarChar		1		Yes	
42	DPVCode1_CASS	☐	VarChar		2		Yes	
43	DPVCode2_CASS	☐	VarChar		2		Yes	
44	DPVCode3_CASS	☐	VarChar		2		Yes	
45	LACSFlag_CASS	☐	VarChar		1		Yes	
46	LACSReturnCode_CASS	☐	VarChar		2		Yes	
47	NameDomain_USPREP	☐	VarChar		100		Yes	

Figure 2-202 Create j11_JOIN_NAB_NAME_AND_ADDR_DATA 13/16



NAB_CONTACT_INFO_WORK_CASS_USPREP_STAN_WORK - Data Set

Stage | Input

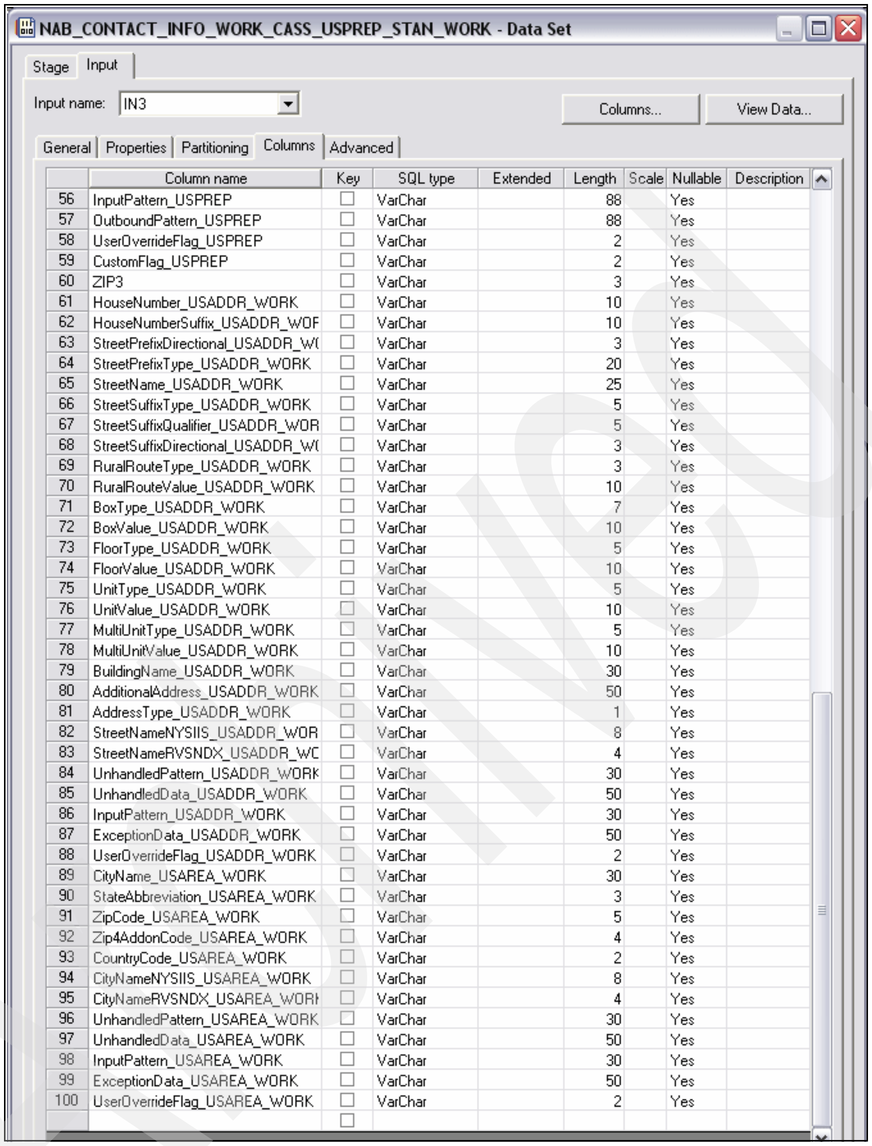
Input name: IN3

Columns... | View Data...

General | Properties | Partitioning | Columns | Advanced

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	Description
47	NameDomain_USPREP	☐	VarChar		100		Yes	
48	AddressDomain_USPREP	☐	VarChar		100		Yes	
49	AreaDomain_USPREP	☐	VarChar		100		Yes	
50	Field1Pattern_USPREP	☐	VarChar		20		Yes	
51	Field2Pattern_USPREP	☐	VarChar		20		Yes	
52	Field3Pattern_USPREP	☐	VarChar		20		Yes	
53	Field4Pattern_USPREP	☐	VarChar		20		Yes	
54	Field5Pattern_USPREP	☐	VarChar		20		Yes	
55	Field6Pattern_USPREP	☐	VarChar		20		Yes	
56	InputPattern_USPREP	☐	VarChar		88		Yes	
57	OutboundPattern_USPREP	☐	VarChar		88		Yes	
58	UserOverrideFlag_USPREP	☐	VarChar		2		Yes	
59	CustomFlag_USPREP	☐	VarChar		2		Yes	
60	ZIP3	☐	VarChar		3		Yes	
61	HouseNumber_USADDR_WORK	☐	VarChar		10		Yes	
62	HouseNumberSuffix_USADDR_WOF	☐	VarChar		10		Yes	
63	StreetPrefixDirectional_USADDR_W(	☐	VarChar		3		Yes	
64	StreetPrefixType_USADDR_WORK	☐	VarChar		20		Yes	
65	StreetName_USADDR_WORK	☐	VarChar		25		Yes	
66	StreetSuffixType_USADDR_WORK	☐	VarChar		5		Yes	
67	StreetSuffixQualifier_USADDR_WOR	☐	VarChar		5		Yes	
68	StreetSuffixDirectional_USADDR_W(	☐	VarChar		3		Yes	
69	RuralRouteType_USADDR_WORK	☐	VarChar		3		Yes	
70	RuralRouteValue_USADDR_WORK	☐	VarChar		10		Yes	
71	BoxType_USADDR_WORK	☐	VarChar		7		Yes	
72	BoxValue_USADDR_WORK	☐	VarChar		10		Yes	
73	FloorType_USADDR_WORK	☐	VarChar		5		Yes	
74	FloorValue_USADDR_WORK	☐	VarChar		10		Yes	
75	UnitType_USADDR_WORK	☐	VarChar		5		Yes	
76	UnitValue_USADDR_WORK	☐	VarChar		10		Yes	
77	MultiUnitType_USADDR_WORK	☐	VarChar		5		Yes	
78	MultiUnitValue_USADDR_WORK	☐	VarChar		10		Yes	
79	BuildingName_USADDR_WORK	☐	VarChar		30		Yes	
80	AdditionalAddress_USADDR_WORK	☐	VarChar		50		Yes	
81	AddressType_USADDR_WORK	☐	VarChar		1		Yes	
82	StreetNameNYSIIS_USADDR_WOR	☐	VarChar		8		Yes	
83	StreetNameRVSNDX_USADDR_WC	☐	VarChar		4		Yes	
84	UnhandledPattern_USADDR_WORK	☐	VarChar		30		Yes	
85	UnhandledData_USADDR_WORK	☐	VarChar		50		Yes	
86	InputPattern_USADDR_WORK	☐	VarChar		30		Yes	
87	ExceptionData_USADDR_WORK	☐	VarChar		50		Yes	
88	UserOverrideFlag_USADDR_WORK	☐	VarChar		2		Yes	
89	CityName_USAREA_WORK	☐	VarChar		30		Yes	
90	StateAbbreviation_USAREA_WORK	☐	VarChar		3		Yes	
91	ZipCode_USAREA_WORK	☐	VarChar		5		Yes	
92	Zip4AddonCode_USAREA_WORK	☐	VarChar		4		Yes	
93	CountryCode_USAREA_WORK	☐	VarChar		2		Yes	

Figure 2-203 Create j11_JOIN_NAB_NAME_AND_ADDR_DATA 14/16


NAB_CONTACT_INFO_WORK_CASS_USPREP_STAN_WORK - Data Set

Stage | Input

Input name: IN3

General | Properties | Partitioning | Columns | Advanced

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	Description
56	InputPattern_USPREP	☐	VarChar		88		Yes	
57	OutboundPattern_USPREP	☐	VarChar		88		Yes	
58	UserOverrideFlag_USPREP	☐	VarChar		2		Yes	
59	CustomFlag_USPREP	☐	VarChar		2		Yes	
60	ZIP3	☐	VarChar		3		Yes	
61	HouseNumber_USADDR_WORK	☐	VarChar		10		Yes	
62	HouseNumberSuffix_USADDR_WOF	☐	VarChar		10		Yes	
63	StreetPrefixDirectional_USADDR_W(	☐	VarChar		3		Yes	
64	StreetPrefixType_USADDR_WORK	☐	VarChar		20		Yes	
65	StreetName_USADDR_WORK	☐	VarChar		25		Yes	
66	StreetSuffixType_USADDR_WORK	☐	VarChar		5		Yes	
67	StreetSuffixQualifier_USADDR_WOR	☐	VarChar		5		Yes	
68	StreetSuffixDirectional_USADDR_W(	☐	VarChar		3		Yes	
69	RuralRouteType_USADDR_WORK	☐	VarChar		3		Yes	
70	RuralRouteValue_USADDR_WORK	☐	VarChar		10		Yes	
71	BoxType_USADDR_WORK	☐	VarChar		7		Yes	
72	BoxValue_USADDR_WORK	☐	VarChar		10		Yes	
73	FloorType_USADDR_WORK	☐	VarChar		5		Yes	
74	FloorValue_USADDR_WORK	☐	VarChar		10		Yes	
75	UnitType_USADDR_WORK	☐	VarChar		5		Yes	
76	UnitValue_USADDR_WORK	☐	VarChar		10		Yes	
77	MultiUnitType_USADDR_WORK	☐	VarChar		5		Yes	
78	MultiUnitValue_USADDR_WORK	☐	VarChar		10		Yes	
79	BuildingName_USADDR_WORK	☐	VarChar		30		Yes	
80	AdditionalAddress_USADDR_WORK	☐	VarChar		50		Yes	
81	AddressType_USADDR_WORK	☐	VarChar		1		Yes	
82	StreetNameNYSIIS_USADDR_WOR	☐	VarChar		8		Yes	
83	StreetNameRVSNDX_USADDR_WC	☐	VarChar		4		Yes	
84	UnhandledPattern_USADDR_WORK	☐	VarChar		30		Yes	
85	UnhandledData_USADDR_WORK	☐	VarChar		50		Yes	
86	InputPattern_USADDR_WORK	☐	VarChar		30		Yes	
87	ExceptionData_USADDR_WORK	☐	VarChar		50		Yes	
88	UserOverrideFlag_USADDR_WORK	☐	VarChar		2		Yes	
89	CityName_USAREA_WORK	☐	VarChar		30		Yes	
90	StateAbbreviation_USAREA_WORK	☐	VarChar		3		Yes	
91	ZipCode_USAREA_WORK	☐	VarChar		5		Yes	
92	Zip4AddonCode_USAREA_WORK	☐	VarChar		4		Yes	
93	CountryCode_USAREA_WORK	☐	VarChar		2		Yes	
94	CityNameNYSIIS_USAREA_WORK	☐	VarChar		8		Yes	
95	CityNameRVSNDX_USAREA_WORI	☐	VarChar		4		Yes	
96	UnhandledPattern_USAREA_WORK	☐	VarChar		30		Yes	
97	UnhandledData_USAREA_WORK	☐	VarChar		50		Yes	
98	InputPattern_USAREA_WORK	☐	VarChar		30		Yes	
99	ExceptionData_USAREA_WORK	☐	VarChar		50		Yes	
100	UserOverrideFlag_USAREA_WORK	☐	VarChar		2		Yes	
		☐						

Figure 2-204 Create j11_JOIN_NAB_NAME_AND_ADDR_DATA 15/16

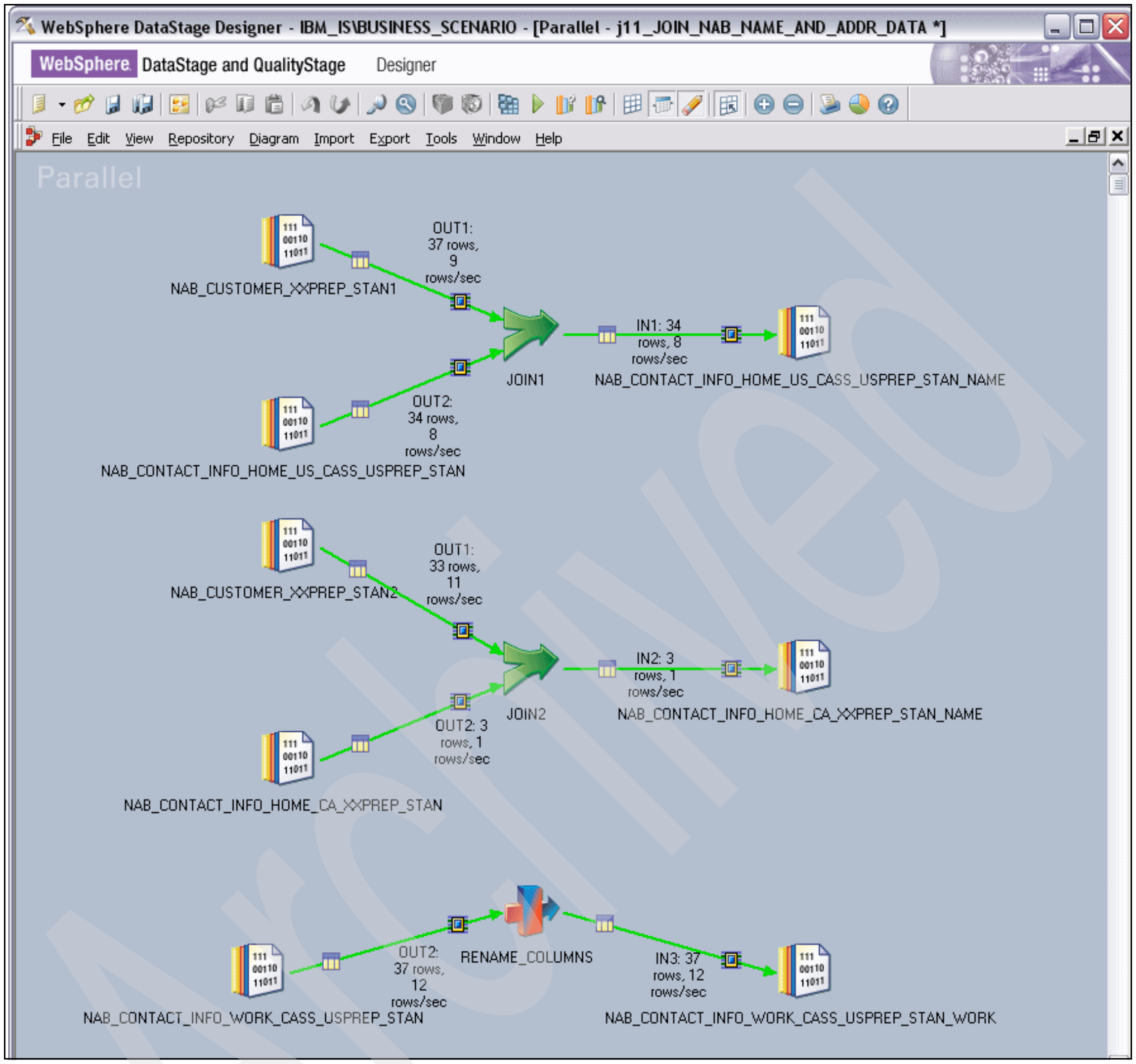

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j11_JOIN_NAB_NAME_AND_ADDR_DATA *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NAB_CUSTOMER_XXPREP_STAN1
OUT1: 37 rows, 9 rows/sec
OUT2: 34 rows, 8 rows/sec
NAB_CONTACT_INFO_HOME_US_CASS_USPREP_STAN
JOIN1
IN1: 34 rows, 8 rows/sec
NAB_CONTACT_INFO_HOME_US_CASS_USPREP_STAN_NAME
NAB_CUSTOMER_XXPREP_STAN2
OUT1: 33 rows, 11 rows/sec
OUT2: 3 rows, 1 rows/sec
NAB_CONTACT_INFO_HOME_CA_XXPREP_STAN
JOIN2
IN2: 3 rows, 1 rows/sec
NAB_CONTACT_INFO_HOME_CA_XXPREP_STAN_NAME
NAB_CONTACT_INFO_WORK_CASS_USPREP_STAN
OUT2: 37 rows, 12 rows/sec
RENAME_COLUMNS
IN3: 37 rows, 12 rows/sec
NAB_CONTACT_INFO_WORK_CASS_USPREP_STAN_WORK

Figure 2-205 Create j11_JOIN_NAB_NAME_AND_ADDR_DATA 16/16

j12_JOIN_NAB_WORK_AND_HOME

In this step, we append the customer's standardized U.S. work address information to the two files that were created in the "j11_JOIN_NAB_NAME_AND_ADDR_DATA" on page 657 step. All the addresses in the work address are U.S. addresses.

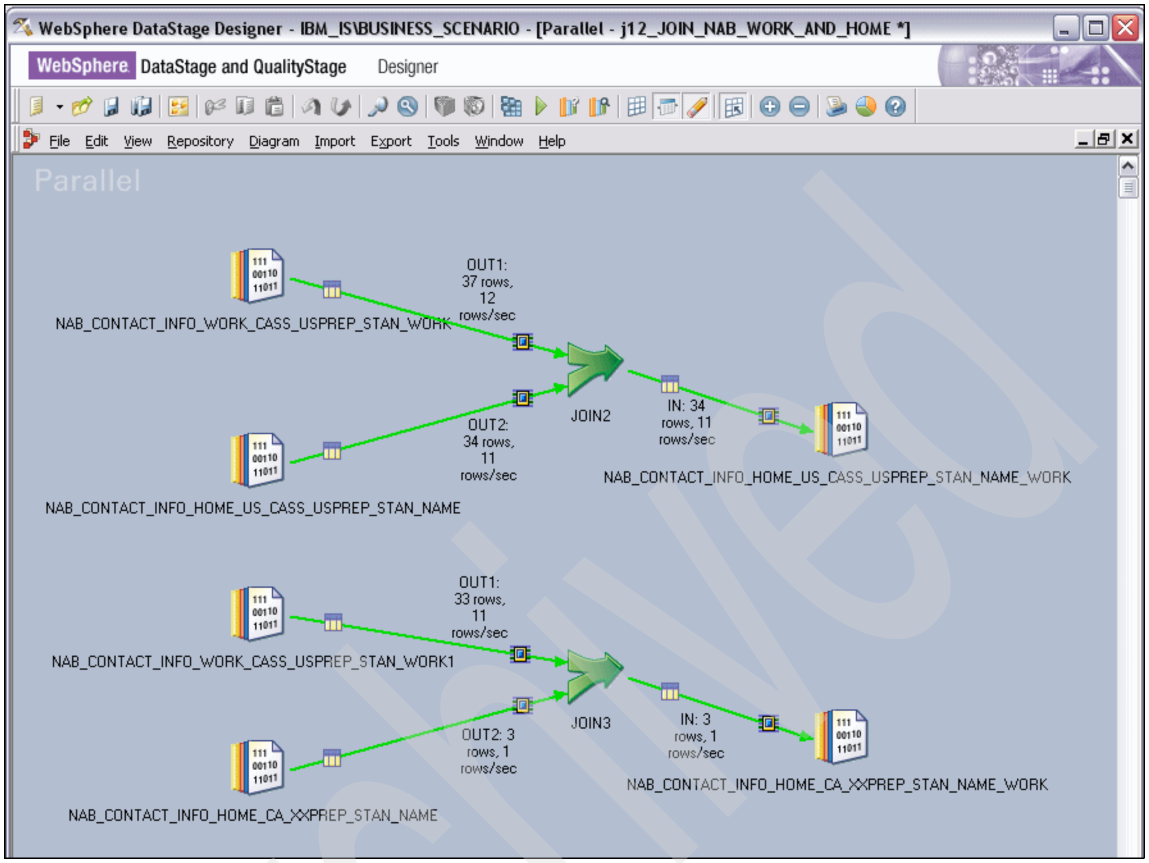
Figure 2-206 on page 676 shows the various stages that are used in this job. It includes the data sets that were created in “j11_JOIN_NAB_NAME_AND_ADDR_DATA” on page 657, a Join stage, and an output Data Set stage for each Join stage. We modified the names of the stages as shown.

Figure 2-207 on page 677 shows the Join2 - Join window, which indicates an inner join of the standardized name and U.S. work addresses customer CONTACT_INFO file (from the previous step) with the standardized name and U.S. home addresses customer CONTACT_INFO file. The metadata of the columns of the joined output is shown in Figure 2-208 on page 678 through Figure 2-211 on page 681.

Figure 2-212 on page 682 through Figure 2-216 on page 686 corresponds the inner join (JOIN3 - Join window) of the standardized name and U.S. work addresses customer CONTACT_INFO file (from the previous step) with the standardized name and Canadian home addresses customer CONTACT_INFO file. The metadata of the columns of the joined output is shown in Figure 2-213 on page 683 through Figure 2-216 on page 686.

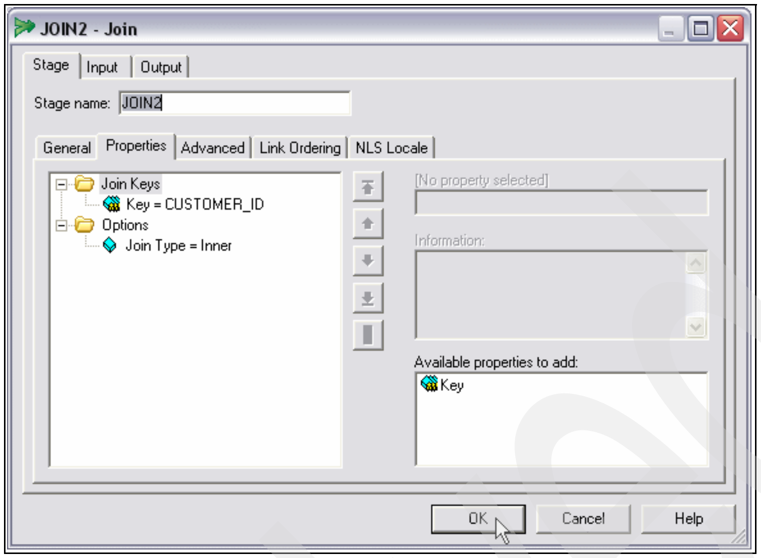
Figure 2-217 on page 687 shows the results after compiling and running this job.

Proceed now to “j13_PREPARE_NAB_DATA_FOR_FUNNEL” on page 687.



WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j12_JOIN_NAB_WORK_AND_HOME *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
OUT1: 37 rows, 12 rows/sec
NAB_CONTACT_INFO_WORK_CASS_USPREP_STAN_WORK
JOIN2
IN: 34 rows, 11 rows/sec
OUT2: 34 rows, 11 rows/sec
NAB_CONTACT_INFO_HOME_US_CASS_USPREP_STAN_NAME_WORK
NAB_CONTACT_INFO_HOME_US_CASS_USPREP_STAN_NAME
OUT1: 33 rows, 11 rows/sec
NAB_CONTACT_INFO_WORK_CASS_USPREP_STAN_WORK1
JOIN3
IN: 3 rows, 1 rows/sec
OUT2: 3 rows, 1 rows/sec
NAB_CONTACT_INFO_HOME_CA_XXPREP_STAN_NAME_WORK
NAB_CONTACT_INFO_HOME_CA_XXPREP_STAN_NAME

Figure 2-206 Create j12_JOIN_NAB_WORK_AND_HOME 1/12


JOIN2 - Join

Stage | Input | Output

Stage name: JOIN2

General | Properties | Advanced | Link Ordering | NLS Locale

Join Keys
  Key = CUSTOMER_ID
Options
  Join Type = Inner

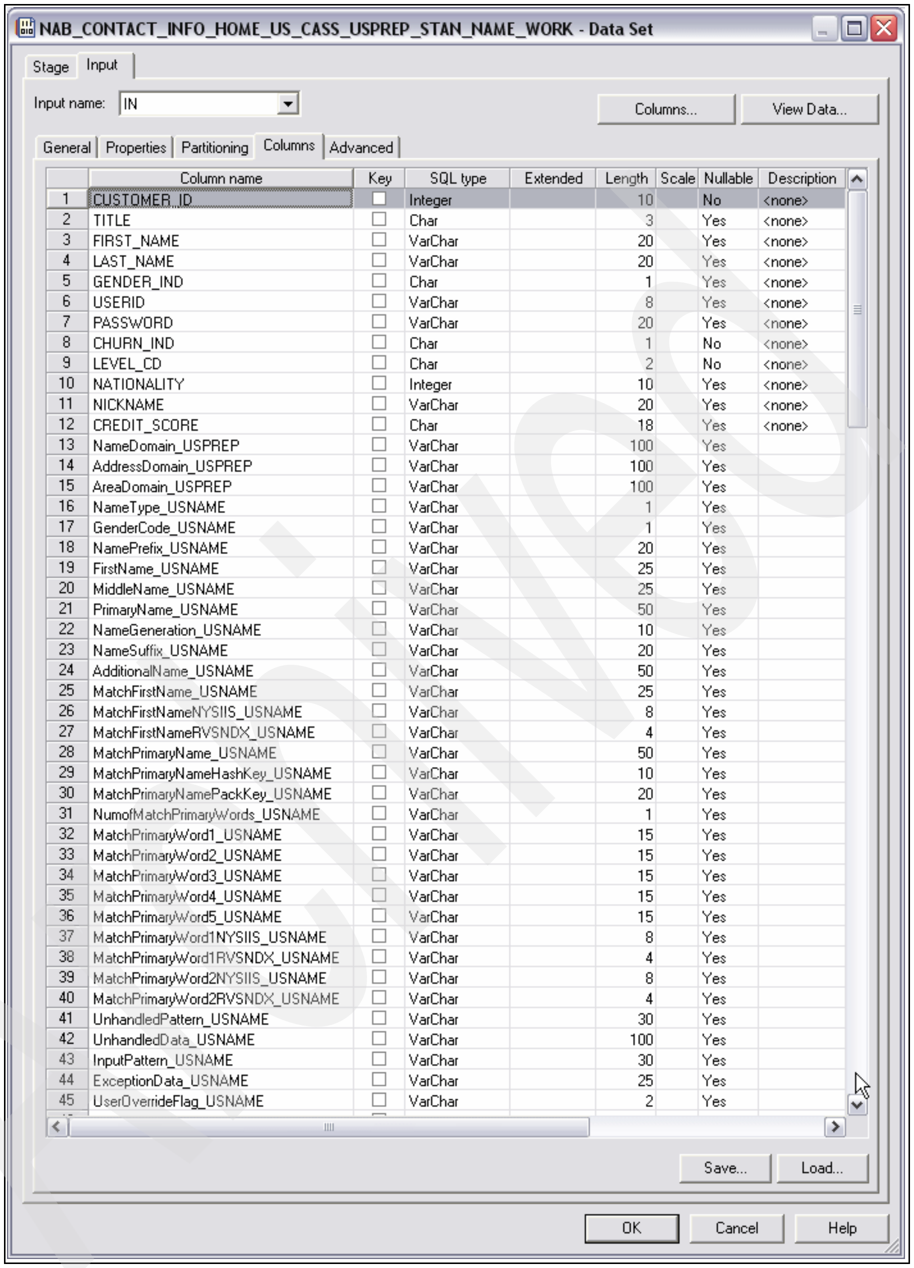
[No property selected]

Information:

Available properties to add:

Key

Figure 2-207 Create j12_JOIN_NAB_WORK_AND_HOME 2/12



NAB_CONTACT_INFO_HOME_US_CASS_USPREP_STAN_NAME_WORK - Data Set

Stage | Input

Input name: IN

Columns... | View Data...

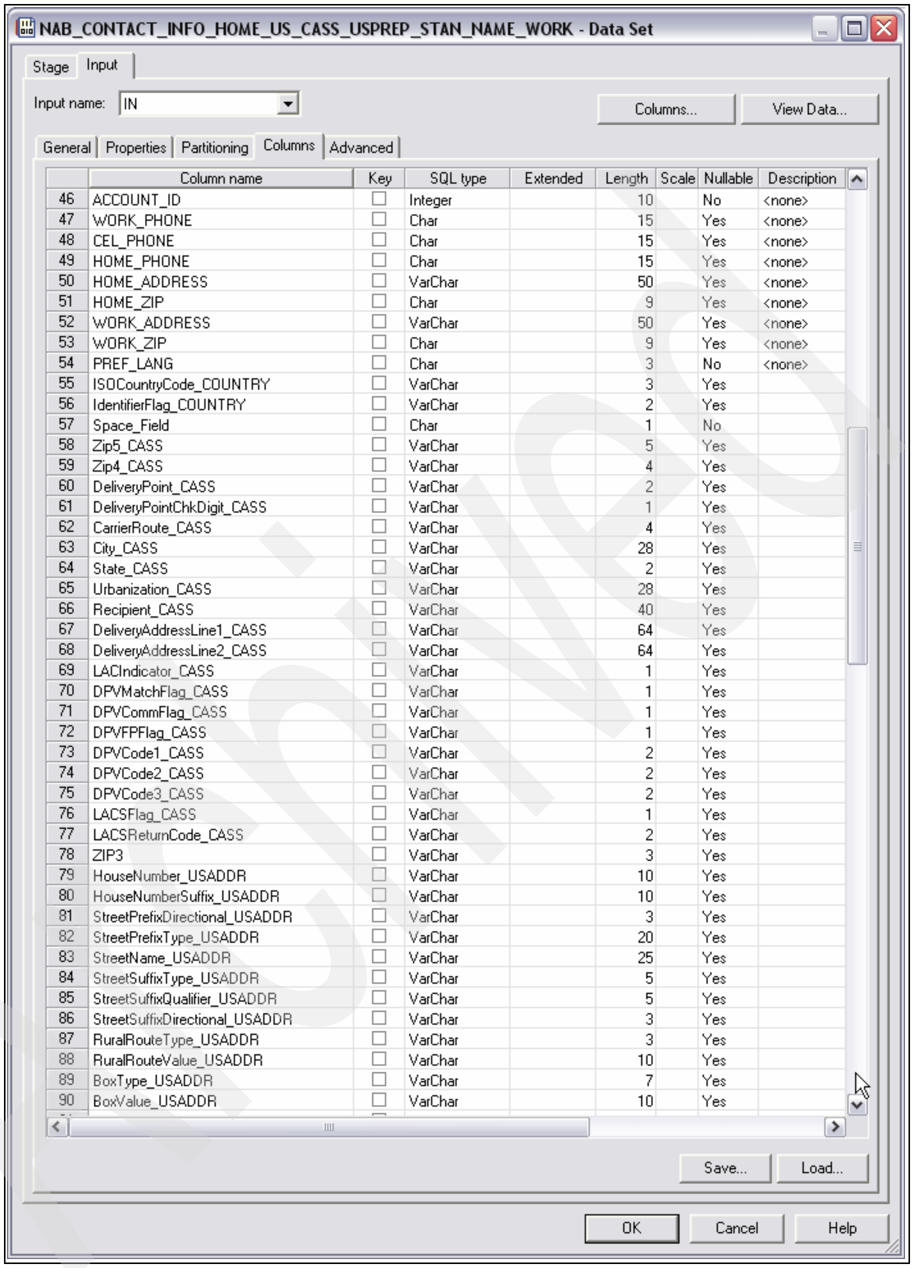
General | Properties | Partitioning | Columns | Advanced

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	Description
1	CUSTOMER_ID	☐	Integer		10		No	<none>
2	TITLE	☐	Char		3		Yes	<none>
3	FIRST_NAME	☐	VarChar		20		Yes	<none>
4	LAST_NAME	☐	VarChar		20		Yes	<none>
5	GENDER_IND	☐	Char		1		Yes	<none>
6	USERID	☐	VarChar		8		Yes	<none>
7	PASSWORD	☐	VarChar		20		Yes	<none>
8	CHURN_IND	☐	Char		1		No	<none>
9	LEVEL_CD	☐	Char		2		No	<none>
10	NATIONALITY	☐	Integer		10		Yes	<none>
11	NICKNAME	☐	VarChar		20		Yes	<none>
12	CREDIT_SCORE	☐	Char		18		Yes	<none>
13	NameDomain_USPREP	☐	VarChar		100		Yes	
14	AddressDomain_USPREP	☐	VarChar		100		Yes	
15	AreaDomain_USPREP	☐	VarChar		100		Yes	
16	NameType_USNAME	☐	VarChar		1		Yes	
17	GenderCode_USNAME	☐	VarChar		1		Yes	
18	NamePrefix_USNAME	☐	VarChar		20		Yes	
19	FirstName_USNAME	☐	VarChar		25		Yes	
20	MiddleName_USNAME	☐	VarChar		25		Yes	
21	PrimaryName_USNAME	☐	VarChar		50		Yes	
22	NameGeneration_USNAME	☐	VarChar		10		Yes	
23	NameSuffix_USNAME	☐	VarChar		20		Yes	
24	AdditionalName_USNAME	☐	VarChar		50		Yes	
25	MatchFirstName_USNAME	☐	VarChar		25		Yes	
26	MatchFirstNameNYSIIS_USNAME	☐	VarChar		8		Yes	
27	MatchFirstNameRVSNDX_USNAME	☐	VarChar		4		Yes	
28	MatchPrimaryName_USNAME	☐	VarChar		50		Yes	
29	MatchPrimaryNameHashKey_USNAME	☐	VarChar		10		Yes	
30	MatchPrimaryNamePackKey_USNAME	☐	VarChar		20		Yes	
31	NumofMatchPrimaryWords_USNAME	☐	VarChar		1		Yes	
32	MatchPrimaryWord1_USNAME	☐	VarChar		15		Yes	
33	MatchPrimaryWord2_USNAME	☐	VarChar		15		Yes	
34	MatchPrimaryWord3_USNAME	☐	VarChar		15		Yes	
35	MatchPrimaryWord4_USNAME	☐	VarChar		15		Yes	
36	MatchPrimaryWord5_USNAME	☐	VarChar		15		Yes	
37	MatchPrimaryWord1NYSIIS_USNAME	☐	VarChar		8		Yes	
38	MatchPrimaryWord1RVSNDX_USNAME	☐	VarChar		4		Yes	
39	MatchPrimaryWord2NYSIIS_USNAME	☐	VarChar		8		Yes	
40	MatchPrimaryWord2RVSNDX_USNAME	☐	VarChar		4		Yes	
41	UnhandledPattern_USNAME	☐	VarChar		30		Yes	
42	UnhandledData_USNAME	☐	VarChar		100		Yes	
43	InputPattern_USNAME	☐	VarChar		30		Yes	
44	ExceptionData_USNAME	☐	VarChar		25		Yes	
45	UserOverrideFlag_USNAME	☐	VarChar		2		Yes	

Save... | Load...

OK | Cancel | Help

Figure 2-208 Create j12_JOIN_NAB_WORK_AND_HOME 3/12



NAB_CONTACT_INFO_HOME_US_CASS_USPREP_STAN_NAME_WORK - Data Set

Stage | Input

Input name: IN

Columns... | View Data...

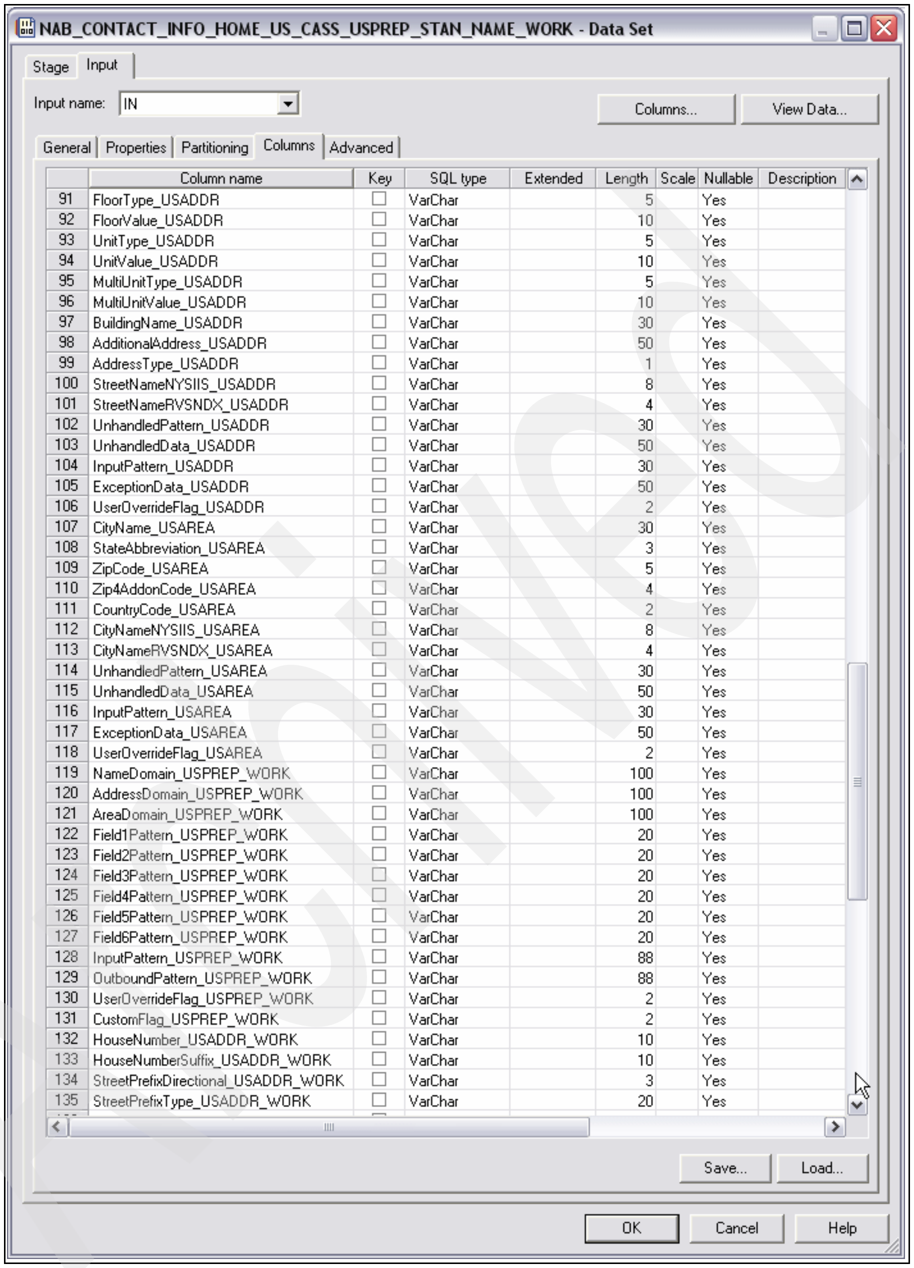
General | Properties | Partitioning | Columns | Advanced

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	Description
46	ACCOUNT_ID	☐	Integer		10		No	<none>
47	WORK_PHONE	☐	Char		15		Yes	<none>
48	CEL_PHONE	☐	Char		15		Yes	<none>
49	HOME_PHONE	☐	Char		15		Yes	<none>
50	HOME_ADDRESS	☐	VarChar		50		Yes	<none>
51	HOME_ZIP	☐	Char		9		Yes	<none>
52	WORK_ADDRESS	☐	VarChar		50		Yes	<none>
53	WORK_ZIP	☐	Char		9		Yes	<none>
54	PREF_LANG	☐	Char		3		No	<none>
55	ISOCountryCode_COUNTRY	☐	VarChar		3		Yes	
56	IdentifierFlag_COUNTRY	☐	VarChar		2		Yes	
57	Space_Field	☐	Char		1		No	
58	Zip5_CASS	☐	VarChar		5		Yes	
59	Zip4_CASS	☐	VarChar		4		Yes	
60	DeliveryPoint_CASS	☐	VarChar		2		Yes	
61	DeliveryPointChkDigit_CASS	☐	VarChar		1		Yes	
62	CarrierRoute_CASS	☐	VarChar		4		Yes	
63	City_CASS	☐	VarChar		28		Yes	
64	State_CASS	☐	VarChar		2		Yes	
65	Urbanization_CASS	☐	VarChar		28		Yes	
66	Recipient_CASS	☐	VarChar		40		Yes	
67	DeliveryAddressLine1_CASS	☐	VarChar		64		Yes	
68	DeliveryAddressLine2_CASS	☐	VarChar		64		Yes	
69	LACIndicator_CASS	☐	VarChar		1		Yes	
70	DPVMatchFlag_CASS	☐	VarChar		1		Yes	
71	DPVCommFlag_CASS	☐	VarChar		1		Yes	
72	DPVFPFlag_CASS	☐	VarChar		1		Yes	
73	DPVCode1_CASS	☐	VarChar		2		Yes	
74	DPVCode2_CASS	☐	VarChar		2		Yes	
75	DPVCode3_CASS	☐	VarChar		2		Yes	
76	LACSFlag_CASS	☐	VarChar		1		Yes	
77	LACSReturnCode_CASS	☐	VarChar		2		Yes	
78	ZIP3	☐	VarChar		3		Yes	
79	HouseNumber_USADDR	☐	VarChar		10		Yes	
80	HouseNumberSuffix_USADDR	☐	VarChar		10		Yes	
81	StreetPrefixDirectional_USADDR	☐	VarChar		3		Yes	
82	StreetPrefixType_USADDR	☐	VarChar		20		Yes	
83	StreetName_USADDR	☐	VarChar		25		Yes	
84	StreetSuffixType_USADDR	☐	VarChar		5		Yes	
85	StreetSuffixQualifier_USADDR	☐	VarChar		5		Yes	
86	StreetSuffixDirectional_USADDR	☐	VarChar		3		Yes	
87	RuralRouteType_USADDR	☐	VarChar		3		Yes	
88	RuralRouteValue_USADDR	☐	VarChar		10		Yes	
89	BoxType_USADDR	☐	VarChar		7		Yes	
90	BoxValue_USADDR	☐	VarChar		10		Yes	

Save... | Load...

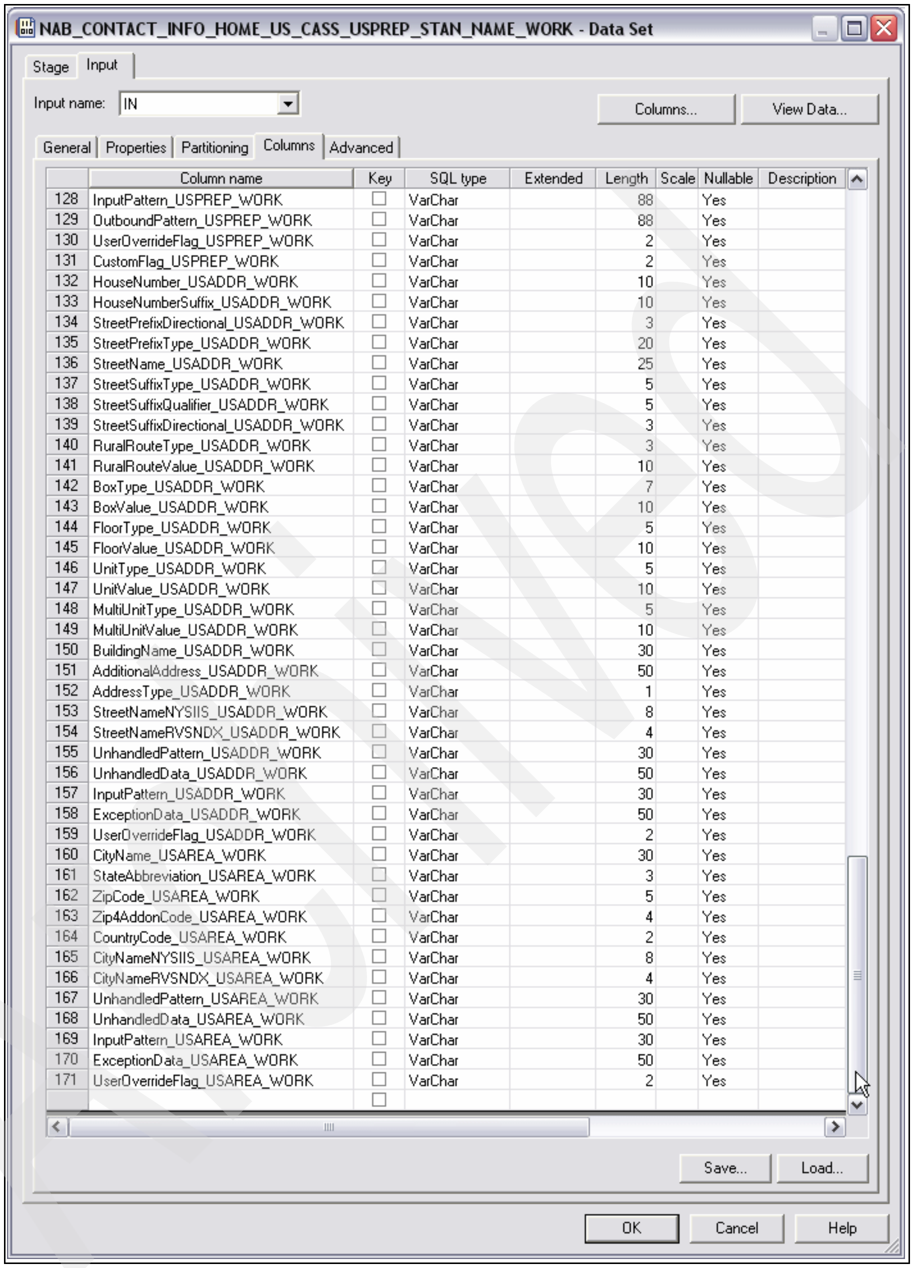
OK | Cancel | Help

Figure 2-209 Create j12_JOIN_NAB_WORK_AND_HOME 4/12


NAB_CONTACT_INFO_HOME_US_CASS_USPREP_STAN_NAME_WORK - Data Set

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	Description
91	FloorType_USADDR	☐	VarChar		5		Yes	
92	FloorValue_USADDR	☐	VarChar		10		Yes	
93	UnitType_USADDR	☐	VarChar		5		Yes	
94	UnitValue_USADDR	☐	VarChar		10		Yes	
95	MultiUnitType_USADDR	☐	VarChar		5		Yes	
96	MultiUnitValue_USADDR	☐	VarChar		10		Yes	
97	BuildingName_USADDR	☐	VarChar		30		Yes	
98	AdditionalAddress_USADDR	☐	VarChar		50		Yes	
99	AddressType_USADDR	☐	VarChar		1		Yes	
100	StreetNameNYSIIS_USADDR	☐	VarChar		8		Yes	
101	StreetNameRVSNDX_USADDR	☐	VarChar		4		Yes	
102	UnhandledPattern_USADDR	☐	VarChar		30		Yes	
103	UnhandledData_USADDR	☐	VarChar		50		Yes	
104	InputPattern_USADDR	☐	VarChar		30		Yes	
105	ExceptionData_USADDR	☐	VarChar		50		Yes	
106	UserOverrideFlag_USADDR	☐	VarChar		2		Yes	
107	CityName_USAREA	☐	VarChar		30		Yes	
108	StateAbbreviation_USAREA	☐	VarChar		3		Yes	
109	ZipCode_USAREA	☐	VarChar		5		Yes	
110	Zip4AddonCode_USAREA	☐	VarChar		4		Yes	
111	CountryCode_USAREA	☐	VarChar		2		Yes	
112	CityNameNYSIIS_USAREA	☐	VarChar		8		Yes	
113	CityNameRVSNDX_USAREA	☐	VarChar		4		Yes	
114	UnhandledPattern_USAREA	☐	VarChar		30		Yes	
115	UnhandledData_USAREA	☐	VarChar		50		Yes	
116	InputPattern_USAREA	☐	VarChar		30		Yes	
117	ExceptionData_USAREA	☐	VarChar		50		Yes	
118	UserOverrideFlag_USAREA	☐	VarChar		2		Yes	
119	NameDomain_USPREP_WORK	☐	VarChar		100		Yes	
120	AddressDomain_USPREP_WORK	☐	VarChar		100		Yes	
121	AreaDomain_USPREP_WORK	☐	VarChar		100		Yes	
122	Field1Pattern_USPREP_WORK	☐	VarChar		20		Yes	
123	Field2Pattern_USPREP_WORK	☐	VarChar		20		Yes	
124	Field3Pattern_USPREP_WORK	☐	VarChar		20		Yes	
125	Field4Pattern_USPREP_WORK	☐	VarChar		20		Yes	
126	Field5Pattern_USPREP_WORK	☐	VarChar		20		Yes	
127	Field6Pattern_USPREP_WORK	☐	VarChar		20		Yes	
128	InputPattern_USPREP_WORK	☐	VarChar		88		Yes	
129	OutboundPattern_USPREP_WORK	☐	VarChar		88		Yes	
130	UserOverrideFlag_USPREP_WORK	☐	VarChar		2		Yes	
131	CustomFlag_USPREP_WORK	☐	VarChar		2		Yes	
132	HouseNumber_USADDR_WORK	☐	VarChar		10		Yes	
133	HouseNumberSuffix_USADDR_WORK	☐	VarChar		10		Yes	
134	StreetPrefixDirectional_USADDR_WORK	☐	VarChar		3		Yes	
135	StreetPrefixType_USADDR_WORK	☐	VarChar		20		Yes	

Figure 2-210 Create j12_JOIN_NAB_WORK_AND_HOME 5/12



NAB_CONTACT_INFO_HOME_US_CASS_USPREP_STAN_NAME_WORK - Data Set

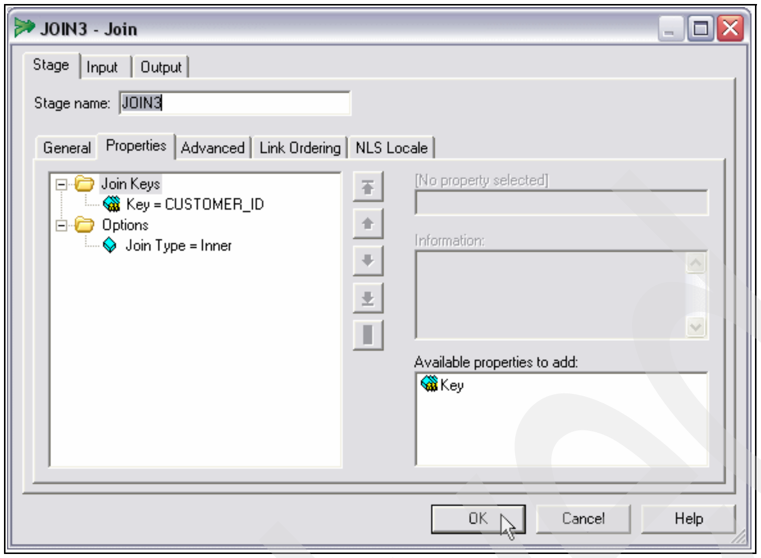
Stage | Input

Input name: IN

General | Properties | Partitioning | Columns | Advanced

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	Description
128	InputPattern_USPREP_WORK	☐	VarChar		88		Yes	
129	OutboundPattern_USPREP_WORK	☐	VarChar		88		Yes	
130	UserOverrideFlag_USPREP_WORK	☐	VarChar		2		Yes	
131	CustomFlag_USPREP_WORK	☐	VarChar		2		Yes	
132	HouseNumber_USADDR_WORK	☐	VarChar		10		Yes	
133	HouseNumberSuffix_USADDR_WORK	☐	VarChar		10		Yes	
134	StreetPrefixDirectional_USADDR_WORK	☐	VarChar		3		Yes	
135	StreetPrefixType_USADDR_WORK	☐	VarChar		20		Yes	
136	StreetName_USADDR_WORK	☐	VarChar		25		Yes	
137	StreetSuffixType_USADDR_WORK	☐	VarChar		5		Yes	
138	StreetSuffixQualifier_USADDR_WORK	☐	VarChar		5		Yes	
139	StreetSuffixDirectional_USADDR_WORK	☐	VarChar		3		Yes	
140	RuralRouteType_USADDR_WORK	☐	VarChar		3		Yes	
141	RuralRouteValue_USADDR_WORK	☐	VarChar		10		Yes	
142	BoxType_USADDR_WORK	☐	VarChar		7		Yes	
143	BoxValue_USADDR_WORK	☐	VarChar		10		Yes	
144	FloorType_USADDR_WORK	☐	VarChar		5		Yes	
145	FloorValue_USADDR_WORK	☐	VarChar		10		Yes	
146	UnitType_USADDR_WORK	☐	VarChar		5		Yes	
147	UnitValue_USADDR_WORK	☐	VarChar		10		Yes	
148	MultiUnitType_USADDR_WORK	☐	VarChar		5		Yes	
149	MultiUnitValue_USADDR_WORK	☐	VarChar		10		Yes	
150	BuildingName_USADDR_WORK	☐	VarChar		30		Yes	
151	AdditionalAddress_USADDR_WORK	☐	VarChar		50		Yes	
152	AddressType_USADDR_WORK	☐	VarChar		1		Yes	
153	StreetNameNYSIIS_USADDR_WORK	☐	VarChar		8		Yes	
154	StreetNameRVSNDX_USADDR_WORK	☐	VarChar		4		Yes	
155	UnhandledPattern_USADDR_WORK	☐	VarChar		30		Yes	
156	UnhandledData_USADDR_WORK	☐	VarChar		50		Yes	
157	InputPattern_USADDR_WORK	☐	VarChar		30		Yes	
158	ExceptionData_USADDR_WORK	☐	VarChar		50		Yes	
159	UserOverrideFlag_USADDR_WORK	☐	VarChar		2		Yes	
160	CityName_USAREA_WORK	☐	VarChar		30		Yes	
161	StateAbbreviation_USAREA_WORK	☐	VarChar		3		Yes	
162	ZipCode_USAREA_WORK	☐	VarChar		5		Yes	
163	Zip4AddonCode_USAREA_WORK	☐	VarChar		4		Yes	
164	CountryCode_USAREA_WORK	☐	VarChar		2		Yes	
165	CityNameNYSIIS_USAREA_WORK	☐	VarChar		8		Yes	
166	CityNameRVSNDX_USAREA_WORK	☐	VarChar		4		Yes	
167	UnhandledPattern_USAREA_WORK	☐	VarChar		30		Yes	
168	UnhandledData_USAREA_WORK	☐	VarChar		50		Yes	
169	InputPattern_USAREA_WORK	☐	VarChar		30		Yes	
170	ExceptionData_USAREA_WORK	☐	VarChar		50		Yes	
171	UserOverrideFlag_USAREA_WORK	☐	VarChar		2		Yes	
		☐						

Figure 2-211   Create j12_JOIN_NAB_WORK_AND_HOME 6/12


JOIN3 - Join

Stage name: JOIN3

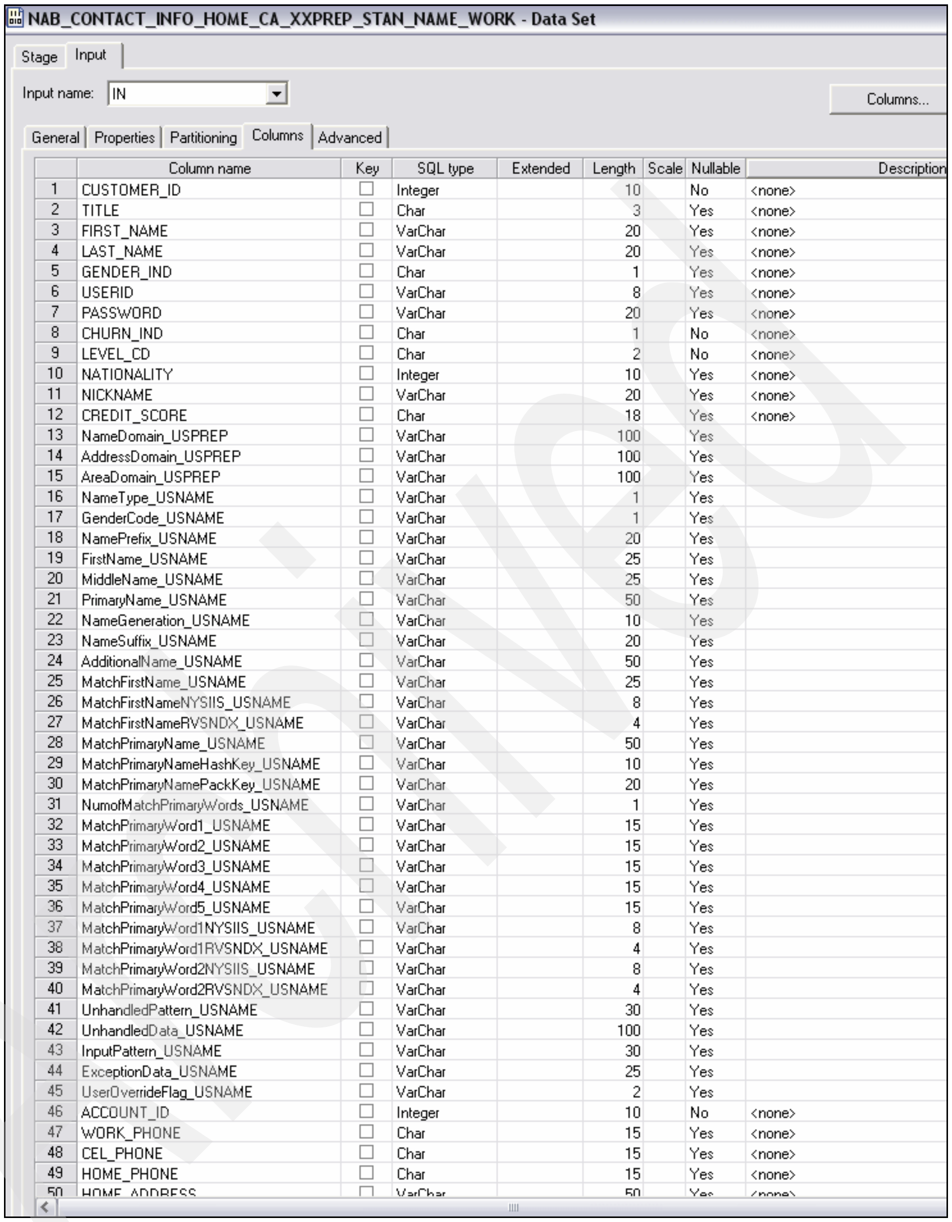
Join Keys
Key = CUSTOMER_ID
Options
Join Type = Inner

[No property selected]

Information:

Available properties to add:
Key

Figure 2-212 Create j12_JOIN_NAB_WORK_AND_HOME 7/12


NAB_CONTACT_INFO_HOME_CA_XXPREP_STAN_NAME_WORK - Data Set

Stage | Input

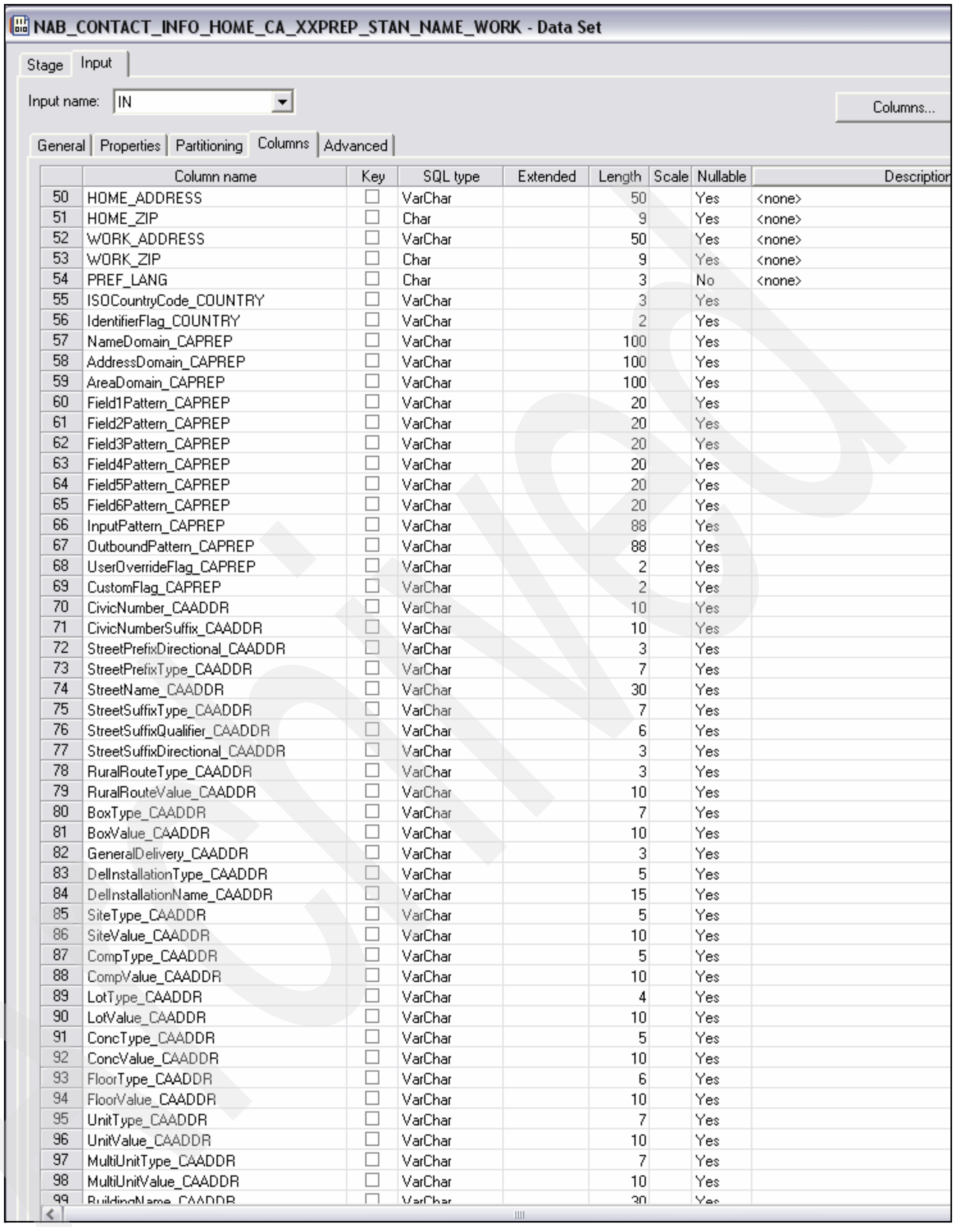
Input name: IN

Columns...

General | Properties | Partitioning | Columns | Advanced

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	Description
1	CUSTOMER_ID	☐	Integer		10		No	<none>
2	TITLE	☐	Char		3		Yes	<none>
3	FIRST_NAME	☐	VarChar		20		Yes	<none>
4	LAST_NAME	☐	VarChar		20		Yes	<none>
5	GENDER_IND	☐	Char		1		Yes	<none>
6	USERID	☐	VarChar		8		Yes	<none>
7	PASSWORD	☐	VarChar		20		Yes	<none>
8	CHURN_IND	☐	Char		1		No	<none>
9	LEVEL_CD	☐	Char		2		No	<none>
10	NATIONALITY	☐	Integer		10		Yes	<none>
11	NICKNAME	☐	VarChar		20		Yes	<none>
12	CREDIT_SCORE	☐	Char		18		Yes	<none>
13	NameDomain_USPREP	☐	VarChar		100		Yes	
14	AddressDomain_USPREP	☐	VarChar		100		Yes	
15	AreaDomain_USPREP	☐	VarChar		100		Yes	
16	NameType_USNAME	☐	VarChar		1		Yes	
17	GenderCode_USNAME	☐	VarChar		1		Yes	
18	NamePrefix_USNAME	☐	VarChar		20		Yes	
19	FirstName_USNAME	☐	VarChar		25		Yes	
20	MiddleName_USNAME	☐	VarChar		25		Yes	
21	PrimaryName_USNAME	☐	VarChar		50		Yes	
22	NameGeneration_USNAME	☐	VarChar		10		Yes	
23	NameSuffix_USNAME	☐	VarChar		20		Yes	
24	AdditionalName_USNAME	☐	VarChar		50		Yes	
25	MatchFirstName_USNAME	☐	VarChar		25		Yes	
26	MatchFirstNameNYSIIS_USNAME	☐	VarChar		8		Yes	
27	MatchFirstNameRVSNDX_USNAME	☐	VarChar		4		Yes	
28	MatchPrimaryName_USNAME	☐	VarChar		50		Yes	
29	MatchPrimaryNameHashKey_USNAME	☐	VarChar		10		Yes	
30	MatchPrimaryNamePackKey_USNAME	☐	VarChar		20		Yes	
31	NumofMatchPrimaryWords_USNAME	☐	VarChar		1		Yes	
32	MatchPrimaryWord1_USNAME	☐	VarChar		15		Yes	
33	MatchPrimaryWord2_USNAME	☐	VarChar		15		Yes	
34	MatchPrimaryWord3_USNAME	☐	VarChar		15		Yes	
35	MatchPrimaryWord4_USNAME	☐	VarChar		15		Yes	
36	MatchPrimaryWord5_USNAME	☐	VarChar		15		Yes	
37	MatchPrimaryWord1NYSIIS_USNAME	☐	VarChar		8		Yes	
38	MatchPrimaryWord1RVSNDX_USNAME	☐	VarChar		4		Yes	
39	MatchPrimaryWord2NYSIIS_USNAME	☐	VarChar		8		Yes	
40	MatchPrimaryWord2RVSNDX_USNAME	☐	VarChar		4		Yes	
41	UnhandledPattern_USNAME	☐	VarChar		30		Yes	
42	UnhandledData_USNAME	☐	VarChar		100		Yes	
43	InputPattern_USNAME	☐	VarChar		30		Yes	
44	ExceptionData_USNAME	☐	VarChar		25		Yes	
45	UserOverrideFlag_USNAME	☐	VarChar		2		Yes	
46	ACCOUNT_ID	☐	Integer		10		No	<none>
47	WORK_PHONE	☐	Char		15		Yes	<none>
48	CEL_PHONE	☐	Char		15		Yes	<none>
49	HOME_PHONE	☐	Char		15		Yes	<none>
50	HOME_ADDRESS	☐	VarChar		50		Yes	<none>

Figure 2-213 Create j12_JOIN_NAB_WORK_AND_HOME 8/12



NAB_CONTACT_INFO_HOME_CA_XXPREP_STAN_NAME_WORK - Data Set

Stage | Input

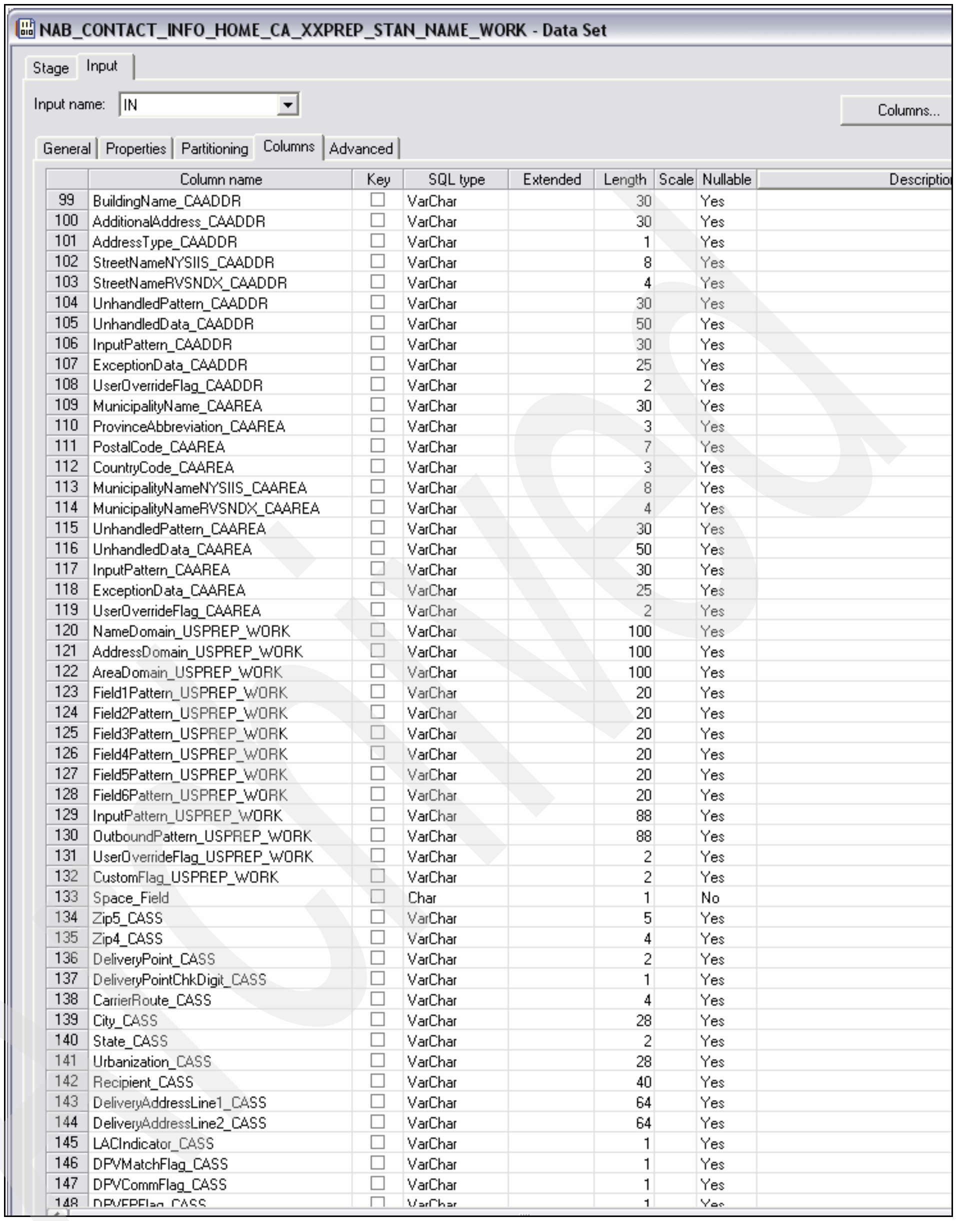
Input name: IN

Columns...

General | Properties | Partitioning | Columns | Advanced

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	Description
50	HOME_ADDRESS	☐	VarChar		50		Yes	<none>
51	HOME_ZIP	☐	Char		9		Yes	<none>
52	WORK_ADDRESS	☐	VarChar		50		Yes	<none>
53	WORK_ZIP	☐	Char		9		Yes	<none>
54	PREF_LANG	☐	Char		3		No	<none>
55	ISOCountryCode_COUNTRY	☐	VarChar		3		Yes	
56	IdentifierFlag_COUNTRY	☐	VarChar		2		Yes	
57	NameDomain_CAPREP	☐	VarChar		100		Yes	
58	AddressDomain_CAPREP	☐	VarChar		100		Yes	
59	AreaDomain_CAPREP	☐	VarChar		100		Yes	
60	Field1Pattern_CAPREP	☐	VarChar		20		Yes	
61	Field2Pattern_CAPREP	☐	VarChar		20		Yes	
62	Field3Pattern_CAPREP	☐	VarChar		20		Yes	
63	Field4Pattern_CAPREP	☐	VarChar		20		Yes	
64	Field5Pattern_CAPREP	☐	VarChar		20		Yes	
65	Field6Pattern_CAPREP	☐	VarChar		20		Yes	
66	InputPattern_CAPREP	☐	VarChar		88		Yes	
67	OutboundPattern_CAPREP	☐	VarChar		88		Yes	
68	UserOverrideFlag_CAPREP	☐	VarChar		2		Yes	
69	CustomFlag_CAPREP	☐	VarChar		2		Yes	
70	CivicNumber_CAADDR	☐	VarChar		10		Yes	
71	CivicNumberSuffix_CAADDR	☐	VarChar		10		Yes	
72	StreetPrefixDirectional_CAADDR	☐	VarChar		3		Yes	
73	StreetPrefixType_CAADDR	☐	VarChar		7		Yes	
74	StreetName_CAADDR	☐	VarChar		30		Yes	
75	StreetSuffixType_CAADDR	☐	VarChar		7		Yes	
76	StreetSuffixQualifier_CAADDR	☐	VarChar		6		Yes	
77	StreetSuffixDirectional_CAADDR	☐	VarChar		3		Yes	
78	RuralRouteType_CAADDR	☐	VarChar		3		Yes	
79	RuralRouteValue_CAADDR	☐	VarChar		10		Yes	
80	BoxType_CAADDR	☐	VarChar		7		Yes	
81	BoxValue_CAADDR	☐	VarChar		10		Yes	
82	GeneralDelivery_CAADDR	☐	VarChar		3		Yes	
83	DelInstallationType_CAADDR	☐	VarChar		5		Yes	
84	DelInstallationName_CAADDR	☐	VarChar		15		Yes	
85	SiteType_CAADDR	☐	VarChar		5		Yes	
86	SiteValue_CAADDR	☐	VarChar		10		Yes	
87	CompType_CAADDR	☐	VarChar		5		Yes	
88	CompValue_CAADDR	☐	VarChar		10		Yes	
89	LotType_CAADDR	☐	VarChar		4		Yes	
90	LotValue_CAADDR	☐	VarChar		10		Yes	
91	ConcType_CAADDR	☐	VarChar		5		Yes	
92	ConcValue_CAADDR	☐	VarChar		10		Yes	
93	FloorType_CAADDR	☐	VarChar		6		Yes	
94	FloorValue_CAADDR	☐	VarChar		10		Yes	
95	UnitType_CAADDR	☐	VarChar		7		Yes	
96	UnitValue_CAADDR	☐	VarChar		10		Yes	
97	MultiUnitType_CAADDR	☐	VarChar		7		Yes	
98	MultiUnitValue_CAADDR	☐	VarChar		10		Yes	
99	BuildingName_CAADDR	☐	VarChar		30		Yes	

Figure 2-214 Create j12_JOIN_NAB_WORK_AND_HOME 9/12

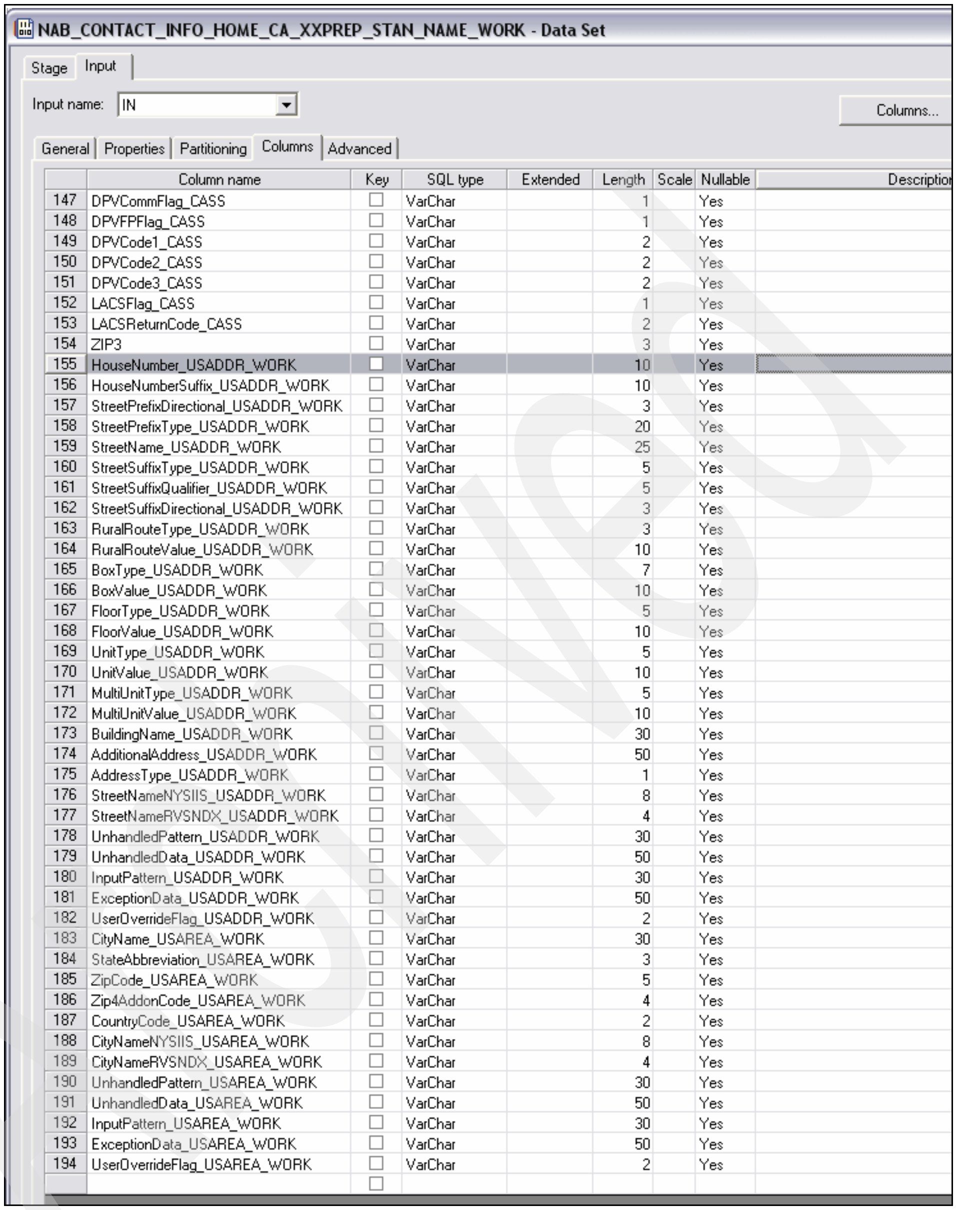


NAB_CONTACT_INFO_HOME_CA_XXPREP_STAN_NAME_WORK - Data Set

Input name: IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	Descriptio
99	BuildingName_CAADDR	☐	VarChar		30		Yes	
100	AdditionalAddress_CAADDR	☐	VarChar		30		Yes	
101	AddressType_CAADDR	☐	VarChar		1		Yes	
102	StreetNameNYSIIS_CAADDR	☐	VarChar		8		Yes	
103	StreetNameRVSNDX_CAADDR	☐	VarChar		4		Yes	
104	UnhandledPattern_CAADDR	☐	VarChar		30		Yes	
105	UnhandledData_CAADDR	☐	VarChar		50		Yes	
106	InputPattern_CAADDR	☐	VarChar		30		Yes	
107	ExceptionData_CAADDR	☐	VarChar		25		Yes	
108	UserOverrideFlag_CAADDR	☐	VarChar		2		Yes	
109	MunicipalityName_CAAREA	☐	VarChar		30		Yes	
110	ProvinceAbbreviation_CAAREA	☐	VarChar		3		Yes	
111	PostalCode_CAAREA	☐	VarChar		7		Yes	
112	CountryCode_CAAREA	☐	VarChar		3		Yes	
113	MunicipalityNameNYSIIS_CAAREA	☐	VarChar		8		Yes	
114	MunicipalityNameRVSNDX_CAAREA	☐	VarChar		4		Yes	
115	UnhandledPattern_CAAREA	☐	VarChar		30		Yes	
116	UnhandledData_CAAREA	☐	VarChar		50		Yes	
117	InputPattern_CAAREA	☐	VarChar		30		Yes	
118	ExceptionData_CAAREA	☐	VarChar		25		Yes	
119	UserOverrideFlag_CAAREA	☐	VarChar		2		Yes	
120	NameDomain_USPREP_WORK	☐	VarChar		100		Yes	
121	AddressDomain_USPREP_WORK	☐	VarChar		100		Yes	
122	AreaDomain_USPREP_WORK	☐	VarChar		100		Yes	
123	Field1Pattern_USPREP_WORK	☐	VarChar		20		Yes	
124	Field2Pattern_USPREP_WORK	☐	VarChar		20		Yes	
125	Field3Pattern_USPREP_WORK	☐	VarChar		20		Yes	
126	Field4Pattern_USPREP_WORK	☐	VarChar		20		Yes	
127	Field5Pattern_USPREP_WORK	☐	VarChar		20		Yes	
128	Field6Pattern_USPREP_WORK	☐	VarChar		20		Yes	
129	InputPattern_USPREP_WORK	☐	VarChar		88		Yes	
130	OutboundPattern_USPREP_WORK	☐	VarChar		88		Yes	
131	UserOverrideFlag_USPREP_WORK	☐	VarChar		2		Yes	
132	CustomFlag_USPREP_WORK	☐	VarChar		2		Yes	
133	Space_Field	☐	Char		1		No	
134	Zip5_CASS	☐	VarChar		5		Yes	
135	Zip4_CASS	☐	VarChar		4		Yes	
136	DeliveryPoint_CASS	☐	VarChar		2		Yes	
137	DeliveryPointChkDigit_CASS	☐	VarChar		1		Yes	
138	CarrierRoute_CASS	☐	VarChar		4		Yes	
139	City_CASS	☐	VarChar		28		Yes	
140	State_CASS	☐	VarChar		2		Yes	
141	Urbanization_CASS	☐	VarChar		28		Yes	
142	Recipient_CASS	☐	VarChar		40		Yes	
143	DeliveryAddressLine1_CASS	☐	VarChar		64		Yes	
144	DeliveryAddressLine2_CASS	☐	VarChar		64		Yes	
145	LACIndicator_CASS	☐	VarChar		1		Yes	
146	DPVMatchFlag_CASS	☐	VarChar		1		Yes	
147	DPVCommFlag_CASS	☐	VarChar		1		Yes	
148	DPVFPFlag_CASS	☐	VarChar		1		Yes	

Figure 2-215 Create j12_JOIN_NAB_WORK_AND_HOME 10/12



NAB_CONTACT_INFO_HOME_CA_XXPREP_STAN_NAME_WORK - Data Set

Stage | Input

Input name: IN

Columns...

General | Properties | Partitioning | Columns | Advanced

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	Description
147	DPVCommFlag_CASS	☐	VarChar		1		Yes	
148	DPVFPFlag_CASS	☐	VarChar		1		Yes	
149	DPVCode1_CASS	☐	VarChar		2		Yes	
150	DPVCode2_CASS	☐	VarChar		2		Yes	
151	DPVCode3_CASS	☐	VarChar		2		Yes	
152	LACSFlag_CASS	☐	VarChar		1		Yes	
153	LACSReturnCode_CASS	☐	VarChar		2		Yes	
154	ZIP3	☐	VarChar		3		Yes	
155	HouseNumber_USADDR_WORK	☐	VarChar		10		Yes	
156	HouseNumberSuffix_USADDR_WORK	☐	VarChar		10		Yes	
157	StreetPrefixDirectional_USADDR_WORK	☐	VarChar		3		Yes	
158	StreetPrefixType_USADDR_WORK	☐	VarChar		20		Yes	
159	StreetName_USADDR_WORK	☐	VarChar		25		Yes	
160	StreetSuffixType_USADDR_WORK	☐	VarChar		5		Yes	
161	StreetSuffixQualifier_USADDR_WORK	☐	VarChar		5		Yes	
162	StreetSuffixDirectional_USADDR_WORK	☐	VarChar		3		Yes	
163	RuralRouteType_USADDR_WORK	☐	VarChar		3		Yes	
164	RuralRouteValue_USADDR_WORK	☐	VarChar		10		Yes	
165	BoxType_USADDR_WORK	☐	VarChar		7		Yes	
166	BoxValue_USADDR_WORK	☐	VarChar		10		Yes	
167	FloorType_USADDR_WORK	☐	VarChar		5		Yes	
168	FloorValue_USADDR_WORK	☐	VarChar		10		Yes	
169	UnitType_USADDR_WORK	☐	VarChar		5		Yes	
170	UnitValue_USADDR_WORK	☐	VarChar		10		Yes	
171	MultiUnitType_USADDR_WORK	☐	VarChar		5		Yes	
172	MultiUnitValue_USADDR_WORK	☐	VarChar		10		Yes	
173	BuildingName_USADDR_WORK	☐	VarChar		30		Yes	
174	AdditionalAddress_USADDR_WORK	☐	VarChar		50		Yes	
175	AddressType_USADDR_WORK	☐	VarChar		1		Yes	
176	StreetNameNYSIIS_USADDR_WORK	☐	VarChar		8		Yes	
177	StreetNameRVSNDX_USADDR_WORK	☐	VarChar		4		Yes	
178	UnhandledPattern_USADDR_WORK	☐	VarChar		30		Yes	
179	UnhandledData_USADDR_WORK	☐	VarChar		50		Yes	
180	InputPattern_USADDR_WORK	☐	VarChar		30		Yes	
181	ExceptionData_USADDR_WORK	☐	VarChar		50		Yes	
182	UserOverrideFlag_USADDR_WORK	☐	VarChar		2		Yes	
183	CityName_USAREA_WORK	☐	VarChar		30		Yes	
184	StateAbbreviation_USAREA_WORK	☐	VarChar		3		Yes	
185	ZipCode_USAREA_WORK	☐	VarChar		5		Yes	
186	Zip4AddonCode_USAREA_WORK	☐	VarChar		4		Yes	
187	CountryCode_USAREA_WORK	☐	VarChar		2		Yes	
188	CityNameNYSIIS_USAREA_WORK	☐	VarChar		8		Yes	
189	CityNameRVSNDX_USAREA_WORK	☐	VarChar		4		Yes	
190	UnhandledPattern_USAREA_WORK	☐	VarChar		30		Yes	
191	UnhandledData_USAREA_WORK	☐	VarChar		50		Yes	
192	InputPattern_USAREA_WORK	☐	VarChar		30		Yes	
193	ExceptionData_USAREA_WORK	☐	VarChar		50		Yes	
194	UserOverrideFlag_USAREA_WORK	☐	VarChar		2		Yes	
		☐						

Figure 2-216 Create j12_JOIN_NAB_WORK_AND_HOME 11/12

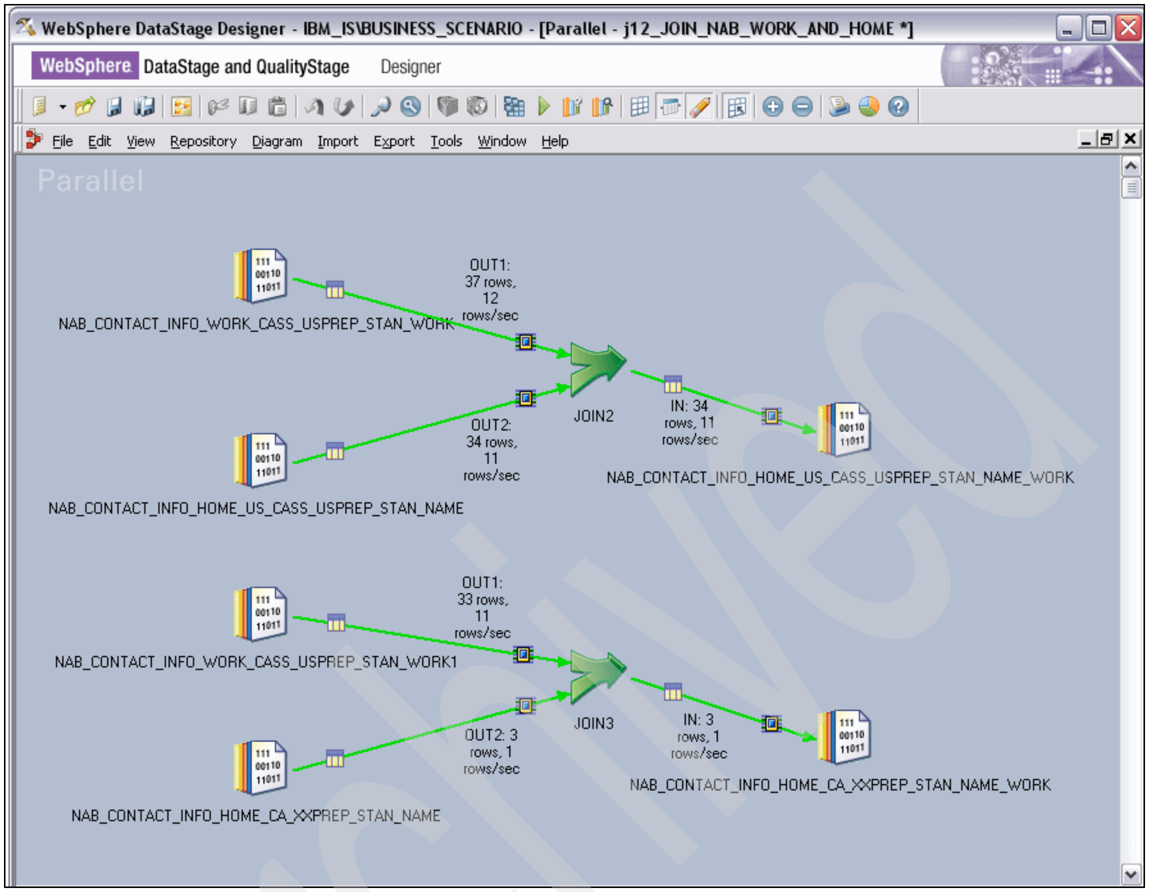

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j12_JOIN_NAB_WORK_AND_HOME *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NAB_CONTACT_INFO_WORK_CASS_USPREP_STAN_WORK
OUT1: 37 rows, 12 rows/sec
NAB_CONTACT_INFO_HOME_US_CASS_USPREP_STAN_NAME
OUT2: 34 rows, 11 rows/sec
JOIN2
IN: 34 rows, 11 rows/sec
NAB_CONTACT_INFO_HOME_US_CASS_USPREP_STAN_NAME_WORK
NAB_CONTACT_INFO_WORK_CASS_USPREP_STAN_WORK1
OUT1: 33 rows, 11 rows/sec
NAB_CONTACT_INFO_HOME_CA_XXPREP_STAN_NAME
OUT2: 3 rows, 1 rows/sec
JOIN3
IN: 3 rows, 1 rows/sec
NAB_CONTACT_INFO_HOME_CA_XXPREP_STAN_NAME_WORK

Figure 2-217 Create j12_JOIN_NAB_WORK_AND_HOME 12/12

j13_PREPARE_NAB_DATA_FOR_FUNNEL

At this point, we have two reconstructed files with standardized names and home and work addresses—one containing U.S. home addresses and the other containing Canadian home addresses. We need to merge these files with the standardized name and address file of the DRIVER in order to identify duplicates (with data from the Northern California Bank's standardized name and addresses from their core and non-core services as described in 2.6.2, "Cleansing Northern California Bank's core and non-core services" on page 706) and to survive the "best" data into the CRM system as described in 2.6.3, "Matching and surviving Northern California Bank and Northern California Bank information" on page 783.

Prior to merging the two files that were created in "j12_JOIN_NAB_WORK_AND_HOME" on page 674 with the standardized name and address DRIVER file that was created in “j09_STAN_CASS” on page 631, we need to prepare the files for merging in the Funnel stage by ensuring that all the files have the same number of columns and same names of columns using a Transformer stage.

Figure 2-218 on page 689 shows the various stages that are used in this job. It includes the data sets that were created in “j12_JOIN_NAB_WORK_AND_HOME” on page 674 and “j09_STAN_CASS” on page 631, a Transformer stage, and an output Data Set stage for each Transformer stage. We modified the names of the stages as shown.

Figure 2-219 on page 690 through Figure 2-223 on page 694 show the PICK_COLUMNS2 - Transformer windows that pick the appropriate columns from the customer name, U.S. home addresses, and the U.S. work addresses file that was created in “j12_JOIN_NAB_WORK_AND_HOME” on page 674. Certain columns are renamed and replaced with an empty string as highlighted. Also, additional columns (that exist in the other files to be merged with such as ZIP3_WORK, NCB_CORE_ID and NCB_NON_CORE_ID) are added with empty string or other values as highlighted.

Figure 2-224 on page 695 through Figure 2-228 on page 699 show the corresponding PICK_COLUMNS3 - Transformer windows that pick the appropriate columns from the customer name, Canadian home addresses, and the U.S. work addresses file that was created in “j12_JOIN_NAB_WORK_AND_HOME” on page 674.

Figure 2-229 on page 700 through Figure 2-233 on page 704 show the corresponding PICK_COLUMNS4 - Transformer windows that pick the appropriate columns from the DRIVER name and address file that was created in “j09_STAN_CASS” on page 631.

Figure 2-234 on page 705 shows the results after compiling and running this job.

Proceed now to “j14_FUNNEL_NAB_DATA_FOR_CRM” on page 705.

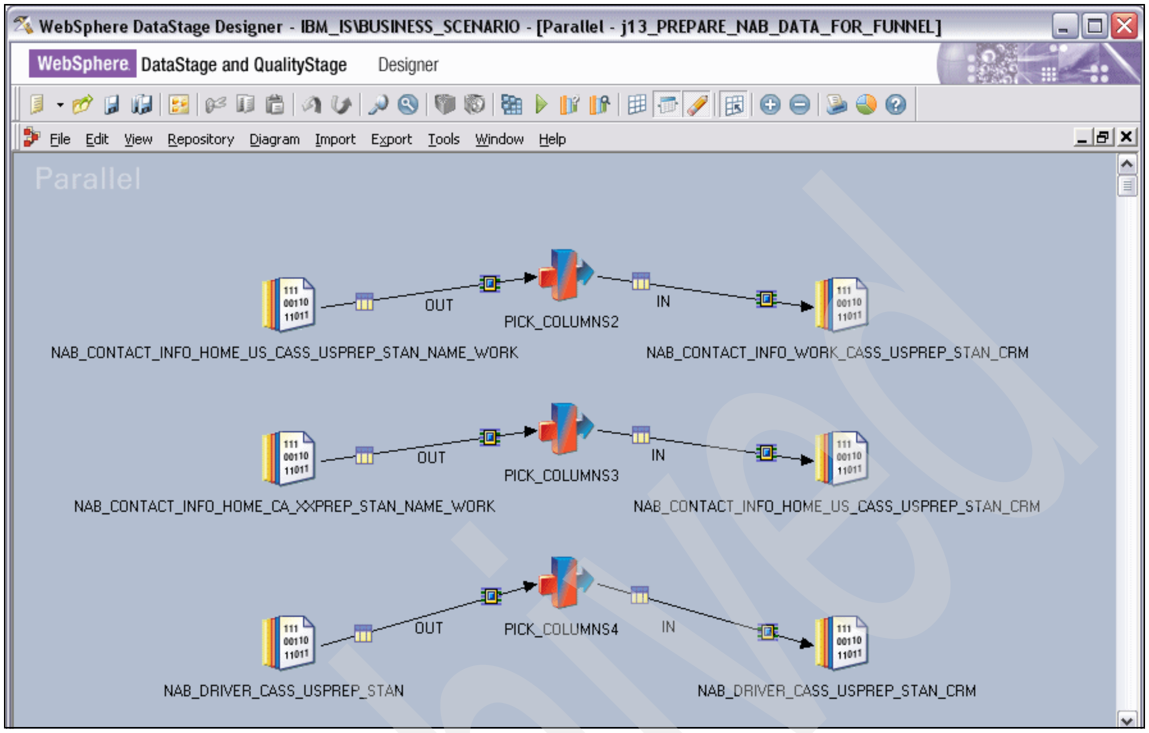
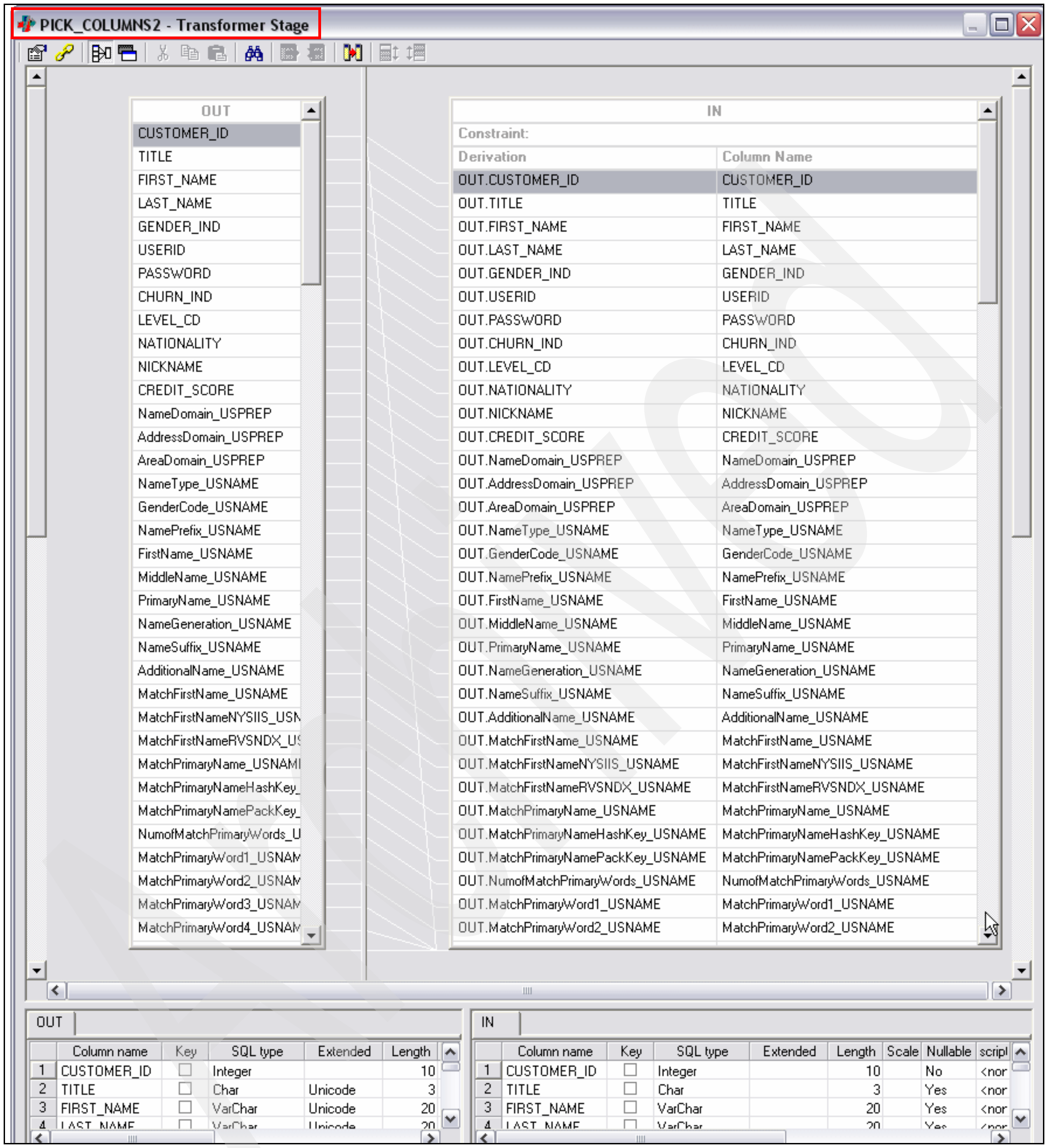

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j13_PREPARE_NAB_DATA_FOR_FUNNEL]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
OUT
PICK_COLUMNS2
IN
NAB_CONTACT_INFO_HOME_US_CASS_USPREP_STAN_NAME_WORK
NAB_CONTACT_INFO_WORK_CASS_USPREP_STAN_CRM
OUT
PICK_COLUMNS3
IN
NAB_CONTACT_INFO_HOME_CA_XXPREP_STAN_NAME_WORK
NAB_CONTACT_INFO_HOME_US_CASS_USPREP_STAN_CRM
OUT
PICK_COLUMNS4
IN
NAB_DRIVER_CASS_USPREP_STAN
NAB_DRIVER_CASS_USPREP_STAN_CRM

Figure 2-218 Create j13_PREPARE_NAB_DATA_FOR_FUNNEL 1/17



PICK_COLUMNS2 - Transformer Stage

OUT
CUSTOMER_ID
TITLE
FIRST_NAME
LAST_NAME
GENDER_IND
USERID
PASSWORD
CHURN_IND
LEVEL_CD
NATIONALITY
NICKNAME
CREDIT_SCORE
NameDomain_USPREP
AddressDomain_USPREP
AreaDomain_USPREP
NameType_USNAME
GenderCode_USNAME
NamePrefix_USNAME
FirstName_USNAME
MiddleName_USNAME
PrimaryName_USNAME
NameGeneration_USNAME
NameSuffix_USNAME
AdditionalName_USNAME
MatchFirstName_USNAME
MatchFirstNameNYSIIS_USN
MatchFirstNameRVSNDX_US
MatchPrimaryName_USNAMI
MatchPrimaryNameHashKey_
MatchPrimaryNamePackKey_
NumofMatchPrimaryWords_U
MatchPrimaryWord1_USNAM
MatchPrimaryWord2_USNAM
MatchPrimaryWord3_USNAM
MatchPrimaryWord4_USNAM

IN

Constraint:

Derivation	Column Name
OUT.CUSTOMER_ID	CUSTOMER_ID
OUT.TITLE	TITLE
OUT.FIRST_NAME	FIRST_NAME
OUT.LAST_NAME	LAST_NAME
OUT.GENDER_IND	GENDER_IND
OUT.USERID	USERID
OUT.PASSWORD	PASSWORD
OUT.CHURN_IND	CHURN_IND
OUT.LEVEL_CD	LEVEL_CD
OUT.NATIONALITY	NATIONALITY
OUT.NICKNAME	NICKNAME
OUT.CREDIT_SCORE	CREDIT_SCORE
OUT.NameDomain_USPREP	NameDomain_USPREP
OUT.AddressDomain_USPREP	AddressDomain_USPREP
OUT.AreaDomain_USPREP	AreaDomain_USPREP
OUT.NameType_USNAME	NameType_USNAME
OUT.GenderCode_USNAME	GenderCode_USNAME
OUT.NamePrefix_USNAME	NamePrefix_USNAME
OUT.FirstName_USNAME	FirstName_USNAME
OUT.MiddleName_USNAME	MiddleName_USNAME
OUT.PrimaryName_USNAME	PrimaryName_USNAME
OUT.NameGeneration_USNAME	NameGeneration_USNAME
OUT.NameSuffix_USNAME	NameSuffix_USNAME
OUT.AdditionalName_USNAME	AdditionalName_USNAME
OUT.MatchFirstName_USNAME	MatchFirstName_USNAME
OUT.MatchFirstNameNYSIIS_USNAME	MatchFirstNameNYSIIS_USNAME
OUT.MatchFirstNameRVSNDX_USNAME	MatchFirstNameRVSNDX_USNAME
OUT.MatchPrimaryName_USNAME	MatchPrimaryName_USNAME
OUT.MatchPrimaryNameHashKey_USNAME	MatchPrimaryNameHashKey_USNAME
OUT.MatchPrimaryNamePackKey_USNAME	MatchPrimaryNamePackKey_USNAME
OUT.NumofMatchPrimaryWords_USNAME	NumofMatchPrimaryWords_USNAME
OUT.MatchPrimaryWord1_USNAME	MatchPrimaryWord1_USNAME
OUT.MatchPrimaryWord2_USNAME	MatchPrimaryWord2_USNAME

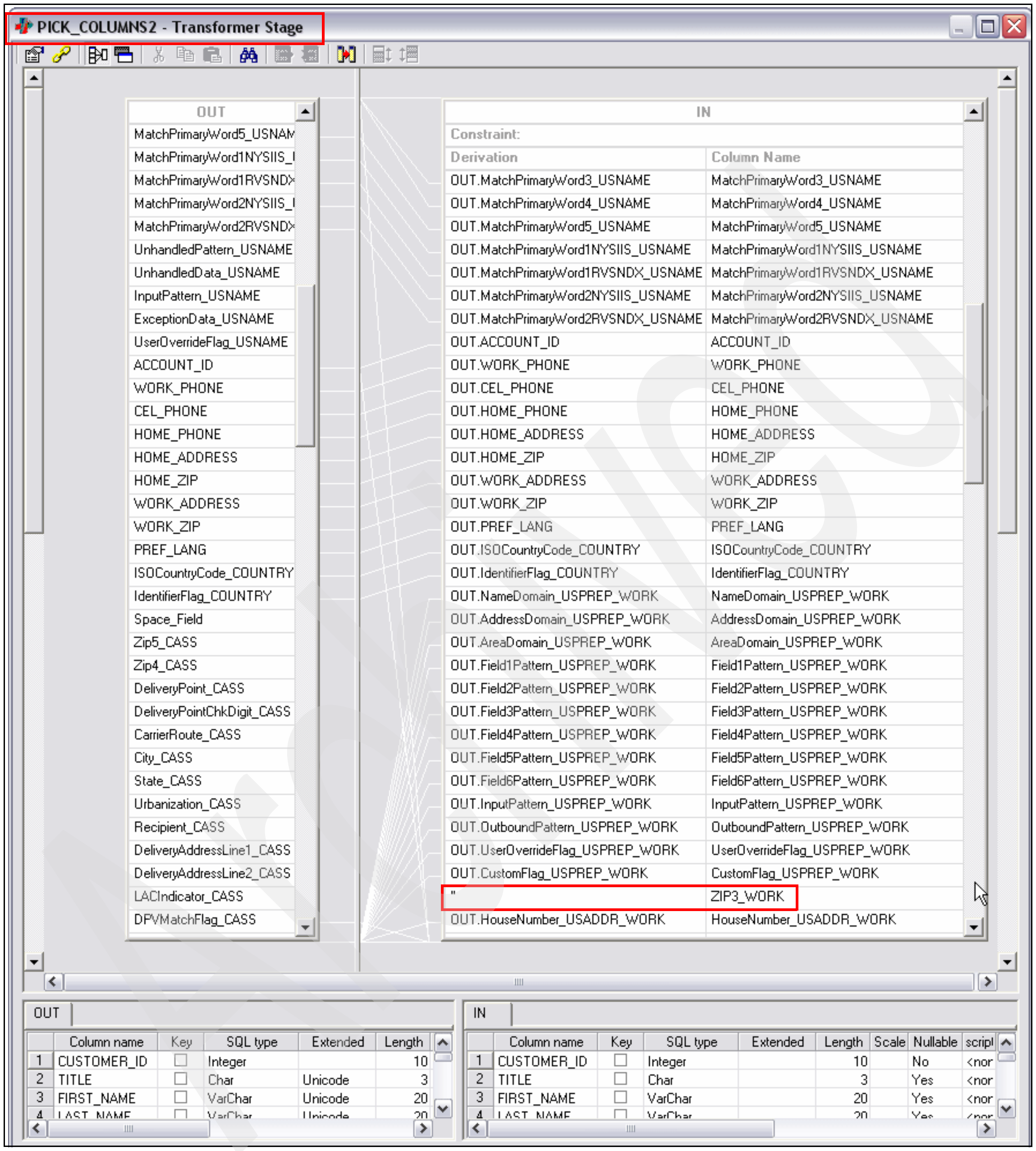
OUT

	Column name	Key	SQL type	Extended	Length
1	CUSTOMER_ID	☐	Integer		10
2	TITLE	☐	Char	Unicode	3
3	FIRST_NAME	☐	VarChar	Unicode	20
4	LAST_NAME	☐	VarChar	Unicode	20

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripl
1	CUSTOMER_ID	☐	Integer		10		No	<nor
2	TITLE	☐	Char		3		Yes	<nor
3	FIRST_NAME	☐	VarChar		20		Yes	<nor
4	LAST_NAME	☐	VarChar		20		Yes	<nor

Figure 2-219 Create j13_PREPARE_NAB_DATA_FOR_FUNNEL 2/17



PICK_COLUMNS2 - Transformer Stage

OUT
MatchPrimaryWord5_USNAM
MatchPrimaryWord1NYSIIS_I
MatchPrimaryWord1RVSNDX
MatchPrimaryWord2NYSIIS_I
MatchPrimaryWord2RVSNDX
UnhandledPattern_USNAME
UnhandledData_USNAME
InputPattern_USNAME
ExceptionData_USNAME
UserOverrideFlag_USNAME
ACCOUNT_ID
WORK_PHONE
CEL_PHONE
HOME_PHONE
HOME_ADDRESS
HOME_ZIP
WORK_ADDRESS
WORK_ZIP
PREF_LANG
ISOCountryCode_COUNTRY
IdentifierFlag_COUNTRY
Space_Field
Zip5_CASS
Zip4_CASS
DeliveryPoint_CASS
DeliveryPointChkDigit_CASS
CarrierRoute_CASS
City_CASS
State_CASS
Urbanization_CASS
Recipient_CASS
DeliveryAddressLine1_CASS
DeliveryAddressLine2_CASS
LACIndicator_CASS
DPVMatchFlag_CASS

IN

Constraint:

Derivation	Column Name
OUT.MatchPrimaryWord3_USNAME	MatchPrimaryWord3_USNAME
OUT.MatchPrimaryWord4_USNAME	MatchPrimaryWord4_USNAME
OUT.MatchPrimaryWord5_USNAME	MatchPrimaryWord5_USNAME
OUT.MatchPrimaryWord1NYSIIS_USNAME	MatchPrimaryWord1NYSIIS_USNAME
OUT.MatchPrimaryWord1RVSNDX_USNAME	MatchPrimaryWord1RVSNDX_USNAME
OUT.MatchPrimaryWord2NYSIIS_USNAME	MatchPrimaryWord2NYSIIS_USNAME
OUT.MatchPrimaryWord2RVSNDX_USNAME	MatchPrimaryWord2RVSNDX_USNAME
OUT.ACCOUNT_ID	ACCOUNT_ID
OUT.WORK_PHONE	WORK_PHONE
OUT.CEL_PHONE	CEL_PHONE
OUT.HOME_PHONE	HOME_PHONE
OUT.HOME_ADDRESS	HOME_ADDRESS
OUT.HOME_ZIP	HOME_ZIP
OUT.WORK_ADDRESS	WORK_ADDRESS
OUT.WORK_ZIP	WORK_ZIP
OUT.PREF_LANG	PREF_LANG
OUT.ISOCountryCode_COUNTRY	ISOCountryCode_COUNTRY
OUT.IdentifierFlag_COUNTRY	IdentifierFlag_COUNTRY
OUT.NameDomain_USPREP_WORK	NameDomain_USPREP_WORK
OUT.AddressDomain_USPREP_WORK	AddressDomain_USPREP_WORK
OUT.AreaDomain_USPREP_WORK	AreaDomain_USPREP_WORK
OUT.Field1Pattern_USPREP_WORK	Field1Pattern_USPREP_WORK
OUT.Field2Pattern_USPREP_WORK	Field2Pattern_USPREP_WORK
OUT.Field3Pattern_USPREP_WORK	Field3Pattern_USPREP_WORK
OUT.Field4Pattern_USPREP_WORK	Field4Pattern_USPREP_WORK
OUT.Field5Pattern_USPREP_WORK	Field5Pattern_USPREP_WORK
OUT.Field6Pattern_USPREP_WORK	Field6Pattern_USPREP_WORK
OUT.InputPattern_USPREP_WORK	InputPattern_USPREP_WORK
OUT.OutboundPattern_USPREP_WORK	OutboundPattern_USPREP_WORK
OUT.UserOverrideFlag_USPREP_WORK	UserOverrideFlag_USPREP_WORK
OUT.CustomFlag_USPREP_WORK	CustomFlag_USPREP_WORK
''	ZIP3_WORK
OUT.HouseNumber_USADDR_WORK	HouseNumber_USADDR_WORK

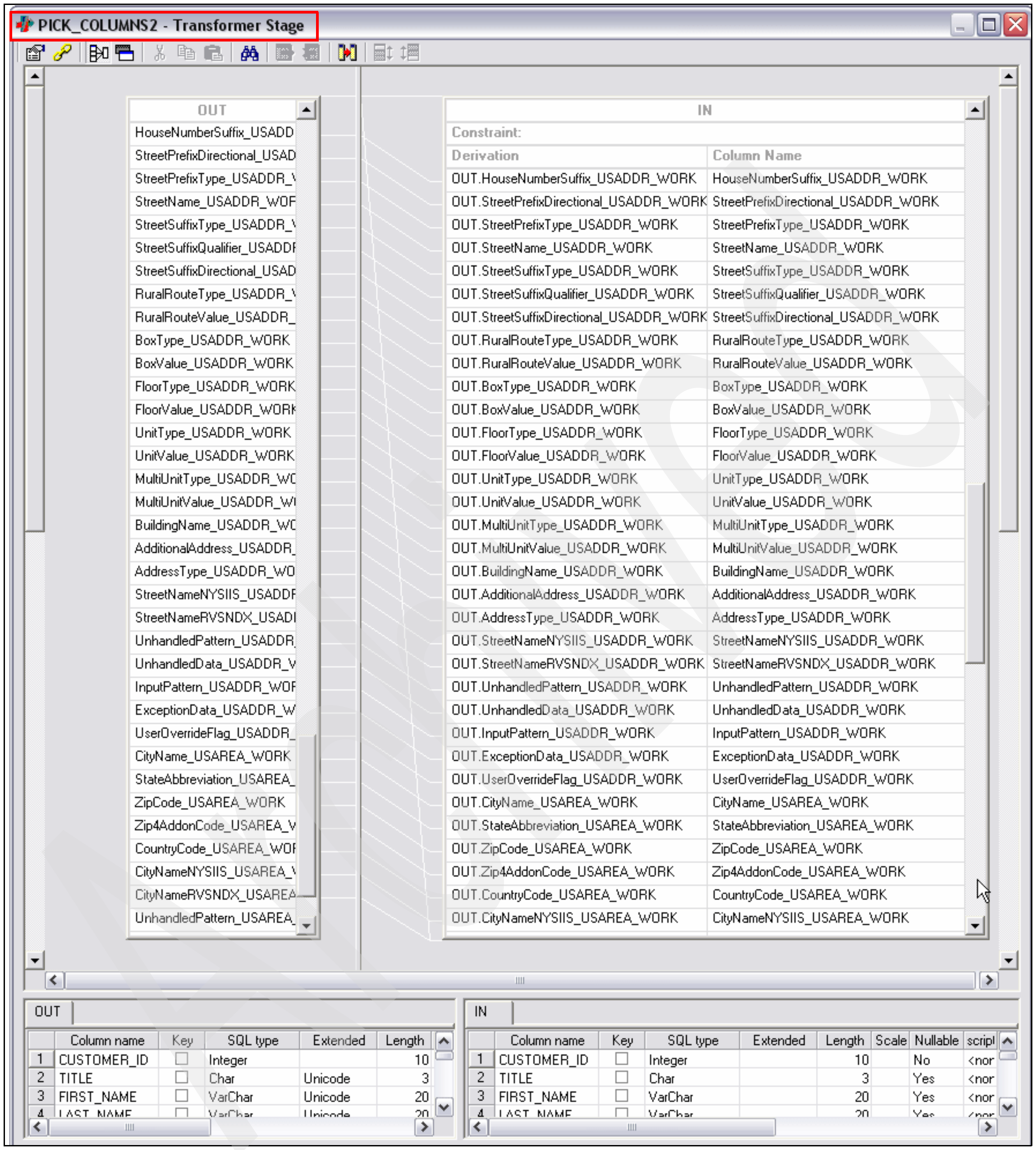
OUT

	Column name	Key	SQL type	Extended	Length
1	CUSTOMER_ID	☐	Integer		10
2	TITLE	☐	Char	Unicode	3
3	FIRST_NAME	☐	VarChar	Unicode	20
4	LAST_NAME	☐	VarChar	Unicode	20

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripl
1	CUSTOMER_ID	☐	Integer		10		No	<nor
2	TITLE	☐	Char		3		Yes	<nor
3	FIRST_NAME	☐	VarChar		20		Yes	<nor
4	LAST_NAME	☐	VarChar		20		Yes	<nor

Figure 2-220 Create j13_PREPARE_NAB_DATA_FOR_FUNNEL 3/17


PICK_COLUMNS2 - Transformer Stage

OUT
HouseNumberSuffix_USADD
StreetPrefixDirectional_USAD
StreetPrefixType_USADDR_\
StreetName_USADDR_WOF
StreetSuffixType_USADDR_\
StreetSuffixQualifier_USADDF
StreetSuffixDirectional_USAD
RuralRouteType_USADDR_\
RuralRouteValue_USADDR_
BoxType_USADDR_WORK
BoxValue_USADDR_WORK
FloorType_USADDR_WORK
FloorValue_USADDR_WORK
UnitType_USADDR_WORK
UnitValue_USADDR_WORK
MultiUnitType_USADDR_WC
MultiUnitValue_USADDR_WI
BuildingName_USADDR_WC
AdditionalAddress_USADDR_
AddressType_USADDR_WO
StreetNameNYSIIS_USADDF
StreetNameRVSNDX_USADI
UnhandledPattern_USADDR_
UnhandledData_USADDR_V
InputPattern_USADDR_WOF
ExceptionData_USADDR_W
UserOverrideFlag_USADDR_
CityName_USAREA_WORK
StateAbbreviation_USAREA_
ZipCode_USAREA_WORK
Zip4AddonCode_USAREA_V
CountryCode_USAREA_WOF
CityNameNYSIIS_USAREA_\
CityNameRVSNDX_USAREA
UnhandledPattern_USAREA_

IN

Constraint:

Derivation	Column Name
OUT.HouseNumberSuffix_USADDR_WORK	HouseNumberSuffix_USADDR_WORK
OUT.StreetPrefixDirectional_USADDR_WORK	StreetPrefixDirectional_USADDR_WORK
OUT.StreetPrefixType_USADDR_WORK	StreetPrefixType_USADDR_WORK
OUT.StreetName_USADDR_WORK	StreetName_USADDR_WORK
OUT.StreetSuffixType_USADDR_WORK	StreetSuffixType_USADDR_WORK
OUT.StreetSuffixQualifier_USADDR_WORK	StreetSuffixQualifier_USADDR_WORK
OUT.StreetSuffixDirectional_USADDR_WORK	StreetSuffixDirectional_USADDR_WORK
OUT.RuralRouteType_USADDR_WORK	RuralRouteType_USADDR_WORK
OUT.RuralRouteValue_USADDR_WORK	RuralRouteValue_USADDR_WORK
OUT.BoxType_USADDR_WORK	BoxType_USADDR_WORK
OUT.BoxValue_USADDR_WORK	BoxValue_USADDR_WORK
OUT.FloorType_USADDR_WORK	FloorType_USADDR_WORK
OUT.FloorValue_USADDR_WORK	FloorValue_USADDR_WORK
OUT.UnitType_USADDR_WORK	UnitType_USADDR_WORK
OUT.UnitValue_USADDR_WORK	UnitValue_USADDR_WORK
OUT.MultiUnitType_USADDR_WORK	MultiUnitType_USADDR_WORK
OUT.MultiUnitValue_USADDR_WORK	MultiUnitValue_USADDR_WORK
OUT.BuildingName_USADDR_WORK	BuildingName_USADDR_WORK
OUT.AdditionalAddress_USADDR_WORK	AdditionalAddress_USADDR_WORK
OUT.AddressType_USADDR_WORK	AddressType_USADDR_WORK
OUT.StreetNameNYSIIS_USADDR_WORK	StreetNameNYSIIS_USADDR_WORK
OUT.StreetNameRVSNDX_USADDR_WORK	StreetNameRVSNDX_USADDR_WORK
OUT.UnhandledPattern_USADDR_WORK	UnhandledPattern_USADDR_WORK
OUT.UnhandledData_USADDR_WORK	UnhandledData_USADDR_WORK
OUT.InputPattern_USADDR_WORK	InputPattern_USADDR_WORK
OUT.ExceptionData_USADDR_WORK	ExceptionData_USADDR_WORK
OUT.UserOverrideFlag_USADDR_WORK	UserOverrideFlag_USADDR_WORK
OUT.CityName_USAREA_WORK	CityName_USAREA_WORK
OUT.StateAbbreviation_USAREA_WORK	StateAbbreviation_USAREA_WORK
OUT.ZipCode_USAREA_WORK	ZipCode_USAREA_WORK
OUT.Zip4AddonCode_USAREA_WORK	Zip4AddonCode_USAREA_WORK
OUT.CountryCode_USAREA_WORK	CountryCode_USAREA_WORK
OUT.CityNameNYSIIS_USAREA_WORK	CityNameNYSIIS_USAREA_WORK

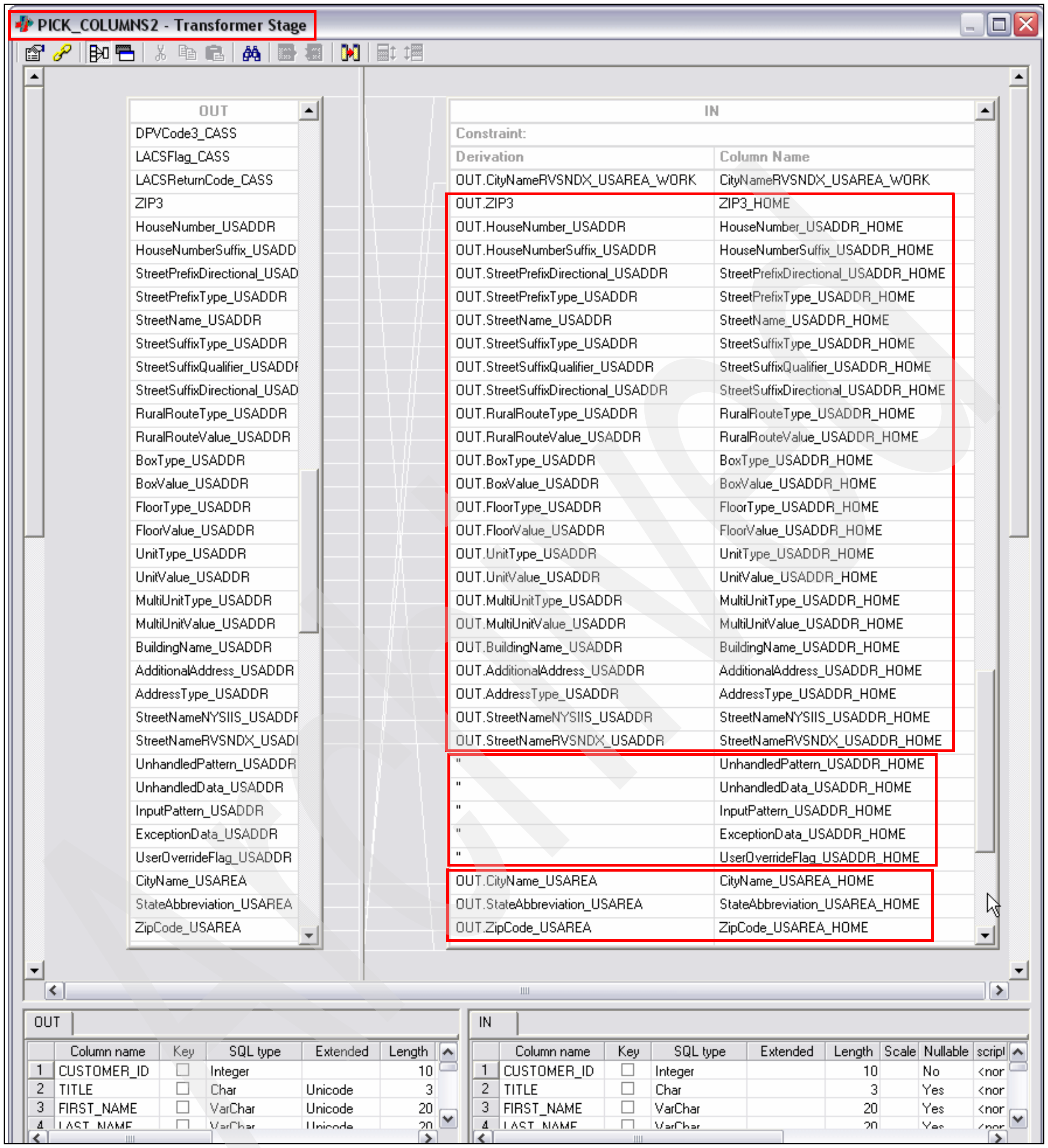
OUT

	Column name	Key	SQL type	Extended	Length
1	CUSTOMER_ID	☐	Integer		10
2	TITLE	☐	Char	Unicode	3
3	FIRST_NAME	☐	VarChar	Unicode	20
4	LAST_NAME	☐	VarChar	Unicode	20

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripl
1	CUSTOMER_ID	☐	Integer		10		No	<nor
2	TITLE	☐	Char		3		Yes	<nor
3	FIRST_NAME	☐	VarChar		20		Yes	<nor
4	LAST_NAME	☐	VarChar		20		Yes	<nor

Figure 2-221 Create j13_PREPARE_NAB_DATA_FOR_FUNNEL 4/17


PICK_COLUMNS2 - Transformer Stage

OUT
DPVCode3_CASS
LACSFlag_CASS
LACSReturnCode_CASS
ZIP3
HouseNumber_USADDR
HouseNumberSuffix_USADD
StreetPrefixDirectional_USAD
StreetPrefixType_USADDR
StreetName_USADDR
StreetSuffixType_USADDR
StreetSuffixQualifier_USADDF
StreetSuffixDirectional_USAD
RuralRouteType_USADDR
RuralRouteValue_USADDR
BoxType_USADDR
BoxValue_USADDR
FloorType_USADDR
FloorValue_USADDR
UnitType_USADDR
UnitValue_USADDR
MultiUnitType_USADDR
MultiUnitValue_USADDR
BuildingName_USADDR
AdditionalAddress_USADDR
AddressType_USADDR
StreetNameNYSIIS_USADDF
StreetNameRVSNDX_USADI
UnhandledPattern_USADDR
UnhandledData_USADDR
InputPattern_USADDR
ExceptionData_USADDR
UserOverrideFlag_USADDR
CityName_USAREA
StateAbbreviation_USAREA
ZipCode_USAREA

IN

Constraint:

Derivation	Column Name
OUT.CityNameRVSNDX_USAREA_WORK	CityNameRVSNDX_USAREA_WORK
OUT.ZIP3	ZIP3_HOME
OUT.HouseNumber_USADDR	HouseNumber_USADDR_HOME
OUT.HouseNumberSuffix_USADDR	HouseNumberSuffix_USADDR_HOME
OUT.StreetPrefixDirectional_USADDR	StreetPrefixDirectional_USADDR_HOME
OUT.StreetPrefixType_USADDR	StreetPrefixType_USADDR_HOME
OUT.StreetName_USADDR	StreetName_USADDR_HOME
OUT.StreetSuffixType_USADDR	StreetSuffixType_USADDR_HOME
OUT.StreetSuffixQualifier_USADDR	StreetSuffixQualifier_USADDR_HOME
OUT.StreetSuffixDirectional_USADDR	StreetSuffixDirectional_USADDR_HOME
OUT.RuralRouteType_USADDR	RuralRouteType_USADDR_HOME
OUT.RuralRouteValue_USADDR	RuralRouteValue_USADDR_HOME
OUT.BoxType_USADDR	BoxType_USADDR_HOME
OUT.BoxValue_USADDR	BoxValue_USADDR_HOME
OUT.FloorType_USADDR	FloorType_USADDR_HOME
OUT.FloorValue_USADDR	FloorValue_USADDR_HOME
OUT.UnitType_USADDR	UnitType_USADDR_HOME
OUT.UnitValue_USADDR	UnitValue_USADDR_HOME
OUT.MultiUnitType_USADDR	MultiUnitType_USADDR_HOME
OUT.MultiUnitValue_USADDR	MultiUnitValue_USADDR_HOME
OUT.BuildingName_USADDR	BuildingName_USADDR_HOME
OUT.AdditionalAddress_USADDR	AdditionalAddress_USADDR_HOME
OUT.AddressType_USADDR	AddressType_USADDR_HOME
OUT.StreetNameNYSIIS_USADDR	StreetNameNYSIIS_USADDR_HOME
OUT.StreetNameRVSNDX_USADDR	StreetNameRVSNDX_USADDR_HOME
''	UnhandledPattern_USADDR_HOME
''	UnhandledData_USADDR_HOME
''	InputPattern_USADDR_HOME
''	ExceptionData_USADDR_HOME
''	UserOverrideFlag_USADDR_HOME
OUT.CityName_USAREA	CityName_USAREA_HOME
OUT.StateAbbreviation_USAREA	StateAbbreviation_USAREA_HOME
OUT.ZipCode_USAREA	ZipCode_USAREA_HOME

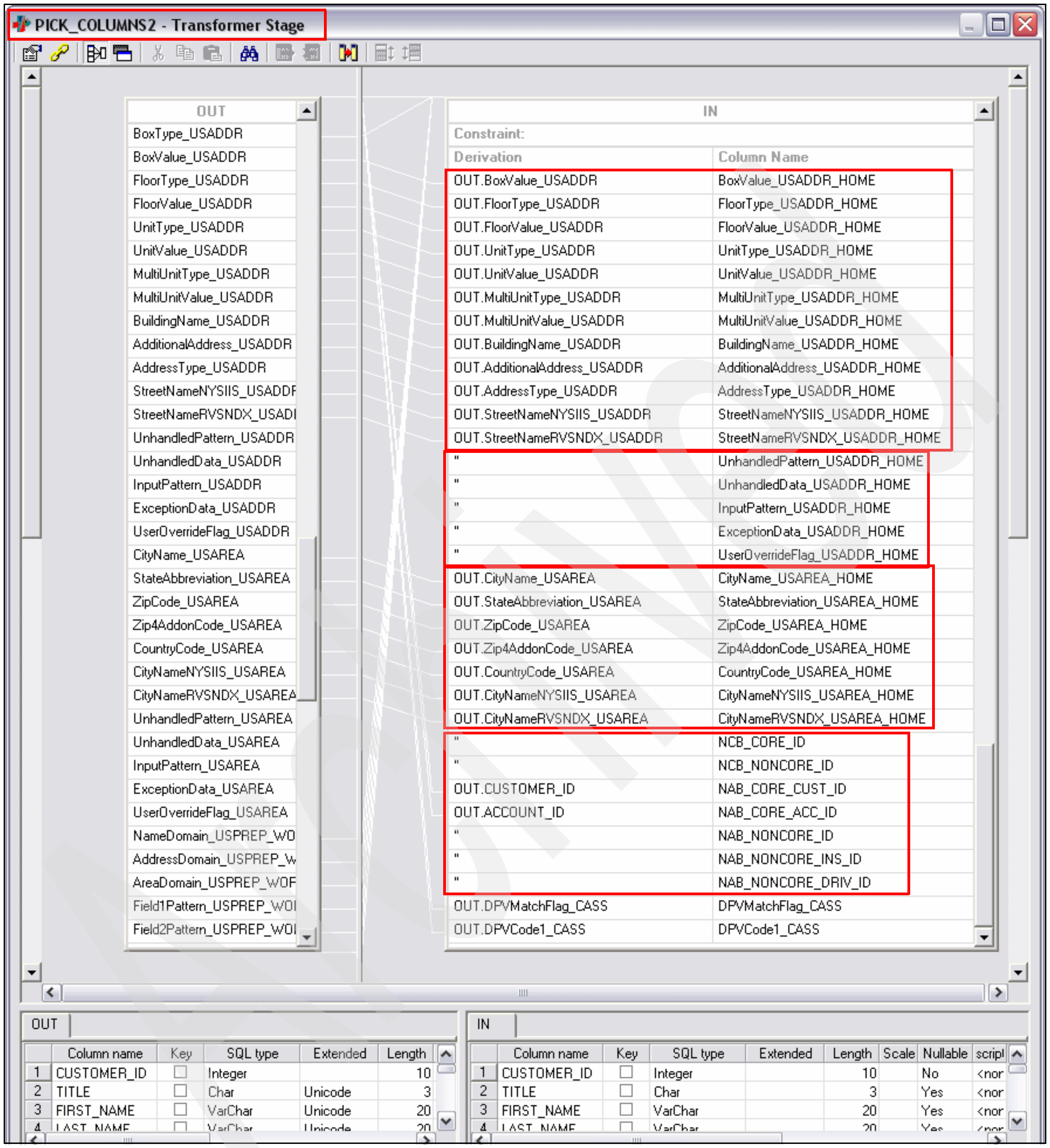
OUT

	Column name	Key	SQL type	Extended	Length
1	CUSTOMER_ID	☐	Integer		10
2	TITLE	☐	Char	Unicode	3
3	FIRST_NAME	☐	VarChar	Unicode	20
4	LAST_NAME	☐	VarChar	Unicode	20

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripl
1	CUSTOMER_ID	☐	Integer		10		No	<nor
2	TITLE	☐	Char		3		Yes	<nor
3	FIRST_NAME	☐	VarChar		20		Yes	<nor
4	LAST_NAME	☐	VarChar		20		Yes	<nor

Figure 2-222 Create j13_PREPARE_NAB_DATA_FOR_FUNNEL 5/17



PICK_COLUMNS2 - Transformer Stage

OUT
BoxType_USADDR
BoxValue_USADDR
FloorType_USADDR
FloorValue_USADDR
UnitType_USADDR
UnitValue_USADDR
MultiUnitType_USADDR
MultiUnitValue_USADDR
BuildingName_USADDR
AdditionalAddress_USADDR
AddressType_USADDR
StreetNameNYSIIS_USADDF
StreetNameRVSNDX_USADI
UnhandledPattern_USADDR
UnhandledData_USADDR
InputPattern_USADDR
ExceptionData_USADDR
UserOverrideFlag_USADDR
CityName_USAREA
StateAbbreviation_USAREA
ZipCode_USAREA
Zip4AddonCode_USAREA
CountryCode_USAREA
CityNameNYSIIS_USAREA
CityNameRVSNDX_USAREA
UnhandledPattern_USAREA
UnhandledData_USAREA
InputPattern_USAREA
ExceptionData_USAREA
UserOverrideFlag_USAREA
NameDomain_USPREP_WO
AddressDomain_USPREP_W
AreaDomain_USPREP_WOF
Field1Pattern_USPREP_WOI
Field2Pattern_USPREP_WOI

IN

Constraint:

Derivation	Column Name
OUT.BoxValue_USADDR	BoxValue_USADDR_HOME
OUT.FloorType_USADDR	FloorType_USADDR_HOME
OUT.FloorValue_USADDR	FloorValue_USADDR_HOME
OUT.UnitType_USADDR	UnitType_USADDR_HOME
OUT.UnitValue_USADDR	UnitValue_USADDR_HOME
OUT.MultiUnitType_USADDR	MultiUnitType_USADDR_HOME
OUT.MultiUnitValue_USADDR	MultiUnitValue_USADDR_HOME
OUT.BuildingName_USADDR	BuildingName_USADDR_HOME
OUT.AdditionalAddress_USADDR	AdditionalAddress_USADDR_HOME
OUT.AddressType_USADDR	AddressType_USADDR_HOME
OUT.StreetNameNYSIIS_USADDR	StreetNameNYSIIS_USADDR_HOME
OUT.StreetNameRVSNDX_USADDR	StreetNameRVSNDX_USADDR_HOME
"	UnhandledPattern_USADDR_HOME
"	UnhandledData_USADDR_HOME
"	InputPattern_USADDR_HOME
"	ExceptionData_USADDR_HOME
"	UserOverrideFlag_USADDR_HOME
OUT.CityName_USAREA	CityName_USAREA_HOME
OUT.StateAbbreviation_USAREA	StateAbbreviation_USAREA_HOME
OUT.ZipCode_USAREA	ZipCode_USAREA_HOME
OUT.Zip4AddonCode_USAREA	Zip4AddonCode_USAREA_HOME
OUT.CountryCode_USAREA	CountryCode_USAREA_HOME
OUT.CityNameNYSIIS_USAREA	CityNameNYSIIS_USAREA_HOME
OUT.CityNameRVSNDX_USAREA	CityNameRVSNDX_USAREA_HOME
"	NCB_CORE_ID
"	NCB_NONCORE_ID
OUT.CUSTOMER_ID	NAB_CORE_CUST_ID
OUT.ACCOUNT_ID	NAB_CORE_ACC_ID
"	NAB_NONCORE_ID
"	NAB_NONCORE_INS_ID
"	NAB_NONCORE_DRIV_ID
OUT.DPVMatchFlag_CASS	DPVMatchFlag_CASS
OUT.DPVCode1_CASS	DPVCode1_CASS

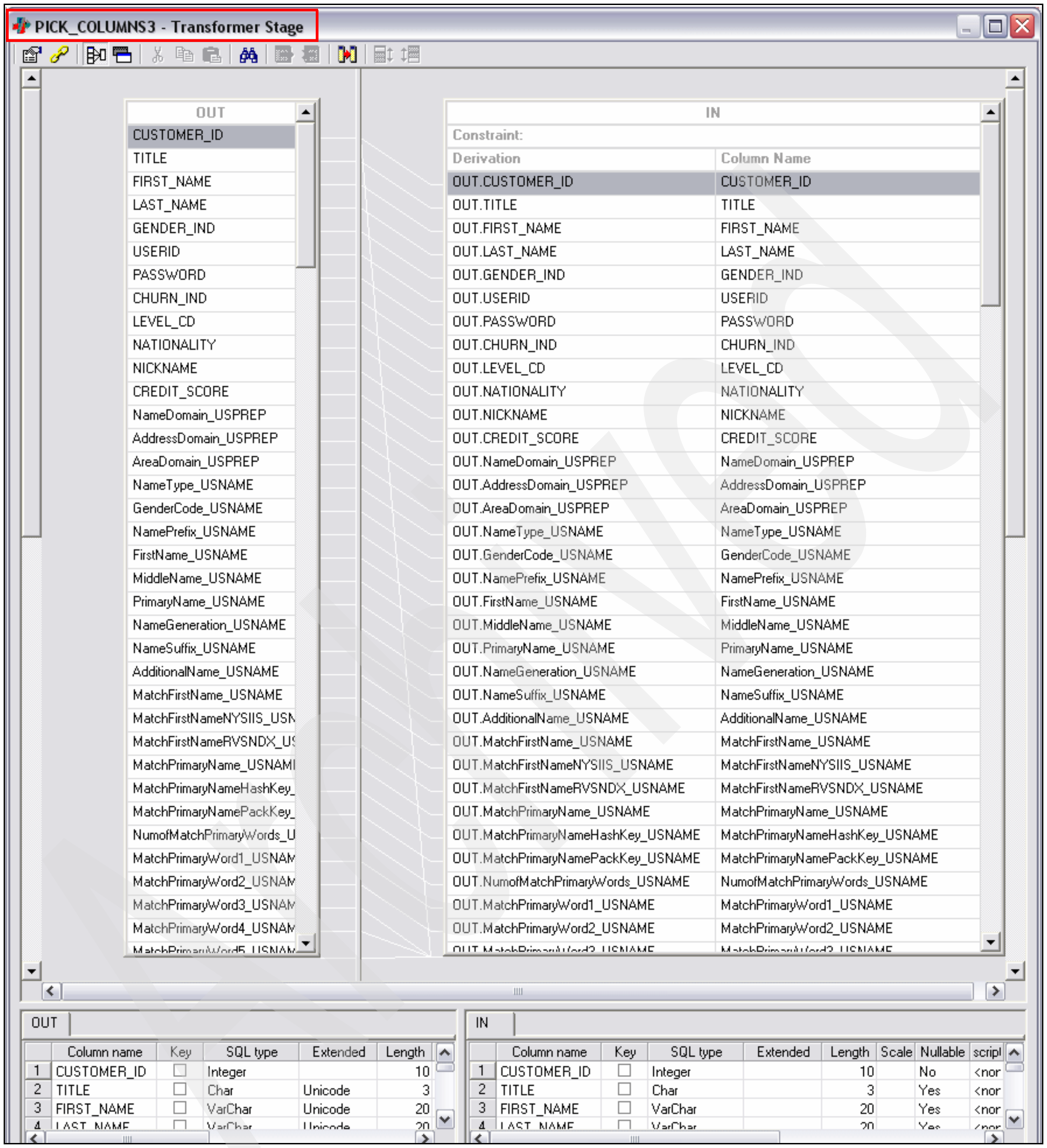
OUT

	Column name	Key	SQL type	Extended	Length
1	CUSTOMER_ID	☐	Integer		10
2	TITLE	☐	Char	Unicode	3
3	FIRST_NAME	☐	VarChar	Unicode	20
4	LAST_NAME	☐	VarChar	Unicode	20

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripl
1	CUSTOMER_ID	☐	Integer		10		No	<nor
2	TITLE	☐	Char		3		Yes	<nor
3	FIRST_NAME	☐	VarChar		20		Yes	<nor
4	LAST_NAME	☐	VarChar		20		Yes	<nor

Figure 2-223 Create j13_PREPARE_NAB_DATA_FOR_FUNNEL 6/17


PICK_COLUMNS3 - Transformer Stage

OUT
CUSTOMER_ID
TITLE
FIRST_NAME
LAST_NAME
GENDER_IND
USERID
PASSWORD
CHURN_IND
LEVEL_CD
NATIONALITY
NICKNAME
CREDIT_SCORE
NameDomain_USPREP
AddressDomain_USPREP
AreaDomain_USPREP
NameType_USNAME
GenderCode_USNAME
NamePrefix_USNAME
FirstName_USNAME
MiddleName_USNAME
PrimaryName_USNAME
NameGeneration_USNAME
NameSuffix_USNAME
AdditionalName_USNAME
MatchFirstName_USNAME
MatchFirstNameNYSIIS_USN
MatchFirstNameRVSNDX_US
MatchPrimaryName_USNAMI
MatchPrimaryNameHashKey_
MatchPrimaryNamePackKey_
NumofMatchPrimaryWords_U
MatchPrimaryWord1_USNAM
MatchPrimaryWord2_USNAM
MatchPrimaryWord3_USNAM
MatchPrimaryWord4_USNAM

IN

Constraint:

Derivation	Column Name
OUT.CUSTOMER_ID	CUSTOMER_ID
OUT.TITLE	TITLE
OUT.FIRST_NAME	FIRST_NAME
OUT.LAST_NAME	LAST_NAME
OUT.GENDER_IND	GENDER_IND
OUT.USERID	USERID
OUT.PASSWORD	PASSWORD
OUT.CHURN_IND	CHURN_IND
OUT.LEVEL_CD	LEVEL_CD
OUT.NATIONALITY	NATIONALITY
OUT.NICKNAME	NICKNAME
OUT.CREDIT_SCORE	CREDIT_SCORE
OUT.NameDomain_USPREP	NameDomain_USPREP
OUT.AddressDomain_USPREP	AddressDomain_USPREP
OUT.AreaDomain_USPREP	AreaDomain_USPREP
OUT.NameType_USNAME	NameType_USNAME
OUT.GenderCode_USNAME	GenderCode_USNAME
OUT.NamePrefix_USNAME	NamePrefix_USNAME
OUT.FirstName_USNAME	FirstName_USNAME
OUT.MiddleName_USNAME	MiddleName_USNAME
OUT.PrimaryName_USNAME	PrimaryName_USNAME
OUT.NameGeneration_USNAME	NameGeneration_USNAME
OUT.NameSuffix_USNAME	NameSuffix_USNAME
OUT.AdditionalName_USNAME	AdditionalName_USNAME
OUT.MatchFirstName_USNAME	MatchFirstName_USNAME
OUT.MatchFirstNameNYSIIS_USNAME	MatchFirstNameNYSIIS_USNAME
OUT.MatchFirstNameRVSNDX_USNAME	MatchFirstNameRVSNDX_USNAME
OUT.MatchPrimaryName_USNAME	MatchPrimaryName_USNAME
OUT.MatchPrimaryNameHashKey_USNAME	MatchPrimaryNameHashKey_USNAME
OUT.MatchPrimaryNamePackKey_USNAME	MatchPrimaryNamePackKey_USNAME
OUT.NumofMatchPrimaryWords_USNAME	NumofMatchPrimaryWords_USNAME
OUT.MatchPrimaryWord1_USNAME	MatchPrimaryWord1_USNAME
OUT.MatchPrimaryWord2_USNAME	MatchPrimaryWord2_USNAME

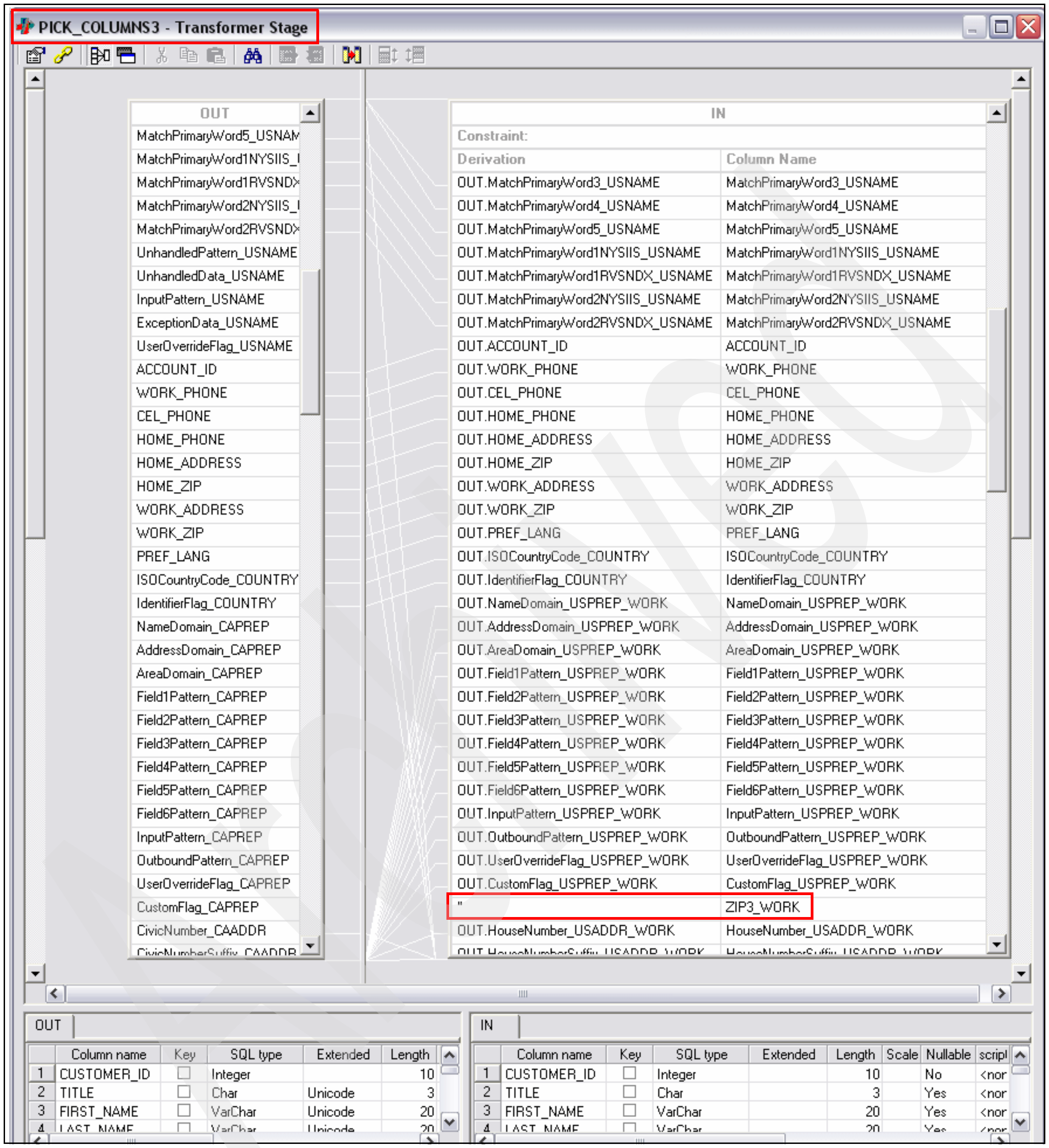
OUT

	Column name	Key	SQL type	Extended	Length
1	CUSTOMER_ID	☐	Integer		10
2	TITLE	☐	Char	Unicode	3
3	FIRST_NAME	☐	VarChar	Unicode	20

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scrip
1	CUSTOMER_ID	☐	Integer		10		No	<nor
2	TITLE	☐	Char		3		Yes	<nor
3	FIRST_NAME	☐	VarChar		20		Yes	<nor

Figure 2-224   Create j13_PREPARE_NAB_DATA_FOR_FUNNEL 7/17


PICK_COLUMNS3 - Transformer Stage

OUT
MatchPrimaryWord5_USNAM
MatchPrimaryWord1NYSIIS_I
MatchPrimaryWord1RVSNDX
MatchPrimaryWord2NYSIIS_I
MatchPrimaryWord2RVSNDX
UnhandledPattern_USNAME
UnhandledData_USNAME
InputPattern_USNAME
ExceptionData_USNAME
UserOverrideFlag_USNAME
ACCOUNT_ID
WORK_PHONE
CEL_PHONE
HOME_PHONE
HOME_ADDRESS
HOME_ZIP
WORK_ADDRESS
WORK_ZIP
PREF_LANG
ISOCountryCode_COUNTRY
IdentifierFlag_COUNTRY
NameDomain_CAPREP
AddressDomain_CAPREP
AreaDomain_CAPREP
Field1Pattern_CAPREP
Field2Pattern_CAPREP
Field3Pattern_CAPREP
Field4Pattern_CAPREP
Field5Pattern_CAPREP
Field6Pattern_CAPREP
InputPattern_CAPREP
OutboundPattern_CAPREP
UserOverrideFlag_CAPREP
CustomFlag_CAPREP
CivicNumber_CAADDR
CivicNumberSuffix_CAADDR

IN

Constraint:

Derivation	Column Name
OUT.MatchPrimaryWord3_USNAME	MatchPrimaryWord3_USNAME
OUT.MatchPrimaryWord4_USNAME	MatchPrimaryWord4_USNAME
OUT.MatchPrimaryWord5_USNAME	MatchPrimaryWord5_USNAME
OUT.MatchPrimaryWord1NYSIIS_USNAME	MatchPrimaryWord1NYSIIS_USNAME
OUT.MatchPrimaryWord1RVSNDX_USNAME	MatchPrimaryWord1RVSNDX_USNAME
OUT.MatchPrimaryWord2NYSIIS_USNAME	MatchPrimaryWord2NYSIIS_USNAME
OUT.MatchPrimaryWord2RVSNDX_USNAME	MatchPrimaryWord2RVSNDX_USNAME
OUT.ACCOUNT_ID	ACCOUNT_ID
OUT.WORK_PHONE	WORK_PHONE
OUT.CEL_PHONE	CEL_PHONE
OUT.HOME_PHONE	HOME_PHONE
OUT.HOME_ADDRESS	HOME_ADDRESS
OUT.HOME_ZIP	HOME_ZIP
OUT.WORK_ADDRESS	WORK_ADDRESS
OUT.WORK_ZIP	WORK_ZIP
OUT.PREF_LANG	PREF_LANG
OUT.ISOCountryCode_COUNTRY	ISOCountryCode_COUNTRY
OUT.IdentifierFlag_COUNTRY	IdentifierFlag_COUNTRY
OUT.NameDomain_USPREP_WORK	NameDomain_USPREP_WORK
OUT.AddressDomain_USPREP_WORK	AddressDomain_USPREP_WORK
OUT.AreaDomain_USPREP_WORK	AreaDomain_USPREP_WORK
OUT.Field1Pattern_USPREP_WORK	Field1Pattern_USPREP_WORK
OUT.Field2Pattern_USPREP_WORK	Field2Pattern_USPREP_WORK
OUT.Field3Pattern_USPREP_WORK	Field3Pattern_USPREP_WORK
OUT.Field4Pattern_USPREP_WORK	Field4Pattern_USPREP_WORK
OUT.Field5Pattern_USPREP_WORK	Field5Pattern_USPREP_WORK
OUT.Field6Pattern_USPREP_WORK	Field6Pattern_USPREP_WORK
OUT.InputPattern_USPREP_WORK	InputPattern_USPREP_WORK
OUT.OutboundPattern_USPREP_WORK	OutboundPattern_USPREP_WORK
OUT.UserOverrideFlag_USPREP_WORK	UserOverrideFlag_USPREP_WORK
OUT.CustomFlag_USPREP_WORK	CustomFlag_USPREP_WORK
''	ZIP3_WORK
OUT.HouseNumber_USADDR_WORK	HouseNumber_USADDR_WORK
OUT.HouseNumberSuffix_USADDR_WORK	HouseNumberSuffix_USADDR_WORK

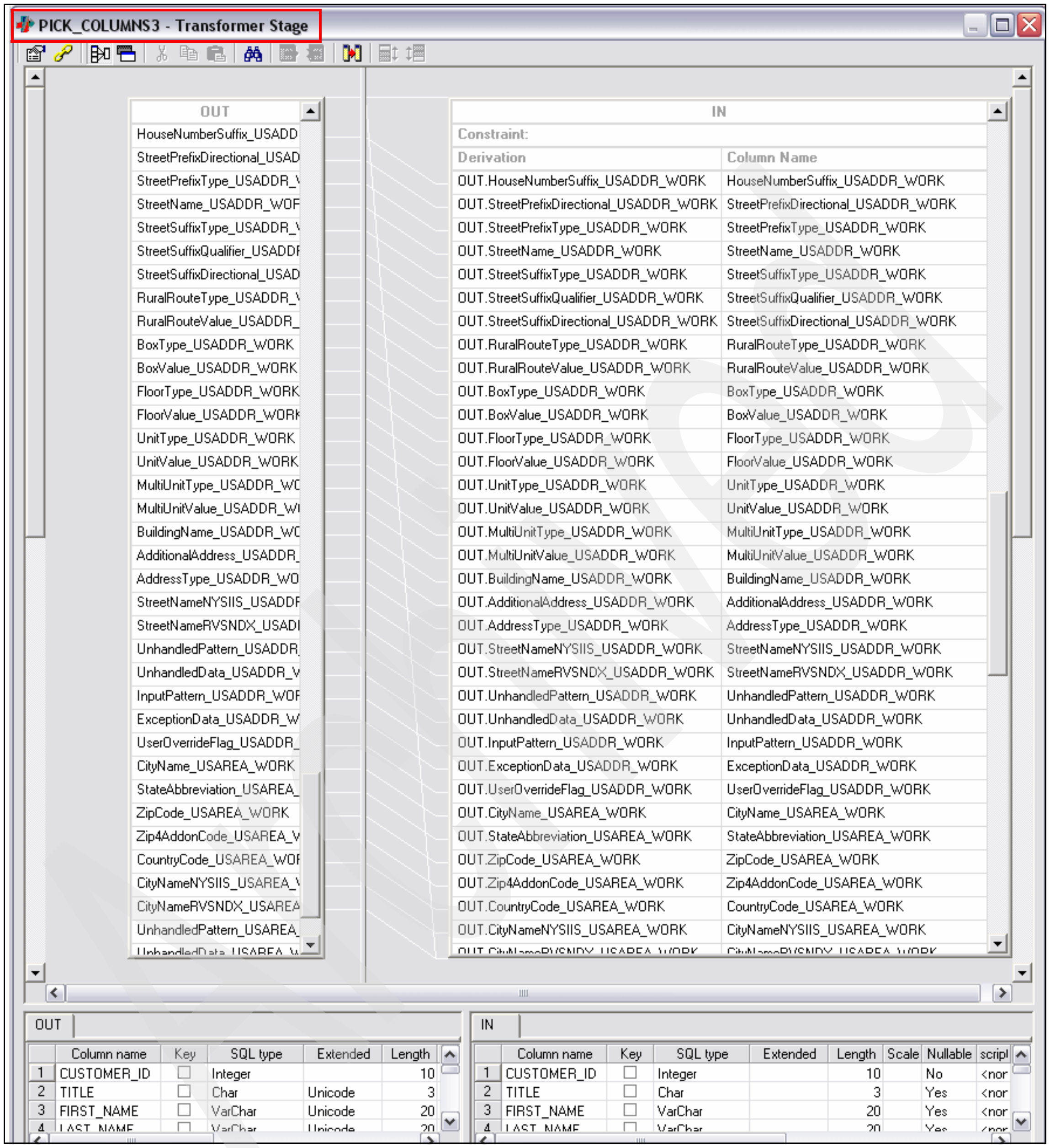
OUT

	Column name	Key	SQL type	Extended	Length
1	CUSTOMER_ID	☐	Integer		10
2	TITLE	☐	Char	Unicode	3
3	FIRST_NAME	☐	VarChar	Unicode	20
4	LAST_NAME	☐	VarChar	Unicode	20

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripl
1	CUSTOMER_ID	☐	Integer		10		No	<nor
2	TITLE	☐	Char		3		Yes	<nor
3	FIRST_NAME	☐	VarChar		20		Yes	<nor
4	LAST_NAME	☐	VarChar		20		Yes	<nor

Figure 2-225   Create j13_PREPARE_NAB_DATA_FOR_FUNNEL 8/17



PICK_COLUMNS3 - Transformer Stage

OUT
HouseNumberSuffix_USADD
StreetPrefixDirectional_USAD
StreetPrefixType_USADDR_\
StreetName_USADDR_WOF
StreetSuffixType_USADDR_\
StreetSuffixQualifier_USADDF
StreetSuffixDirectional_USAD
RuralRouteType_USADDR_\
RuralRouteValue_USADDR_
BoxType_USADDR_WORK
BoxValue_USADDR_WORK
FloorType_USADDR_WORK
FloorValue_USADDR_WORK
UnitType_USADDR_WORK
UnitValue_USADDR_WORK
MultiUnitType_USADDR_WC
MultiUnitValue_USADDR_WI
BuildingName_USADDR_WC
AdditionalAddress_USADDR_
AddressType_USADDR_WO
StreetNameNYSIIS_USADDF
StreetNameRVSNDX_USADI
UnhandledPattern_USADDR_
UnhandledData_USADDR_V
InputPattern_USADDR_WOF
ExceptionData_USADDR_W
UserOverrideFlag_USADDR_
CityName_USAREA_WORK
StateAbbreviation_USAREA_
ZipCode_USAREA_WORK
Zip4AddonCode_USAREA_V
CountryCode_USAREA_WOF
CityNameNYSIIS_USAREA_\
CityNameRVSNDX_USAREA
UnhandledPattern_USAREA_
UnhandledData_USAREA_W

IN

Constraint:

Derivation	Column Name
OUT.HouseNumberSuffix_USADDR_WORK	HouseNumberSuffix_USADDR_WORK
OUT.StreetPrefixDirectional_USADDR_WORK	StreetPrefixDirectional_USADDR_WORK
OUT.StreetPrefixType_USADDR_WORK	StreetPrefixType_USADDR_WORK
OUT.StreetName_USADDR_WORK	StreetName_USADDR_WORK
OUT.StreetSuffixType_USADDR_WORK	StreetSuffixType_USADDR_WORK
OUT.StreetSuffixQualifier_USADDR_WORK	StreetSuffixQualifier_USADDR_WORK
OUT.StreetSuffixDirectional_USADDR_WORK	StreetSuffixDirectional_USADDR_WORK
OUT.RuralRouteType_USADDR_WORK	RuralRouteType_USADDR_WORK
OUT.RuralRouteValue_USADDR_WORK	RuralRouteValue_USADDR_WORK
OUT.BoxType_USADDR_WORK	BoxType_USADDR_WORK
OUT.BoxValue_USADDR_WORK	BoxValue_USADDR_WORK
OUT.FloorType_USADDR_WORK	FloorType_USADDR_WORK
OUT.FloorValue_USADDR_WORK	FloorValue_USADDR_WORK
OUT.UnitType_USADDR_WORK	UnitType_USADDR_WORK
OUT.UnitValue_USADDR_WORK	UnitValue_USADDR_WORK
OUT.MultiUnitType_USADDR_WORK	MultiUnitType_USADDR_WORK
OUT.MultiUnitValue_USADDR_WORK	MultiUnitValue_USADDR_WORK
OUT.BuildingName_USADDR_WORK	BuildingName_USADDR_WORK
OUT.AdditionalAddress_USADDR_WORK	AdditionalAddress_USADDR_WORK
OUT.AddressType_USADDR_WORK	AddressType_USADDR_WORK
OUT.StreetNameNYSIIS_USADDR_WORK	StreetNameNYSIIS_USADDR_WORK
OUT.StreetNameRVSNDX_USADDR_WORK	StreetNameRVSNDX_USADDR_WORK
OUT.UnhandledPattern_USADDR_WORK	UnhandledPattern_USADDR_WORK
OUT.UnhandledData_USADDR_WORK	UnhandledData_USADDR_WORK
OUT.InputPattern_USADDR_WORK	InputPattern_USADDR_WORK
OUT.ExceptionData_USADDR_WORK	ExceptionData_USADDR_WORK
OUT.UserOverrideFlag_USADDR_WORK	UserOverrideFlag_USADDR_WORK
OUT.CityName_USAREA_WORK	CityName_USAREA_WORK
OUT.StateAbbreviation_USAREA_WORK	StateAbbreviation_USAREA_WORK
OUT.ZipCode_USAREA_WORK	ZipCode_USAREA_WORK
OUT.Zip4AddonCode_USAREA_WORK	Zip4AddonCode_USAREA_WORK
OUT.CountryCode_USAREA_WORK	CountryCode_USAREA_WORK
OUT.CityNameNYSIIS_USAREA_WORK	CityNameNYSIIS_USAREA_WORK
OUT.CityNameRVSNDX_USAREA_WORK	CityNameRVSNDX_USAREA_WORK

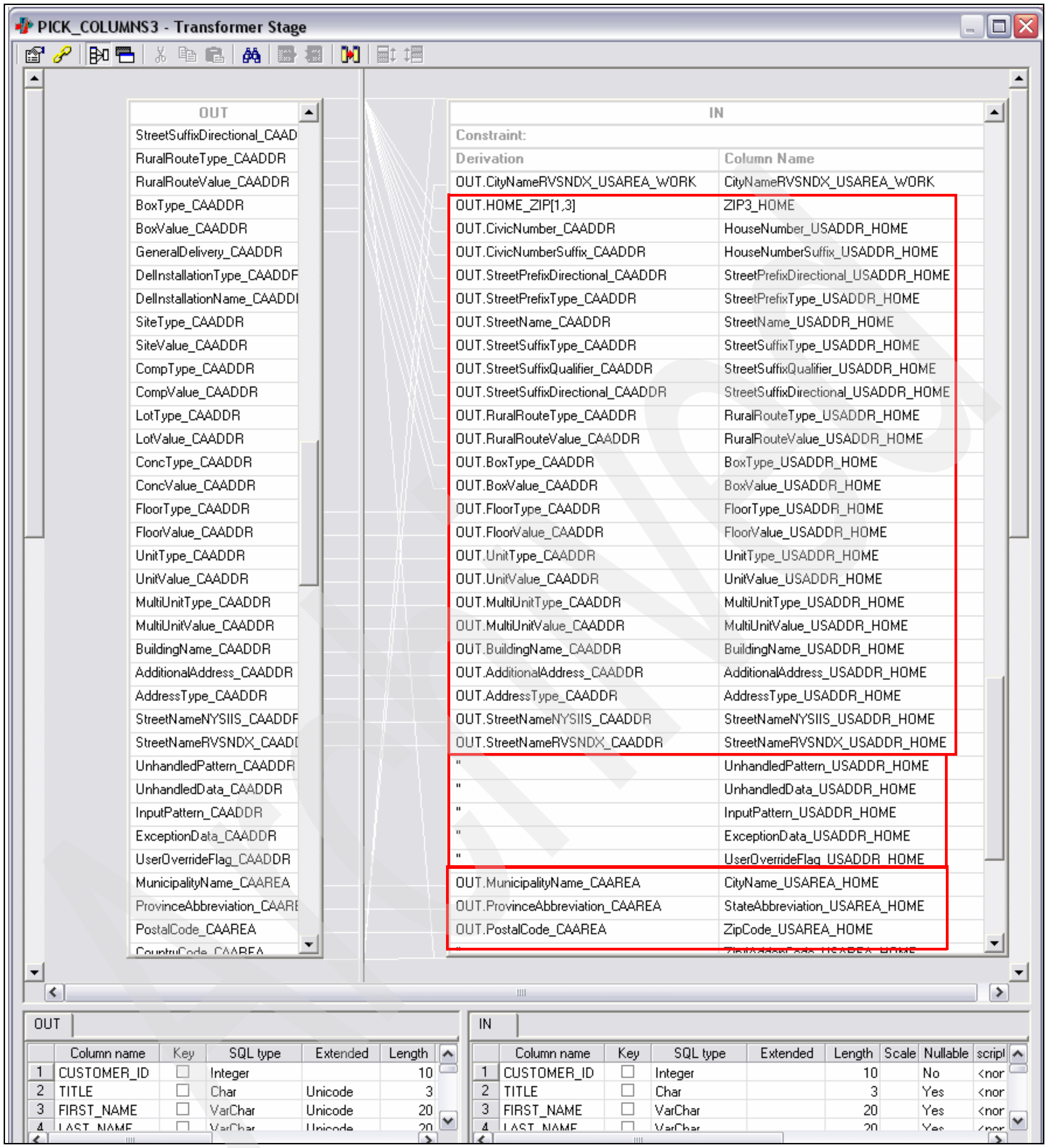
OUT

	Column name	Key	SQL type	Extended	Length
1	CUSTOMER_ID	☐	Integer		10
2	TITLE	☐	Char	Unicode	3
3	FIRST_NAME	☐	VarChar	Unicode	20
4	LAST_NAME	☐	VarChar	Unicode	20

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripl
1	CUSTOMER_ID	☐	Integer		10		No	<nor
2	TITLE	☐	Char		3		Yes	<nor
3	FIRST_NAME	☐	VarChar		20		Yes	<nor
4	LAST_NAME	☐	VarChar		20		Yes	<nor

Figure 2-226 Create j13_PREPARE_NAB_DATA_FOR_FUNNEL 9/17


PICK_COLUMNS3 - Transformer Stage

OUT
StreetSuffixDirectional_CAAD
RuralRouteType_CAADDR
RuralRouteValue_CAADDR
BoxType_CAADDR
BoxValue_CAADDR
GeneralDelivery_CAADDR
DelInstallationType_CAADDF
DelInstallationName_CAADDI
SiteType_CAADDR
SiteValue_CAADDR
CompType_CAADDR
CompValue_CAADDR
LotType_CAADDR
LotValue_CAADDR
ConcType_CAADDR
ConcValue_CAADDR
FloorType_CAADDR
FloorValue_CAADDR
UnitType_CAADDR
UnitValue_CAADDR
MultiUnitType_CAADDR
MultiUnitValue_CAADDR
BuildingName_CAADDR
AdditionalAddress_CAADDR
AddressType_CAADDR
StreetNameNYSIIS_CAADDF
StreetNameRVSNDX_CAADI
UnhandledPattern_CAADDR
UnhandledData_CAADDR
InputPattern_CAADDR
ExceptionData_CAADDR
UserOverrideFlag_CAADDR
MunicipalityName_CAAREA
ProvinceAbbreviation_CAARE
PostalCode_CAAREA
CountryCode_CAAREA

IN

Constraint:

Derivation	Column Name
OUT.CityNameRVSNDX_USAREA_WORK	CityNameRVSNDX_USAREA_WORK
OUT.HOME_ZIP[1,3]	ZIP3_HOME
OUT.CivicNumber_CAADDR	HouseNumber_USADDR_HOME
OUT.CivicNumberSuffix_CAADDR	HouseNumberSuffix_USADDR_HOME
OUT.StreetPrefixDirectional_CAADDR	StreetPrefixDirectional_USADDR_HOME
OUT.StreetPrefixType_CAADDR	StreetPrefixType_USADDR_HOME
OUT.StreetName_CAADDR	StreetName_USADDR_HOME
OUT.StreetSuffixType_CAADDR	StreetSuffixType_USADDR_HOME
OUT.StreetSuffixQualifier_CAADDR	StreetSuffixQualifier_USADDR_HOME
OUT.StreetSuffixDirectional_CAADDR	StreetSuffixDirectional_USADDR_HOME
OUT.RuralRouteType_CAADDR	RuralRouteType_USADDR_HOME
OUT.RuralRouteValue_CAADDR	RuralRouteValue_USADDR_HOME
OUT.BoxType_CAADDR	BoxType_USADDR_HOME
OUT.BoxValue_CAADDR	BoxValue_USADDR_HOME
OUT.FloorType_CAADDR	FloorType_USADDR_HOME
OUT.FloorValue_CAADDR	FloorValue_USADDR_HOME
OUT.UnitType_CAADDR	UnitType_USADDR_HOME
OUT.UnitValue_CAADDR	UnitValue_USADDR_HOME
OUT.MultiUnitType_CAADDR	MultiUnitType_USADDR_HOME
OUT.MultiUnitValue_CAADDR	MultiUnitValue_USADDR_HOME
OUT.BuildingName_CAADDR	BuildingName_USADDR_HOME
OUT.AdditionalAddress_CAADDR	AdditionalAddress_USADDR_HOME
OUT.AddressType_CAADDR	AddressType_USADDR_HOME
OUT.StreetNameNYSIIS_CAADDR	StreetNameNYSIIS_USADDR_HOME
OUT.StreetNameRVSNDX_CAADDR	StreetNameRVSNDX_USADDR_HOME
''	UnhandledPattern_USADDR_HOME
''	UnhandledData_USADDR_HOME
''	InputPattern_USADDR_HOME
''	ExceptionData_USADDR_HOME
''	UserOverrideFlag_USADDR_HOME
OUT.MunicipalityName_CAAREA	CityName_USAREA_HOME
OUT.ProvinceAbbreviation_CAAREA	StateAbbreviation_USAREA_HOME
OUT.PostalCode_CAAREA	ZipCode_USAREA_HOME
''	ZipAddressCode_USAREA_HOME

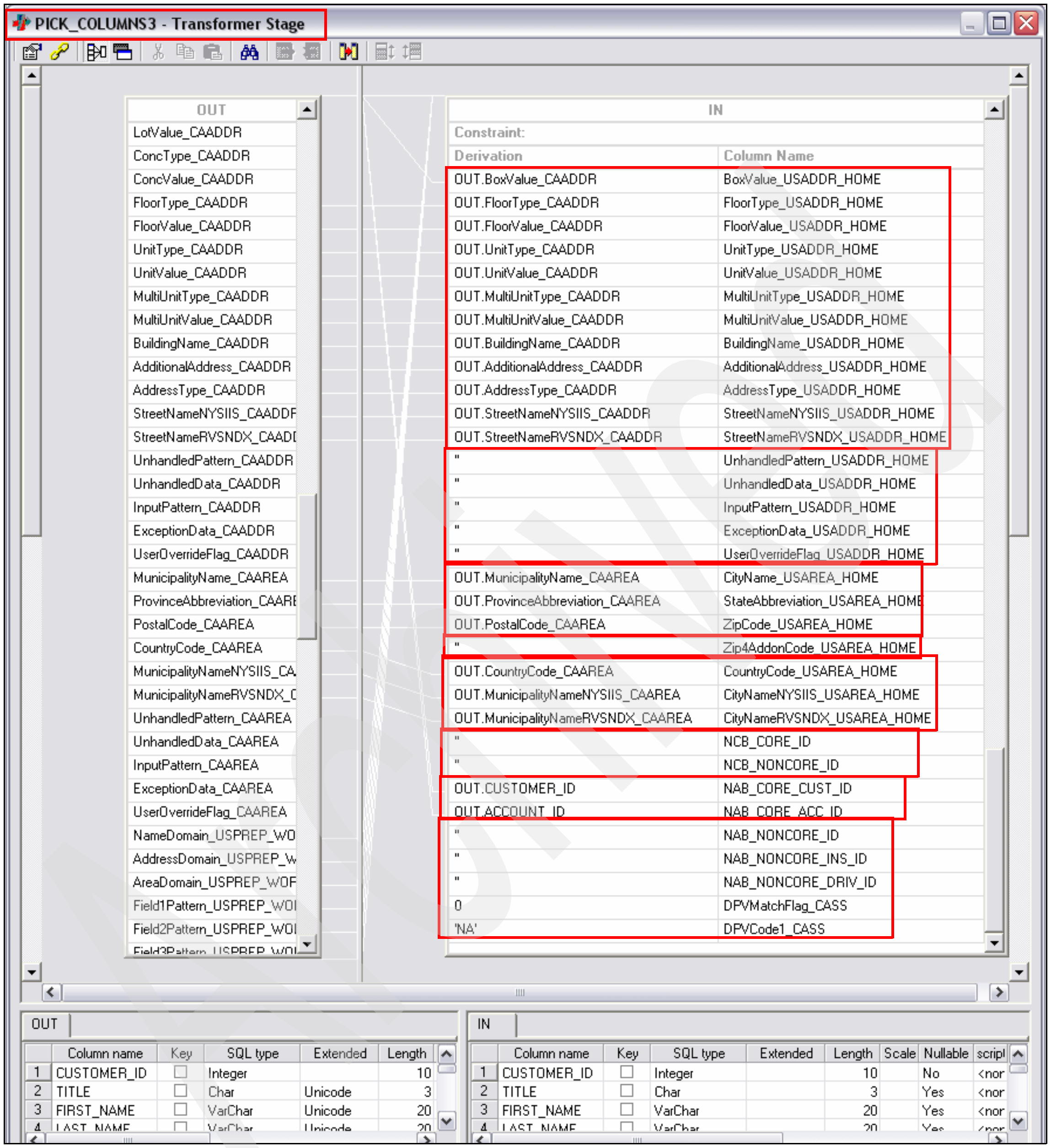
OUT

	Column name	Key	SQL type	Extended	Length
1	CUSTOMER_ID	☐	Integer		10
2	TITLE	☐	Char	Unicode	3
3	FIRST_NAME	☐	VarChar	Unicode	20
4	LAST_NAME	☐	VarChar	Unicode	20

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripl
1	CUSTOMER_ID	☐	Integer		10		No	<nor
2	TITLE	☐	Char		3		Yes	<nor
3	FIRST_NAME	☐	VarChar		20		Yes	<nor
4	LAST_NAME	☐	VarChar		20		Yes	<nor

Figure 2-227 Create j13_PREPARE_NAB_DATA_FOR_FUNNEL 10/17



PICK_COLUMNS3 - Transformer Stage

OUT
LotValue_CAADDR
ConcType_CAADDR
ConcValue_CAADDR
FloorType_CAADDR
FloorValue_CAADDR
UnitType_CAADDR
UnitValue_CAADDR
MultiUnitType_CAADDR
MultiUnitValue_CAADDR
BuildingName_CAADDR
AdditionalAddress_CAADDR
AddressType_CAADDR
StreetNameNYSIIS_CAADDF
StreetNameRVSNDX_CAADI
UnhandledPattern_CAADDR
UnhandledData_CAADDR
InputPattern_CAADDR
ExceptionData_CAADDR
UserOverrideFlag_CAADDR
MunicipalityName_CAAREA
ProvinceAbbreviation_CAARE
PostalCode_CAAREA
CountryCode_CAAREA
MunicipalityNameNYSIIS_CA
MunicipalityNameRVSNDX_C
UnhandledPattern_CAAREA
UnhandledData_CAAREA
InputPattern_CAAREA
ExceptionData_CAAREA
UserOverrideFlag_CAAREA
NameDomain_USPREP_WO
AddressDomain_USPREP_W
AreaDomain_USPREP_WOF
Field1Pattern_USPREP_WOI
Field2Pattern_USPREP_WOI
Field3Pattern_USPREP_WOI

IN

Constraint:

Derivation	Column Name
OUT.BoxValue_CAADDR	BoxValue_USADDR_HOME
OUT.FloorType_CAADDR	FloorType_USADDR_HOME
OUT.FloorValue_CAADDR	FloorValue_USADDR_HOME
OUT.UnitType_CAADDR	UnitType_USADDR_HOME
OUT.UnitValue_CAADDR	UnitValue_USADDR_HOME
OUT.MultiUnitType_CAADDR	MultiUnitType_USADDR_HOME
OUT.MultiUnitValue_CAADDR	MultiUnitValue_USADDR_HOME
OUT.BuildingName_CAADDR	BuildingName_USADDR_HOME
OUT.AdditionalAddress_CAADDR	AdditionalAddress_USADDR_HOME
OUT.AddressType_CAADDR	AddressType_USADDR_HOME
OUT.StreetNameNYSIIS_CAADDR	StreetNameNYSIIS_USADDR_HOME
OUT.StreetNameRVSNDX_CAADDR	StreetNameRVSNDX_USADDR_HOME
''	UnhandledPattern_USADDR_HOME
''	UnhandledData_USADDR_HOME
''	InputPattern_USADDR_HOME
''	ExceptionData_USADDR_HOME
''	UserOverrideFlag_USADDR_HOME
OUT.MunicipalityName_CAAREA	CityName_USAREA_HOME
OUT.ProvinceAbbreviation_CAAREA	StateAbbreviation_USAREA_HOME
OUT.PostalCode_CAAREA	ZipCode_USAREA_HOME
''	Zip4AddonCode_USAREA_HOME
OUT.CountryCode_CAAREA	CountryCode_USAREA_HOME
OUT.MunicipalityNameNYSIIS_CAAREA	CityNameNYSIIS_USAREA_HOME
OUT.MunicipalityNameRVSNDX_CAAREA	CityNameRVSNDX_USAREA_HOME
''	NCB_CORE_ID
''	NCB_NONCORE_ID
OUT.CUSTOMER_ID	NAB_CORE_CUST_ID
OUT.ACCOUNT_ID	NAB_CORE_ACC_ID
''	NAB_NONCORE_ID
''	NAB_NONCORE_INS_ID
''	NAB_NONCORE_DRIV_ID
0	DPVMatchFlag_CASS
'NA'	DPVCode1_CASS

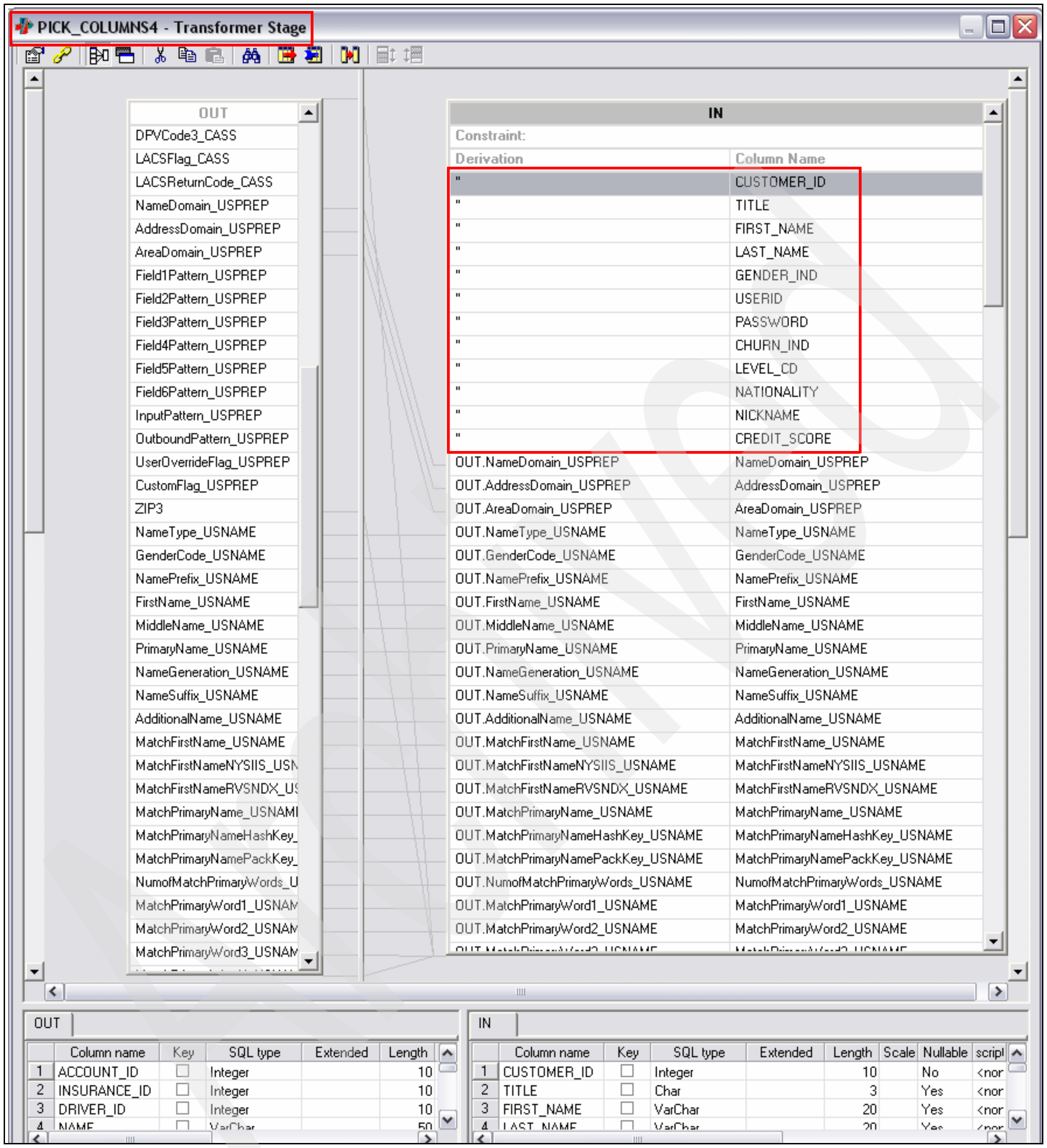
OUT

	Column name	Key	SQL type	Extended	Length
1	CUSTOMER_ID	☐	Integer		10
2	TITLE	☐	Char	Unicode	3
3	FIRST_NAME	☐	VarChar	Unicode	20
4	LAST_NAME	☐	VarChar	Unicode	20

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripl
1	CUSTOMER_ID	☐	Integer		10		No	<nor
2	TITLE	☐	Char		3		Yes	<nor
3	FIRST_NAME	☐	VarChar		20		Yes	<nor
4	LAST_NAME	☐	VarChar		20		Yes	<nor

Figure 2-228 Create j13_PREPARE_NAB_DATA_FOR_FUNNEL 11/17



PICK_COLUMNS4 - Transformer Stage

OUT
DPVCode3_CASS
LACSFlag_CASS
LACSReturnCode_CASS
NameDomain_USPREP
AddressDomain_USPREP
AreaDomain_USPREP
Field1Pattern_USPREP
Field2Pattern_USPREP
Field3Pattern_USPREP
Field4Pattern_USPREP
Field5Pattern_USPREP
Field6Pattern_USPREP
InputPattern_USPREP
OutboundPattern_USPREP
UserOverrideFlag_USPREP
CustomFlag_USPREP
ZIP3
NameType_USNAME
GenderCode_USNAME
NamePrefix_USNAME
FirstName_USNAME
MiddleName_USNAME
PrimaryName_USNAME
NameGeneration_USNAME
NameSuffix_USNAME
AdditionalName_USNAME
MatchFirstName_USNAME
MatchFirstNameNYSIIS_USN
MatchFirstNameRVSNDX_US
MatchPrimaryName_USNAMI
MatchPrimaryNameHashKey_
MatchPrimaryNamePackKey_
NumofMatchPrimaryWords_U
MatchPrimaryWord1_USNAM
MatchPrimaryWord2_USNAM
MatchPrimaryWord3_USNAM

IN

Constraint:

Derivation	Column Name
"	CUSTOMER_ID
"	TITLE
"	FIRST_NAME
"	LAST_NAME
"	GENDER_IND
"	USERID
"	PASSWORD
"	CHURN_IND
"	LEVEL_CD
"	NATIONALITY
"	NICKNAME
"	CREDIT_SCORE
OUT.NameDomain_USPREP	NameDomain_USPREP
OUT.AddressDomain_USPREP	AddressDomain_USPREP
OUT.AreaDomain_USPREP	AreaDomain_USPREP
OUT.NameType_USNAME	NameType_USNAME
OUT.GenderCode_USNAME	GenderCode_USNAME
OUT.NamePrefix_USNAME	NamePrefix_USNAME
OUT.FirstName_USNAME	FirstName_USNAME
OUT.MiddleName_USNAME	MiddleName_USNAME
OUT.PrimaryName_USNAME	PrimaryName_USNAME
OUT.NameGeneration_USNAME	NameGeneration_USNAME
OUT.NameSuffix_USNAME	NameSuffix_USNAME
OUT.AdditionalName_USNAME	AdditionalName_USNAME
OUT.MatchFirstName_USNAME	MatchFirstName_USNAME
OUT.MatchFirstNameNYSIIS_USNAME	MatchFirstNameNYSIIS_USNAME
OUT.MatchFirstNameRVSNDX_USNAME	MatchFirstNameRVSNDX_USNAME
OUT.MatchPrimaryName_USNAME	MatchPrimaryName_USNAME
OUT.MatchPrimaryNameHashKey_USNAME	MatchPrimaryNameHashKey_USNAME
OUT.MatchPrimaryNamePackKey_USNAME	MatchPrimaryNamePackKey_USNAME
OUT.NumofMatchPrimaryWords_USNAME	NumofMatchPrimaryWords_USNAME
OUT.MatchPrimaryWord1_USNAME	MatchPrimaryWord1_USNAME
OUT.MatchPrimaryWord2_USNAME	MatchPrimaryWord2_USNAME

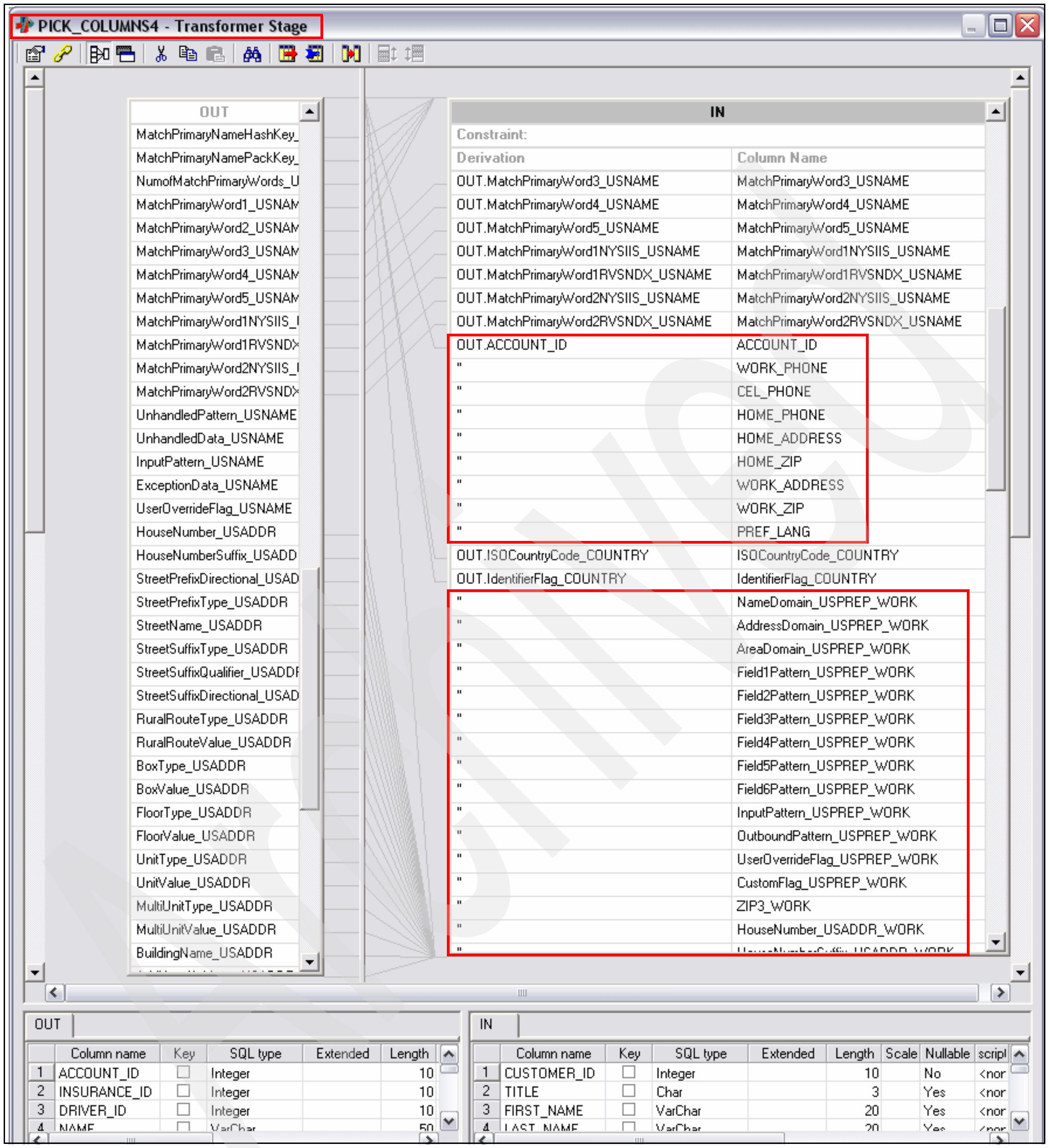
OUT

	Column name	Key	SQL type	Extended	Length
1	ACCOUNT_ID	☐	Integer		10
2	INSURANCE_ID	☐	Integer		10
3	DRIVER_ID	☐	Integer		10
4	NAME	☐	VarChar		50

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripl
1	CUSTOMER_ID	☐	Integer		10		No	<nor
2	TITLE	☐	Char		3		Yes	<nor
3	FIRST_NAME	☐	VarChar		20		Yes	<nor
4	LAST_NAME	☐	VarChar		20		Yes	<nor

Figure 2-229 Create j13_PREPARE_NAB_DATA_FOR_FUNNEL 12/17



PICK_COLUMNS4 - Transformer Stage

OUT
MatchPrimaryNameHashKey_
MatchPrimaryNamePackKey_
NumofMatchPrimaryWords_U
MatchPrimaryWord1_USNAM
MatchPrimaryWord2_USNAM
MatchPrimaryWord3_USNAM
MatchPrimaryWord4_USNAM
MatchPrimaryWord5_USNAM
MatchPrimaryWord1NYSIIS_
MatchPrimaryWord1RVSNDX
MatchPrimaryWord2NYSIIS_
MatchPrimaryWord2RVSNDX
UnhandledPattern_USNAME
UnhandledData_USNAME
InputPattern_USNAME
ExceptionData_USNAME
UserOverrideFlag_USNAME
HouseNumber_USADDR
HouseNumberSuffix_USADD
StreetPrefixDirectional_USAD
StreetPrefixType_USADDR
StreetName_USADDR
StreetSuffixType_USADDR
StreetSuffixQualifier_USADDF
StreetSuffixDirectional_USAD
RuralRouteType_USADDR
RuralRouteValue_USADDR
BoxType_USADDR
BoxValue_USADDR
FloorType_USADDR
FloorValue_USADDR
UnitType_USADDR
UnitValue_USADDR
MultiUnitType_USADDR
MultiUnitValue_USADDR
BuildingName_USADDR

IN

Constraint:

Derivation	Column Name
OUT.MatchPrimaryWord3_USNAME	MatchPrimaryWord3_USNAME
OUT.MatchPrimaryWord4_USNAME	MatchPrimaryWord4_USNAME
OUT.MatchPrimaryWord5_USNAME	MatchPrimaryWord5_USNAME
OUT.MatchPrimaryWord1NYSIIS_USNAME	MatchPrimaryWord1NYSIIS_USNAME
OUT.MatchPrimaryWord1RVSNDX_USNAME	MatchPrimaryWord1RVSNDX_USNAME
OUT.MatchPrimaryWord2NYSIIS_USNAME	MatchPrimaryWord2NYSIIS_USNAME
OUT.MatchPrimaryWord2RVSNDX_USNAME	MatchPrimaryWord2RVSNDX_USNAME
OUT.ACCOUNT_ID	ACCOUNT_ID
"	WORK_PHONE
"	CEL_PHONE
"	HOME_PHONE
"	HOME_ADDRESS
"	HOME_ZIP
"	WORK_ADDRESS
"	WORK_ZIP
"	PREF_LANG
OUT.ISOCountryCode_COUNTRY	ISOCountryCode_COUNTRY
OUT.IdentifierFlag_COUNTRY	IdentifierFlag_COUNTRY
"	NameDomain_USPREP_WORK
"	AddressDomain_USPREP_WORK
"	AreaDomain_USPREP_WORK
"	Field1Pattern_USPREP_WORK
"	Field2Pattern_USPREP_WORK
"	Field3Pattern_USPREP_WORK
"	Field4Pattern_USPREP_WORK
"	Field5Pattern_USPREP_WORK
"	Field6Pattern_USPREP_WORK
"	InputPattern_USPREP_WORK
"	OutboundPattern_USPREP_WORK
"	UserOverrideFlag_USPREP_WORK
"	CustomFlag_USPREP_WORK
"	ZIP3_WORK
"	HouseNumber_USADDR_WORK

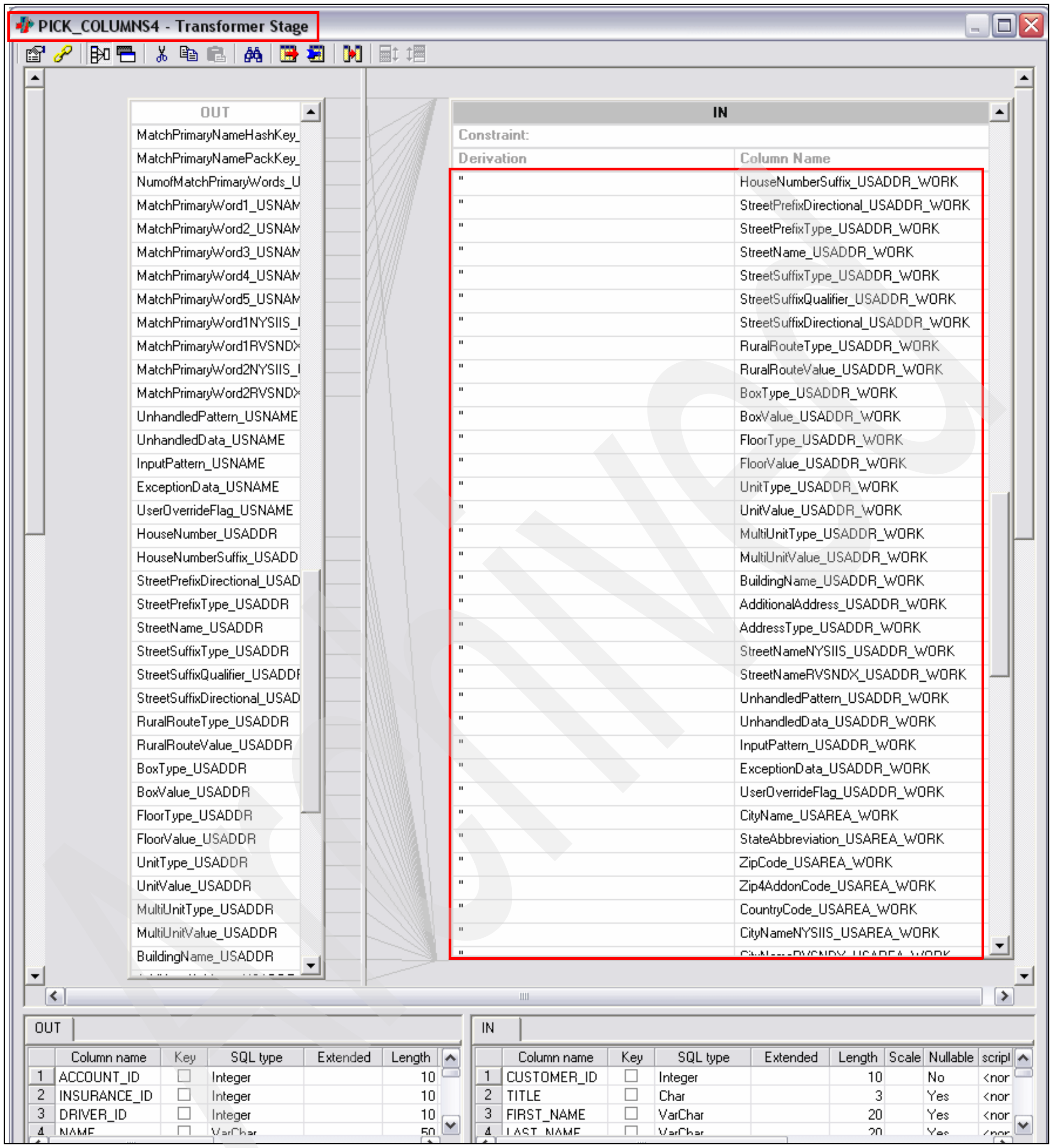
OUT

	Column name	Key	SQL type	Extended	Length
1	ACCOUNT_ID	☐	Integer		10
2	INSURANCE_ID	☐	Integer		10
3	DRIVER_ID	☐	Integer		10

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripl
1	CUSTOMER_ID	☐	Integer		10		No	<nor
2	TITLE	☐	Char		3		Yes	<nor
3	FIRST_NAME	☐	VarChar		20		Yes	<nor

Figure 2-230 Create j13_PREPARE_NAB_DATA_FOR_FUNNEL 13/17



PICK_COLUMNS4 - Transformer Stage

OUT

MatchPrimaryNameHashKey_
MatchPrimaryNamePackKey_
NumofMatchPrimaryWords_U
MatchPrimaryWord1_USNAM
MatchPrimaryWord2_USNAM
MatchPrimaryWord3_USNAM
MatchPrimaryWord4_USNAM
MatchPrimaryWord5_USNAM
MatchPrimaryWord1NYSIIS_I
MatchPrimaryWord1RVSNDX
MatchPrimaryWord2NYSIIS_I
MatchPrimaryWord2RVSNDX
UnhandledPattern_USNAME
UnhandledData_USNAME
InputPattern_USNAME
ExceptionData_USNAME
UserOverrideFlag_USNAME
HouseNumber_USADDR
HouseNumberSuffix_USADD
StreetPrefixDirectional_USAD
StreetPrefixType_USADDR
StreetName_USADDR
StreetSuffixType_USADDR
StreetSuffixQualifier_USADDF
StreetSuffixDirectional_USAD
RuralRouteType_USADDR
RuralRouteValue_USADDR
BoxType_USADDR
BoxValue_USADDR
FloorType_USADDR
FloorValue_USADDR
UnitType_USADDR
UnitValue_USADDR
MultiUnitType_USADDR
MultiUnitValue_USADDR
BuildingName_USADDR

IN

Constraint:

Derivation	Column Name
"	HouseNumberSuffix_USADDR_WORK
"	StreetPrefixDirectional_USADDR_WORK
"	StreetPrefixType_USADDR_WORK
"	StreetName_USADDR_WORK
"	StreetSuffixType_USADDR_WORK
"	StreetSuffixQualifier_USADDR_WORK
"	StreetSuffixDirectional_USADDR_WORK
"	RuralRouteType_USADDR_WORK
"	RuralRouteValue_USADDR_WORK
"	BoxType_USADDR_WORK
"	BoxValue_USADDR_WORK
"	FloorType_USADDR_WORK
"	FloorValue_USADDR_WORK
"	UnitType_USADDR_WORK
"	UnitValue_USADDR_WORK
"	MultiUnitType_USADDR_WORK
"	MultiUnitValue_USADDR_WORK
"	BuildingName_USADDR_WORK
"	AdditionalAddress_USADDR_WORK
"	AddressType_USADDR_WORK
"	StreetNameNYSIIS_USADDR_WORK
"	StreetNameRVSNDX_USADDR_WORK
"	UnhandledPattern_USADDR_WORK
"	UnhandledData_USADDR_WORK
"	InputPattern_USADDR_WORK
"	ExceptionData_USADDR_WORK
"	UserOverrideFlag_USADDR_WORK
"	CityName_USAREA_WORK
"	StateAbbreviation_USAREA_WORK
"	ZipCode_USAREA_WORK
"	Zip4AddonCode_USAREA_WORK
"	CountryCode_USAREA_WORK
"	CityNameNYSIIS_USAREA_WORK

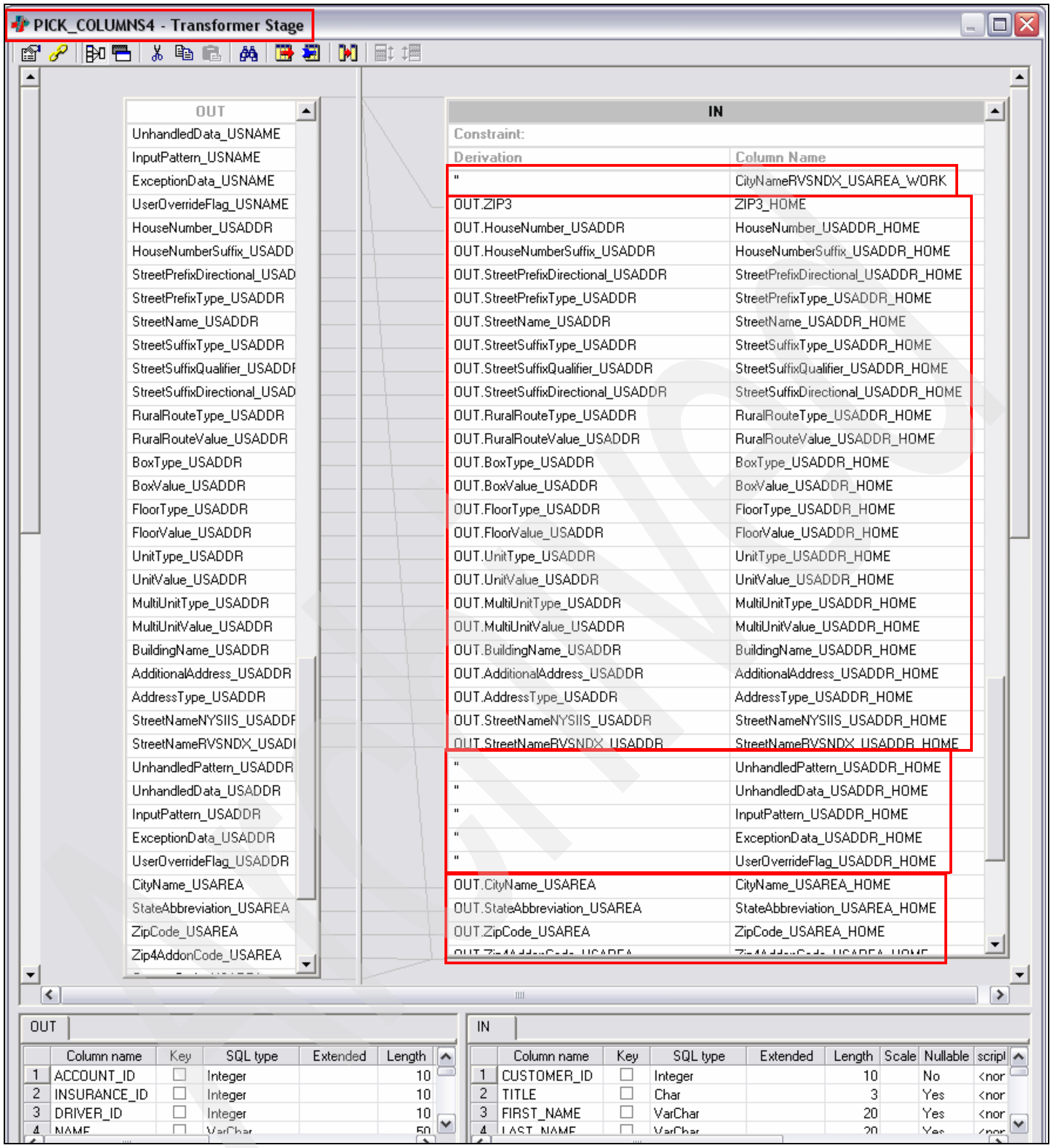
OUT

	Column name	Key	SQL type	Extended	Length
1	ACCOUNT_ID	☐	Integer		10
2	INSURANCE_ID	☐	Integer		10
3	DRIVER_ID	☐	Integer		10
4	NAME	☐	VarChar		50

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripl
1	CUSTOMER_ID	☐	Integer		10		No	<nor
2	TITLE	☐	Char		3		Yes	<nor
3	FIRST_NAME	☐	VarChar		20		Yes	<nor
4	LAST_NAME	☐	VarChar		20		Yes	<nor

Figure 2-231 Create j13_PREPARE_NAB_DATA_FOR_FUNNEL 14/17



PICK_COLUMNS4 - Transformer Stage

OUT
UnhandledData_USNAME
InputPattern_USNAME
ExceptionData_USNAME
UserOverrideFlag_USNAME
HouseNumber_USADDR
HouseNumberSuffix_USADD
StreetPrefixDirectional_USAD
StreetPrefixType_USADDR
StreetName_USADDR
StreetSuffixType_USADDR
StreetSuffixQualifier_USADDF
StreetSuffixDirectional_USAD
RuralRouteType_USADDR
RuralRouteValue_USADDR
BoxType_USADDR
BoxValue_USADDR
FloorType_USADDR
FloorValue_USADDR
UnitType_USADDR
UnitValue_USADDR
MultiUnitType_USADDR
MultiUnitValue_USADDR
BuildingName_USADDR
AdditionalAddress_USADDR
AddressType_USADDR
StreetNameNYSIIS_USADDF
StreetNameRVSNDX_USADI
UnhandledPattern_USADDR
UnhandledData_USADDR
InputPattern_USADDR
ExceptionData_USADDR
UserOverrideFlag_USADDR
CityName_USAREA
StateAbbreviation_USAREA
ZipCode_USAREA
Zip4AddonCode_USAREA

IN

Constraint:

Derivation	Column Name
''	CityNameRVSNDX_USAREA_WORK
OUT.ZIP3	ZIP3_HOME
OUT.HouseNumber_USADDR	HouseNumber_USADDR_HOME
OUT.HouseNumberSuffix_USADDR	HouseNumberSuffix_USADDR_HOME
OUT.StreetPrefixDirectional_USADDR	StreetPrefixDirectional_USADDR_HOME
OUT.StreetPrefixType_USADDR	StreetPrefixType_USADDR_HOME
OUT.StreetName_USADDR	StreetName_USADDR_HOME
OUT.StreetSuffixType_USADDR	StreetSuffixType_USADDR_HOME
OUT.StreetSuffixQualifier_USADDR	StreetSuffixQualifier_USADDR_HOME
OUT.StreetSuffixDirectional_USADDR	StreetSuffixDirectional_USADDR_HOME
OUT.RuralRouteType_USADDR	RuralRouteType_USADDR_HOME
OUT.RuralRouteValue_USADDR	RuralRouteValue_USADDR_HOME
OUT.BoxType_USADDR	BoxType_USADDR_HOME
OUT.BoxValue_USADDR	BoxValue_USADDR_HOME
OUT.FloorType_USADDR	FloorType_USADDR_HOME
OUT.FloorValue_USADDR	FloorValue_USADDR_HOME
OUT.UnitType_USADDR	UnitType_USADDR_HOME
OUT.UnitValue_USADDR	UnitValue_USADDR_HOME
OUT.MultiUnitType_USADDR	MultiUnitType_USADDR_HOME
OUT.MultiUnitValue_USADDR	MultiUnitValue_USADDR_HOME
OUT.BuildingName_USADDR	BuildingName_USADDR_HOME
OUT.AdditionalAddress_USADDR	AdditionalAddress_USADDR_HOME
OUT.AddressType_USADDR	AddressType_USADDR_HOME
OUT.StreetNameNYSIIS_USADDR	StreetNameNYSIIS_USADDR_HOME
OUT.StreetNameRVSNDX_USADDR	StreetNameRVSNDX_USADDR_HOME
''	UnhandledPattern_USADDR_HOME
''	UnhandledData_USADDR_HOME
''	InputPattern_USADDR_HOME
''	ExceptionData_USADDR_HOME
''	UserOverrideFlag_USADDR_HOME
OUT.CityName_USAREA	CityName_USAREA_HOME
OUT.StateAbbreviation_USAREA	StateAbbreviation_USAREA_HOME
OUT.ZipCode_USAREA	ZipCode_USAREA_HOME

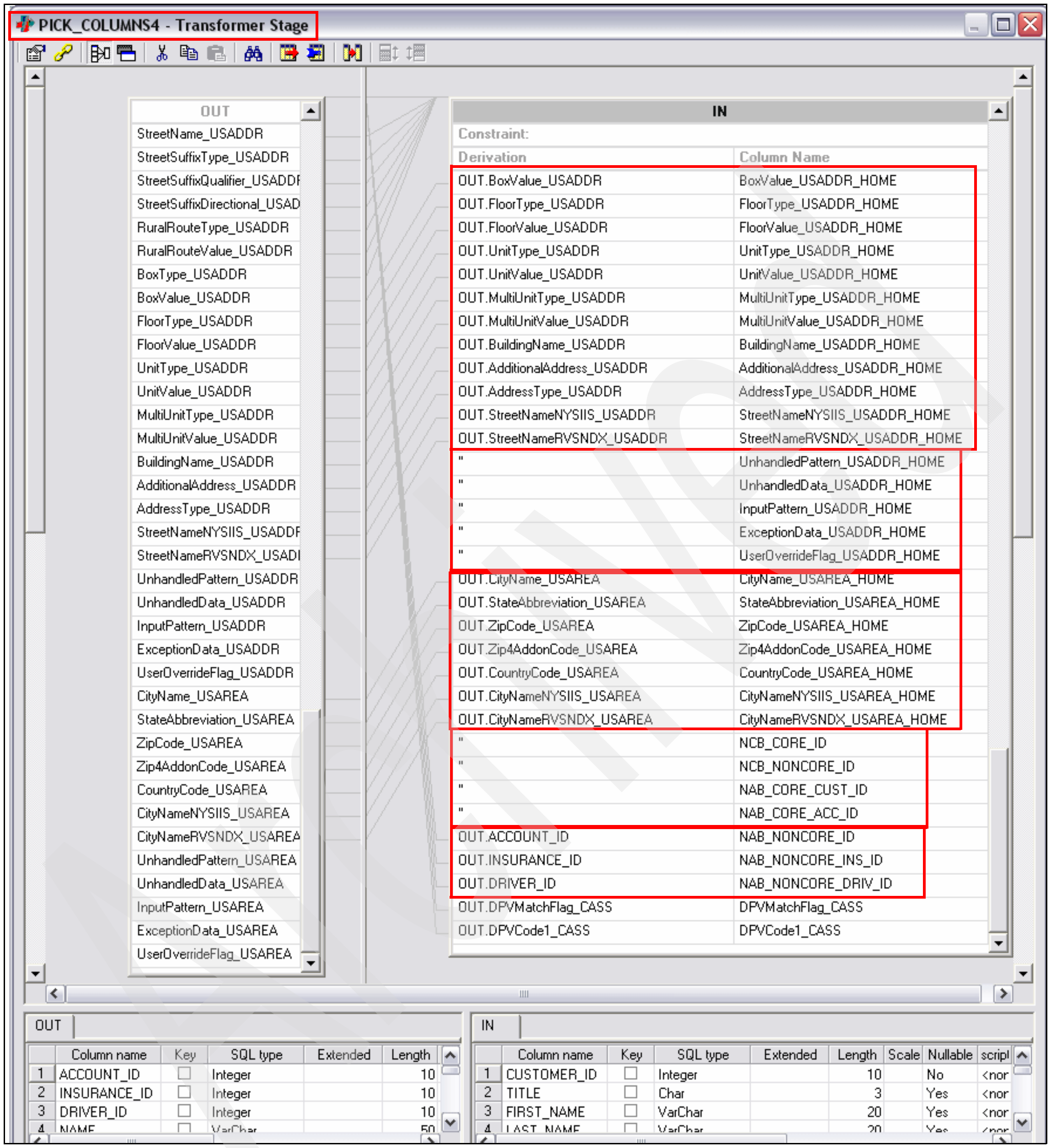
OUT

	Column name	Key	SQL type	Extended	Length
1	ACCOUNT_ID	☐	Integer		10
2	INSURANCE_ID	☐	Integer		10
3	DRIVER_ID	☐	Integer		10

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripl
1	CUSTOMER_ID	☐	Integer		10		No	<nor
2	TITLE	☐	Char		3		Yes	<nor
3	FIRST_NAME	☐	VarChar		20		Yes	<nor

Figure 2-232 Create j13_PREPARE_NAB_DATA_FOR_FUNNEL 15/17


PICK_COLUMNS4 - Transformer Stage

OUT
StreetName_USADDR
StreetSuffixType_USADDR
StreetSuffixQualifier_USADDF
StreetSuffixDirectional_USAD
RuralRouteType_USADDR
RuralRouteValue_USADDR
BoxType_USADDR
BoxValue_USADDR
FloorType_USADDR
FloorValue_USADDR
UnitType_USADDR
UnitValue_USADDR
MultiUnitType_USADDR
MultiUnitValue_USADDR
BuildingName_USADDR
AdditionalAddress_USADDR
AddressType_USADDR
StreetNameNYSIIS_USADDF
StreetNameRVSNDX_USADI
UnhandledPattern_USADDR
UnhandledData_USADDR
InputPattern_USADDR
ExceptionData_USADDR
UserOverrideFlag_USADDR
CityName_USAREA
StateAbbreviation_USAREA
ZipCode_USAREA
Zip4AddonCode_USAREA
CountryCode_USAREA
CityNameNYSIIS_USAREA
CityNameRVSNDX_USAREA
UnhandledPattern_USAREA
UnhandledData_USAREA
InputPattern_USAREA
ExceptionData_USAREA
UserOverrideFlag_USAREA

IN

Constraint:

Derivation	Column Name
OUT.BoxValue_USADDR	BoxValue_USADDR_HOME
OUT.FloorType_USADDR	FloorType_USADDR_HOME
OUT.FloorValue_USADDR	FloorValue_USADDR_HOME
OUT.UnitType_USADDR	UnitType_USADDR_HOME
OUT.UnitValue_USADDR	UnitValue_USADDR_HOME
OUT.MultiUnitType_USADDR	MultiUnitType_USADDR_HOME
OUT.MultiUnitValue_USADDR	MultiUnitValue_USADDR_HOME
OUT.BuildingName_USADDR	BuildingName_USADDR_HOME
OUT.AdditionalAddress_USADDR	AdditionalAddress_USADDR_HOME
OUT.AddressType_USADDR	AddressType_USADDR_HOME
OUT.StreetNameNYSIIS_USADDR	StreetNameNYSIIS_USADDR_HOME
OUT.StreetNameRVSNDX_USADDR	StreetNameRVSNDX_USADDR_HOME
"	UnhandledPattern_USADDR_HOME
"	UnhandledData_USADDR_HOME
"	InputPattern_USADDR_HOME
"	ExceptionData_USADDR_HOME
"	UserOverrideFlag_USADDR_HOME
OUT.CityName_USAREA	CityName_USAREA_HOME
OUT.StateAbbreviation_USAREA	StateAbbreviation_USAREA_HOME
OUT.ZipCode_USAREA	ZipCode_USAREA_HOME
OUT.Zip4AddonCode_USAREA	Zip4AddonCode_USAREA_HOME
OUT.CountryCode_USAREA	CountryCode_USAREA_HOME
OUT.CityNameNYSIIS_USAREA	CityNameNYSIIS_USAREA_HOME
OUT.CityNameRVSNDX_USAREA	CityNameRVSNDX_USAREA_HOME
"	NCB_CORE_ID
"	NCB_NONCORE_ID
"	NAB_CORE_CUST_ID
"	NAB_CORE_ACC_ID
OUT.ACCOUNT_ID	NAB_NONCORE_ID
OUT.INSURANCE_ID	NAB_NONCORE_INS_ID
OUT.DRIVER_ID	NAB_NONCORE_DRIV_ID
OUT.DPVMatchFlag_CASS	DPVMatchFlag_CASS
OUT.DPVCode1_CASS	DPVCode1_CASS

OUT

	Column name	Key	SQL type	Extended	Length
1	ACCOUNT_ID	☐	Integer		10
2	INSURANCE_ID	☐	Integer		10
3	DRIVER_ID	☐	Integer		10
4	NAME	☐	VarChar		50

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripl
1	CUSTOMER_ID	☐	Integer		10		No	<nor
2	TITLE	☐	Char		3		Yes	<nor
3	FIRST_NAME	☐	VarChar		20		Yes	<nor
4	LAST_NAME	☐	VarChar		20		Yes	<nor

Figure 2-233 Create j13_PREPARE_NAB_DATA_FOR_FUNNEL 16/17

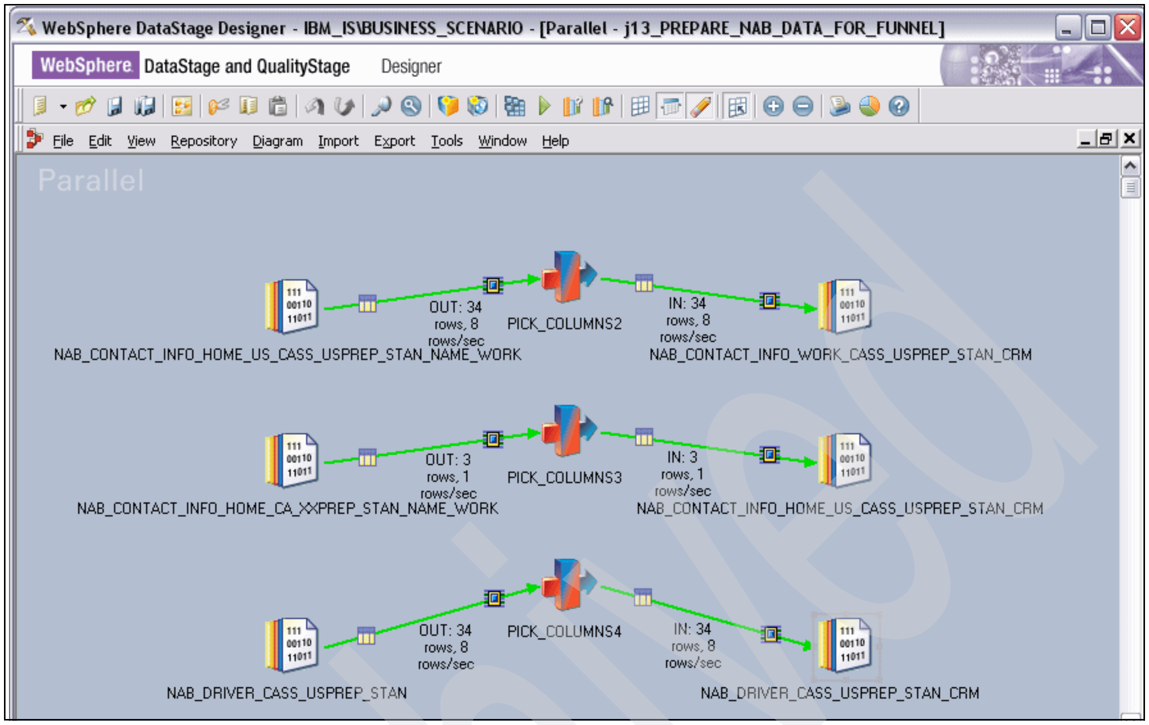

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j13_PREPARE_NAB_DATA_FOR_FUNNEL]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
OUT: 34 rows, 8 rows/sec
PICK_COLUMNS2
IN: 34 rows, 8 rows/sec
NAB_CONTACT_INFO_HOME_US_CASS_USPREP_STAN_NAME_WORK
NAB_CONTACT_INFO_WORK_CASS_USPREP_STAN_CRM
OUT: 3 rows, 1 rows/sec
PICK_COLUMNS3
IN: 3 rows, 1 rows/sec
NAB_CONTACT_INFO_HOME_CA_XXPREP_STAN_NAME_WORK
NAB_CONTACT_INFO_HOME_US_CASS_USPREP_STAN_CRM
OUT: 34 rows, 8 rows/sec
PICK_COLUMNS4
IN: 34 rows, 8 rows/sec
NAB_DRIVER_CASS_USPREP_STAN
NAB_DRIVER_CASS_USPREP_STAN_CRM

Figure 2-234 Create j13_PREPARE_NAB_DATA_FOR_FUNNEL 17/17

j14_FUNNEL_NAB_DATA_FOR_CRM

In this step, we merge the three files that were created in "j13_PREPARE_NAB_DATA_FOR_FUNNEL" on page 687 using the Funnel stage. The output of this step is then included for matching and surviving the "best" data as described in 2.6.3, "Matching and surviving Northern California Bank and Northern California Bank information" on page 783.

Figure 2-218 on page 689 shows the various stages that are used in this job. It includes the data sets that were created in "j13_PREPARE_NAB_DATA_FOR_FUNNEL" on page 687, a Funnel stage, and an output Data Set stage. We modified the stage names as shown.

Because we have discuss these configurations previously, we do not repeat it here.

Figure 2-235 on page 706 shows the results after compiling and running this job.

Proceed now to 2.6.3, "Matching and surviving Northern California Bank and Northern California Bank information" on page 783.

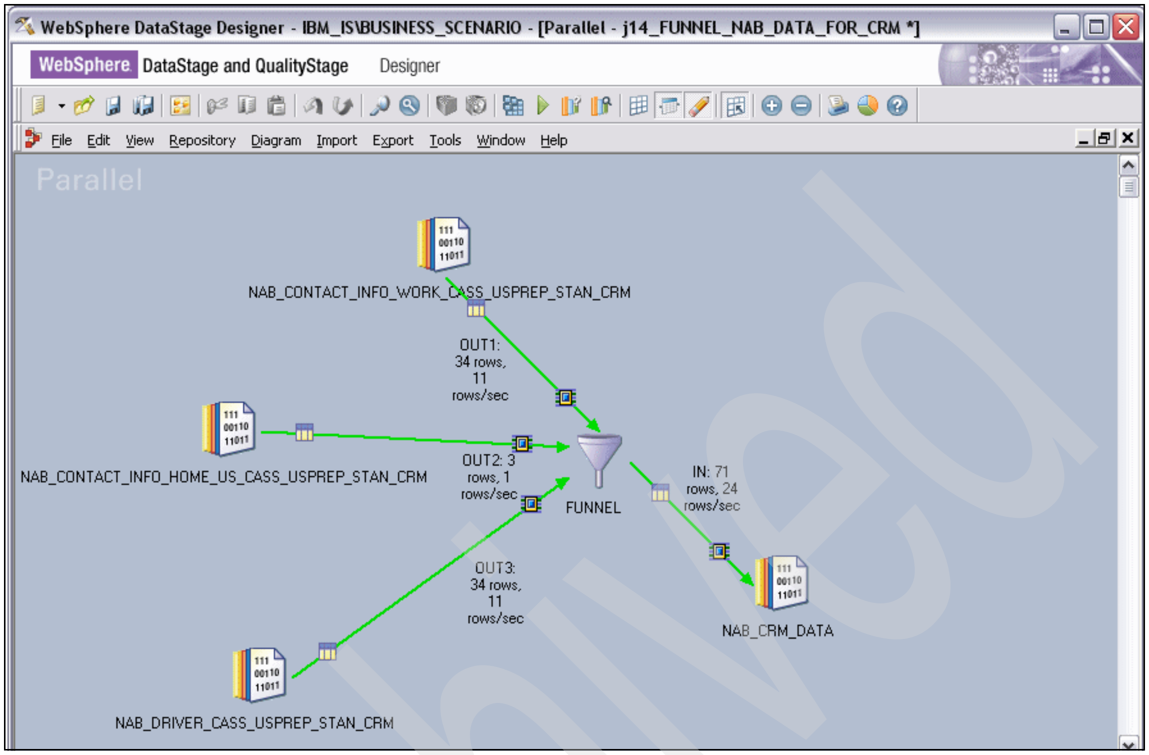

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j14_FUNNEL_NAB_DATA_FOR_CRM *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NAB_CONTACT_INFO_WORK_CASS_USPREP_STAN_CRM
OUT1: 34 rows, 11 rows/sec
NAB_CONTACT_INFO_HOME_US_CASS_USPREP_STAN_CRM
OUT2: 3 rows, 1 rows/sec
FUNNEL
IN: 71 rows, 24 rows/sec
OUT3: 34 rows, 11 rows/sec
NAB_CRM_DATA
NAB_DRIVER_CASS_USPREP_STAN_CRM

Figure 2-235 Create j14_FUNNEL_NAB_DATA_FOR_CRM

2.6.2 Cleansing Northern California Bank's core and non-core services

Figure 2-7 on page 498 shows the processing flow and jobs that are used for cleansing name and addresses in Northern California Bank's core and non-core services.

We describe the steps briefly here:

1. We began by extracting all the CUSTOMER, BCUSTOMER, and BRANCH data from the DB2 database and loading it into data sets to isolate it from changes during analysis. We also introduced a Transformer stage to provide for pre-processing the source for analysis, such as trimming out any extra spaces in the name and address fields using the TRIM function and changing default values to nulls using the IsNull ..then..else function.

   Job "j00_SRC_NCB" on page 709 performs this step.

2. Next, we analyzed the addresses in appropriate data sets (CUSTOMER and BCUSTOMER) to determine their (ISO code) country using the COUNTRY rule set in the Standardize stage.

   > **Note:** The CRM system has only the name of the BRANCH included in its data model (see Figure 2-3 on page 485). Because BRANCH was a very small file, we visually verified and corrected the name and address fields in the records without using IBM WebSphere QualityStage.

   Job “j01_STAN_COUNTRY_NCB” on page 719 performs this step.

3. The ISO codes that were generated by the previous step were analyzed by the Investigate stage using the character concatenate investigate option with the *C* mask to obtain frequency distribution. This step identified whether the addresses in the appropriate data sets belonged to more than one country, and identified the codes of the countries in the addresses.

   In this case, the addresses were all U.S. addresses.

   Job “j02_INVCD_ISOCODE_NCB” on page 726 performs this step.

4. Next, we used the Standardize stage with the USPREP domain-preprocessor rule set to move name and address data into Name, Address, and Area domains for the CUSTOMER and BCUSTOMER files.

   Job “j03_STAN_USPREP_NCB” on page 729 performs this step.

5. We then used the Investigate stage using the character concatenate investigate option on the input patterns generated by the Standardize stage in the previous step, to determine the degree of success achieved by the USPREP domain-preprocessor rule set in parsing the tokens in the name/address fields into the correct domains.

   A visual analysis of the pattern reports that were generated by the Investigate stage indicated no errors in the movement of data to the name, address, and area domains. However, some spelling errors (such as 321 Curie Drivee of Torben Anderson) were detected, which we decided to defer addressing till after subsequent stages. We, therefore, proceeded to the next step, “j05_CASS_USPREP_NCB” on page 742.

   Job “j04_INVCC_USPREP_INPUT_PATTERN_NCB” on page 735 performs this step.

6. Next, we used the CASS stage to validate, correct, and standardize the U.S. addresses in the Address domain. We included a Transformer stage to add a second address line column to customer file because CASS requires two address lines as input for its processing.

Note: CASS is a separately priced module that requires installation of the CASS module.

Job "j05_CASS_USPREP_NCB" on page 742 performs this step.

7. We then ran the Investigate stage with character concatenate option on the results of job "j05_CASS_USPREP_NCB" on page 742 step to determine addresses not recognized by CASS (delivery point verification or DPV).

   Note: Due to a bug with handling nulls, we introduced a Transformer stage to convert nulls to a space using column derivation.

   We investigated the generated columns DPVMATCHFLAG_CASS and DPVCODE1_CASS using a *C* mask. A value of `A1` in the DPVCODE1_CASS field indicated a potential problem.

   Job "j06_INVCC_CASS_NCB" on page 752 performs this step.

8. The next step standardized the address contents of the output of job "j05_CASS_USPREP_NCB" on page 742 using the domain-preprocessor USPREP rule set. This parsed and moved the name, address, and area content from selected input columns to the AddressDomain_USPREP and AreaDomain_USPREP columns. We also added a column to the output that only had the first three characters of the ZIP code using a Transformer stage. This new column (ZIP3) was used as a blocking variable in an upcoming matching stage.

   Job "j07_PREP_CASS_NCB" on page 755 performs this step.

9. In this step, we standardized the name and address contents of the output of job "j07_PREP_CASS_NCB" on page 755 using the domain-specific rule sets USNAME (with column NameDomain_USPREP), USADDR (with column AddressDomain_USPREP), and USAREA (with column AreaDomain_USPREP). Separate processes were defined—one for each rule set.

   Note: Here again, we had to introduce a null handling Transformer stage to circumvent a bug.

   Job "j08_STAN_CASS_NCB" on page 763 performs this step.

10. The next step identified unhandled patterns and classifications in the previous step in the CUSTOMER and BCUSTOMER files. We ran a series of Investigate stage with the character concatenate option using the *C* mask on the unhandled pattern column from the results of "j08_STAN_CASS_NCB" on

page 763. The columns investigated corresponded to the name, address, and area domains. Unhandled patterns were reviewed and classification and input pattern overrides were generated to rectify the problem. The Standardize stage was then rerun with the overrides in place and verified using Investigate that all the unhandled patterns had been resolved.

Jobs "j09a_INVCC_CUSTOMER_STAN_CASS_NCB" on page 765 and "j09b_INVCC_BCUSTOMER_STAN_CASS_NCB" on page 771 perform these steps.

11. After the unhandled patterns were handled, we merged the standardized name and address file of the CUSTOMER and BCUSTOMER in order to identify duplicates (with data from the North American Bank's standardized name and addresses from their core and non-core services as described in 2.6.1, "Cleansing North American Bank's core and non-core services" on page 508) and survive the "best" data into the CRM system as described in 2.6.3, "Matching and surviving Northern California Bank and Northern California Bank information" on page 783.

    Prior to merging the two files that were created in the "j08_STAN_CASS_NCB" on page 763 step, we needed to prepare the two files for merging in the Funnel stage by ensuring that all the files have the same number of columns and same names of columns using a Transformer stage.

    Job "j10_PREPARE_NCB_DATA_FOR_FUNNEL" on page 773 performs this step.

12. In this step, we merged the two files that were created in the "j10_PREPARE_NCB_DATA_FOR_FUNNEL" on page 773 step using the Funnel stage.

    Job "j11_FUNNEL_NCB_DATA" on page 782 performs this step.

    The output of this step was then included for matching and surviving the "best" data as described in 2.6.3, "Matching and surviving Northern California Bank and Northern California Bank information" on page 783.

These jobs are described in more detail in the following sections.

j00_SRC_NCB

We begin by extracting all the CUSTOMER, BCUSTOMER, and BRANCH data from the DB2 database and loading it into data sets to isolate it from changes during analysis. We also introduce a Transformer stage to provide for pre-processing the source for analysis such as trimming out any extra spaces in the name and address fields using the TRIM function and changing default values to nulls using the IsNull ..then..else function.

Figure 2-236 on page 711 shows the various stages that are used in this job, including a DB2 UDB API stage used to access the data in the CUSTOMER, BCUSTOMER, and BRANCH tables, a Transformer stage for every input table to remove blanks from name and address text fields, and an output Data Set stage for input table accessed. We modified the names of the stages as shown.

Because the configuration of this job is very similar to that described in "J00_SRC_CUSTOMER" on page 142, we do not repeat it here. However, some of the configurations of interest are as follows:

- Figure 2-237 on page 712 shows the configuration of the TRIM_FIELDS - Transformer Stage that removes trailing blanks from name and address columns retrieved from the CUSTOMER table using the TRIMOUT function. We also replaced nulls in date fields (such as the UPDATED column) with a value of `1901-01-01` and moved all the input columns directly to the output. We do not show the corresponding configurations for BCUSTOMER and BRANCH here.
- After saving, compiling, and running the job (not shown here), the results are shown in Figure 2-238 on page 713 through Figure 2-244 on page 719. The CUSTOMER file had 50 records, the BCUSTOMER file had 39 records, and the BRANCH file had 37 records.

Proceed now to "j01_STAN_COUNTRY_NCB" on page 719.

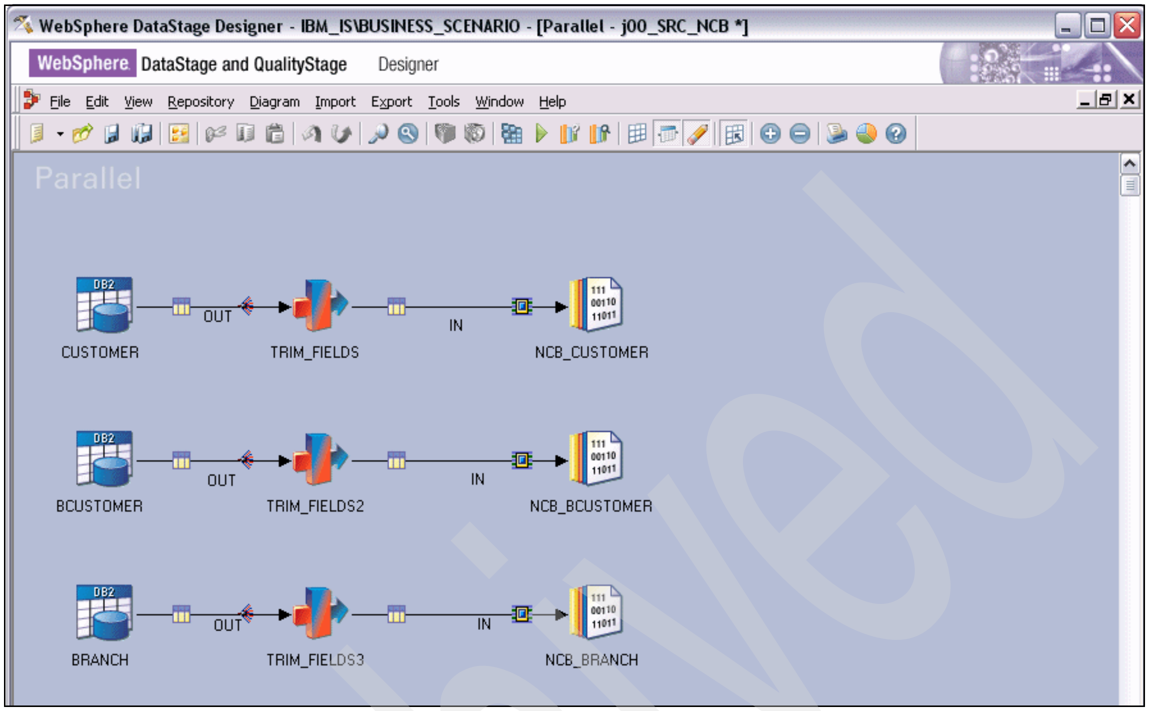
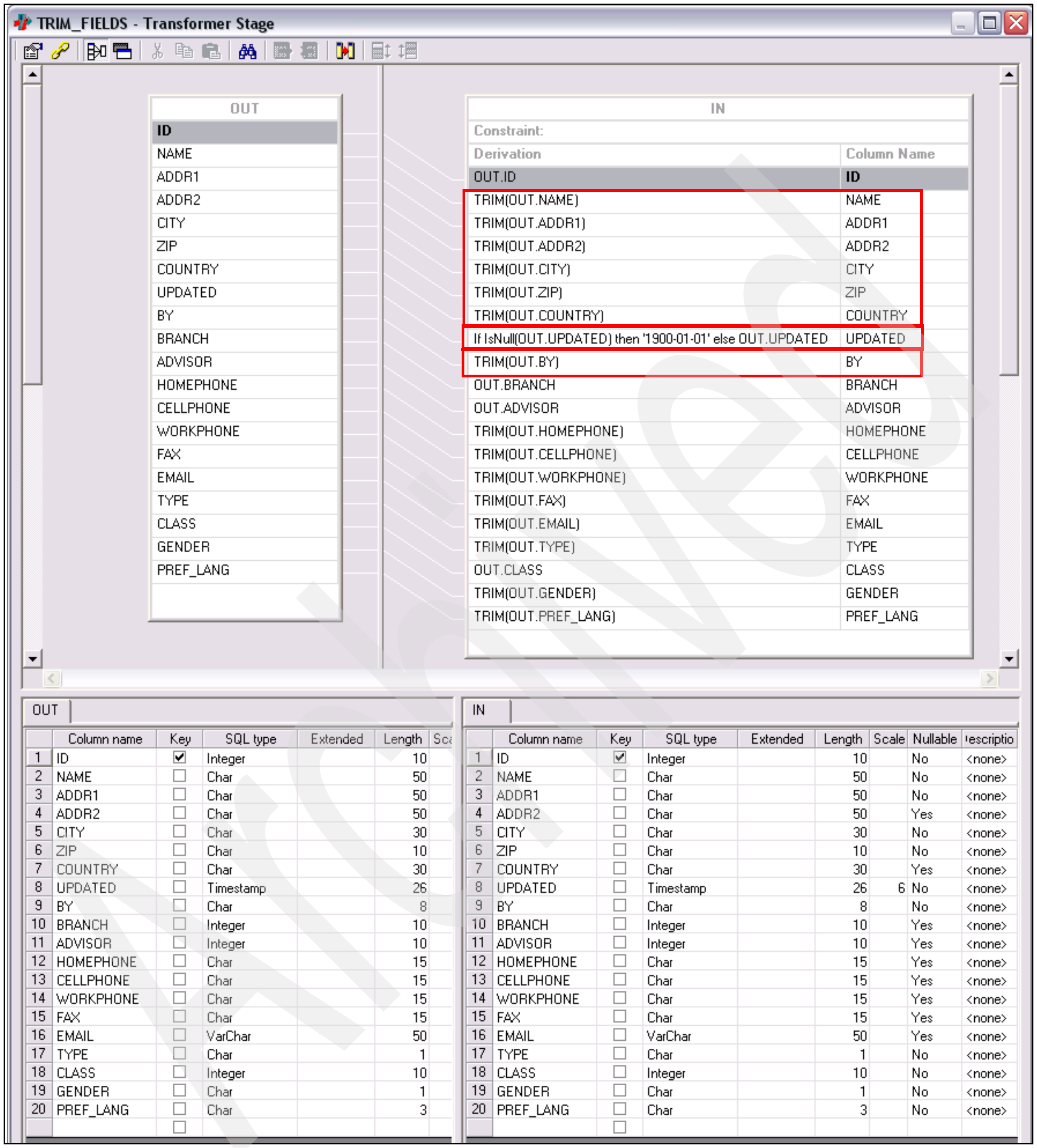

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j00_SRC_NCB *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
CUSTOMER
OUT
TRIM_FIELDS
IN
NCB_CUSTOMER
BCUSTOMER
OUT
TRIM_FIELDS2
IN
NCB_BCUSTOMER
BRANCH
OUT
TRIM_FIELDS3
IN
NCB_BRANCH

Figure 2-236 Create j00_SRC_NCB 1/9


TRIM_FIELDS - Transformer Stage

OUT
ID
NAME
ADDR1
ADDR2
CITY
ZIP
COUNTRY
UPDATED
BY
BRANCH
ADVISOR
HOMEPHONE
CELLPHONE
WORKPHONE
FAX
EMAIL
TYPE
CLASS
GENDER
PREF_LANG

IN

Constraint:

Derivation	Column Name
OUT.ID	ID
TRIM(OUT.NAME)	NAME
TRIM(OUT.ADDR1)	ADDR1
TRIM(OUT.ADDR2)	ADDR2
TRIM(OUT.CITY)	CITY
TRIM(OUT.ZIP)	ZIP
TRIM(OUT.COUNTRY)	COUNTRY
If IsNull(OUT.UPDATED) then '1900-01-01' else OUT.UPDATED	UPDATED
TRIM(OUT.BY)	BY
OUT.BRANCH	BRANCH
OUT.ADVISOR	ADVISOR
TRIM(OUT.HOMEPHONE)	HOMEPHONE
TRIM(OUT.CELLPHONE)	CELLPHONE
TRIM(OUT.WORKPHONE)	WORKPHONE
TRIM(OUT.FAX)	FAX
TRIM(OUT.EMAIL)	EMAIL
TRIM(OUT.TYPE)	TYPE
OUT.CLASS	CLASS
TRIM(OUT.GENDER)	GENDER
TRIM(OUT.PREF_LANG)	PREF_LANG

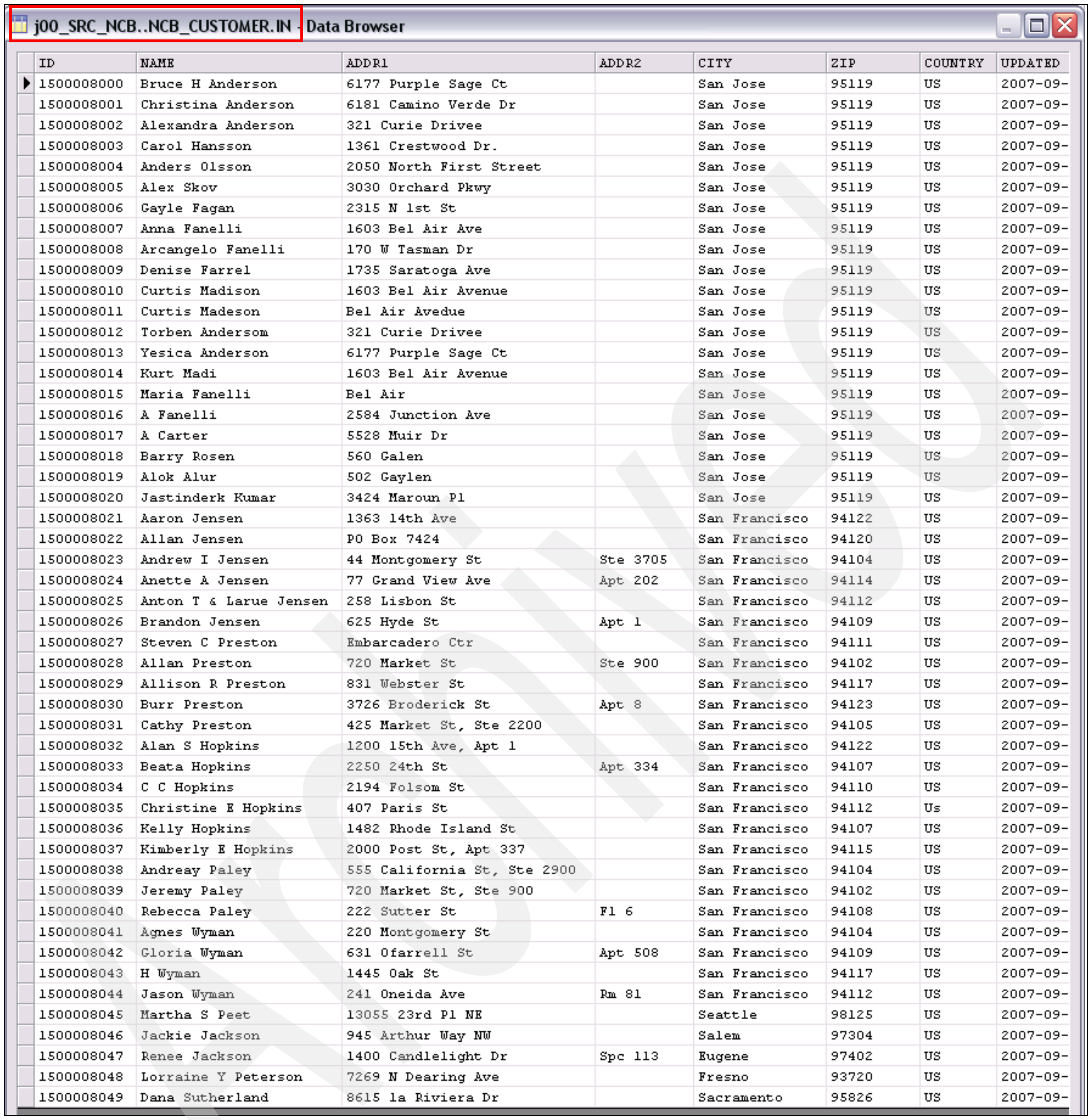
OUT

	Column name	Key	SQL type	Extended	Length	Sca
1	ID	✓	Integer		10	
2	NAME	☐	Char		50	
3	ADDR1	☐	Char		50	
4	ADDR2	☐	Char		50	
5	CITY	☐	Char		30	
6	ZIP	☐	Char		10	
7	COUNTRY	☐	Char		30	
8	UPDATED	☐	Timestamp		26	
9	BY	☐	Char		8	
10	BRANCH	☐	Integer		10	
11	ADVISOR	☐	Integer		10	
12	HOMEPHONE	☐	Char		15	
13	CELLPHONE	☐	Char		15	
14	WORKPHONE	☐	Char		15	
15	FAX	☐	Char		15	
16	EMAIL	☐	VarChar		50	
17	TYPE	☐	Char		1	
18	CLASS	☐	Integer		10	
19	GENDER	☐	Char		1	
20	PREF_LANG	☐	Char		3	

IN

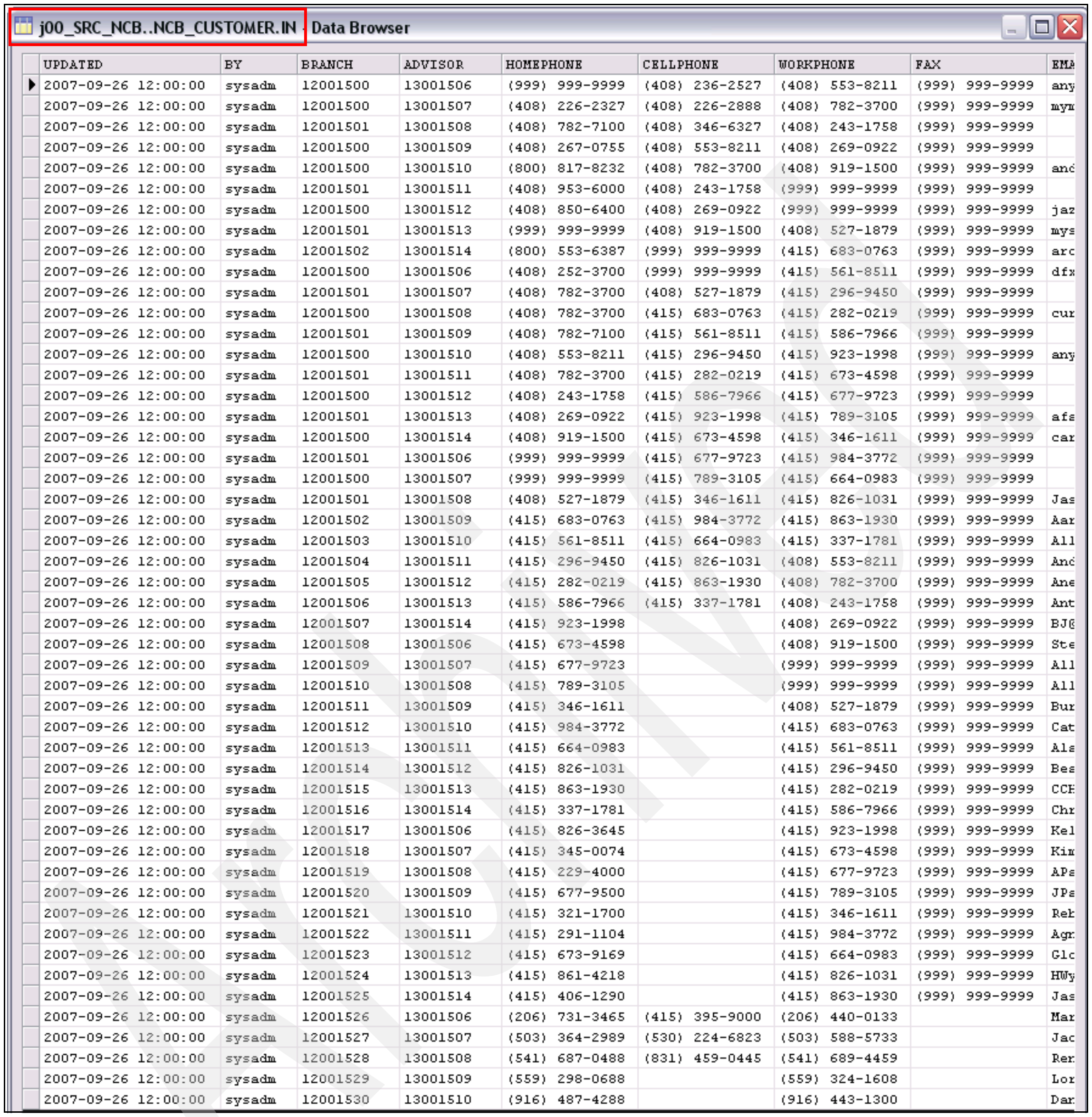
	Column name	Key	SQL type	Extended	Length	Scale	Nullable	escriptio
1	ID	✓	Integer		10		No	<none>
2	NAME	☐	Char		50		No	<none>
3	ADDR1	☐	Char		50		No	<none>
4	ADDR2	☐	Char		50		Yes	<none>
5	CITY	☐	Char		30		No	<none>
6	ZIP	☐	Char		10		No	<none>
7	COUNTRY	☐	Char		30		Yes	<none>
8	UPDATED	☐	Timestamp		26	6	No	<none>
9	BY	☐	Char		8		No	<none>
10	BRANCH	☐	Integer		10		Yes	<none>
11	ADVISOR	☐	Integer		10		Yes	<none>
12	HOMEPHONE	☐	Char		15		Yes	<none>
13	CELLPHONE	☐	Char		15		Yes	<none>
14	WORKPHONE	☐	Char		15		Yes	<none>
15	FAX	☐	Char		15		Yes	<none>
16	EMAIL	☐	VarChar		50		Yes	<none>
17	TYPE	☐	Char		1		No	<none>
18	CLASS	☐	Integer		10		No	<none>
19	GENDER	☐	Char		1		No	<none>
20	PREF_LANG	☐	Char		3		No	<none>

Figure 2-237 Create j00_SRC_NCB 2/9


j00_SRC_NCB..NCB_CUSTOMER.IN - Data Browser

ID	NAME	ADDR1	ADDR2	CITY	ZIP	COUNTRY	UPDATED
1500008000	Bruce H Anderson	6177 Purple Sage Ct		San Jose	95119	US	2007-09-
1500008001	Christina Anderson	6181 Camino Verde Dr		San Jose	95119	US	2007-09-
1500008002	Alexandra Anderson	321 Curie Drivee		San Jose	95119	US	2007-09-
1500008003	Carol Hansson	1361 Crestwood Dr.		San Jose	95119	US	2007-09-
1500008004	Anders Olsson	2050 North First Street		San Jose	95119	US	2007-09-
1500008005	Alex Skov	3030 Orchard Pkwy		San Jose	95119	US	2007-09-
1500008006	Gayle Fagan	2315 N 1st St		San Jose	95119	US	2007-09-
1500008007	Anna Fanelli	1603 Bel Air Ave		San Jose	95119	US	2007-09-
1500008008	Arcangelo Fanelli	170 W Tasman Dr		San Jose	95119	US	2007-09-
1500008009	Denise Farrel	1735 Saratoga Ave		San Jose	95119	US	2007-09-
1500008010	Curtis Madison	1603 Bel Air Avenue		San Jose	95119	US	2007-09-
1500008011	Curtis Madeson	Bel Air Avedue		San Jose	95119	US	2007-09-
1500008012	Torben Andersom	321 Curie Drivee		San Jose	95119	US	2007-09-
1500008013	Yesica Anderson	6177 Purple Sage Ct		San Jose	95119	US	2007-09-
1500008014	Kurt Madi	1603 Bel Air Avenue		San Jose	95119	US	2007-09-
1500008015	Maria Fanelli	Bel Air		San Jose	95119	US	2007-09-
1500008016	A Fanelli	2584 Junction Ave		San Jose	95119	US	2007-09-
1500008017	A Carter	5528 Muir Dr		San Jose	95119	US	2007-09-
1500008018	Barry Rosen	560 Galen		San Jose	95119	US	2007-09-
1500008019	Alok Alur	502 Gaylen		San Jose	95119	US	2007-09-
1500008020	Jastinderk Kumar	3424 Maroun Pl		San Jose	95119	US	2007-09-
1500008021	Aaron Jensen	1363 14th Ave		San Francisco	94122	US	2007-09-
1500008022	Allan Jensen	PO Box 7424		San Francisco	94120	US	2007-09-
1500008023	Andrew I Jensen	44 Montgomery St	Ste 3705	San Francisco	94104	US	2007-09-
1500008024	Anette A Jensen	77 Grand View Ave	Apt 202	San Francisco	94114	US	2007-09-
1500008025	Anton T & Larue Jensen	258 Lisbon St		San Francisco	94112	US	2007-09-
1500008026	Brandon Jensen	625 Hyde St	Apt 1	San Francisco	94109	US	2007-09-
1500008027	Steven C Preston	Embarcadero Ctr		San Francisco	94111	US	2007-09-
1500008028	Allan Preston	720 Market St	Ste 900	San Francisco	94102	US	2007-09-
1500008029	Allison R Preston	831 Webster St		San Francisco	94117	US	2007-09-
1500008030	Burr Preston	3726 Broderick St	Apt 8	San Francisco	94123	US	2007-09-
1500008031	Cathy Preston	425 Market St, Ste 2200		San Francisco	94105	US	2007-09-
1500008032	Alan S Hopkins	1200 15th Ave, Apt 1		San Francisco	94122	US	2007-09-
1500008033	Beata Hopkins	2250 24th St	Apt 334	San Francisco	94107	US	2007-09-
1500008034	C C Hopkins	2194 Folsom St		San Francisco	94110	US	2007-09-
1500008035	Christine E Hopkins	407 Paris St		San Francisco	94112	Us	2007-09-
1500008036	Kelly Hopkins	1482 Rhode Island St		San Francisco	94107	US	2007-09-
1500008037	Kimberly E Hopkins	2000 Post St, Apt 337		San Francisco	94115	US	2007-09-
1500008038	Andreay Paley	555 California St, Ste 2900		San Francisco	94104	US	2007-09-
1500008039	Jeremy Paley	720 Market St, Ste 900		San Francisco	94102	US	2007-09-
1500008040	Rebecca Paley	222 Sutter St	Fl 6	San Francisco	94108	US	2007-09-
1500008041	Agnes Wyman	220 Montgomery St		San Francisco	94104	US	2007-09-
1500008042	Gloria Wyman	631 Ofarrell St	Apt 508	San Francisco	94109	US	2007-09-
1500008043	H Wyman	1445 Oak St		San Francisco	94117	US	2007-09-
1500008044	Jason Wyman	241 Oneida Ave	Rm 81	San Francisco	94112	US	2007-09-
1500008045	Martha S Peet	13055 23rd Pl NE		Seattle	98125	US	2007-09-
1500008046	Jackie Jackson	945 Arthur Way NW		Salem	97304	US	2007-09-
1500008047	Renee Jackson	1400 Candlelight Dr	Spc 113	Eugene	97402	US	2007-09-
1500008048	Lorraine Y Peterson	7269 N Dearing Ave		Fresno	93720	US	2007-09-
1500008049	Dana Sutherland	8615 la Riviera Dr		Sacramento	95826	US	2007-09-

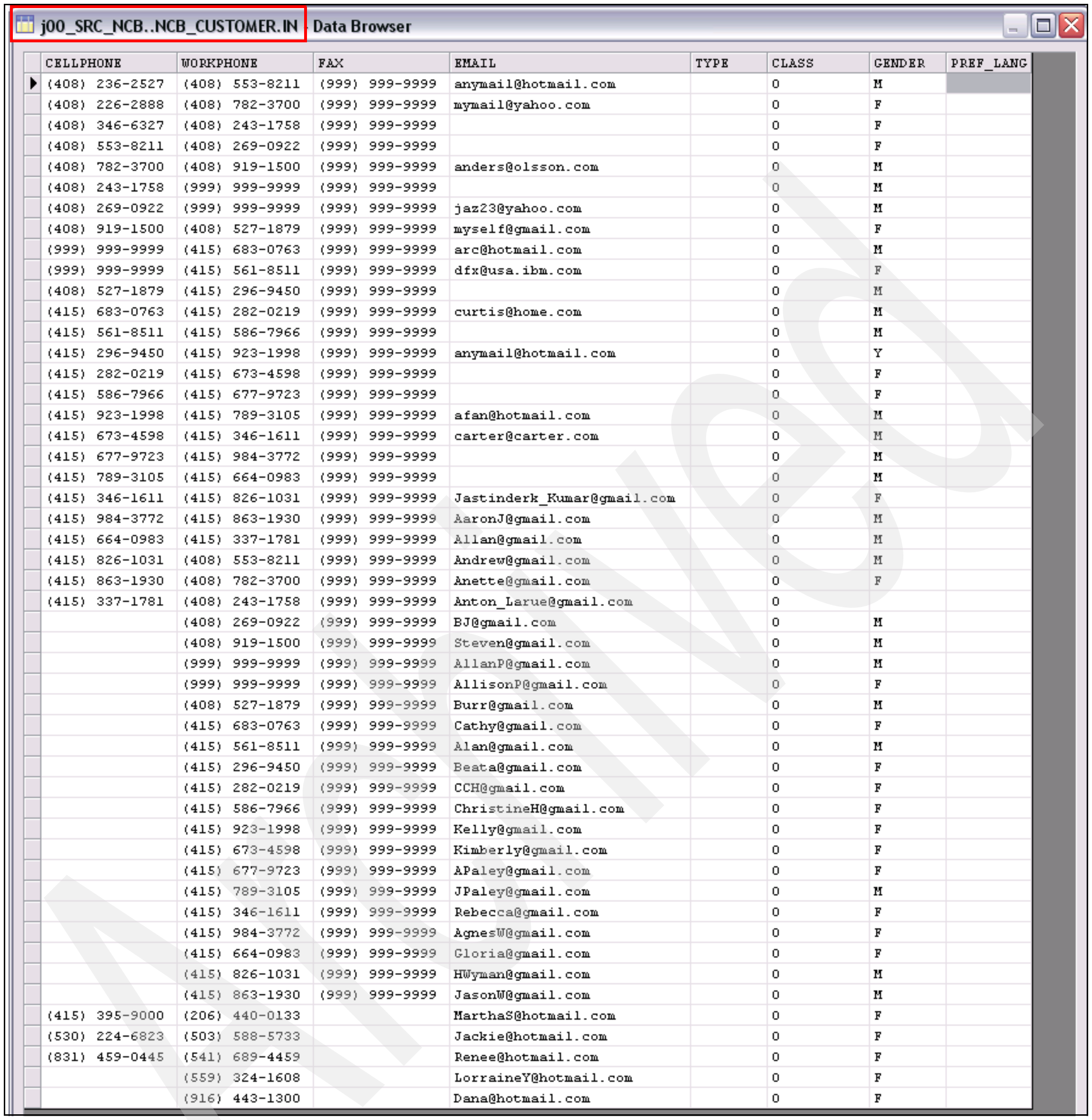
Figure 2-238 Create j00_SRC_NCB 3/9



j00_SRC_NCB..NCB_CUSTOMER.IN - Data Browser

UPDATED	BY	BRANCH	ADVISOR	HOMEPHONE	CELLPHONE	WORKPHONE	FAX	EMA
2007-09-26 12:00:00	sysadm	12001500	13001506	(999) 999-9999	(408) 236-2527	(408) 553-8211	(999) 999-9999	any
2007-09-26 12:00:00	sysadm	12001500	13001507	(408) 226-2327	(408) 226-2888	(408) 782-3700	(999) 999-9999	mym
2007-09-26 12:00:00	sysadm	12001501	13001508	(408) 782-7100	(408) 346-6327	(408) 243-1758	(999) 999-9999	
2007-09-26 12:00:00	sysadm	12001500	13001509	(408) 267-0755	(408) 553-8211	(408) 269-0922	(999) 999-9999	
2007-09-26 12:00:00	sysadm	12001500	13001510	(800) 817-8232	(408) 782-3700	(408) 919-1500	(999) 999-9999	and
2007-09-26 12:00:00	sysadm	12001501	13001511	(408) 953-6000	(408) 243-1758	(999) 999-9999	(999) 999-9999	
2007-09-26 12:00:00	sysadm	12001500	13001512	(408) 850-6400	(408) 269-0922	(999) 999-9999	(999) 999-9999	jaz
2007-09-26 12:00:00	sysadm	12001501	13001513	(999) 999-9999	(408) 919-1500	(408) 527-1879	(999) 999-9999	mys
2007-09-26 12:00:00	sysadm	12001502	13001514	(800) 553-6387	(999) 999-9999	(415) 683-0763	(999) 999-9999	arc
2007-09-26 12:00:00	sysadm	12001500	13001506	(408) 252-3700	(999) 999-9999	(415) 561-8511	(999) 999-9999	dfx
2007-09-26 12:00:00	sysadm	12001501	13001507	(408) 782-3700	(408) 527-1879	(415) 296-9450	(999) 999-9999	
2007-09-26 12:00:00	sysadm	12001500	13001508	(408) 782-3700	(415) 683-0763	(415) 282-0219	(999) 999-9999	cur
2007-09-26 12:00:00	sysadm	12001501	13001509	(408) 782-7100	(415) 561-8511	(415) 586-7966	(999) 999-9999	
2007-09-26 12:00:00	sysadm	12001500	13001510	(408) 553-8211	(415) 296-9450	(415) 923-1998	(999) 999-9999	any
2007-09-26 12:00:00	sysadm	12001501	13001511	(408) 782-3700	(415) 282-0219	(415) 673-4598	(999) 999-9999	
2007-09-26 12:00:00	sysadm	12001500	13001512	(408) 243-1758	(415) 586-7966	(415) 677-9723	(999) 999-9999	
2007-09-26 12:00:00	sysadm	12001501	13001513	(408) 269-0922	(415) 923-1998	(415) 789-3105	(999) 999-9999	afs
2007-09-26 12:00:00	sysadm	12001500	13001514	(408) 919-1500	(415) 673-4598	(415) 346-1611	(999) 999-9999	car
2007-09-26 12:00:00	sysadm	12001501	13001506	(999) 999-9999	(415) 677-9723	(415) 984-3772	(999) 999-9999	
2007-09-26 12:00:00	sysadm	12001500	13001507	(999) 999-9999	(415) 789-3105	(415) 664-0983	(999) 999-9999	
2007-09-26 12:00:00	sysadm	12001501	13001508	(408) 527-1879	(415) 346-1611	(415) 826-1031	(999) 999-9999	Jas
2007-09-26 12:00:00	sysadm	12001502	13001509	(415) 683-0763	(415) 984-3772	(415) 863-1930	(999) 999-9999	Aar
2007-09-26 12:00:00	sysadm	12001503	13001510	(415) 561-8511	(415) 664-0983	(415) 337-1781	(999) 999-9999	All
2007-09-26 12:00:00	sysadm	12001504	13001511	(415) 296-9450	(415) 826-1031	(408) 553-8211	(999) 999-9999	And
2007-09-26 12:00:00	sysadm	12001505	13001512	(415) 282-0219	(415) 863-1930	(408) 782-3700	(999) 999-9999	Ane
2007-09-26 12:00:00	sysadm	12001506	13001513	(415) 586-7966	(415) 337-1781	(408) 243-1758	(999) 999-9999	Ant
2007-09-26 12:00:00	sysadm	12001507	13001514	(415) 923-1998		(408) 269-0922	(999) 999-9999	BJ@
2007-09-26 12:00:00	sysadm	12001508	13001506	(415) 673-4598		(408) 919-1500	(999) 999-9999	Ste
2007-09-26 12:00:00	sysadm	12001509	13001507	(415) 677-9723		(999) 999-9999	(999) 999-9999	All
2007-09-26 12:00:00	sysadm	12001510	13001508	(415) 789-3105		(999) 999-9999	(999) 999-9999	All
2007-09-26 12:00:00	sysadm	12001511	13001509	(415) 346-1611		(408) 527-1879	(999) 999-9999	Bur
2007-09-26 12:00:00	sysadm	12001512	13001510	(415) 984-3772		(415) 683-0763	(999) 999-9999	Cat
2007-09-26 12:00:00	sysadm	12001513	13001511	(415) 664-0983		(415) 561-8511	(999) 999-9999	Als
2007-09-26 12:00:00	sysadm	12001514	13001512	(415) 826-1031		(415) 296-9450	(999) 999-9999	Bes
2007-09-26 12:00:00	sysadm	12001515	13001513	(415) 863-1930		(415) 282-0219	(999) 999-9999	CCH
2007-09-26 12:00:00	sysadm	12001516	13001514	(415) 337-1781		(415) 586-7966	(999) 999-9999	Chr
2007-09-26 12:00:00	sysadm	12001517	13001506	(415) 826-3645		(415) 923-1998	(999) 999-9999	Kel
2007-09-26 12:00:00	sysadm	12001518	13001507	(415) 345-0074		(415) 673-4598	(999) 999-9999	Kim
2007-09-26 12:00:00	sysadm	12001519	13001508	(415) 229-4000		(415) 677-9723	(999) 999-9999	APs
2007-09-26 12:00:00	sysadm	12001520	13001509	(415) 677-9500		(415) 789-3105	(999) 999-9999	JPs
2007-09-26 12:00:00	sysadm	12001521	13001510	(415) 321-1700		(415) 346-1611	(999) 999-9999	Reb
2007-09-26 12:00:00	sysadm	12001522	13001511	(415) 291-1104		(415) 984-3772	(999) 999-9999	Agr
2007-09-26 12:00:00	sysadm	12001523	13001512	(415) 673-9169		(415) 664-0983	(999) 999-9999	Glc
2007-09-26 12:00:00	sysadm	12001524	13001513	(415) 861-4218		(415) 826-1031	(999) 999-9999	HWy
2007-09-26 12:00:00	sysadm	12001525	13001514	(415) 406-1290		(415) 863-1930	(999) 999-9999	Jas
2007-09-26 12:00:00	sysadm	12001526	13001506	(206) 731-3465	(415) 395-9000	(206) 440-0133		Mar
2007-09-26 12:00:00	sysadm	12001527	13001507	(503) 364-2989	(530) 224-6823	(503) 588-5733		Jac
2007-09-26 12:00:00	sysadm	12001528	13001508	(541) 687-0488	(831) 459-0445	(541) 689-4459		Ren
2007-09-26 12:00:00	sysadm	12001529	13001509	(559) 298-0688		(559) 324-1608		Lor
2007-09-26 12:00:00	sysadm	12001530	13001510	(916) 487-4288		(916) 443-1300		Dar

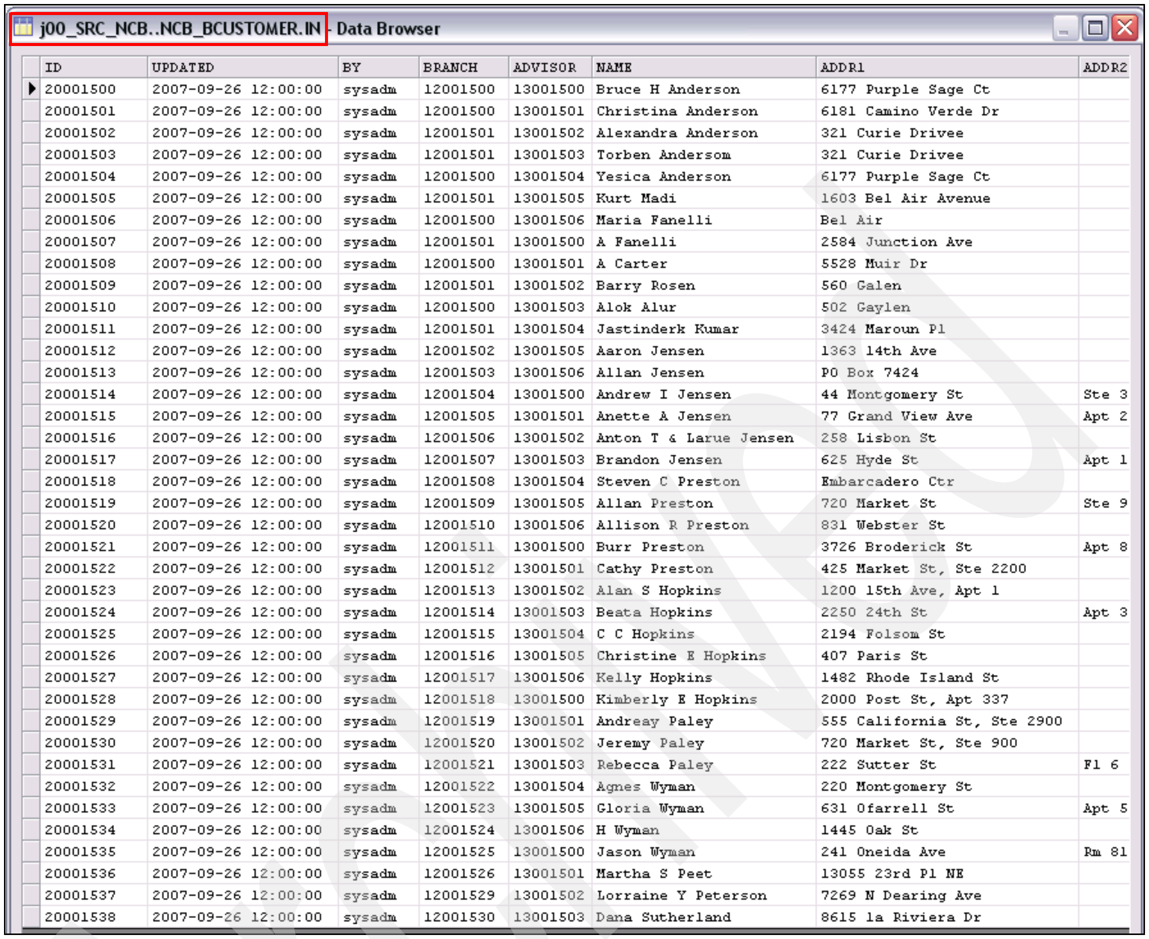
Figure 2-239 Create j00_SRC_NCB 4/9



j00_SRC_NCB..NCB_CUSTOMER.IN Data Browser

CELLPHONE	WORKPHONE	FAX	EMAIL	TYPE	CLASS	GENDER	PREF_LANG
(408) 236-2527	(408) 553-8211	(999) 999-9999	anymail@hotmail.com		0	M	
(408) 226-2888	(408) 782-3700	(999) 999-9999	mymail@yahoo.com		0	F	
(408) 346-6327	(408) 243-1758	(999) 999-9999			0	F	
(408) 553-8211	(408) 269-0922	(999) 999-9999			0	F	
(408) 782-3700	(408) 919-1500	(999) 999-9999	anders@olsson.com		0	M	
(408) 243-1758	(999) 999-9999	(999) 999-9999			0	M	
(408) 269-0922	(999) 999-9999	(999) 999-9999	jaz23@yahoo.com		0	M	
(408) 919-1500	(408) 527-1879	(999) 999-9999	myself@gmail.com		0	F	
(999) 999-9999	(415) 683-0763	(999) 999-9999	arc@hotmail.com		0	M	
(999) 999-9999	(415) 561-8511	(999) 999-9999	dfx@usa.ibm.com		0	F	
(408) 527-1879	(415) 296-9450	(999) 999-9999			0	M	
(415) 683-0763	(415) 282-0219	(999) 999-9999	curtis@home.com		0	M	
(415) 561-8511	(415) 586-7966	(999) 999-9999			0	M	
(415) 296-9450	(415) 923-1998	(999) 999-9999	anymail@hotmail.com		0	Y	
(415) 282-0219	(415) 673-4598	(999) 999-9999			0	F	
(415) 586-7966	(415) 677-9723	(999) 999-9999			0	F	
(415) 923-1998	(415) 789-3105	(999) 999-9999	afan@hotmail.com		0	M	
(415) 673-4598	(415) 346-1611	(999) 999-9999	carter@carter.com		0	M	
(415) 677-9723	(415) 984-3772	(999) 999-9999			0	M	
(415) 789-3105	(415) 664-0983	(999) 999-9999			0	M	
(415) 346-1611	(415) 826-1031	(999) 999-9999	Jastinderk_Kumar@gmail.com		0	F	
(415) 984-3772	(415) 863-1930	(999) 999-9999	AaronJ@gmail.com		0	M	
(415) 664-0983	(415) 337-1781	(999) 999-9999	Allan@gmail.com		0	M	
(415) 826-1031	(408) 553-8211	(999) 999-9999	Andrew@gmail.com		0	M	
(415) 863-1930	(408) 782-3700	(999) 999-9999	Anette@gmail.com		0	F	
(415) 337-1781	(408) 243-1758	(999) 999-9999	Anton_Larue@gmail.com		0		
	(408) 269-0922	(999) 999-9999	BJ@gmail.com		0	M	
	(408) 919-1500	(999) 999-9999	Steven@gmail.com		0	M	
	(999) 999-9999	(999) 999-9999	AllanP@gmail.com		0	M	
	(999) 999-9999	(999) 999-9999	AllisonP@gmail.com		0	F	
	(408) 527-1879	(999) 999-9999	Burr@gmail.com		0	M	
	(415) 683-0763	(999) 999-9999	Cathy@gmail.com		0	F	
	(415) 561-8511	(999) 999-9999	Alan@gmail.com		0	M	
	(415) 296-9450	(999) 999-9999	Beata@gmail.com		0	F	
	(415) 282-0219	(999) 999-9999	CCH@gmail.com		0	F	
	(415) 586-7966	(999) 999-9999	ChristineH@gmail.com		0	F	
	(415) 923-1998	(999) 999-9999	Kelly@gmail.com		0	F	
	(415) 673-4598	(999) 999-9999	Kimberly@gmail.com		0	F	
	(415) 677-9723	(999) 999-9999	APaley@gmail.com		0	F	
	(415) 789-3105	(999) 999-9999	JPaley@gmail.com		0	M	
	(415) 346-1611	(999) 999-9999	Rebecca@gmail.com		0	F	
	(415) 984-3772	(999) 999-9999	AgnesW@gmail.com		0	F	
	(415) 664-0983	(999) 999-9999	Gloria@gmail.com		0	F	
	(415) 826-1031	(999) 999-9999	HWyman@gmail.com		0	M	
	(415) 863-1930	(999) 999-9999	JasonW@gmail.com		0	M	
(415) 395-9000	(206) 440-0133		MarthaS@hotmail.com		0	F	
(530) 224-6823	(503) 588-5733		Jackie@hotmail.com		0	F	
(831) 459-0445	(541) 689-4459		Renee@hotmail.com		0	F	
	(559) 324-1608		LorraineY@hotmail.com		0	F	
	(916) 443-1300		Dana@hotmail.com		0	F	

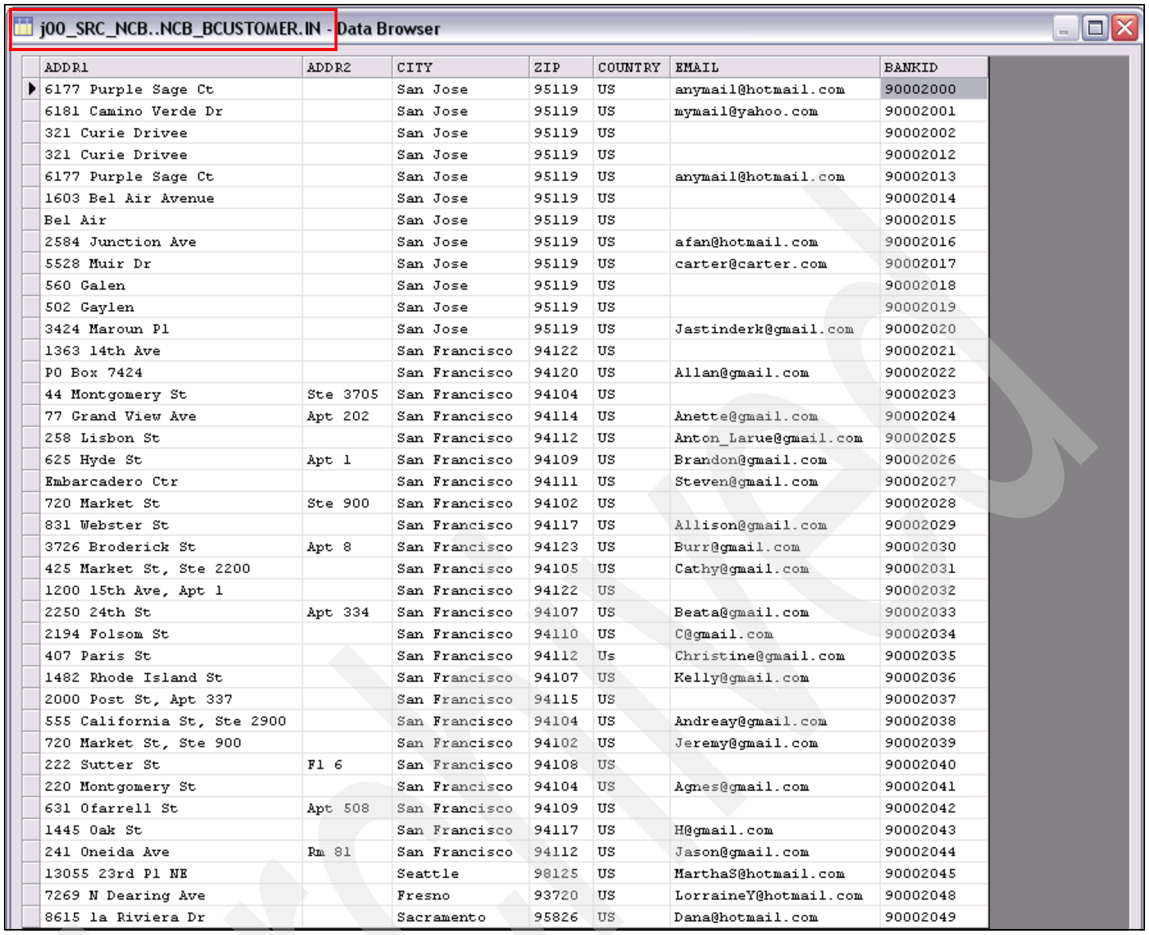
Figure 2-240 Create j00_SRC_NCB 5/9



j00_SRC_NCB..NCB_BCUSTOMER.IN - Data Browser

ID	UPDATED	BY	BRANCH	ADVISOR	NAME	ADDR1	ADDR2
20001500	2007-09-26 12:00:00	sysadm	12001500	13001500	Bruce H Anderson	6177 Purple Sage Ct	
20001501	2007-09-26 12:00:00	sysadm	12001500	13001501	Christina Anderson	6181 Camino Verde Dr	
20001502	2007-09-26 12:00:00	sysadm	12001501	13001502	Alexandra Anderson	321 Curie Drivee	
20001503	2007-09-26 12:00:00	sysadm	12001501	13001503	Torben Andersom	321 Curie Drivee	
20001504	2007-09-26 12:00:00	sysadm	12001500	13001504	Yesica Anderson	6177 Purple Sage Ct	
20001505	2007-09-26 12:00:00	sysadm	12001501	13001505	Kurt Madi	1603 Bel Air Avenue	
20001506	2007-09-26 12:00:00	sysadm	12001500	13001506	Maria Fanelli	Bel Air	
20001507	2007-09-26 12:00:00	sysadm	12001501	13001500	A Fanelli	2584 Junction Ave	
20001508	2007-09-26 12:00:00	sysadm	12001500	13001501	A Carter	5528 Muir Dr	
20001509	2007-09-26 12:00:00	sysadm	12001501	13001502	Barry Rosen	560 Galen	
20001510	2007-09-26 12:00:00	sysadm	12001500	13001503	Alok Alur	502 Gaylen	
20001511	2007-09-26 12:00:00	sysadm	12001501	13001504	Jastinderk Kumar	3424 Maroun Pl	
20001512	2007-09-26 12:00:00	sysadm	12001502	13001505	Aaron Jensen	1363 14th Ave	
20001513	2007-09-26 12:00:00	sysadm	12001503	13001506	Allan Jensen	PO Box 7424	
20001514	2007-09-26 12:00:00	sysadm	12001504	13001500	Andrew I Jensen	44 Montgomery St	Ste 3
20001515	2007-09-26 12:00:00	sysadm	12001505	13001501	Anette A Jensen	77 Grand View Ave	Apt 2
20001516	2007-09-26 12:00:00	sysadm	12001506	13001502	Anton T & Larue Jensen	258 Lisbon St	
20001517	2007-09-26 12:00:00	sysadm	12001507	13001503	Brandon Jensen	625 Hyde St	Apt 1
20001518	2007-09-26 12:00:00	sysadm	12001508	13001504	Steven C Preston	Embarcadero Ctr	
20001519	2007-09-26 12:00:00	sysadm	12001509	13001505	Allan Preston	720 Market St	Ste 9
20001520	2007-09-26 12:00:00	sysadm	12001510	13001506	Allison R Preston	831 Webster St	
20001521	2007-09-26 12:00:00	sysadm	12001511	13001500	Burr Preston	3726 Broderick St	Apt 8
20001522	2007-09-26 12:00:00	sysadm	12001512	13001501	Cathy Preston	425 Market St, Ste 2200	
20001523	2007-09-26 12:00:00	sysadm	12001513	13001502	Alan S Hopkins	1200 15th Ave, Apt 1	
20001524	2007-09-26 12:00:00	sysadm	12001514	13001503	Beata Hopkins	2250 24th St	Apt 3
20001525	2007-09-26 12:00:00	sysadm	12001515	13001504	C C Hopkins	2194 Folsom St	
20001526	2007-09-26 12:00:00	sysadm	12001516	13001505	Christine E Hopkins	407 Paris St	
20001527	2007-09-26 12:00:00	sysadm	12001517	13001506	Kelly Hopkins	1482 Rhode Island St	
20001528	2007-09-26 12:00:00	sysadm	12001518	13001500	Kimberly E Hopkins	2000 Post St, Apt 337	
20001529	2007-09-26 12:00:00	sysadm	12001519	13001501	Andreay Paley	555 California St, Ste 2900	
20001530	2007-09-26 12:00:00	sysadm	12001520	13001502	Jeremy Paley	720 Market St, Ste 900	
20001531	2007-09-26 12:00:00	sysadm	12001521	13001503	Rebecca Paley	222 Sutter St	Fl 6
20001532	2007-09-26 12:00:00	sysadm	12001522	13001504	Agnes Wyman	220 Montgomery St	
20001533	2007-09-26 12:00:00	sysadm	12001523	13001505	Gloria Wyman	631 Ofarrell St	Apt 5
20001534	2007-09-26 12:00:00	sysadm	12001524	13001506	H Wyman	1445 Oak St	
20001535	2007-09-26 12:00:00	sysadm	12001525	13001500	Jason Wyman	241 Oneida Ave	Rm 81
20001536	2007-09-26 12:00:00	sysadm	12001526	13001501	Martha S Peet	13055 23rd Pl NE	
20001537	2007-09-26 12:00:00	sysadm	12001529	13001502	Lorraine Y Peterson	7269 N Dearing Ave	
20001538	2007-09-26 12:00:00	sysadm	12001530	13001503	Dana Sutherland	8615 la Riviera Dr	

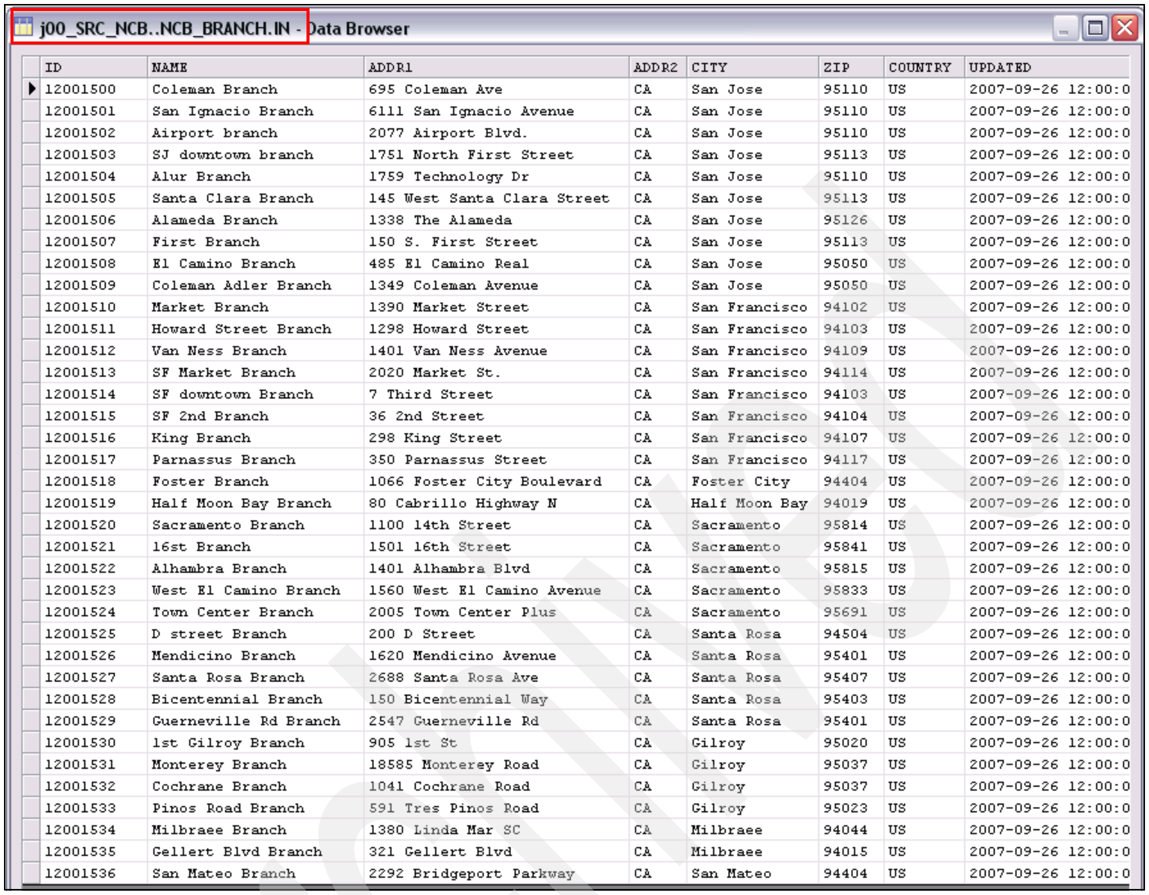
Figure 2-241 Create j00_SRC_NCB 6/9



j00_SRC_NCB..NCB_BCUSTOMER.IN - Data Browser

ADDR1	ADDR2	CITY	ZIP	COUNTRY	EMAIL	BANKID
6177 Purple Sage Ct		San Jose	95119	US	anymail@hotmail.com	90002000
6181 Camino Verde Dr		San Jose	95119	US	mymail@yahoo.com	90002001
321 Curie Drivee		San Jose	95119	US		90002002
321 Curie Drivee		San Jose	95119	US		90002012
6177 Purple Sage Ct		San Jose	95119	US	anymail@hotmail.com	90002013
1603 Bel Air Avenue		San Jose	95119	US		90002014
Bel Air		San Jose	95119	US		90002015
2584 Junction Ave		San Jose	95119	US	afan@hotmail.com	90002016
5528 Muir Dr		San Jose	95119	US	carter@carter.com	90002017
560 Galen		San Jose	95119	US		90002018
502 Gaylen		San Jose	95119	US		90002019
3424 Maroun Pl		San Jose	95119	US	Jastinderk@gmail.com	90002020
1363 14th Ave		San Francisco	94122	US		90002021
PO Box 7424		San Francisco	94120	US	Allan@gmail.com	90002022
44 Montgomery St	Ste 3705	San Francisco	94104	US		90002023
77 Grand View Ave	Apt 202	San Francisco	94114	US	Anette@gmail.com	90002024
258 Lisbon St		San Francisco	94112	US	Anton_Larue@gmail.com	90002025
625 Hyde St	Apt 1	San Francisco	94109	US	Brandon@gmail.com	90002026
Embarcadero Ctr		San Francisco	94111	US	Steven@gmail.com	90002027
720 Market St	Ste 900	San Francisco	94102	US		90002028
831 Webster St		San Francisco	94117	US	Allison@gmail.com	90002029
3726 Broderick St	Apt 8	San Francisco	94123	US	Burr@gmail.com	90002030
425 Market St, Ste 2200		San Francisco	94105	US	Cathy@gmail.com	90002031
1200 15th Ave, Apt 1		San Francisco	94122	US		90002032
2250 24th St	Apt 334	San Francisco	94107	US	Beata@gmail.com	90002033
2194 Folsom St		San Francisco	94110	US	C@gmail.com	90002034
407 Paris St		San Francisco	94112	Us	Christine@gmail.com	90002035
1482 Rhode Island St		San Francisco	94107	US	Kelly@gmail.com	90002036
2000 Post St, Apt 337		San Francisco	94115	US		90002037
555 California St, Ste 2900		San Francisco	94104	US	Andreay@gmail.com	90002038
720 Market St, Ste 900		San Francisco	94102	US	Jeremy@gmail.com	90002039
222 Sutter St	Fl 6	San Francisco	94108	US		90002040
220 Montgomery St		San Francisco	94104	US	Agnes@gmail.com	90002041
631 Ofarrell St	Apt 508	San Francisco	94109	US		90002042
1445 Oak St		San Francisco	94117	US	H@gmail.com	90002043
241 Oneida Ave	Rm 81	San Francisco	94112	US	Jason@gmail.com	90002044
13055 23rd Pl NE		Seattle	98125	US	MarthaS@hotmail.com	90002045
7269 N Dearing Ave		Fresno	93720	US	LorraineY@hotmail.com	90002048
8615 la Riviera Dr		Sacramento	95826	US	Dana@hotmail.com	90002049

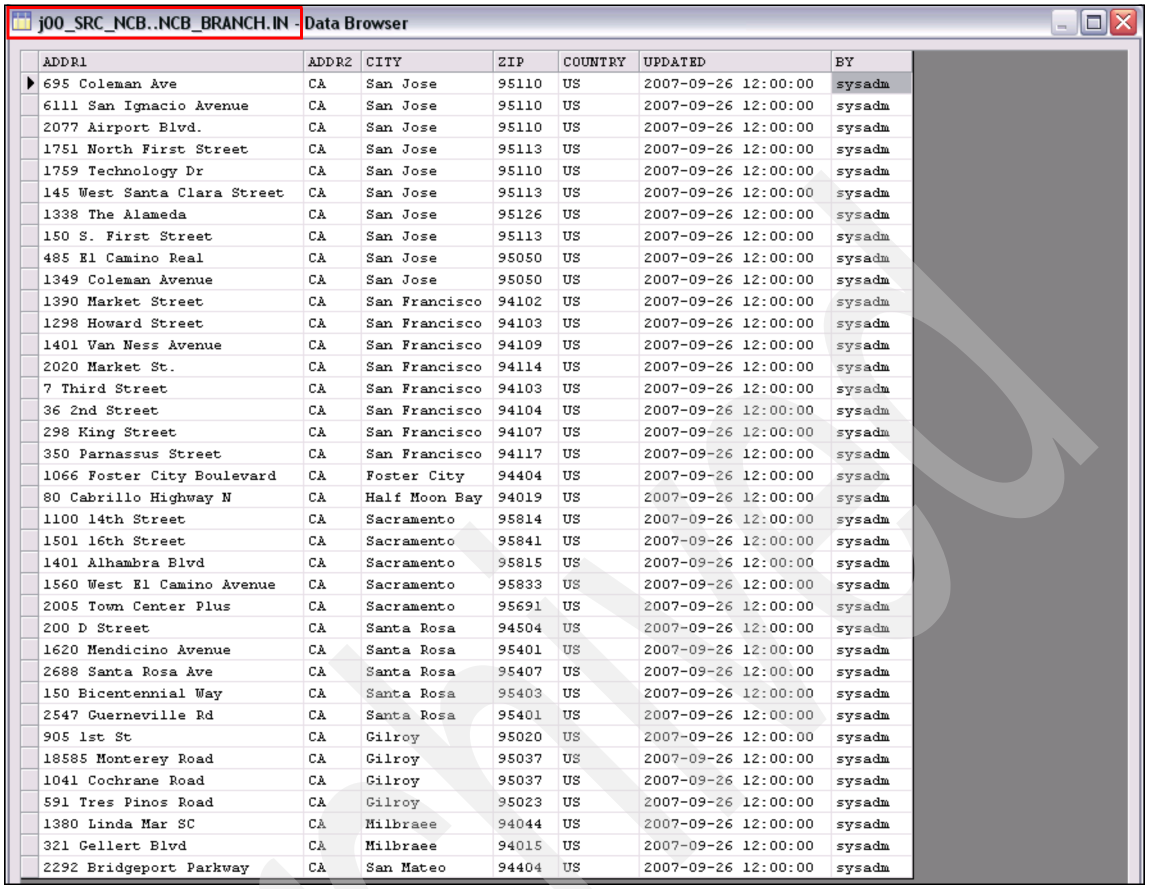
Figure 2-242 Create j00_SRC_NCB 7/9



j00_SRC_NCB..NCB_BRANCH.IN - Data Browser

ID	NAME	ADDR1	ADDR2	CITY	ZIP	COUNTRY	UPDATED
12001500	Coleman Branch	695 Coleman Ave	CA	San Jose	95110	US	2007-09-26 12:00:0
12001501	San Ignacio Branch	6111 San Ignacio Avenue	CA	San Jose	95110	US	2007-09-26 12:00:0
12001502	Airport branch	2077 Airport Blvd.	CA	San Jose	95110	US	2007-09-26 12:00:0
12001503	SJ downtown branch	1751 North First Street	CA	San Jose	95113	US	2007-09-26 12:00:0
12001504	Alur Branch	1759 Technology Dr	CA	San Jose	95110	US	2007-09-26 12:00:0
12001505	Santa Clara Branch	145 West Santa Clara Street	CA	San Jose	95113	US	2007-09-26 12:00:0
12001506	Alameda Branch	1338 The Alameda	CA	San Jose	95126	US	2007-09-26 12:00:0
12001507	First Branch	150 S. First Street	CA	San Jose	95113	US	2007-09-26 12:00:0
12001508	El Camino Branch	485 El Camino Real	CA	San Jose	95050	US	2007-09-26 12:00:0
12001509	Coleman Adler Branch	1349 Coleman Avenue	CA	San Jose	95050	US	2007-09-26 12:00:0
12001510	Market Branch	1390 Market Street	CA	San Francisco	94102	US	2007-09-26 12:00:0
12001511	Howard Street Branch	1298 Howard Street	CA	San Francisco	94103	US	2007-09-26 12:00:0
12001512	Van Ness Branch	1401 Van Ness Avenue	CA	San Francisco	94109	US	2007-09-26 12:00:0
12001513	SF Market Branch	2020 Market St.	CA	San Francisco	94114	US	2007-09-26 12:00:0
12001514	SF downtown Branch	7 Third Street	CA	San Francisco	94103	US	2007-09-26 12:00:0
12001515	SF 2nd Branch	36 2nd Street	CA	San Francisco	94104	US	2007-09-26 12:00:0
12001516	King Branch	298 King Street	CA	San Francisco	94107	US	2007-09-26 12:00:0
12001517	Parnassus Branch	350 Parnassus Street	CA	San Francisco	94117	US	2007-09-26 12:00:0
12001518	Foster Branch	1066 Foster City Boulevard	CA	Foster City	94404	US	2007-09-26 12:00:0
12001519	Half Moon Bay Branch	80 Cabrillo Highway N	CA	Half Moon Bay	94019	US	2007-09-26 12:00:0
12001520	Sacramento Branch	1100 14th Street	CA	Sacramento	95814	US	2007-09-26 12:00:0
12001521	16st Branch	1501 16th Street	CA	Sacramento	95841	US	2007-09-26 12:00:0
12001522	Alhambra Branch	1401 Alhambra Blvd	CA	Sacramento	95815	US	2007-09-26 12:00:0
12001523	West El Camino Branch	1560 West El Camino Avenue	CA	Sacramento	95833	US	2007-09-26 12:00:0
12001524	Town Center Branch	2005 Town Center Plus	CA	Sacramento	95691	US	2007-09-26 12:00:0
12001525	D street Branch	200 D Street	CA	Santa Rosa	94504	US	2007-09-26 12:00:0
12001526	Mendicino Branch	1620 Mendicino Avenue	CA	Santa Rosa	95401	US	2007-09-26 12:00:0
12001527	Santa Rosa Branch	2688 Santa Rosa Ave	CA	Santa Rosa	95407	US	2007-09-26 12:00:0
12001528	Bicentennial Branch	150 Bicentennial Way	CA	Santa Rosa	95403	US	2007-09-26 12:00:0
12001529	Guerneville Rd Branch	2547 Guerneville Rd	CA	Santa Rosa	95401	US	2007-09-26 12:00:0
12001530	1st Gilroy Branch	905 1st St	CA	Gilroy	95020	US	2007-09-26 12:00:0
12001531	Monterey Branch	18585 Monterey Road	CA	Gilroy	95037	US	2007-09-26 12:00:0
12001532	Cochrane Branch	1041 Cochrane Road	CA	Gilroy	95037	US	2007-09-26 12:00:0
12001533	Pinos Road Branch	591 Tres Pinos Road	CA	Gilroy	95023	US	2007-09-26 12:00:0
12001534	Milbraee Branch	1380 Linda Mar SC	CA	Milbraee	94044	US	2007-09-26 12:00:0
12001535	Gellert Blvd Branch	321 Gellert Blvd	CA	Milbraee	94015	US	2007-09-26 12:00:0
12001536	San Mateo Branch	2292 Bridgeport Parkway	CA	San Mateo	94404	US	2007-09-26 12:00:0

Figure 2-243 Create j00_SRC_NCB 8/9



ADDR1	ADDR2	CITY	ZIP	COUNTRY	UPDATED	BY
695 Coleman Ave	CA	San Jose	95110	US	2007-09-26 12:00:00	sysadm
6111 San Ignacio Avenue	CA	San Jose	95110	US	2007-09-26 12:00:00	sysadm
2077 Airport Blvd.	CA	San Jose	95110	US	2007-09-26 12:00:00	sysadm
1751 North First Street	CA	San Jose	95113	US	2007-09-26 12:00:00	sysadm
1759 Technology Dr	CA	San Jose	95110	US	2007-09-26 12:00:00	sysadm
145 West Santa Clara Street	CA	San Jose	95113	US	2007-09-26 12:00:00	sysadm
1338 The Alameda	CA	San Jose	95126	US	2007-09-26 12:00:00	sysadm
150 S. First Street	CA	San Jose	95113	US	2007-09-26 12:00:00	sysadm
485 El Camino Real	CA	San Jose	95050	US	2007-09-26 12:00:00	sysadm
1349 Coleman Avenue	CA	San Jose	95050	US	2007-09-26 12:00:00	sysadm
1390 Market Street	CA	San Francisco	94102	US	2007-09-26 12:00:00	sysadm
1298 Howard Street	CA	San Francisco	94103	US	2007-09-26 12:00:00	sysadm
1401 Van Ness Avenue	CA	San Francisco	94109	US	2007-09-26 12:00:00	sysadm
2020 Market St.	CA	San Francisco	94114	US	2007-09-26 12:00:00	sysadm
7 Third Street	CA	San Francisco	94103	US	2007-09-26 12:00:00	sysadm
36 2nd Street	CA	San Francisco	94104	US	2007-09-26 12:00:00	sysadm
298 King Street	CA	San Francisco	94107	US	2007-09-26 12:00:00	sysadm
350 Parnassus Street	CA	San Francisco	94117	US	2007-09-26 12:00:00	sysadm
1066 Foster City Boulevard	CA	Foster City	94404	US	2007-09-26 12:00:00	sysadm
80 Cabrillo Highway N	CA	Half Moon Bay	94019	US	2007-09-26 12:00:00	sysadm
1100 14th Street	CA	Sacramento	95814	US	2007-09-26 12:00:00	sysadm
1501 16th Street	CA	Sacramento	95841	US	2007-09-26 12:00:00	sysadm
1401 Alhambra Blvd	CA	Sacramento	95815	US	2007-09-26 12:00:00	sysadm
1560 West El Camino Avenue	CA	Sacramento	95833	US	2007-09-26 12:00:00	sysadm
2005 Town Center Plus	CA	Sacramento	95691	US	2007-09-26 12:00:00	sysadm
200 D Street	CA	Santa Rosa	94504	US	2007-09-26 12:00:00	sysadm
1620 Mendicino Avenue	CA	Santa Rosa	95401	US	2007-09-26 12:00:00	sysadm
2688 Santa Rosa Ave	CA	Santa Rosa	95407	US	2007-09-26 12:00:00	sysadm
150 Bicentennial Way	CA	Santa Rosa	95403	US	2007-09-26 12:00:00	sysadm
2547 Guerneville Rd	CA	Santa Rosa	95401	US	2007-09-26 12:00:00	sysadm
905 1st St	CA	Gilroy	95020	US	2007-09-26 12:00:00	sysadm
18585 Monterey Road	CA	Gilroy	95037	US	2007-09-26 12:00:00	sysadm
1041 Cochrane Road	CA	Gilroy	95037	US	2007-09-26 12:00:00	sysadm
591 Tres Pinos Road	CA	Gilroy	95023	US	2007-09-26 12:00:00	sysadm
1380 Linda Mar SC	CA	Milbraee	94044	US	2007-09-26 12:00:00	sysadm
321 Gellert Blvd	CA	Milbraee	94015	US	2007-09-26 12:00:00	sysadm
2292 Bridgeport Parkway	CA	San Mateo	94404	US	2007-09-26 12:00:00	sysadm

Figure 2-244 Create j00_SRC_NCB 9/9

j01_STAN_COUNTRY_NCB

Next, we analyze the addresses in appropriate data sets (CUSTOMER and BCUSTOMER) to determine the (ISO code) country using the COUNTRY rule set in the Standardize stage.

Note: The CRM system only has the name of the BRANCH included in its data model (see Figure 2-3 on page 485). Because BRANCH was a very small file, we verified it visually and corrected the name and address fields in the records without using IBM WebSphere QualityStage.

Figure 2-245 on page 720 shows the various stages that were used in this job. It includes the data sets that were created in “j00_SRC_NCB” on page 709, a

Standardize stage and an output Data Set stage for each input Data Set. We modified the names of the stages as shown.

Because the configuration of this job is very similar to that described in "J01_STAN_COUNTRY" on page 168, we do not repeat it here. However, some of the configurations of interest are as follows:

- Figure 2-246 on page 721 shows the Standardize Rule Process window with the configured COUNTRY rule set, and the literal ZQUSZQ followed by the CITY, ZIP, and COUNTRY columns of the CUSTOMER file in the Selected Columns list.
- Figure 2-247 on page 721 shows the Standardize Rule Process window with the configured COUNTRY rule set, and the literal ZQUSZQ followed by the CITY, ZIP, and COUNTRY columns of the BCUSTOMER file in the Selected Columns list.
- After saving, compiling, and running the job, Figure 2-248 on page 722 shows the results of the execution.
- Figure 2-249 on page 723 through Figure 2-252 on page 726 show the reports produced for the two sources. Because the volumes are small in this case, we can view these reports to see the distribution of country addresses. In the real word where large volumes of data are involved, you need to run Investigate to determine the countries detected as described in "j02_INVCD_ISOCODE_NCB" on page 726.

Proceed now to "j02_INVCD_ISOCODE_NCB" on page 726.

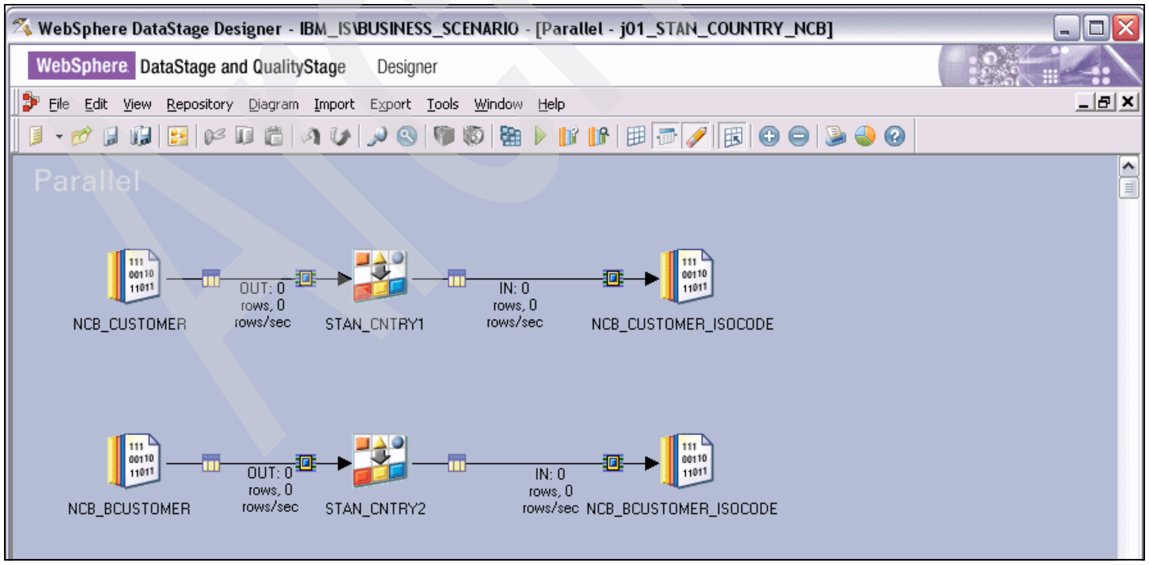
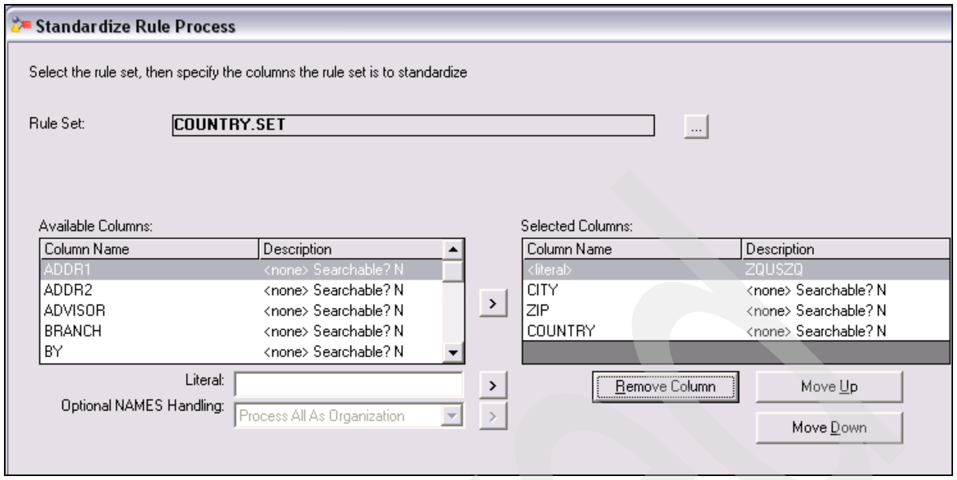

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j01_STAN_COUNTRY_NCB]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NCB_CUSTOMER
OUT: 0 rows, 0 rows/sec
STAN_CNTRY1
IN: 0 rows, 0 rows/sec
NCB_CUSTOMER_ISOCODE
NCB_BCUSTOMER
OUT: 0 rows, 0 rows/sec
STAN_CNTRY2
IN: 0 rows, 0 rows/sec
NCB_BCUSTOMER_ISOCODE

Figure 2-245 Create j01_STAN_COUNTRY_NCB 1/8


Standardize Rule Process

Select the rule set, then specify the columns the rule set is to standardize

Rule Set: COUNTRY.SET

Available Columns:

Column Name	Description
ADDR1	<none> Searchable? N
ADDR2	<none> Searchable? N
ADVISOR	<none> Searchable? N
BRANCH	<none> Searchable? N
BY	<none> Searchable? N

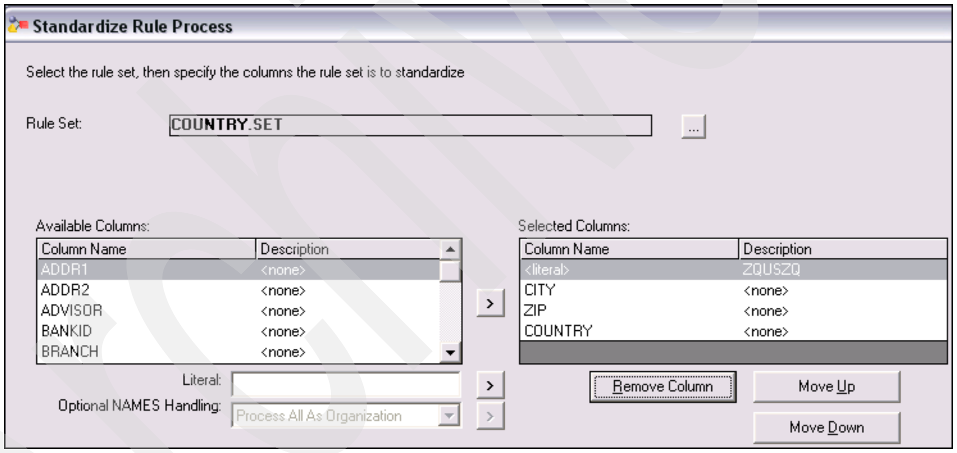
Selected Columns:

Column Name	Description
<literal>	ZQUSZQ
CITY	<none> Searchable? N
ZIP	<none> Searchable? N
COUNTRY	<none> Searchable? N

Literal:

Optional NAMES Handling: Process All As Organization

Figure 2-246 Create j01_STAN_COUNTRY_NCB 2/8


Standardize Rule Process

Select the rule set, then specify the columns the rule set is to standardize

Rule Set: COUNTRY.SET

Available Columns:

Column Name	Description
ADDR1	<none>
ADDR2	<none>
ADVISOR	<none>
BANKID	<none>
BRANCH	<none>

Selected Columns:

Column Name	Description
<literal>	ZQUSZQ
CITY	<none>
ZIP	<none>
COUNTRY	<none>

Literal:

Optional NAMES Handling: Process All As Organization

Figure 2-247 Create j01_STAN_COUNTRY_NCB 3/8

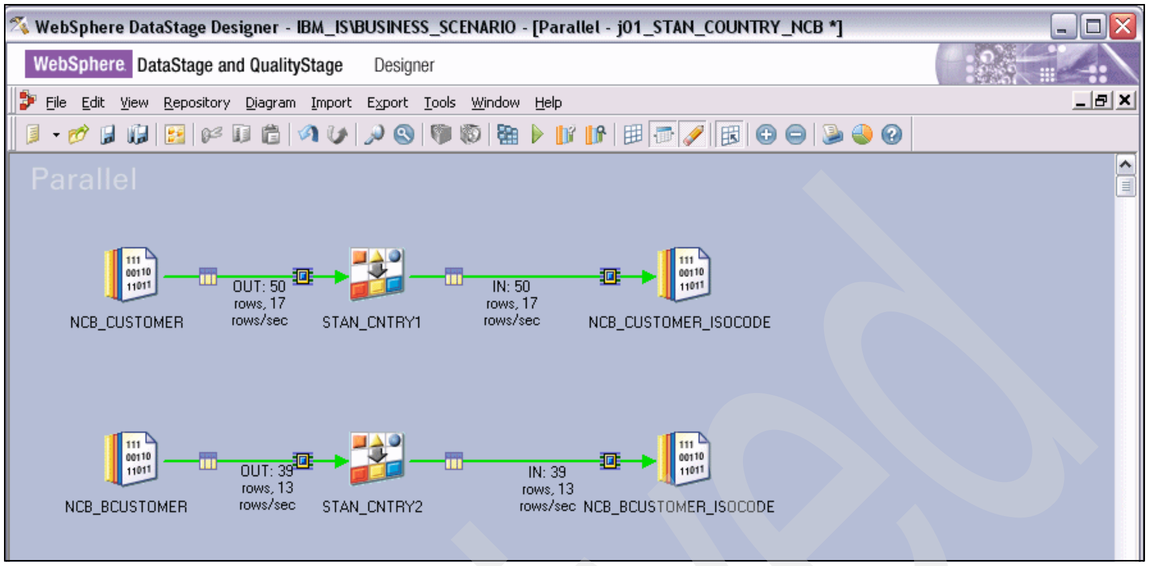
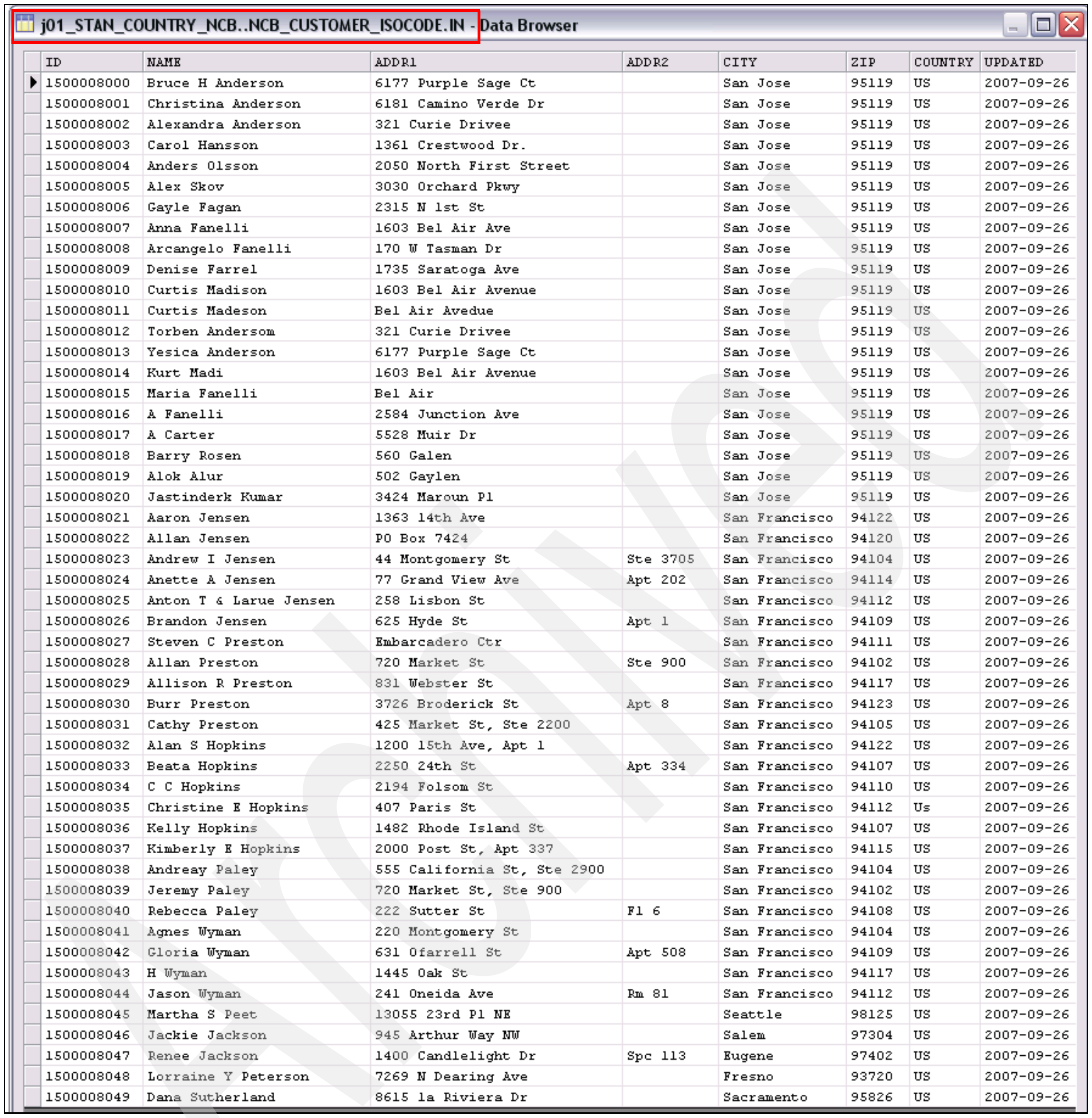

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j01_STAN_COUNTRY_NCB *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NCB_CUSTOMER
OUT: 50 rows, 17 rows/sec
STAN_CNTRY1
IN: 50 rows, 17 rows/sec
NCB_CUSTOMER_ISOCODE
NCB_BCUSTOMER
OUT: 39 rows, 13 rows/sec
STAN_CNTRY2
IN: 39 rows, 13 rows/sec
NCB_BCUSTOMER_ISOCODE

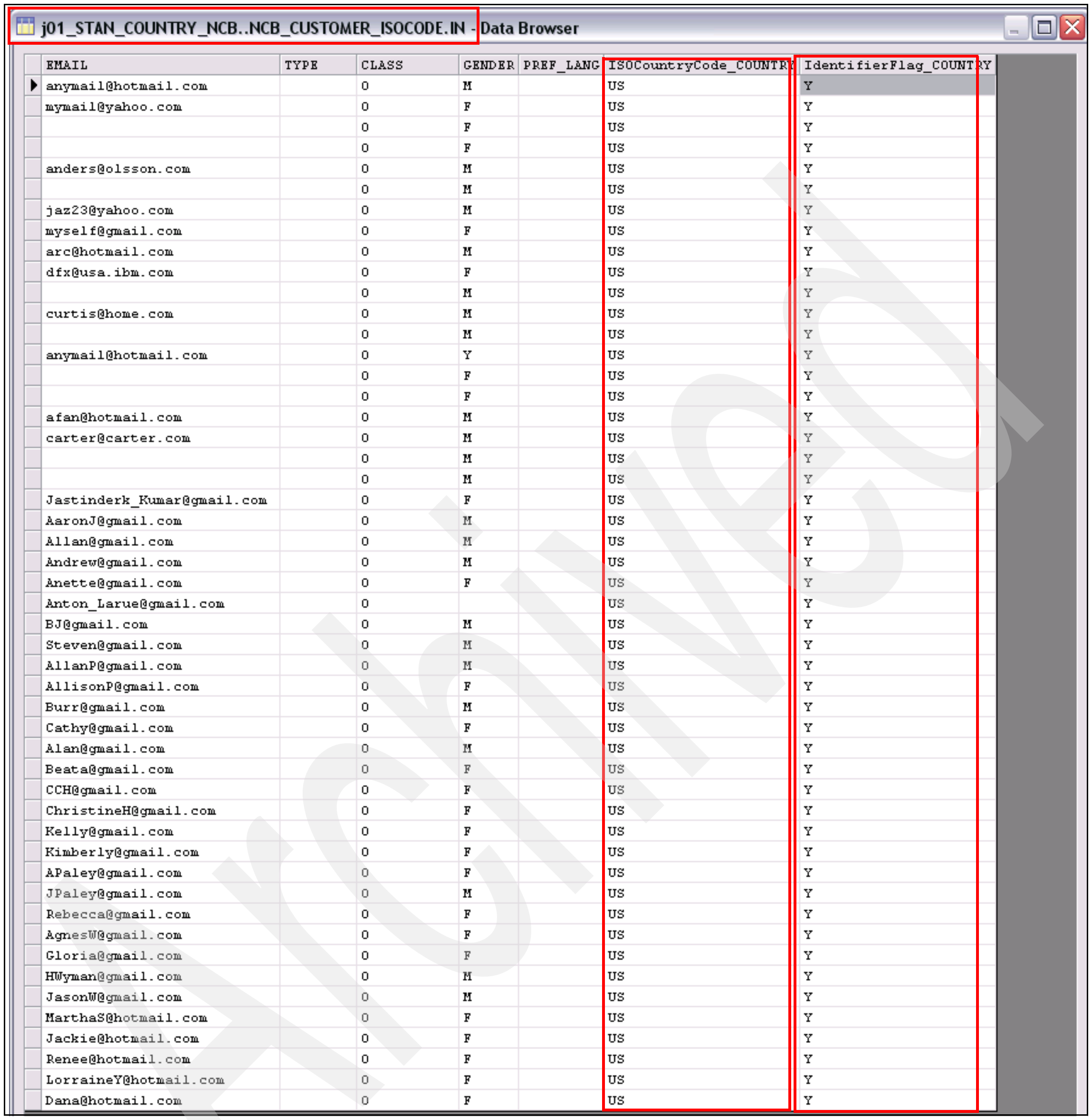
Figure 2-248 Create j01_STAN_COUNTRY_NCB 4/8



j01_STAN_COUNTRY_NCB..NCB_CUSTOMER_ISOCODE.IN - Data Browser

ID	NAME	ADDR1	ADDR2	CITY	ZIP	COUNTRY	UPDATED
1500008000	Bruce H Anderson	6177 Purple Sage Ct		San Jose	95119	US	2007-09-26
1500008001	Christina Anderson	6181 Camino Verde Dr		San Jose	95119	US	2007-09-26
1500008002	Alexandra Anderson	321 Curie Drivee		San Jose	95119	US	2007-09-26
1500008003	Carol Hansson	1361 Crestwood Dr.		San Jose	95119	US	2007-09-26
1500008004	Anders Olsson	2050 North First Street		San Jose	95119	US	2007-09-26
1500008005	Alex Skov	3030 Orchard Pkwy		San Jose	95119	US	2007-09-26
1500008006	Gayle Fagan	2315 N 1st St		San Jose	95119	US	2007-09-26
1500008007	Anna Fanelli	1603 Bel Air Ave		San Jose	95119	US	2007-09-26
1500008008	Arcangelo Fanelli	170 W Tasman Dr		San Jose	95119	US	2007-09-26
1500008009	Denise Farrel	1735 Saratoga Ave		San Jose	95119	US	2007-09-26
1500008010	Curtis Madison	1603 Bel Air Avenue		San Jose	95119	US	2007-09-26
1500008011	Curtis Madeson	Bel Air Avedue		San Jose	95119	US	2007-09-26
1500008012	Torben Andersom	321 Curie Drivee		San Jose	95119	US	2007-09-26
1500008013	Yesica Anderson	6177 Purple Sage Ct		San Jose	95119	US	2007-09-26
1500008014	Kurt Madi	1603 Bel Air Avenue		San Jose	95119	US	2007-09-26
1500008015	Maria Fanelli	Bel Air		San Jose	95119	US	2007-09-26
1500008016	A Fanelli	2584 Junction Ave		San Jose	95119	US	2007-09-26
1500008017	A Carter	5528 Muir Dr		San Jose	95119	US	2007-09-26
1500008018	Barry Rosen	560 Galen		San Jose	95119	US	2007-09-26
1500008019	Alok Alur	502 Gaylen		San Jose	95119	US	2007-09-26
1500008020	Jastinderk Kumar	3424 Maroun Pl		San Jose	95119	US	2007-09-26
1500008021	Aaron Jensen	1363 14th Ave		San Francisco	94122	US	2007-09-26
1500008022	Allan Jensen	PO Box 7424		San Francisco	94120	US	2007-09-26
1500008023	Andrew I Jensen	44 Montgomery St	Ste 3705	San Francisco	94104	US	2007-09-26
1500008024	Anette A Jensen	77 Grand View Ave	Apt 202	San Francisco	94114	US	2007-09-26
1500008025	Anton T & Larue Jensen	258 Lisbon St		San Francisco	94112	US	2007-09-26
1500008026	Brandon Jensen	625 Hyde St	Apt 1	San Francisco	94109	US	2007-09-26
1500008027	Steven C Preston	Embarcadero Ctr		San Francisco	94111	US	2007-09-26
1500008028	Allan Preston	720 Market St	Ste 900	San Francisco	94102	US	2007-09-26
1500008029	Allison R Preston	831 Webster St		San Francisco	94117	US	2007-09-26
1500008030	Burr Preston	3726 Broderick St	Apt 8	San Francisco	94123	US	2007-09-26
1500008031	Cathy Preston	425 Market St, Ste 2200		San Francisco	94105	US	2007-09-26
1500008032	Alan S Hopkins	1200 15th Ave, Apt 1		San Francisco	94122	US	2007-09-26
1500008033	Beata Hopkins	2250 24th St	Apt 334	San Francisco	94107	US	2007-09-26
1500008034	C C Hopkins	2194 Folsom St		San Francisco	94110	US	2007-09-26
1500008035	Christine E Hopkins	407 Paris St		San Francisco	94112	Us	2007-09-26
1500008036	Kelly Hopkins	1482 Rhode Island St		San Francisco	94107	US	2007-09-26
1500008037	Kimberly E Hopkins	2000 Post St, Apt 337		San Francisco	94115	US	2007-09-26
1500008038	Andreay Paley	555 California St, Ste 2900		San Francisco	94104	US	2007-09-26
1500008039	Jeremy Paley	720 Market St, Ste 900		San Francisco	94102	US	2007-09-26
1500008040	Rebecca Paley	222 Sutter St	Fl 6	San Francisco	94108	US	2007-09-26
1500008041	Agnes Wyman	220 Montgomery St		San Francisco	94104	US	2007-09-26
1500008042	Gloria Wyman	631 Ofarrell St	Apt 508	San Francisco	94109	US	2007-09-26
1500008043	H Wyman	1445 Oak St		San Francisco	94117	US	2007-09-26
1500008044	Jason Wyman	241 Oneida Ave	Rm 81	San Francisco	94112	US	2007-09-26
1500008045	Martha S Peet	13055 23rd Pl NE		Seattle	98125	US	2007-09-26
1500008046	Jackie Jackson	945 Arthur Way NW		Salem	97304	US	2007-09-26
1500008047	Renee Jackson	1400 Candlelight Dr	Spc 113	Eugene	97402	US	2007-09-26
1500008048	Lorraine Y Peterson	7269 N Dearing Ave		Fresno	93720	US	2007-09-26
1500008049	Dana Sutherland	8615 la Riviera Dr		Sacramento	95826	US	2007-09-26

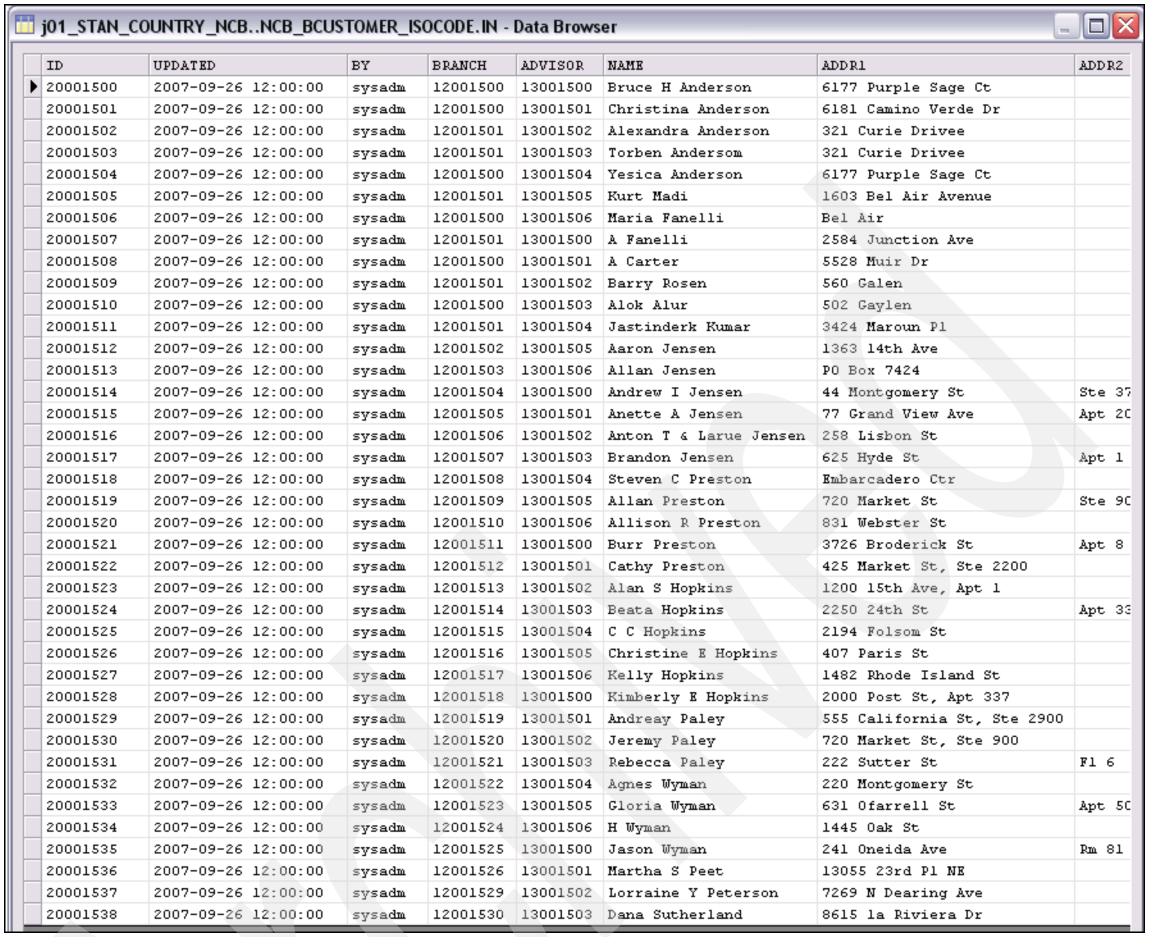
Figure 2-249 Create j01_STAN_COUNTRY_NCB 5/8



j01_STAN_COUNTRY_NCB..NCB_CUSTOMER_ISOCODE.IN - Data Browser

EMAIL	TYPE	CLASS	GENDER	PREF_LANG	ISOCountryCode_COUNTRY	IdentifierFlag_COUNTRY
anymail@hotmail.com		0	M		US	Y
mymail@yahoo.com		0	F		US	Y
		0	F		US	Y
		0	F		US	Y
anders@olsson.com		0	M		US	Y
		0	M		US	Y
jaz23@yahoo.com		0	M		US	Y
myself@gmail.com		0	F		US	Y
arc@hotmail.com		0	M		US	Y
dfx@usa.ibm.com		0	F		US	Y
		0	M		US	Y
curtis@home.com		0	M		US	Y
		0	M		US	Y
anymail@hotmail.com		0	Y		US	Y
		0	F		US	Y
		0	F		US	Y
afan@hotmail.com		0	M		US	Y
carter@carter.com		0	M		US	Y
		0	M		US	Y
		0	M		US	Y
Jastinderk_Kumar@gmail.com		0	F		US	Y
AaronJ@gmail.com		0	M		US	Y
Allan@gmail.com		0	M		US	Y
Andrew@gmail.com		0	M		US	Y
Anette@gmail.com		0	F		US	Y
Anton_Larue@gmail.com		0			US	Y
BJ@gmail.com		0	M		US	Y
Steven@gmail.com		0	M		US	Y
AllanP@gmail.com		0	M		US	Y
AllisonP@gmail.com		0	F		US	Y
Burr@gmail.com		0	M		US	Y
Cathy@gmail.com		0	F		US	Y
Alan@gmail.com		0	M		US	Y
Beata@gmail.com		0	F		US	Y
CCH@gmail.com		0	F		US	Y
ChristineH@gmail.com		0	F		US	Y
Kelly@gmail.com		0	F		US	Y
Kimberly@gmail.com		0	F		US	Y
APaley@gmail.com		0	F		US	Y
JPaley@gmail.com		0	M		US	Y
Rebecca@gmail.com		0	F		US	Y
AgnesW@gmail.com		0	F		US	Y
Gloria@gmail.com		0	F		US	Y
HWyman@gmail.com		0	M		US	Y
JasonW@gmail.com		0	M		US	Y
MarthaS@hotmail.com		0	F		US	Y
Jackie@hotmail.com		0	F		US	Y
Renee@hotmail.com		0	F		US	Y
LorraineY@hotmail.com		0	F		US	Y
Dana@hotmail.com		0	F		US	Y

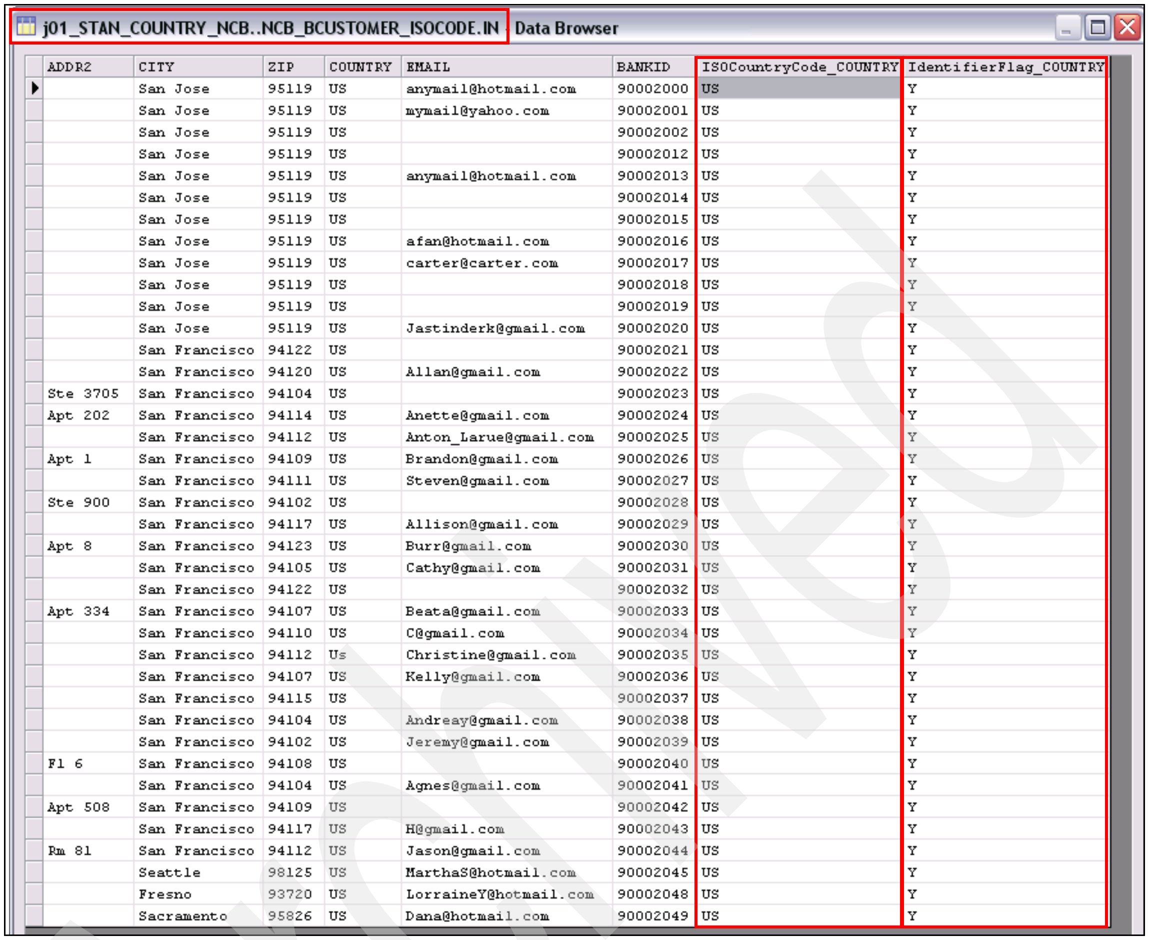
Figure 2-250 Create j01_STAN_COUNTRY_NCB 6/8



j01_STAN_COUNTRY_NCB..NCB_BCUSTOMER_ISOCODE.IN - Data Browser

ID	UPDATED	BY	BRANCH	ADVISOR	NAME	ADDR1	ADDR2
20001500	2007-09-26 12:00:00	sysadm	12001500	13001500	Bruce H Anderson	6177 Purple Sage Ct	
20001501	2007-09-26 12:00:00	sysadm	12001500	13001501	Christina Anderson	6181 Camino Verde Dr	
20001502	2007-09-26 12:00:00	sysadm	12001501	13001502	Alexandra Anderson	321 Curie Drivee	
20001503	2007-09-26 12:00:00	sysadm	12001501	13001503	Torben Andersom	321 Curie Drivee	
20001504	2007-09-26 12:00:00	sysadm	12001500	13001504	Yesica Anderson	6177 Purple Sage Ct	
20001505	2007-09-26 12:00:00	sysadm	12001501	13001505	Kurt Madi	1603 Bel Air Avenue	
20001506	2007-09-26 12:00:00	sysadm	12001500	13001506	Maria Fanelli	Bel Air	
20001507	2007-09-26 12:00:00	sysadm	12001501	13001500	A Fanelli	2584 Junction Ave	
20001508	2007-09-26 12:00:00	sysadm	12001500	13001501	A Carter	5528 Muir Dr	
20001509	2007-09-26 12:00:00	sysadm	12001501	13001502	Barry Rosen	560 Galen	
20001510	2007-09-26 12:00:00	sysadm	12001500	13001503	Alok Alur	502 Gaylen	
20001511	2007-09-26 12:00:00	sysadm	12001501	13001504	Jastinderk Kumar	3424 Maroun Pl	
20001512	2007-09-26 12:00:00	sysadm	12001502	13001505	Aaron Jensen	1363 14th Ave	
20001513	2007-09-26 12:00:00	sysadm	12001503	13001506	Allan Jensen	PO Box 7424	
20001514	2007-09-26 12:00:00	sysadm	12001504	13001500	Andrew I Jensen	44 Montgomery St	Ste 37
20001515	2007-09-26 12:00:00	sysadm	12001505	13001501	Anette A Jensen	77 Grand View Ave	Apt 20
20001516	2007-09-26 12:00:00	sysadm	12001506	13001502	Anton T & Larue Jensen	258 Lisbon St	
20001517	2007-09-26 12:00:00	sysadm	12001507	13001503	Brandon Jensen	625 Hyde St	Apt 1
20001518	2007-09-26 12:00:00	sysadm	12001508	13001504	Steven C Preston	Embarcadero Ctr	
20001519	2007-09-26 12:00:00	sysadm	12001509	13001505	Allan Preston	720 Market St	Ste 90
20001520	2007-09-26 12:00:00	sysadm	12001510	13001506	Allison R Preston	831 Webster St	
20001521	2007-09-26 12:00:00	sysadm	12001511	13001500	Burr Preston	3726 Broderick St	Apt 8
20001522	2007-09-26 12:00:00	sysadm	12001512	13001501	Cathy Preston	425 Market St, Ste 2200	
20001523	2007-09-26 12:00:00	sysadm	12001513	13001502	Alan S Hopkins	1200 15th Ave, Apt 1	
20001524	2007-09-26 12:00:00	sysadm	12001514	13001503	Beata Hopkins	2250 24th St	Apt 33
20001525	2007-09-26 12:00:00	sysadm	12001515	13001504	C C Hopkins	2194 Folsom St	
20001526	2007-09-26 12:00:00	sysadm	12001516	13001505	Christine E Hopkins	407 Paris St	
20001527	2007-09-26 12:00:00	sysadm	12001517	13001506	Kelly Hopkins	1482 Rhode Island St	
20001528	2007-09-26 12:00:00	sysadm	12001518	13001500	Kimberly E Hopkins	2000 Post St, Apt 337	
20001529	2007-09-26 12:00:00	sysadm	12001519	13001501	Andreay Paley	555 California St, Ste 2900	
20001530	2007-09-26 12:00:00	sysadm	12001520	13001502	Jeremy Paley	720 Market St, Ste 900	
20001531	2007-09-26 12:00:00	sysadm	12001521	13001503	Rebecca Paley	222 Sutter St	Fl 6
20001532	2007-09-26 12:00:00	sysadm	12001522	13001504	Agnes Wyman	220 Montgomery St	
20001533	2007-09-26 12:00:00	sysadm	12001523	13001505	Gloria Wyman	631 Ofarrell St	Apt 50
20001534	2007-09-26 12:00:00	sysadm	12001524	13001506	H Wyman	1445 Oak St	
20001535	2007-09-26 12:00:00	sysadm	12001525	13001500	Jason Wyman	241 Oneida Ave	Rm 81
20001536	2007-09-26 12:00:00	sysadm	12001526	13001501	Martha S Peet	13055 23rd Pl NE	
20001537	2007-09-26 12:00:00	sysadm	12001529	13001502	Lorraine Y Peterson	7269 N Dearing Ave	
20001538	2007-09-26 12:00:00	sysadm	12001530	13001503	Dana Sutherland	8615 la Riviera Dr	

Figure 2-251 Create j01_STAN_COUNTRY_NCB 7/8



j01_STAN_COUNTRY_NCB..NCB_BCUSTOMER_ISOCODE.IN - Data Browser

ADDR2	CITY	ZIP	COUNTRY	EMAIL	BANKID	ISOCountryCode_COUNTRY	IdentifierFlag_COUNTRY
	San Jose	95119	US	anymail@hotmail.com	90002000	US	Y
	San Jose	95119	US	mymail@yahoo.com	90002001	US	Y
	San Jose	95119	US		90002002	US	Y
	San Jose	95119	US		90002012	US	Y
	San Jose	95119	US	anymail@hotmail.com	90002013	US	Y
	San Jose	95119	US		90002014	US	Y
	San Jose	95119	US		90002015	US	Y
	San Jose	95119	US	afan@hotmail.com	90002016	US	Y
	San Jose	95119	US	carter@carter.com	90002017	US	Y
	San Jose	95119	US		90002018	US	Y
	San Jose	95119	US		90002019	US	Y
	San Jose	95119	US	Jastinderk@gmail.com	90002020	US	Y
	San Francisco	94122	US		90002021	US	Y
	San Francisco	94120	US	Allan@gmail.com	90002022	US	Y
Ste 3705	San Francisco	94104	US		90002023	US	Y
Apt 202	San Francisco	94114	US	Anette@gmail.com	90002024	US	Y
	San Francisco	94112	US	Anton_Larue@gmail.com	90002025	US	Y
Apt 1	San Francisco	94109	US	Brandon@gmail.com	90002026	US	Y
	San Francisco	94111	US	Steven@gmail.com	90002027	US	Y
Ste 900	San Francisco	94102	US		90002028	US	Y
	San Francisco	94117	US	Allison@gmail.com	90002029	US	Y
Apt 8	San Francisco	94123	US	Burr@gmail.com	90002030	US	Y
	San Francisco	94105	US	Cathy@gmail.com	90002031	US	Y
	San Francisco	94122	US		90002032	US	Y
Apt 334	San Francisco	94107	US	Beata@gmail.com	90002033	US	Y
	San Francisco	94110	US	C@gmail.com	90002034	US	Y
	San Francisco	94112	US	Christine@gmail.com	90002035	US	Y
	San Francisco	94107	US	Kelly@gmail.com	90002036	US	Y
	San Francisco	94115	US		90002037	US	Y
	San Francisco	94104	US	Andreay@gmail.com	90002038	US	Y
	San Francisco	94102	US	Jeremy@gmail.com	90002039	US	Y
Fl 6	San Francisco	94108	US		90002040	US	Y
	San Francisco	94104	US	Agnes@gmail.com	90002041	US	Y
Apt 508	San Francisco	94109	US		90002042	US	Y
	San Francisco	94117	US	H@gmail.com	90002043	US	Y
Rm 81	San Francisco	94112	US	Jason@gmail.com	90002044	US	Y
	Seattle	98125	US	MarthaS@hotmail.com	90002045	US	Y
	Fresno	93720	US	LorraineY@hotmail.com	90002048	US	Y
	Sacramento	95826	US	Dana@hotmail.com	90002049	US	Y

Figure 2-252 Create j01_STAN_COUNTRY_NCB 8/8

j02_INVCD_ISOCODE_NCB

In this step, we analyze the ISO codes that were generated by the previous step by the Investigate stage using the character concatenate investigate option with the *C* mask to obtain frequency distribution. This step identifies whether the addresses in the appropriate data sets belonged to more than one country, and identifies the codes of the countries in the addresses. In this case, the addresses are all U.S. addresses.

Figure 2-253 on page 727 shows the various stages that are used in this job. It includes the data sets that were created in "j01_STAN_COUNTRY_NCB" on page 719, an Investigate stage and an output Sequential File stage for each of the input files. We modified the names of the stages as shown.

Because the configuration of this job is very similar to that described in "J02_INVCC_ISCODE" on page 186, we do not repeat it here. However, some of the configurations of interest are as follows:

- Figure 2-254 on page 728 shows the INV_CC_ISOCODE1 - Investigate Stage window with the Character Concatenate Investigate option, and columns ISOCountryCode_Country and IdentifierFlag_COUNTRY selected with the *C* masks for each character.

  The configuration of the INV_CC_ISOCODE2 - Investigate Stage window for the BCUSTOMER file is similar to this investigate. We do not repeat it here.

- After saving, compiling, and running this job, the job statistics are shown in Figure 2-255 on page 728.
- The output of the Investigate stage written to the sequential files are shown in Figure 2-256 on page 728 and Figure 2-257 on page 729 as follows:
  - Figure 2-256 on page 728 shows the CUSTOMER address report with a concatenated value of `US  Y` in 100% of the records.
  - Figure 2-257 on page 729 shows the BCUSTOMER address report with a concatenated value of `US  Y` in 100% of the records.

Proceed now to "j03_STAN_USPREP_NCB" on page 729.

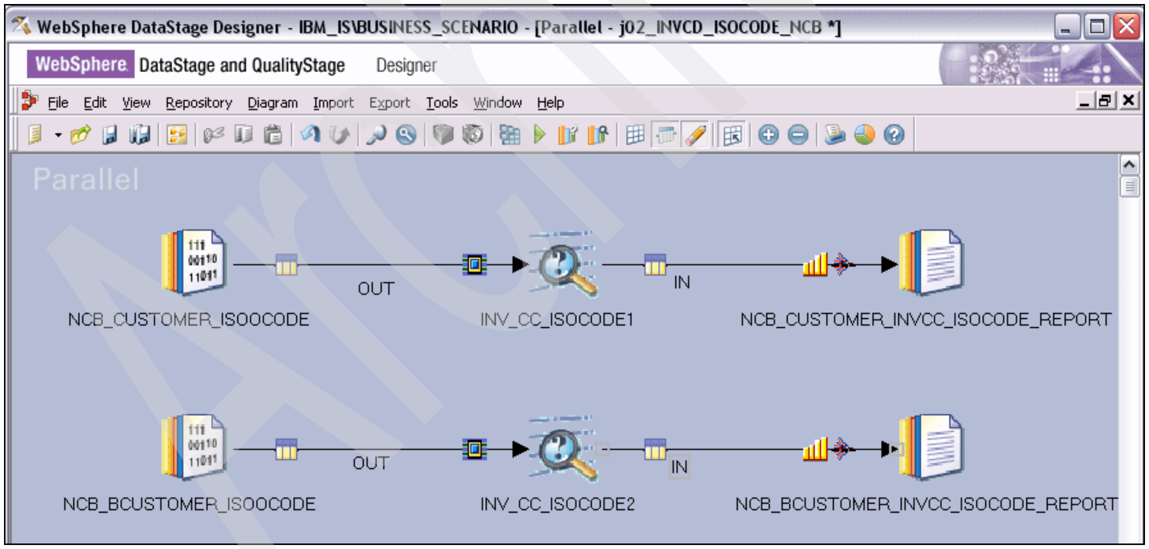
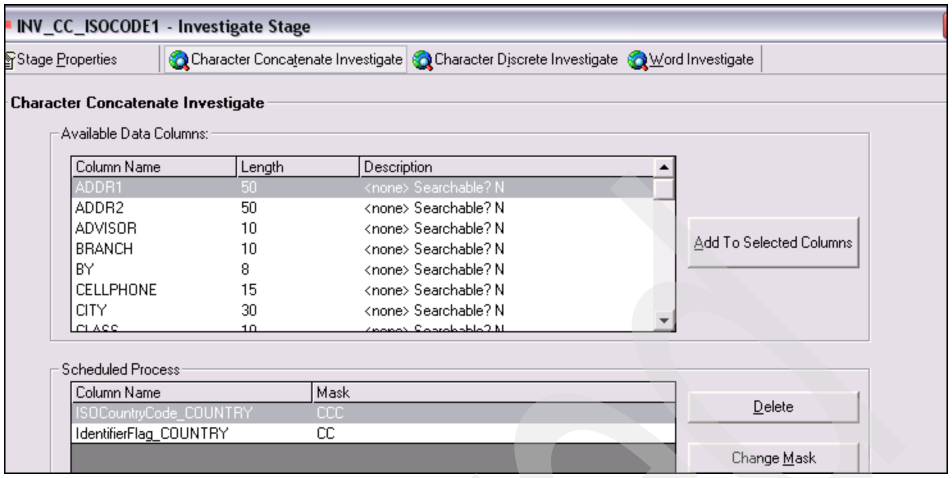

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j02_INVCD_ISOCODE_NCB *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NCB_CUSTOMER_ISOOCODE
OUT
INV_CC_ISOCODE1
IN
NCB_CUSTOMER_INVCC_ISOCODE_REPORT
NCB_BCUSTOMER_ISOOCODE
OUT
INV_CC_ISOCODE2
IN
NCB_BCUSTOMER_INVCC_ISOCODE_REPORT

Figure 2-253 Create j02_INVCD_ISOCODE_NCB 1/5


INV_CC_ISOCODE1 - Investigate Stage

Stage Properties
Character Concatenate Investigate
Character Discrete Investigate
Word Investigate

Character Concatenate Investigate

Available Data Columns:

Column Name	Length	Description
ADDR1	50	<none> Searchable? N
ADDR2	50	<none> Searchable? N
ADVISOR	10	<none> Searchable? N
BRANCH	10	<none> Searchable? N
BY	8	<none> Searchable? N
CELLPHONE	15	<none> Searchable? N
CITY	30	<none> Searchable? N

Scheduled Process

Column Name	Mask
ISOCountryCode_COUNTRY	CCC
IdentifierFlag_COUNTRY	CC

Figure 2-254 Create j02_INVCD_ISOCODE_NCB 2/5

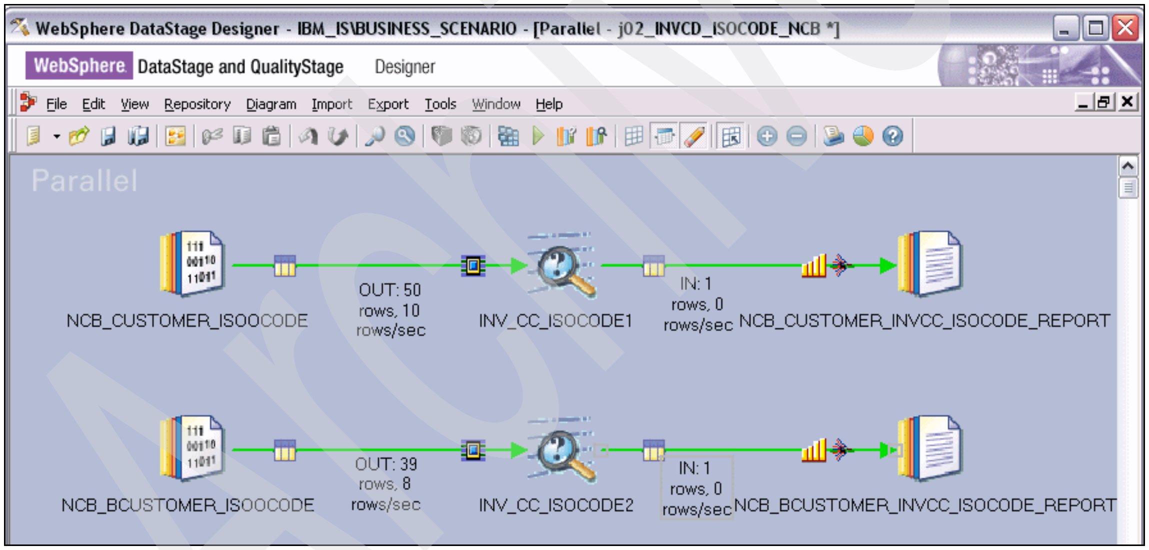
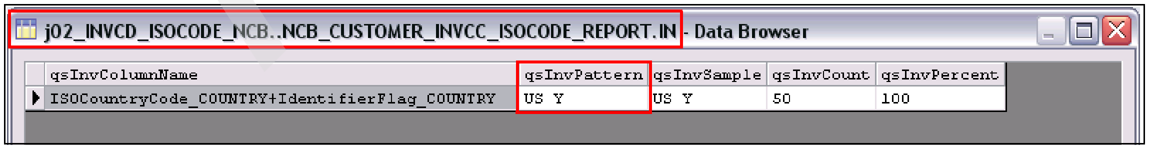
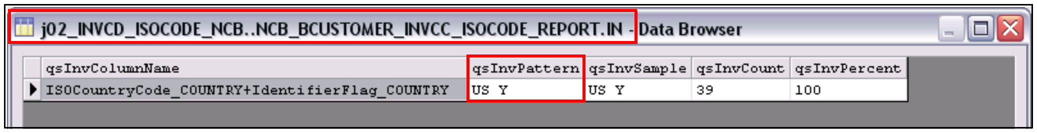

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j02_INVCD_ISOCODE_NCB *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NCB_CUSTOMER_ISOOCODE
OUT: 50 rows, 10 rows/sec
INV_CC_ISOCODE1
IN: 1 rows, 0 rows/sec
NCB_CUSTOMER_INVCC_ISOCODE_REPORT
NCB_BCUSTOMER_ISOOCODE
OUT: 39 rows, 8 rows/sec
INV_CC_ISOCODE2
IN: 1 rows, 0 rows/sec
NCB_BCUSTOMER_INVCC_ISOCODE_REPORT

Figure 2-255 Create j02_INVCD_ISOCODE_NCB 3/5


j02_INVCD_ISOCODE_NCB..NCB_CUSTOMER_INVCC_ISOCODE_REPORT.IN - Data Browser

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
ISOCountryCode_COUNTRY+IdentifierFlag_COUNTRY	US Y	US Y	50	100

Figure 2-256 Create j02_INVCD_ISOCODE_NCB 4/5


j02_INVCD_ISOCODE_NCB..NCB_BCUSTOMER_INVCC_ISOCODE_REPORT.IN - Data Browser

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
ISOCountryCode_COUNTRY+IdentifierFlag_COUNTRY	US Y	US Y	39	100

Figure 2-257 Create j02_INVCD_ISOCODE_NCB 5/5

j03_STAN_USPREP_NCB

In this step, we use the Standardize stage with the USPREP domain-preprocessor rule set to move name and address data into Name, Address, and Area domains for the CUSTOMER and BCUSTOMER files.

Figure 2-258 on page 730 shows the various stages that are used in this job. It includes the data sets that were created in the “j01_STAN_COUNTRY_NCB” on page 719 step, one Standardize stage, and one output Data Set stage for each Standardize stage. We modified the names of the stages as shown.

Because the configuration of this job is very similar to that described in “J03_STAN_USPREP” on page 196, we do not repeat it here. However, some of the configurations of interest are as follows:

- The Standardize Rule Process windows of Figure 2-259 on page 730 and Figure 2-260 on page 731 show the USPREP rule set and the literals and columns selected for analysis. Figure 2-259 on page 730 and Figure 2-260 on page 731 show the Standardize Rule Process window with the configured USPREP rule set, and the following string in the Selected Columns list for the CUSTOMER file:

  ```
  ZQNAMEZQ NAME ZQADDRZQ ADDR1 ADDR2 ZQAREAZQ CITY ZIP COUNTRY
  ```

  The configuration of the Standardize Rule Process window for the BCUSTOMER file is similar, and we do not repeat it here.

- After saving, compiling, and running this job, view the results as shown in Figure 2-261 on page 731.

- Figure 2-262 on page 732 through Figure 2-265 on page 735 show the contents of the standardized output for each data set.

  The partial reports show the columns Columns NameDomain_USPREP (contains prefix, first name, last name, suffix tokens), AddressDomain_USPREP (contains apartment, street name, and street type tokens), and AreaDomain_USPREP (contains city, state, and ZIP code tokens) that were parsed from the input columns.

Proceed now to “j04_INVCC_USPREP_INPUT_PATTERN_NCB” on page 735.

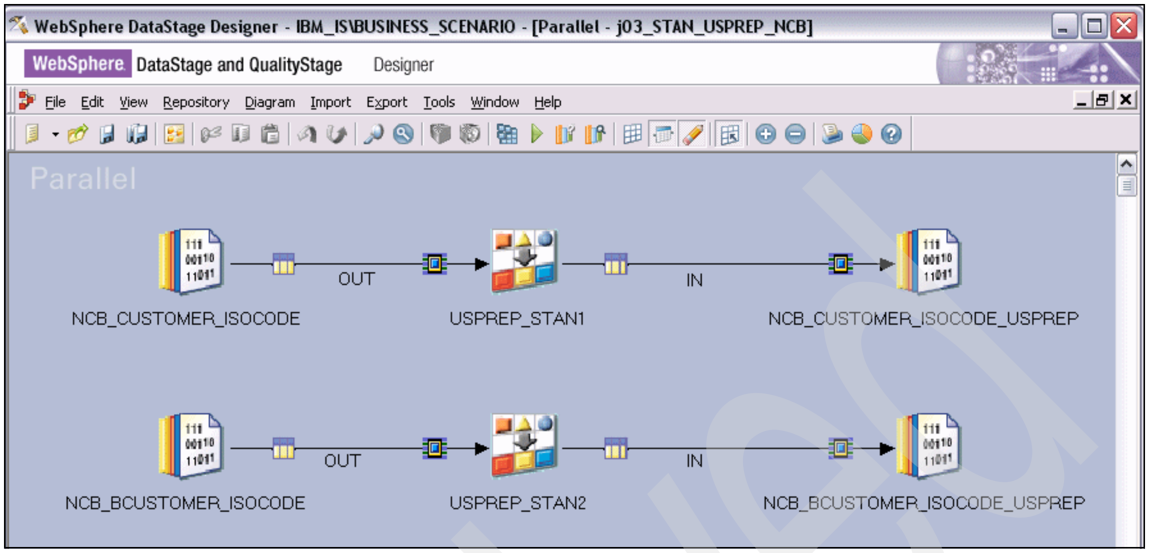
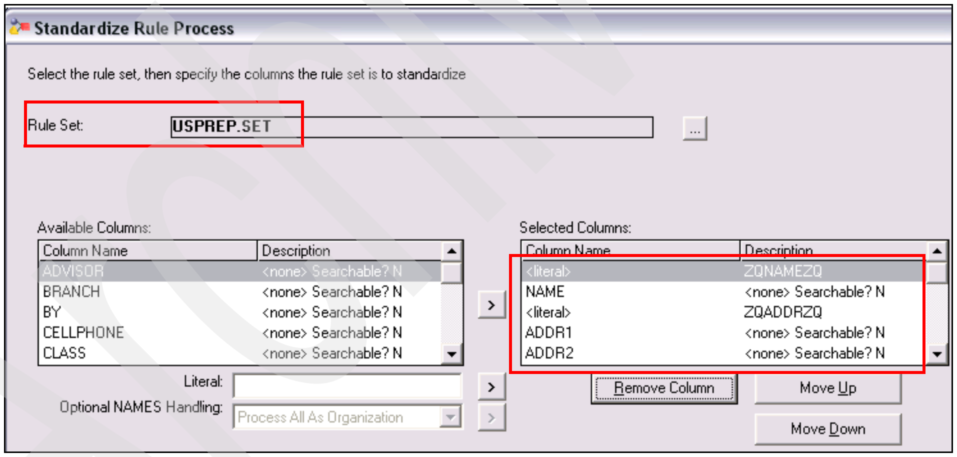

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j03_STAN_USPREP_NCB]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NCB_CUSTOMER_ISOCODE
OUT
USPREP_STAN1
IN
NCB_CUSTOMER_ISOCODE_USPREP
NCB_BCUSTOMER_ISOCODE
OUT
USPREP_STAN2
IN
NCB_BCUSTOMER_ISOCODE_USPREP

Figure 2-258   Create j03_STAN_USPREP_NCB 1/8



Select the rule set, then specify the columns the rule set is to standardize

Rule Set: USPREP.SET

Available Columns:

Column Name	Description
ADVISOR	<none> Searchable? N
BRANCH	<none> Searchable? N
BY	<none> Searchable? N
CELLPHONE	<none> Searchable? N
CLASS	<none> Searchable? N

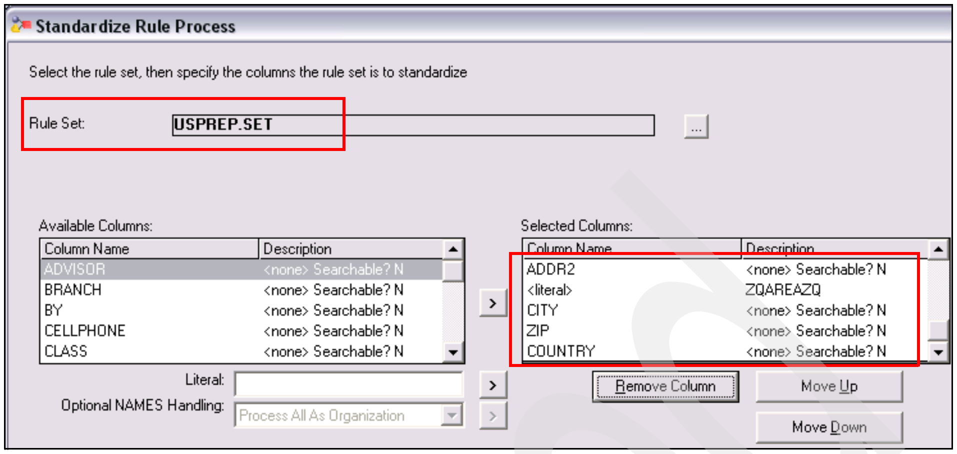
Selected Columns:

Column Name	Description
<literal>	ZQNAMEZQ
NAME	<none> Searchable? N
<literal>	ZQADDRZQ
ADDR1	<none> Searchable? N
ADDR2	<none> Searchable? N

Literal:

Optional NAMES Handling: Process All As Organization

Figure 2-259   Create j03_STAN_USPREP_NCB 2/8


Standardize Rule Process

Select the rule set, then specify the columns the rule set is to standardize

Rule Set: USPREP.SET

Available Columns:

Column Name	Description
ADVISOR	<none> Searchable? N
BRANCH	<none> Searchable? N
BY	<none> Searchable? N
CELLPHONE	<none> Searchable? N
CLASS	<none> Searchable? N

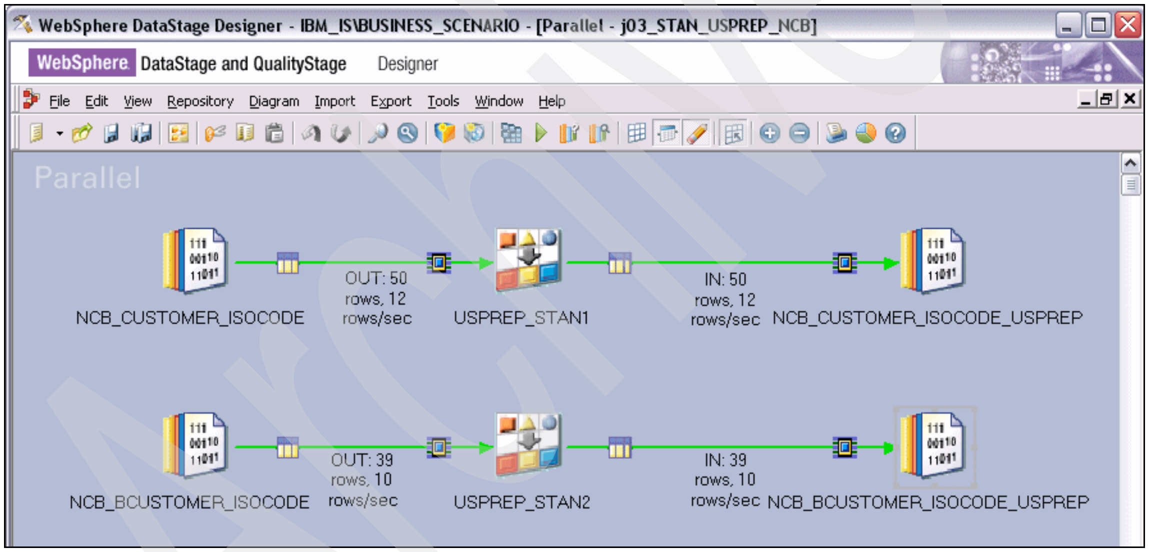
Selected Columns:

Column Name	Description
ADDR2	<none> Searchable? N
<literal>	ZQAREAZQ
CITY	<none> Searchable? N
ZIP	<none> Searchable? N
COUNTRY	<none> Searchable? N

Literal:

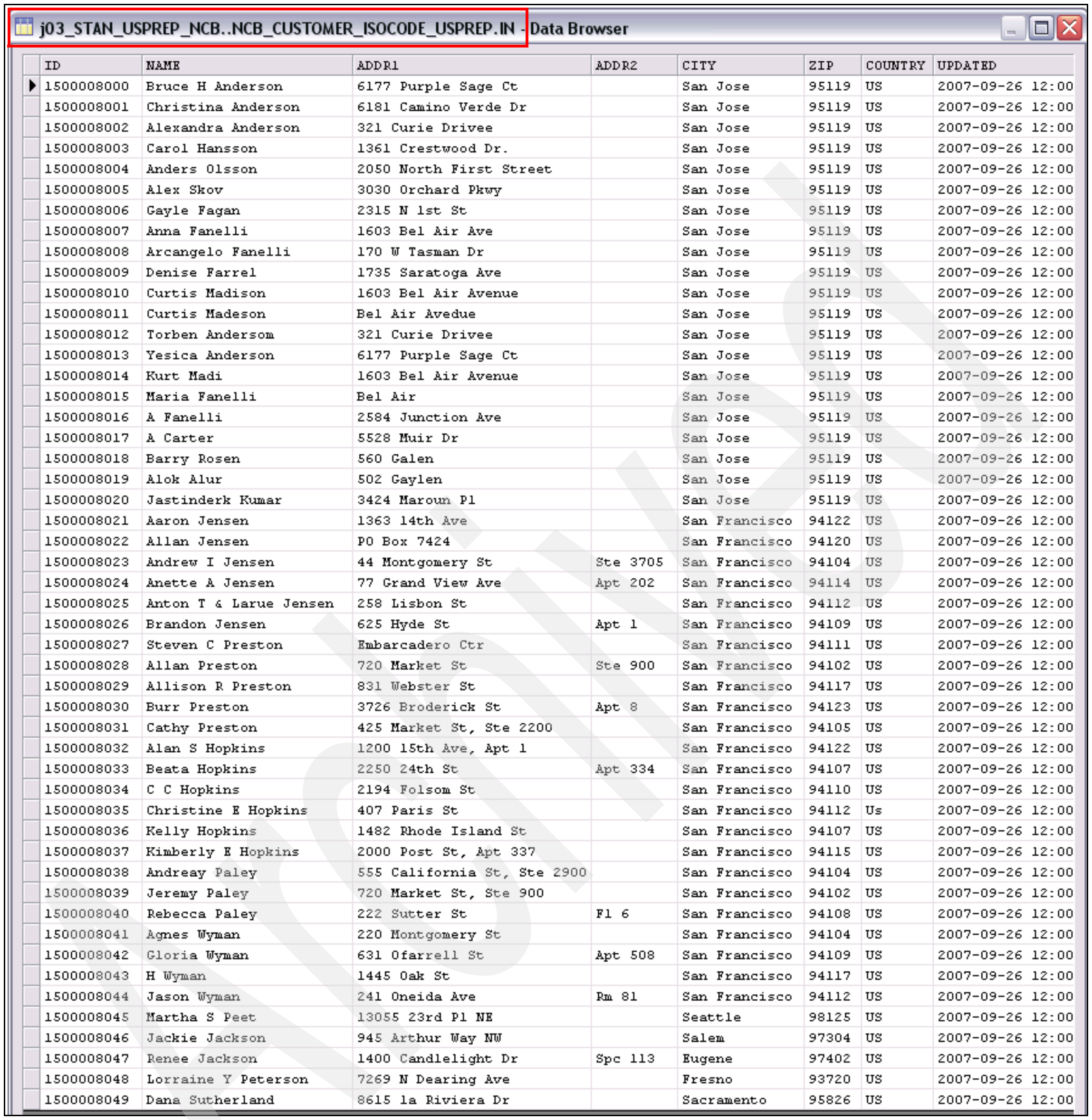
Optional NAMES Handling: Process All As Organization

Figure 2-260   Create j03_STAN_USPREP_NCB 3/8



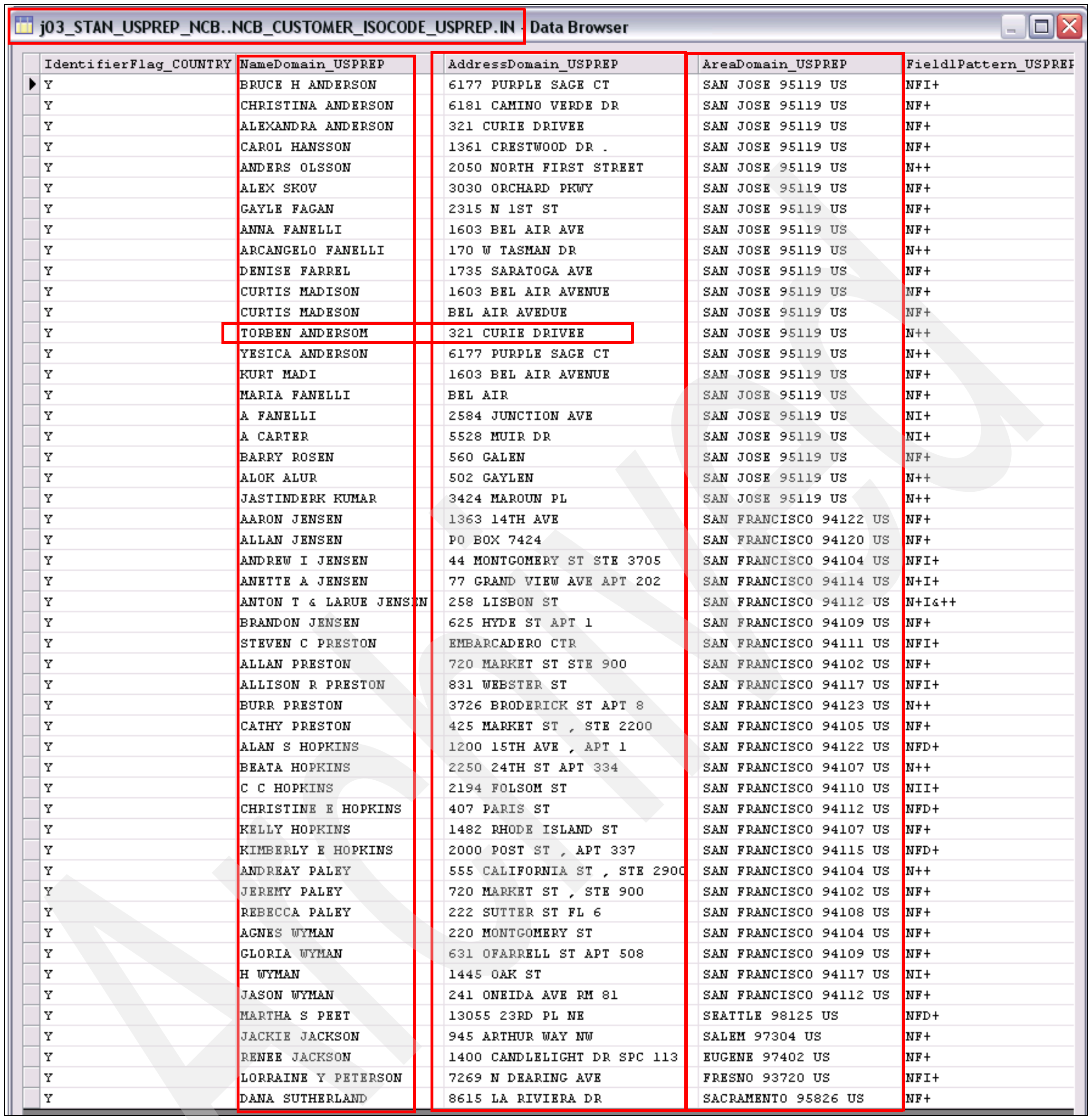
WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j03_STAN_USPREP_NCB]
WebSphere DataStage and QualityStage Designer
File
Edit
View
Repository
Diagram
Import
Export
Tools
Window
Help
Parallel
NCB_CUSTOMER_ISOCODE
OUT: 50 rows, 12 rows/sec
USPREP_STAN1
IN: 50 rows, 12 rows/sec
NCB_CUSTOMER_ISOCODE_USPREP
NCB_BCUSTOMER_ISOCODE
OUT: 39 rows, 10 rows/sec
USPREP_STAN2
IN: 39 rows, 10 rows/sec
NCB_BCUSTOMER_ISOCODE_USPREP

Figure 2-261   Create j03_STAN_USPREP_NCB 4/8


j03_STAN_USPREP_NCB..NCB_CUSTOMER_ISOCODE_USPREP.IN - Data Browser

ID	NAME	ADDR1	ADDR2	CITY	ZIP	COUNTRY	UPDATED
1500008000	Bruce H Anderson	6177 Purple Sage Ct		San Jose	95119	US	2007-09-26 12:00
1500008001	Christina Anderson	6181 Camino Verde Dr		San Jose	95119	US	2007-09-26 12:00
1500008002	Alexandra Anderson	321 Curie Drivee		San Jose	95119	US	2007-09-26 12:00
1500008003	Carol Hansson	1361 Crestwood Dr.		San Jose	95119	US	2007-09-26 12:00
1500008004	Anders Olsson	2050 North First Street		San Jose	95119	US	2007-09-26 12:00
1500008005	Alex Skov	3030 Orchard Pkwy		San Jose	95119	US	2007-09-26 12:00
1500008006	Gayle Fagan	2315 N 1st St		San Jose	95119	US	2007-09-26 12:00
1500008007	Anna Fanelli	1603 Bel Air Ave		San Jose	95119	US	2007-09-26 12:00
1500008008	Arcangelo Fanelli	170 W Tasman Dr		San Jose	95119	US	2007-09-26 12:00
1500008009	Denise Farrel	1735 Saratoga Ave		San Jose	95119	US	2007-09-26 12:00
1500008010	Curtis Madison	1603 Bel Air Avenue		San Jose	95119	US	2007-09-26 12:00
1500008011	Curtis Madeson	Bel Air Avedue		San Jose	95119	US	2007-09-26 12:00
1500008012	Torben Andersom	321 Curie Drivee		San Jose	95119	US	2007-09-26 12:00
1500008013	Yesica Anderson	6177 Purple Sage Ct		San Jose	95119	US	2007-09-26 12:00
1500008014	Kurt Madi	1603 Bel Air Avenue		San Jose	95119	US	2007-09-26 12:00
1500008015	Maria Fanelli	Bel Air		San Jose	95119	US	2007-09-26 12:00
1500008016	A Fanelli	2584 Junction Ave		San Jose	95119	US	2007-09-26 12:00
1500008017	A Carter	5528 Muir Dr		San Jose	95119	US	2007-09-26 12:00
1500008018	Barry Rosen	560 Galen		San Jose	95119	US	2007-09-26 12:00
1500008019	Alok Alur	502 Gaylen		San Jose	95119	US	2007-09-26 12:00
1500008020	Jastinderk Kumar	3424 Maroun Pl		San Jose	95119	US	2007-09-26 12:00
1500008021	Aaron Jensen	1363 14th Ave		San Francisco	94122	US	2007-09-26 12:00
1500008022	Allan Jensen	PO Box 7424		San Francisco	94120	US	2007-09-26 12:00
1500008023	Andrew I Jensen	44 Montgomery St	Ste 3705	San Francisco	94104	US	2007-09-26 12:00
1500008024	Anette A Jensen	77 Grand View Ave	Apt 202	San Francisco	94114	US	2007-09-26 12:00
1500008025	Anton T & Larue Jensen	258 Lisbon St		San Francisco	94112	US	2007-09-26 12:00
1500008026	Brandon Jensen	625 Hyde St	Apt 1	San Francisco	94109	US	2007-09-26 12:00
1500008027	Steven C Preston	Embarcadero Ctr		San Francisco	94111	US	2007-09-26 12:00
1500008028	Allan Preston	720 Market St	Ste 900	San Francisco	94102	US	2007-09-26 12:00
1500008029	Allison R Preston	831 Webster St		San Francisco	94117	US	2007-09-26 12:00
1500008030	Burr Preston	3726 Broderick St	Apt 8	San Francisco	94123	US	2007-09-26 12:00
1500008031	Cathy Preston	425 Market St, Ste 2200		San Francisco	94105	US	2007-09-26 12:00
1500008032	Alan S Hopkins	1200 15th Ave, Apt 1		San Francisco	94122	US	2007-09-26 12:00
1500008033	Beata Hopkins	2250 24th St	Apt 334	San Francisco	94107	US	2007-09-26 12:00
1500008034	C C Hopkins	2194 Folsom St		San Francisco	94110	US	2007-09-26 12:00
1500008035	Christine E Hopkins	407 Paris St		San Francisco	94112	Us	2007-09-26 12:00
1500008036	Kelly Hopkins	1482 Rhode Island St		San Francisco	94107	US	2007-09-26 12:00
1500008037	Kimberly E Hopkins	2000 Post St, Apt 337		San Francisco	94115	US	2007-09-26 12:00
1500008038	Andreay Paley	555 California St, Ste 2900		San Francisco	94104	US	2007-09-26 12:00
1500008039	Jeremy Paley	720 Market St, Ste 900		San Francisco	94102	US	2007-09-26 12:00
1500008040	Rebecca Paley	222 Sutter St	Fl 6	San Francisco	94108	US	2007-09-26 12:00
1500008041	Agnes Wyman	220 Montgomery St		San Francisco	94104	US	2007-09-26 12:00
1500008042	Gloria Wyman	631 Ofarrell St	Apt 508	San Francisco	94109	US	2007-09-26 12:00
1500008043	H Wyman	1445 Oak St		San Francisco	94117	US	2007-09-26 12:00
1500008044	Jason Wyman	241 Oneida Ave	Rm 81	San Francisco	94112	US	2007-09-26 12:00
1500008045	Martha S Peet	13055 23rd Pl NE		Seattle	98125	US	2007-09-26 12:00
1500008046	Jackie Jackson	945 Arthur Way NW		Salem	97304	US	2007-09-26 12:00
1500008047	Renee Jackson	1400 Candlelight Dr	Spc 113	Eugene	97402	US	2007-09-26 12:00
1500008048	Lorraine Y Peterson	7269 N Dearing Ave		Fresno	93720	US	2007-09-26 12:00
1500008049	Dana Sutherland	8615 la Riviera Dr		Sacramento	95826	US	2007-09-26 12:00

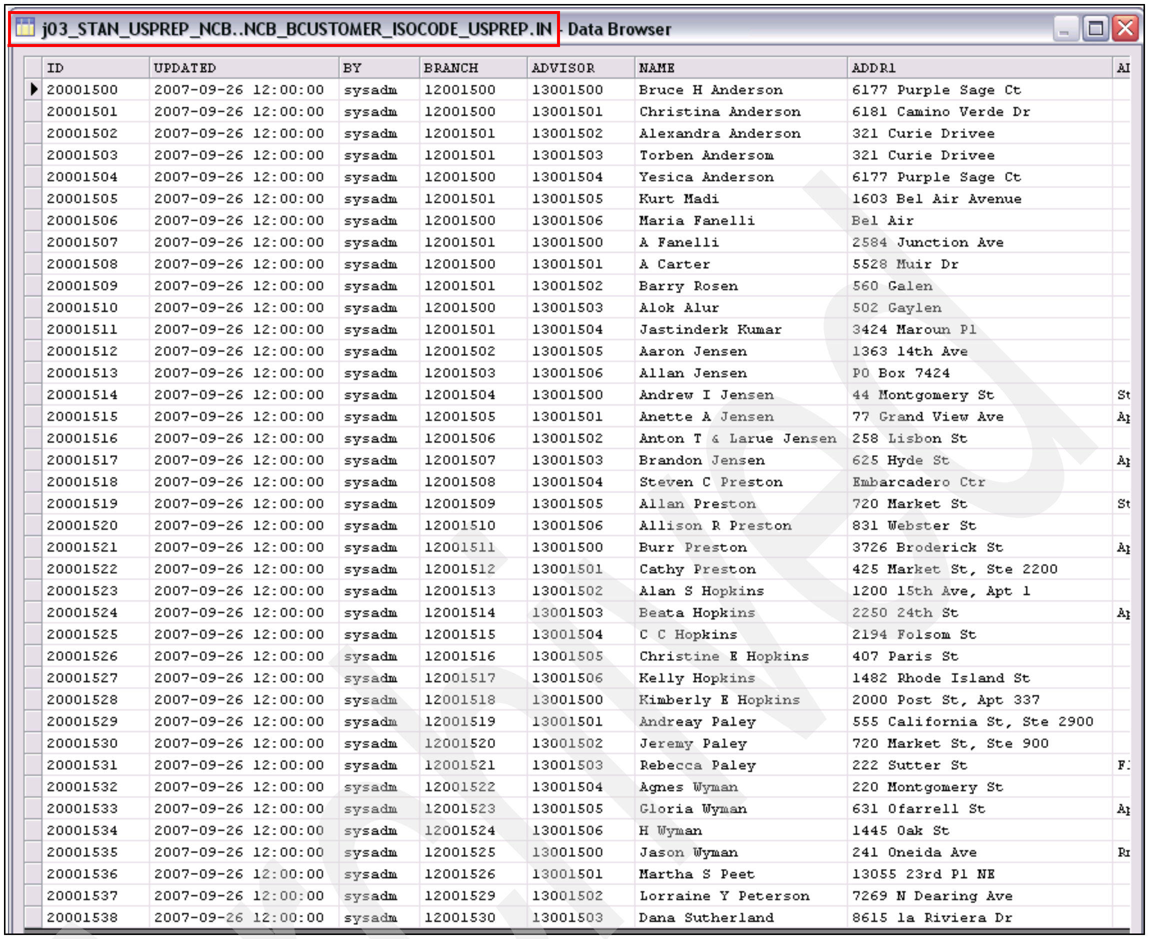
Figure 2-262 Create j03_STAN_USPREP_NCB 5/8



j03_STAN_USPREP_NCB..NCB_CUSTOMER_ISOCODE_USPREP.IN - Data Browser

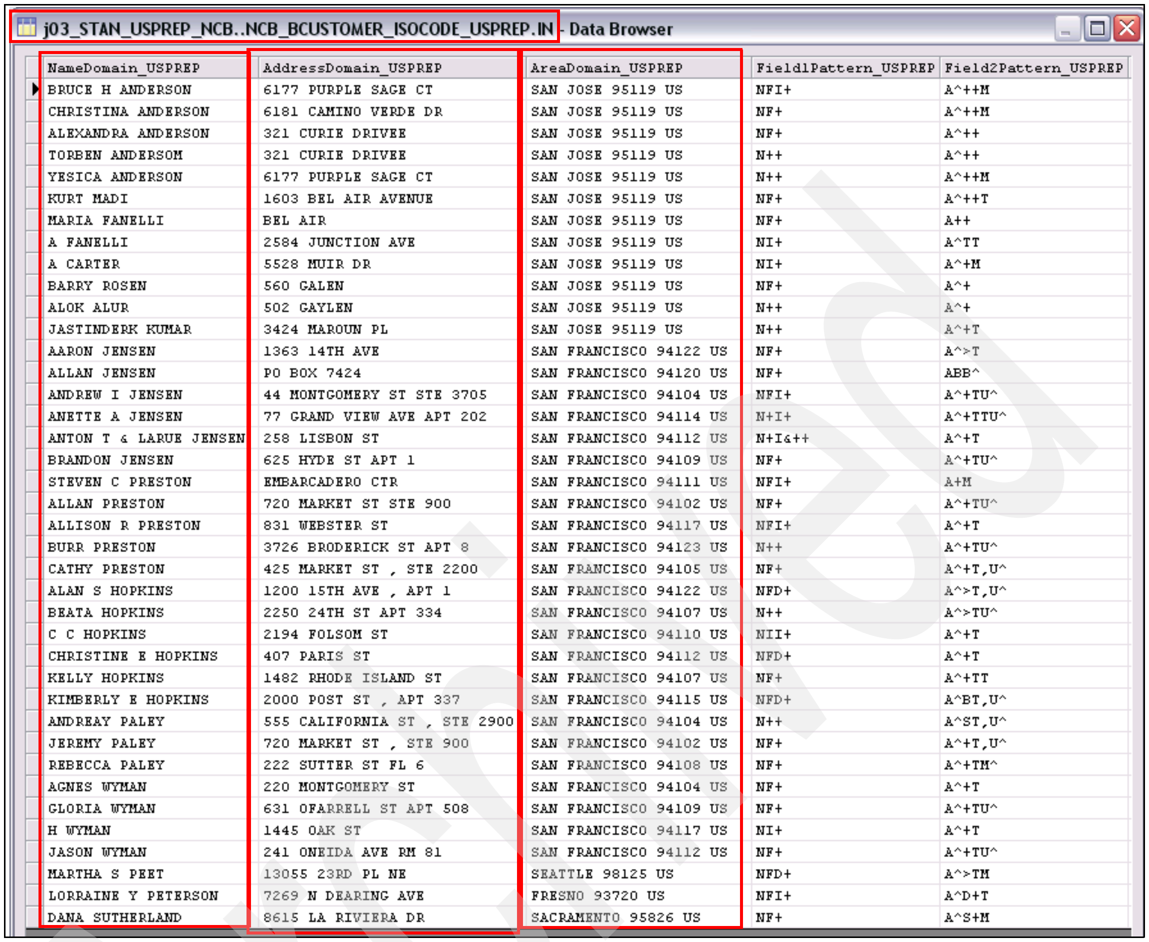
IdentifierFlag_COUNTRY	NameDomain_USPREP	AddressDomain_USPREP	AreaDomain_USPREP	Field1Pattern_USPREP
Y	BRUCE H ANDERSON	6177 PURPLE SAGE CT	SAN JOSE 95119 US	NFI+
Y	CHRISTINA ANDERSON	6181 CAMINO VERDE DR	SAN JOSE 95119 US	NF+
Y	ALEXANDRA ANDERSON	321 CURIE DRIVEE	SAN JOSE 95119 US	NF+
Y	CAROL HANSSON	1361 CRESTWOOD DR .	SAN JOSE 95119 US	NF+
Y	ANDERS OLSSON	2050 NORTH FIRST STREET	SAN JOSE 95119 US	N++
Y	ALEX SKOV	3030 ORCHARD PKWY	SAN JOSE 95119 US	NF+
Y	GAYLE FAGAN	2315 N 1ST ST	SAN JOSE 95119 US	NF+
Y	ANNA FANELLI	1603 BEL AIR AVE	SAN JOSE 95119 US	NF+
Y	ARCANGELO FANELLI	170 W TASMAN DR	SAN JOSE 95119 US	N++
Y	DENISE FARREL	1735 SARATOGA AVE	SAN JOSE 95119 US	NF+
Y	CURTIS MADISON	1603 BEL AIR AVENUE	SAN JOSE 95119 US	NF+
Y	CURTIS MADESON	BEL AIR AVEDUE	SAN JOSE 95119 US	NF+
Y	TORBEN ANDERSOM	321 CURIE DRIVEE	SAN JOSE 95119 US	N++
Y	YESICA ANDERSON	6177 PURPLE SAGE CT	SAN JOSE 95119 US	N++
Y	KURT MADI	1603 BEL AIR AVENUE	SAN JOSE 95119 US	NF+
Y	MARIA FANELLI	BEL AIR	SAN JOSE 95119 US	NF+
Y	A FANELLI	2584 JUNCTION AVE	SAN JOSE 95119 US	NI+
Y	A CARTER	5528 MUIR DR	SAN JOSE 95119 US	NI+
Y	BARRY ROSEN	560 GALEN	SAN JOSE 95119 US	NF+
Y	ALOK ALUR	502 GAYLEN	SAN JOSE 95119 US	N++
Y	JASTINDERK KUMAR	3424 MAROUN PL	SAN JOSE 95119 US	N++
Y	AARON JENSEN	1363 14TH AVE	SAN FRANCISCO 94122 US	NF+
Y	ALLAN JENSEN	PO BOX 7424	SAN FRANCISCO 94120 US	NF+
Y	ANDREW I JENSEN	44 MONTGOMERY ST STE 3705	SAN FRANCISCO 94104 US	NFI+
Y	ANETTE A JENSEN	77 GRAND VIEW AVE APT 202	SAN FRANCISCO 94114 US	N+I+
Y	ANTON T & LARUE JENSEN	258 LISBON ST	SAN FRANCISCO 94112 US	N+I&++
Y	BRANDON JENSEN	625 HYDE ST APT 1	SAN FRANCISCO 94109 US	NF+
Y	STEVEN C PRESTON	EMBARCADERO CTR	SAN FRANCISCO 94111 US	NFI+
Y	ALLAN PRESTON	720 MARKET ST STE 900	SAN FRANCISCO 94102 US	NF+
Y	ALLISON R PRESTON	831 WEBSTER ST	SAN FRANCISCO 94117 US	NFI+
Y	BURR PRESTON	3726 BRODERICK ST APT 8	SAN FRANCISCO 94123 US	N++
Y	CATHY PRESTON	425 MARKET ST , STE 2200	SAN FRANCISCO 94105 US	NF+
Y	ALAN S HOPKINS	1200 15TH AVE , APT 1	SAN FRANCISCO 94122 US	NFD+
Y	BEATA HOPKINS	2250 24TH ST APT 334	SAN FRANCISCO 94107 US	N++
Y	C C HOPKINS	2194 FOLSOM ST	SAN FRANCISCO 94110 US	NII+
Y	CHRISTINE E HOPKINS	407 PARIS ST	SAN FRANCISCO 94112 US	NFD+
Y	KELLY HOPKINS	1482 RHODE ISLAND ST	SAN FRANCISCO 94107 US	NF+
Y	KIMBERLY E HOPKINS	2000 POST ST , APT 337	SAN FRANCISCO 94115 US	NFD+
Y	ANDREAY PALEY	555 CALIFORNIA ST , STE 2900	SAN FRANCISCO 94104 US	N++
Y	JEREMY PALEY	720 MARKET ST , STE 900	SAN FRANCISCO 94102 US	NF+
Y	REBECCA PALEY	222 SUTTER ST FL 6	SAN FRANCISCO 94108 US	NF+
Y	AGNES WYMAN	220 MONTGOMERY ST	SAN FRANCISCO 94104 US	NF+
Y	GLORIA WYMAN	631 OFARRELL ST APT 508	SAN FRANCISCO 94109 US	NF+
Y	H WYMAN	1445 OAK ST	SAN FRANCISCO 94117 US	NI+
Y	JASON WYMAN	241 ONEIDA AVE RM 81	SAN FRANCISCO 94112 US	NF+
Y	MARTHA S PEET	13055 23RD PL NE	SEATTLE 98125 US	NFD+
Y	JACKIE JACKSON	945 ARTHUR WAY NW	SALEM 97304 US	NF+
Y	RENEE JACKSON	1400 CANDLELIGHT DR SPC 113	EUGENE 97402 US	NF+
Y	LORRAINE Y PETERSON	7269 N DEARING AVE	FRESNO 93720 US	NFI+
Y	DANA SUTHERLAND	8615 LA RIVIERA DR	SACRAMENTO 95826 US	NF+

Figure 2-263 Create j03_STAN_USPREP_NCB 6/8


j03_STAN_USPREP_NCB..NCB_BCUSTOMER_ISOCODE_USPREP.IN - Data Browser

ID	UPDATED	BY	BRANCH	ADVISOR	NAME	ADDR1	AI
20001500	2007-09-26 12:00:00	sysadm	12001500	13001500	Bruce H Anderson	6177 Purple Sage Ct	
20001501	2007-09-26 12:00:00	sysadm	12001500	13001501	Christina Anderson	6181 Camino Verde Dr	
20001502	2007-09-26 12:00:00	sysadm	12001501	13001502	Alexandra Anderson	321 Curie Drivee	
20001503	2007-09-26 12:00:00	sysadm	12001501	13001503	Torben Andersom	321 Curie Drivee	
20001504	2007-09-26 12:00:00	sysadm	12001500	13001504	Yesica Anderson	6177 Purple Sage Ct	
20001505	2007-09-26 12:00:00	sysadm	12001501	13001505	Kurt Madi	1603 Bel Air Avenue	
20001506	2007-09-26 12:00:00	sysadm	12001500	13001506	Maria Fanelli	Bel Air	
20001507	2007-09-26 12:00:00	sysadm	12001501	13001500	A Fanelli	2584 Junction Ave	
20001508	2007-09-26 12:00:00	sysadm	12001500	13001501	A Carter	5528 Muir Dr	
20001509	2007-09-26 12:00:00	sysadm	12001501	13001502	Barry Rosen	560 Galen	
20001510	2007-09-26 12:00:00	sysadm	12001500	13001503	Alok Alur	502 Gaylen	
20001511	2007-09-26 12:00:00	sysadm	12001501	13001504	Jastinderk Kumar	3424 Maroun Pl	
20001512	2007-09-26 12:00:00	sysadm	12001502	13001505	Aaron Jensen	1363 14th Ave	
20001513	2007-09-26 12:00:00	sysadm	12001503	13001506	Allan Jensen	PO Box 7424	
20001514	2007-09-26 12:00:00	sysadm	12001504	13001500	Andrew I Jensen	44 Montgomery St	St
20001515	2007-09-26 12:00:00	sysadm	12001505	13001501	Anette A Jensen	77 Grand View Ave	A
20001516	2007-09-26 12:00:00	sysadm	12001506	13001502	Anton T & Larue Jensen	258 Lisbon St	
20001517	2007-09-26 12:00:00	sysadm	12001507	13001503	Brandon Jensen	625 Hyde St	A
20001518	2007-09-26 12:00:00	sysadm	12001508	13001504	Steven C Preston	Embarcadero Ctr	
20001519	2007-09-26 12:00:00	sysadm	12001509	13001505	Allan Preston	720 Market St	St
20001520	2007-09-26 12:00:00	sysadm	12001510	13001506	Allison R Preston	831 Webster St	
20001521	2007-09-26 12:00:00	sysadm	12001511	13001500	Burr Preston	3726 Broderick St	A
20001522	2007-09-26 12:00:00	sysadm	12001512	13001501	Cathy Preston	425 Market St, Ste 2200	
20001523	2007-09-26 12:00:00	sysadm	12001513	13001502	Alan S Hopkins	1200 15th Ave, Apt 1	
20001524	2007-09-26 12:00:00	sysadm	12001514	13001503	Beata Hopkins	2250 24th St	A
20001525	2007-09-26 12:00:00	sysadm	12001515	13001504	C C Hopkins	2194 Folsom St	
20001526	2007-09-26 12:00:00	sysadm	12001516	13001505	Christine E Hopkins	407 Paris St	
20001527	2007-09-26 12:00:00	sysadm	12001517	13001506	Kelly Hopkins	1482 Rhode Island St	
20001528	2007-09-26 12:00:00	sysadm	12001518	13001500	Kimberly E Hopkins	2000 Post St, Apt 337	
20001529	2007-09-26 12:00:00	sysadm	12001519	13001501	Andreay Paley	555 California St, Ste 2900	
20001530	2007-09-26 12:00:00	sysadm	12001520	13001502	Jeremy Paley	720 Market St, Ste 900	
20001531	2007-09-26 12:00:00	sysadm	12001521	13001503	Rebecca Paley	222 Sutter St	F
20001532	2007-09-26 12:00:00	sysadm	12001522	13001504	Agnes Wyman	220 Montgomery St	
20001533	2007-09-26 12:00:00	sysadm	12001523	13001505	Gloria Wyman	631 Ofarrell St	A
20001534	2007-09-26 12:00:00	sysadm	12001524	13001506	H Wyman	1445 Oak St	
20001535	2007-09-26 12:00:00	sysadm	12001525	13001500	Jason Wyman	241 Oneida Ave	R
20001536	2007-09-26 12:00:00	sysadm	12001526	13001501	Martha S Peet	13055 23rd Pl NE	
20001537	2007-09-26 12:00:00	sysadm	12001529	13001502	Lorraine Y Peterson	7269 N Dearing Ave	
20001538	2007-09-26 12:00:00	sysadm	12001530	13001503	Dana Sutherland	8615 la Riviera Dr	

Figure 2-264 Create j03_STAN_USPREP_NCB 7/8


j03_STAN_USPREP_NCB..NCB_BCUSTOMER_ISOCODE_USPREP.IN - Data Browser

NameDomain_USPREP	AddressDomain_USPREP	AreaDomain_USPREP	Field1Pattern_USPREP	Field2Pattern_USPREP
BRUCE H ANDERSON	6177 PURPLE SAGE CT	SAN JOSE 95119 US	NFI+	A^++M
CHRISTINA ANDERSON	6181 CAMINO VERDE DR	SAN JOSE 95119 US	NF+	A^++M
ALEXANDRA ANDERSON	321 CURIE DRIVEE	SAN JOSE 95119 US	NF+	A^++
TORBEN ANDERSOM	321 CURIE DRIVEE	SAN JOSE 95119 US	N++	A^++
YESICA ANDERSON	6177 PURPLE SAGE CT	SAN JOSE 95119 US	N++	A^++M
KURT MADI	1603 BEL AIR AVENUE	SAN JOSE 95119 US	NF+	A^++T
MARIA FANELLI	BEL AIR	SAN JOSE 95119 US	NF+	A++
A FANELLI	2584 JUNCTION AVE	SAN JOSE 95119 US	NI+	A^TT
A CARTER	5528 MUIR DR	SAN JOSE 95119 US	NI+	A^+M
BARRY ROSEN	560 GALEN	SAN JOSE 95119 US	NF+	A^+
ALOK ALUR	502 GAYLEN	SAN JOSE 95119 US	N++	A^+
JASTINDERK KUMAR	3424 MAROUN PL	SAN JOSE 95119 US	N++	A^+T
AARON JENSEN	1363 14TH AVE	SAN FRANCISCO 94122 US	NF+	A^>T
ALLAN JENSEN	PO BOX 7424	SAN FRANCISCO 94120 US	NF+	ABB^
ANDREW I JENSEN	44 MONTGOMERY ST STE 3705	SAN FRANCISCO 94104 US	NFI+	A^+TU^
ANETTE A JENSEN	77 GRAND VIEW AVE APT 202	SAN FRANCISCO 94114 US	N+I+	A^+TTU^
ANTON T & LARUE JENSEN	258 LISBON ST	SAN FRANCISCO 94112 US	N+I&++	A^+T
BRANDON JENSEN	625 HYDE ST APT 1	SAN FRANCISCO 94109 US	NF+	A^+TU^
STEVEN C PRESTON	EMBARCADERO CTR	SAN FRANCISCO 94111 US	NFI+	A+M
ALLAN PRESTON	720 MARKET ST STE 900	SAN FRANCISCO 94102 US	NF+	A^+TU^
ALLISON R PRESTON	831 WEBSTER ST	SAN FRANCISCO 94117 US	NFI+	A^+T
BURR PRESTON	3726 BRODERICK ST APT 8	SAN FRANCISCO 94123 US	N++	A^+TU^
CATHY PRESTON	425 MARKET ST , STE 2200	SAN FRANCISCO 94105 US	NF+	A^+T,U^
ALAN S HOPKINS	1200 15TH AVE , APT 1	SAN FRANCISCO 94122 US	NFD+	A^>T,U^
BEATA HOPKINS	2250 24TH ST APT 334	SAN FRANCISCO 94107 US	N++	A^>TU^
C C HOPKINS	2194 FOLSOM ST	SAN FRANCISCO 94110 US	NII+	A^+T
CHRISTINE E HOPKINS	407 PARIS ST	SAN FRANCISCO 94112 US	NFD+	A^+T
KELLY HOPKINS	1482 RHODE ISLAND ST	SAN FRANCISCO 94107 US	NF+	A^+TT
KIMBERLY E HOPKINS	2000 POST ST , APT 337	SAN FRANCISCO 94115 US	NFD+	A^BT,U^
ANDREAY PALEY	555 CALIFORNIA ST , STE 2900	SAN FRANCISCO 94104 US	N++	A^ST,U^
JEREMY PALEY	720 MARKET ST , STE 900	SAN FRANCISCO 94102 US	NF+	A^+T,U^
REBECCA PALEY	222 SUTTER ST FL 6	SAN FRANCISCO 94108 US	NF+	A^+TM^
AGNES WYMAN	220 MONTGOMERY ST	SAN FRANCISCO 94104 US	NF+	A^+T
GLORIA WYMAN	631 OFARRELL ST APT 508	SAN FRANCISCO 94109 US	NF+	A^+TU^
H WYMAN	1445 OAK ST	SAN FRANCISCO 94117 US	NI+	A^+T
JASON WYMAN	241 ONEIDA AVE RM 81	SAN FRANCISCO 94112 US	NF+	A^+TU^
MARTHA S PEET	13055 23RD PL NE	SEATTLE 98125 US	NFD+	A^>TM
LORRAINE Y PETERSON	7269 N DEARING AVE	FRESNO 93720 US	NFI+	A^D+T
DANA SUTHERLAND	8615 LA RIVIERA DR	SACRAMENTO 95826 US	NF+	A^S+M

Figure 2-265 Create j03_STAN_USPREP_NCB 8/8

j04_INVCC_USPREP_INPUT_PATTERN_NCB

We then use the Investigate stage using the character concatenate investigate option on the input patterns that were generated by the Standardize stage in the previous step to determine the degree of success achieved by the USPREP domain-preprocessor rule set in parsing the tokens in the name and address fields into the correct domains.

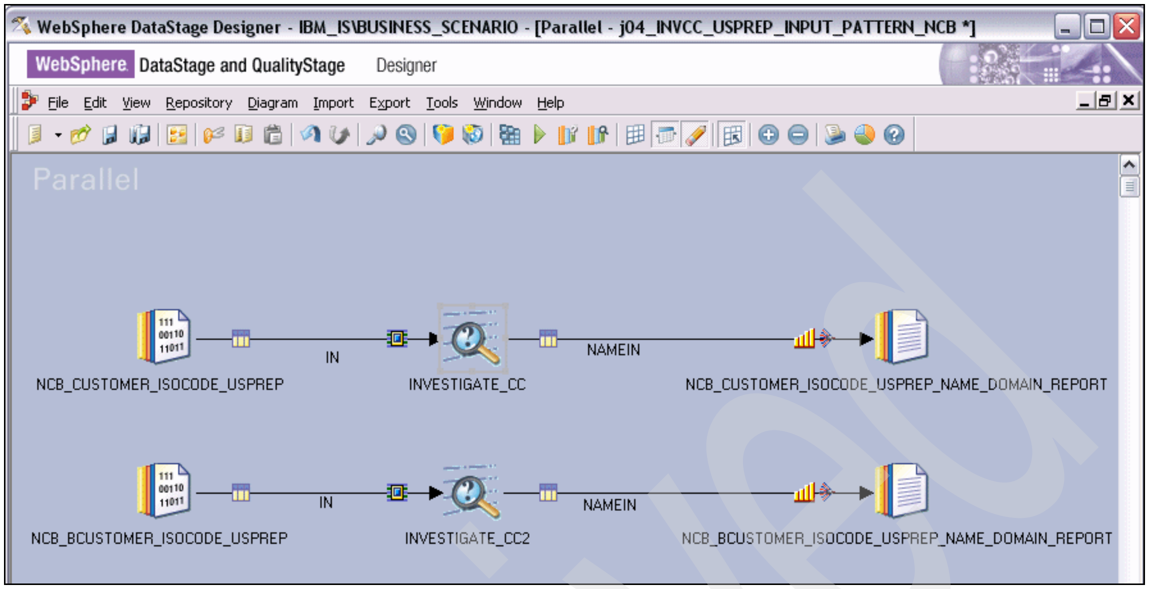
Figure 2-266 on page 737 shows the various stages that are used in this job. It includes the data sets that were created in “j03_STAN_USPREP_NCB” on page 729, one Investigate stage, and one output Sequential File stage for each Investigate stage. We modified the names of the stages as shown.

Because the configuration of this job is very similar to that described in "J02_INVCC_ISCODE" on page 186, we do not repeat it here. However, some of the configurations of interest are as follows:

- Figure 2-267 on page 737 and Figure 2-268 on page 738 show the INVESTIGATE_CC - Investigate Stage window with the Character Concatenate Investigate option, and columns InputPattern_USPREP with the *C* mask for each character and *X* masks on the NameDomain_USPREP, AddressDomain_USPREP, AreaDomain_USPREP, OutboundPattern_USPREP, NAME, ADDR1, ADDR2, CITY and ZIP columns in the CUSTOMER file.

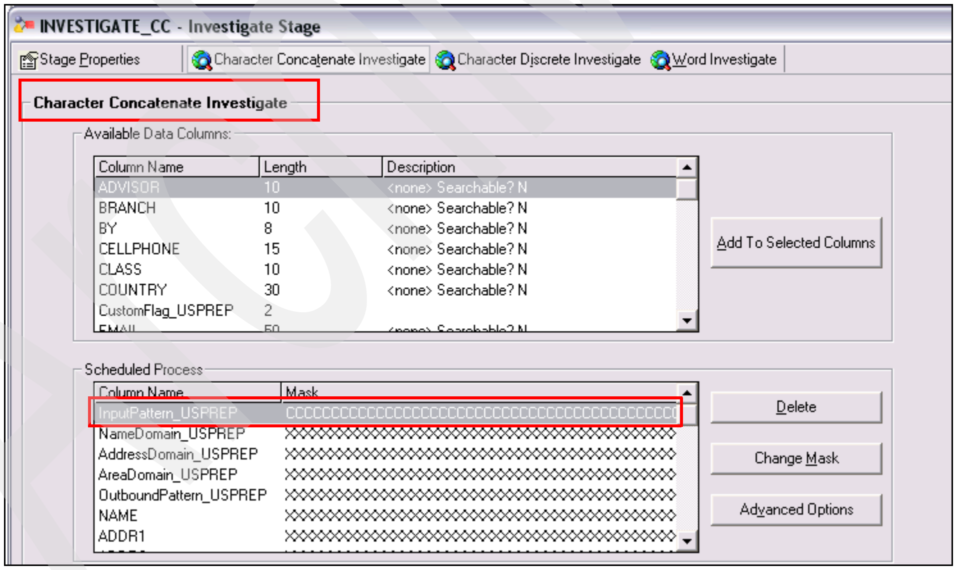
  Because the configuration of the INVESTIGATE_CC2 - Investigate Stage window for the BCUSTOMER file is similar, we do not repeat it here.
- After saving, compiling, and running this job, the job statistics are shown in Figure 2-269 on page 738.
- The output of the Investigate stages written to the sequential files are shown in Figure 2-270 on page 739 through Figure 2-273 on page 742 as follows:
  - Figure 2-270 on page 739 and Figure 2-271 on page 740 show the report corresponding to the name and address of the CUSTOMER file. The name and addresses seem to have been parsed correctly.
  - Figure 2-272 on page 741 and Figure 2-273 on page 742 show the report corresponding to the name and address of the BCUSTOMER file. The

A visual analysis of the pattern reports generated by the Investigate stage indicates no errors in the movement of data to the name, address, and area domains. However, some spelling errors (such as 321 Curie Drivee of Torben Anderson) are detected, which we decide to defer addressing till after subsequent stages. We, therefore, proceed to the next step, "j05_CASS_USPREP_NCB" on page 742.



WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j04_INVCC_USPREP_INPUT_PATTERN_NCB *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NCB_CUSTOMER_ISOCODE_USPREP
IN
INVESTIGATE_CC
NAMEIN
NCB_CUSTOMER_ISOCODE_USPREP_NAME_DOMAIN_REPORT
NCB_BCUSTOMER_ISOCODE_USPREP
IN
INVESTIGATE_CC2
NAMEIN
NCB_BCUSTOMER_ISOCODE_USPREP_NAME_DOMAIN_REPORT

Figure 2-266 Create j04_INVCC_USPREP_INPUT_PATTERN_NCB 1/8



INVESTIGATE_CC - Investigate Stage
Stage Properties
Character Concatenate Investigate
Character Discrete Investigate
Word Investigate
Add To Selected Columns
Delete
Change Mask
Advanced Options

Character Concatenate Investigate

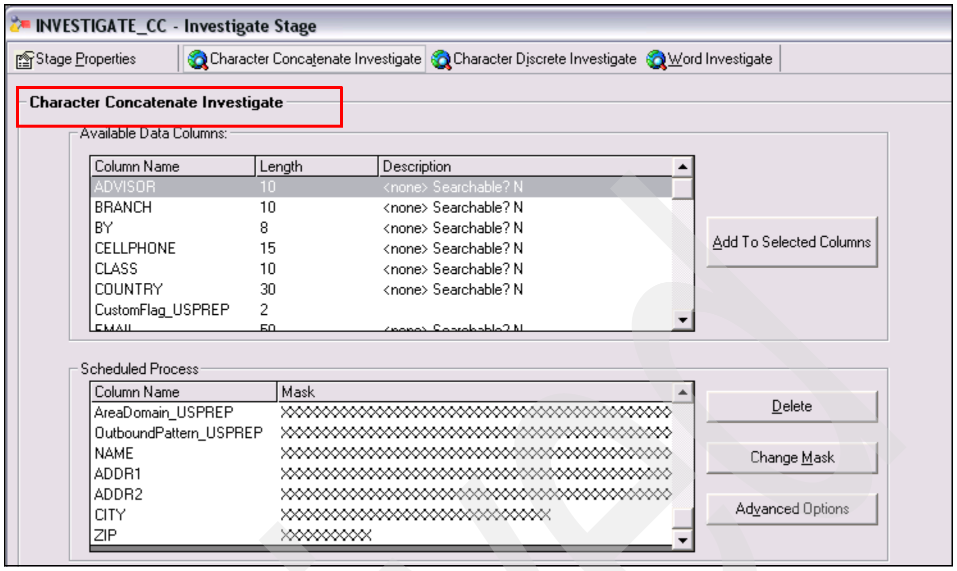
Available Data Columns:

Column Name	Length	Description
ADVISOR	10	<none> Searchable? N
BRANCH	10	<none> Searchable? N
BY	8	<none> Searchable? N
CELLPHONE	15	<none> Searchable? N
CLASS	10	<none> Searchable? N
COUNTRY	30	<none> Searchable? N
CustomFlag_USPREP	2	
EMAIL	50	<none> Searchable? N

Scheduled Process

Column Name	Mask
InputPattern_USPREP	CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
NameDomain_USPREP	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
AddressDomain_USPREP	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
AreaDomain_USPREP	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
OutboundPattern_USPREP	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
NAME	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
ADDR1	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Figure 2-267 Create j04_INVCC_USPREP_INPUT_PATTERN_NCB 2/8


INVESTIGATE_CC - Investigate Stage

Stage Properties | Character Concatenate Investigate | Character Discrete Investigate | Word Investigate

Character Concatenate Investigate

Available Data Columns:

Column Name	Length	Description
ADVISOR	10	<none> Searchable? N
BRANCH	10	<none> Searchable? N
BY	8	<none> Searchable? N
CELLPHONE	15	<none> Searchable? N
CLASS	10	<none> Searchable? N
COUNTRY	30	<none> Searchable? N
CustomFlag_USPREP	2	

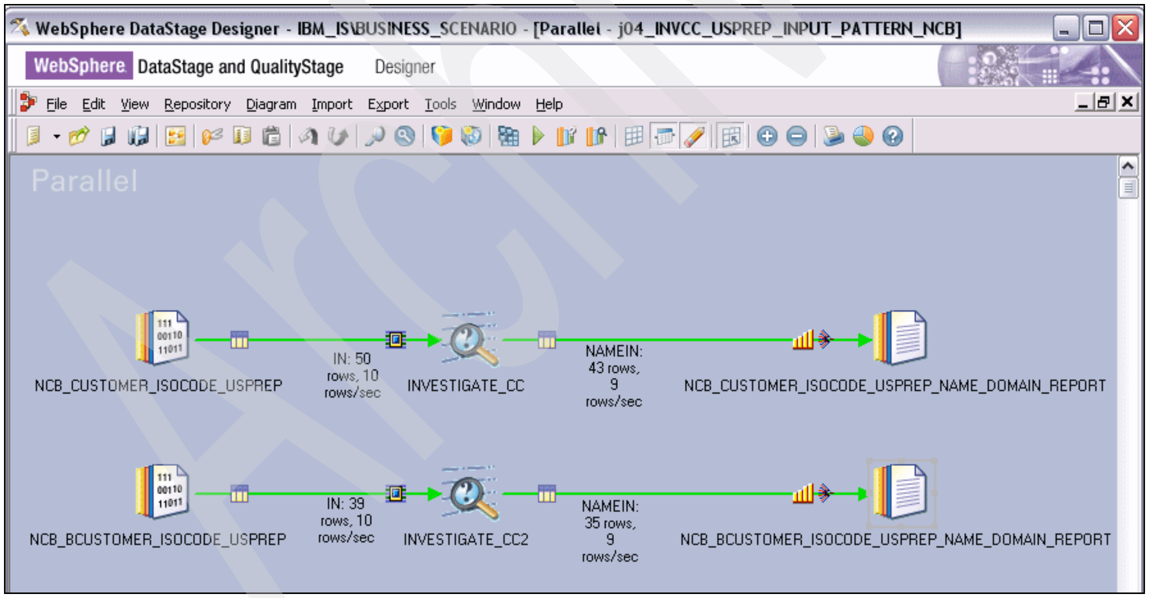
Add To Selected Columns

Scheduled Process

Column Name	Mask
AreaDomain_USPREP	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
OutboundPattern_USPREP	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
NAME	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
ADDR1	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
ADDR2	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
CITY	XXXXXXXXXXXXXXXXXXXXXXXXXXXX
ZIP	XXXXXXXXXX

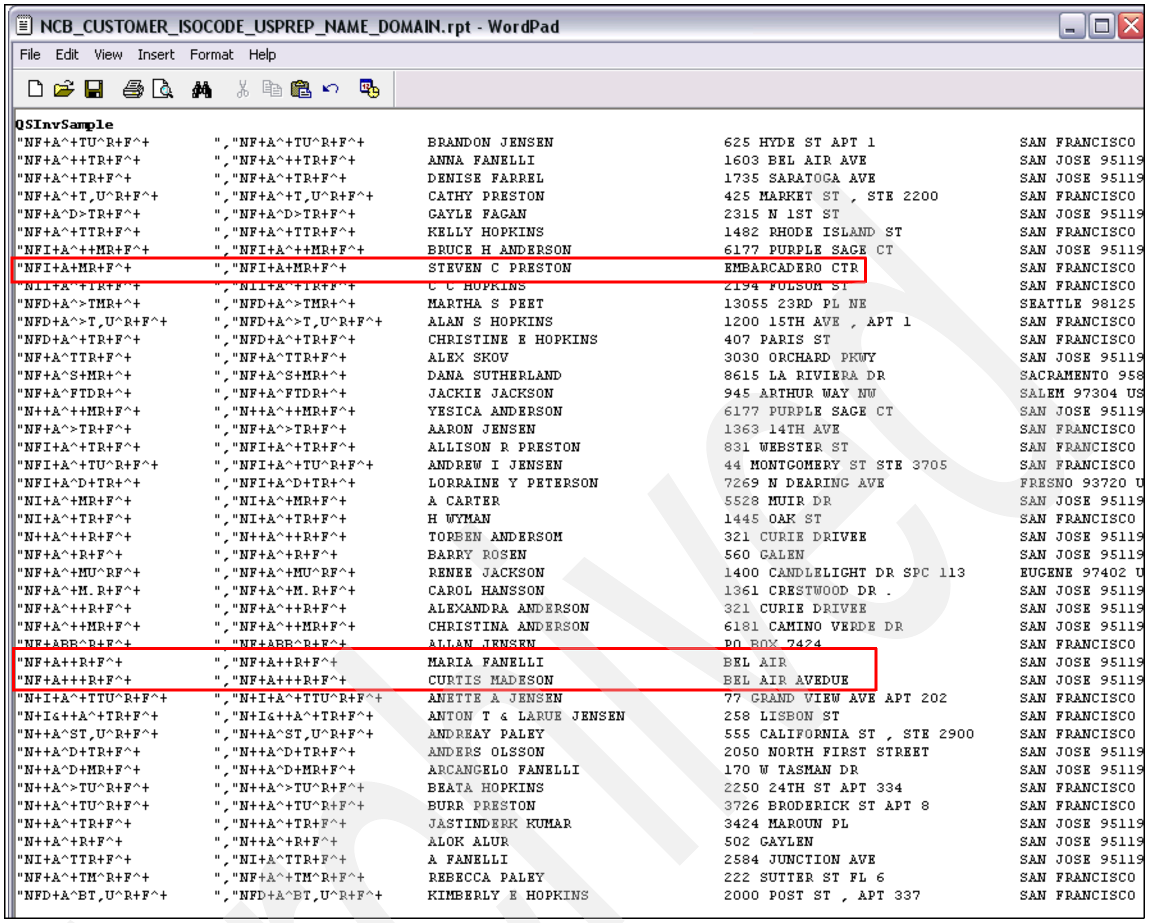
Delete
Change Mask
Advanced Options

Figure 2-268   Create j04_INVCC_USPREP_INPUT_PATTERN_NCB 3/8


WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j04_INVCC_USPREP_INPUT_PATTERN_NCB]

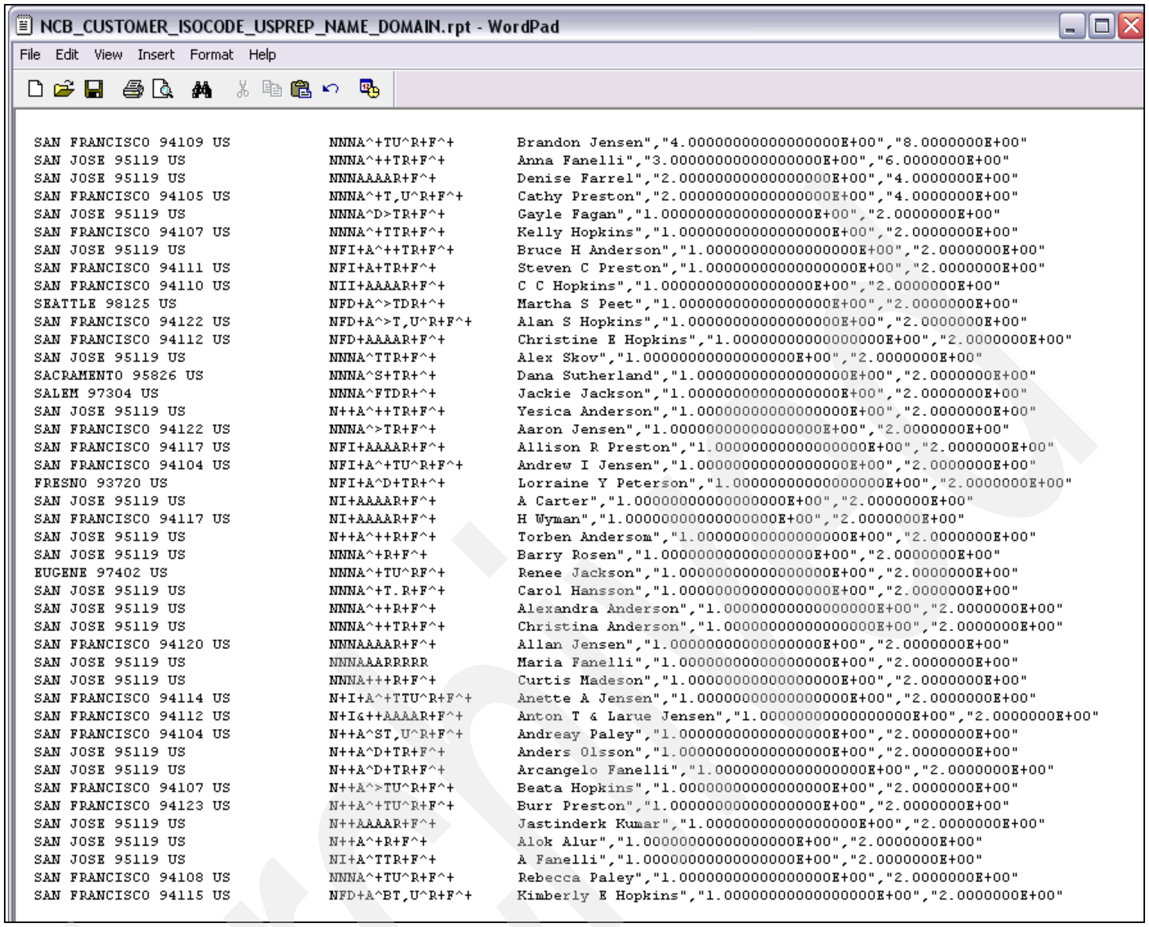
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NCB_CUSTOMER_ISOCODE_USPREP
IN: 50 rows, 10 rows/sec
INVESTIGATE_CC
NAMEIN: 43 rows, 9 rows/sec
NCB_CUSTOMER_ISOCODE_USPREP_NAME_DOMAIN_REPORT
NCB_BCUSTOMER_ISOCODE_USPREP
IN: 39 rows, 10 rows/sec
INVESTIGATE_CC2
NAMEIN: 35 rows, 9 rows/sec
NCB_BCUSTOMER_ISOCODE_USPREP_NAME_DOMAIN_REPORT

Figure 2-269   Create j04_INVCC_USPREP_INPUT_PATTERN_NCB 4/8


NCB_CUSTOMER_ISOCODE_USPREP_NAME_DOMAIN.rpt - WordPad
File Edit View Insert Format Help

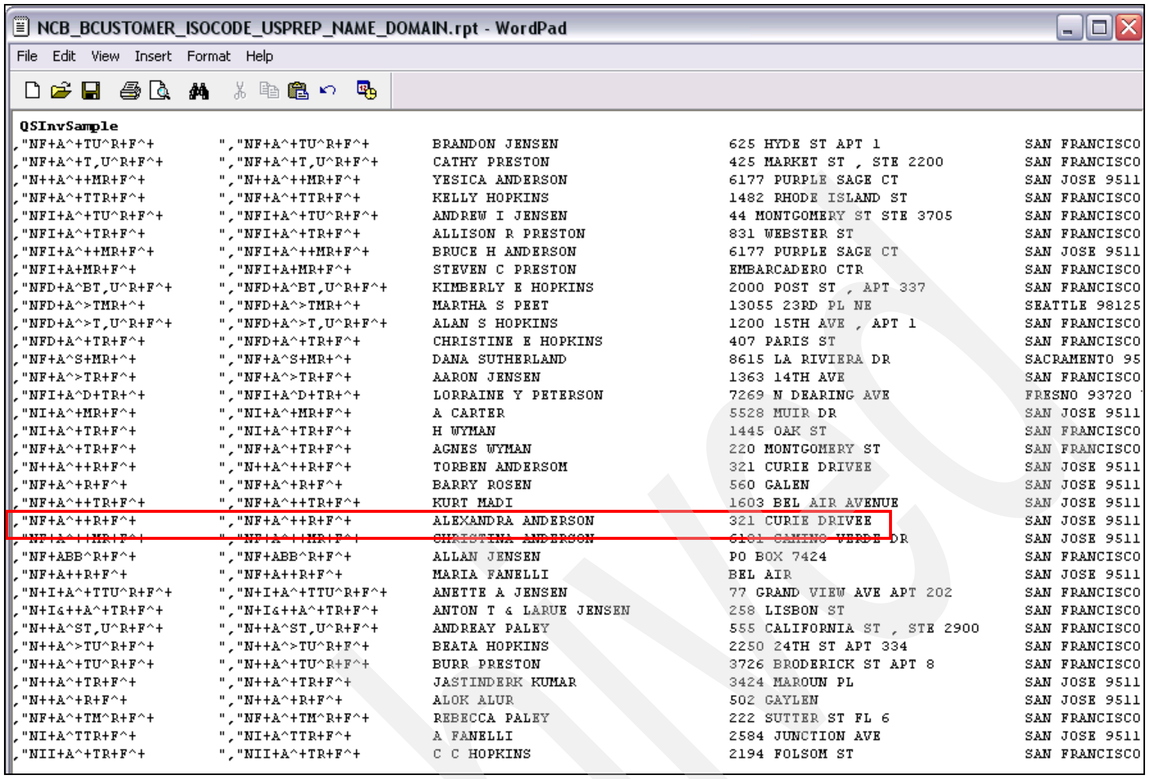
```
QSInvSample
"NF+A^+TU^R+F^+        ","NF+A^+TU^R+F^+        BRANDON JENSEN                  625 HYDE ST APT 1               SAN FRANCISCO
"NF+A^++TR+F^+         ","NF+A^++TR+F^+         ANNA FANELLI                    1603 BEL AIR AVE                SAN JOSE 95119
"NF+A^+TR+F^+          ","NF+A^+TR+F^+          DENISE FARREL                   1735 SARATOGA AVE               SAN JOSE 95119
"NF+A^+T,U^R+F^+       ","NF+A^+T,U^R+F^+       CATHY PRESTON                   425 MARKET ST , STE 2200        SAN FRANCISCO
"NF+A^D>TR+F^+         ","NF+A^D>TR+F^+         GAYLE FAGAN                     2315 N 1ST ST                   SAN JOSE 95119
"NF+A^+TTR+F^+         ","NF+A^+TTR+F^+         KELLY HOPKINS                   1482 RHODE ISLAND ST            SAN FRANCISCO
"NFI+A^++MR+F^+        ","NFI+A^++MR+F^+        BRUCE H ANDERSON                6177 PURPLE SAGE CT             SAN JOSE 95119
"NFI+A+MR+F^+          ","NFI+A+MR+F^+          STEVEN C PRESTON                EMBARCADERO CTR                 SAN FRANCISCO
"NII+A^+TR+F^+         ","NII+A^+TR+F^+         C C HOPKINS                     2194 FOLSOM ST                  SAN FRANCISCO
"NFD+A^>TMR+^+         ","NFD+A^>TMR+^+         MARTHA S PEET                   13055 23RD PL NE                SEATTLE 98125
"NFD+A^>T,U^R+F^+      ","NFD+A^>T,U^R+F^+      ALAN S HOPKINS                  1200 15TH AVE , APT 1           SAN FRANCISCO
"NFD+A^+TR+F^+         ","NFD+A^+TR+F^+         CHRISTINE E HOPKINS             407 PARIS ST                    SAN FRANCISCO
"NF+A^TTR+F^+          ","NF+A^TTR+F^+          ALEX SKOV                       3030 ORCHARD PKWY               SAN JOSE 95119
"NF+A^S+MR+^+          ","NF+A^S+MR+^+          DANA SUTHERLAND                 8615 LA RIVIERA DR              SACRAMENTO 958
"NF+A^FTDR+^+          ","NF+A^FTDR+^+          JACKIE JACKSON                  945 ARTHUR WAY NW               SALEM 97304 US
"N++A^++MR+F^+         ","N++A^++MR+F^+         YESICA ANDERSON                 6177 PURPLE SAGE CT             SAN JOSE 95119
"NF+A^>TR+F^+          ","NF+A^>TR+F^+          AARON JENSEN                    1363 14TH AVE                   SAN FRANCISCO
"NFI+A^+TR+F^+         ","NFI+A^+TR+F^+         ALLISON R PRESTON               831 WEBSTER ST                  SAN FRANCISCO
"NFI+A^+TU^R+F^+       ","NFI+A^+TU^R+F^+       ANDREW I JENSEN                 44 MONTGOMERY ST STE 3705       SAN FRANCISCO
"NFI+A^D+TR+^+         ","NFI+A^D+TR+^+         LORRAINE Y PETERSON             7269 N DEARING AVE              FRESNO 93720 U
"NI+A^+MR+F^+          ","NI+A^+MR+F^+          A CARTER                        5528 MUIR DR                    SAN JOSE 95119
"NI+A^+TR+F^+          ","NI+A^+TR+F^+          H WYMAN                         1445 OAK ST                     SAN FRANCISCO
"N++A^++R+F^+          ","N++A^++R+F^+          TORBEN ANDERSOM                 321 CURIE DRIVEE                SAN JOSE 95119
"NF+A^+R+F^+           ","NF+A^+R+F^+           BARRY ROSEN                     560 GALEN                       SAN JOSE 95119
"NF+A^+MU^RF^+         ","NF+A^+MU^RF^+         RENEE JACKSON                   1400 CANDLELIGHT DR SPC 113     EUGENE 97402 U
"NF+A^+M.R+F^+         ","NF+A^+M.R+F^+         CAROL HANSSON                   1361 CRESTWOOD DR .             SAN JOSE 95119
"NF+A^++R+F^+          ","NF+A^++R+F^+          ALEXANDRA ANDERSON              321 CURIE DRIVEE                SAN JOSE 95119
"NF+A^++MR+F^+         ","NF+A^++MR+F^+         CHRISTINA ANDERSON              6181 CAMINO VERDE DR            SAN JOSE 95119
"NF+ABB^R+F^+          ","NF+ABB^R+F^+          ALLAN JENSEN                    PO BOX 7424                     SAN FRANCISCO
"NF+A++R+F^+           ","NF+A++R+F^+           MARIA FANELLI                   BEL AIR                         SAN JOSE 95119
"NF+A+++R+F^+          ","NF+A+++R+F^+          CURTIS MADESON                  BEL AIR AVEDUE                  SAN JOSE 95119
"N+I+A^+TTU^R+F^+      ","N+I+A^+TTU^R+F^+      ANETTE A JENSEN                 77 GRAND VIEW AVE APT 202       SAN FRANCISCO
"N+I&++A^+TR+F^+       ","N+I&++A^+TR+F^+       ANTON T & LARUE JENSEN          258 LISBON ST                   SAN FRANCISCO
"N++A^ST,U^R+F^+       ","N++A^ST,U^R+F^+       ANDREAY PALEY                   555 CALIFORNIA ST , STE 2900    SAN FRANCISCO
"N++A^D+TR+F^+         ","N++A^D+TR+F^+         ANDERS OLSSON                   2050 NORTH FIRST STREET         SAN JOSE 95119
"N++A^D+MR+F^+         ","N++A^D+MR+F^+         ARCANGELO FANELLI               170 W TASMAN DR                 SAN JOSE 95119
"N++A^>TU^R+F^+        ","N++A^>TU^R+F^+        BEATA HOPKINS                   2250 24TH ST APT 334            SAN FRANCISCO
"N++A^+TU^R+F^+        ","N++A^+TU^R+F^+        BURR PRESTON                    3726 BRODERICK ST APT 8         SAN FRANCISCO
"N++A^+TR+F^+          ","N++A^+TR+F^+          JASTINDERK KUMAR                3424 MAROUN PL                  SAN JOSE 95119
"N++A^+R+F^+           ","N++A^+R+F^+           ALOK ALUR                       502 GAYLEN                      SAN JOSE 95119
"NI+A^TTR+F^+          ","NI+A^TTR+F^+          A FANELLI                       2584 JUNCTION AVE               SAN JOSE 95119
"NF+A^+TM^R+F^+        ","NF+A^+TM^R+F^+        REBECCA PALEY                   222 SUTTER ST FL 6              SAN FRANCISCO
"NFD+A^BT,U^R+F^+      ","NFD+A^BT,U^R+F^+      KIMBERLY E HOPKINS              2000 POST ST , APT 337          SAN FRANCISCO
```

Figure 2-270 Create j04_INVCC_USPREP_INPUT_PATTERN_NCB 5/8


NCB_CUSTOMER_ISOCODE_USPREP_NAME_DOMAIN.rpt - WordPad

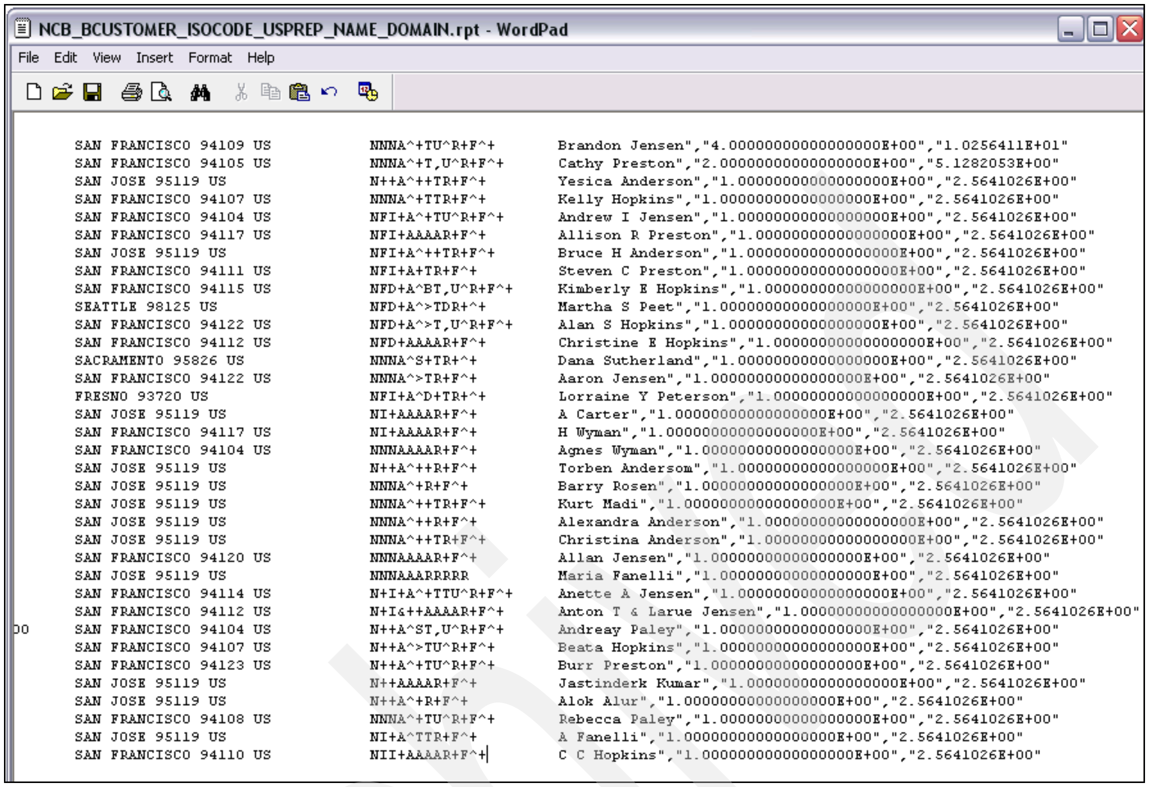
```
SAN FRANCISCO 94109 US          NNNA^+TU^R+F^+      Brandon Jensen","4.00000000000000000E+00","8.0000000E+00"
SAN JOSE 95119 US               NNNA^++TR+F^+       Anna Fanelli","3.00000000000000000E+00","6.0000000E+00"
SAN JOSE 95119 US               NNNAAAAR+F^+        Denise Farrel","2.00000000000000000E+00","4.0000000E+00"
SAN FRANCISCO 94105 US          NNNA^+T,U^R+F^+     Cathy Preston","2.00000000000000000E+00","4.0000000E+00"
SAN JOSE 95119 US               NNNA^D>TR+F^+       Gayle Fagan","1.00000000000000000E+00","2.0000000E+00"
SAN FRANCISCO 94107 US          NNNA^+TTR+F^+       Kelly Hopkins","1.00000000000000000E+00","2.0000000E+00"
SAN JOSE 95119 US               NFI+A^++TR+F^+      Bruce H Anderson","1.00000000000000000E+00","2.0000000E+00"
SAN FRANCISCO 94111 US          NFI+A+TR+F^+        Steven C Preston","1.00000000000000000E+00","2.0000000E+00"
SAN FRANCISCO 94110 US          NII+AAAAR+F^+       C C Hopkins","1.00000000000000000E+00","2.0000000E+00"
SEATTLE 98125 US                NFD+A^>TDR+^+       Martha S Peet","1.00000000000000000E+00","2.0000000E+00"
SAN FRANCISCO 94122 US          NFD+A^>T,U^R+F^+    Alan S Hopkins","1.00000000000000000E+00","2.0000000E+00"
SAN FRANCISCO 94112 US          NFD+AAAAR+F^+       Christine E Hopkins","1.00000000000000000E+00","2.0000000E+00"
SAN JOSE 95119 US               NNNA^TTR+F^+        Alex Skov","1.00000000000000000E+00","2.0000000E+00"
SACRAMENTO 95826 US             NNNA^S+TR+^+        Dana Sutherland","1.00000000000000000E+00","2.0000000E+00"
SALEM 97304 US                  NNNA^FTDR+^+        Jackie Jackson","1.00000000000000000E+00","2.0000000E+00"
SAN JOSE 95119 US               N++A^++TR+F^+       Yesica Anderson","1.00000000000000000E+00","2.0000000E+00"
SAN FRANCISCO 94122 US          NNNA^>TR+F^+        Aaron Jensen","1.00000000000000000E+00","2.0000000E+00"
SAN FRANCISCO 94117 US          NFI+AAAAR+F^+       Allison R Preston","1.00000000000000000E+00","2.0000000E+00"
SAN FRANCISCO 94104 US          NFI+A^+TU^R+F^+     Andrew I Jensen","1.00000000000000000E+00","2.0000000E+00"
FRESNO 93720 US                 NFI+A^D+TR+^+       Lorraine Y Peterson","1.00000000000000000E+00","2.0000000E+00"
SAN JOSE 95119 US               NI+AAAAR+F^+        A Carter","1.00000000000000000E+00","2.0000000E+00"
SAN FRANCISCO 94117 US          NI+AAAAR+F^+        H Wyman","1.00000000000000000E+00","2.0000000E+00"
SAN JOSE 95119 US               N++A^++R+F^+        Torben Andersom","1.00000000000000000E+00","2.0000000E+00"
SAN JOSE 95119 US               NNNA^+R+F^+         Barry Rosen","1.00000000000000000E+00","2.0000000E+00"
EUGENE 97402 US                 NNNA^+TU^RF^+       Renee Jackson","1.00000000000000000E+00","2.0000000E+00"
SAN JOSE 95119 US               NNNA^+T.R+F^+       Carol Hansson","1.00000000000000000E+00","2.0000000E+00"
SAN JOSE 95119 US               NNNA^++R+F^+        Alexandra Anderson","1.00000000000000000E+00","2.0000000E+00"
SAN JOSE 95119 US               NNNA^++TR+F^+       Christina Anderson","1.00000000000000000E+00","2.0000000E+00"
SAN FRANCISCO 94120 US          NNNAAAAR+F^+        Allan Jensen","1.00000000000000000E+00","2.0000000E+00"
SAN JOSE 95119 US               NNNAAARRRRR         Maria Fanelli","1.00000000000000000E+00","2.0000000E+00"
SAN JOSE 95119 US               NNNA+++R+F^+        Curtis Madeson","1.00000000000000000E+00","2.0000000E+00"
SAN FRANCISCO 94114 US          N+I+A^+TTU^R+F^+    Anette A Jensen","1.00000000000000000E+00","2.0000000E+00"
SAN FRANCISCO 94112 US          N+I&++AAAAR+F^+     Anton T & Larue Jensen","1.00000000000000000E+00","2.0000000E+00"
SAN FRANCISCO 94104 US          N++A^ST,U^R+F^+     Andreay Paley","1.00000000000000000E+00","2.0000000E+00"
SAN JOSE 95119 US               N++A^D+TR+F^+       Anders Olsson","1.00000000000000000E+00","2.0000000E+00"
SAN JOSE 95119 US               N++A^D+TR+F^+       Arcangelo Fanelli","1.00000000000000000E+00","2.0000000E+00"
SAN FRANCISCO 94107 US          N++A^>TU^R+F^+      Beata Hopkins","1.00000000000000000E+00","2.0000000E+00"
SAN FRANCISCO 94123 US          N++A^+TU^R+F^+      Burr Preston","1.00000000000000000E+00","2.0000000E+00"
SAN JOSE 95119 US               N++AAAAR+F^+        Jastinderk Kumar","1.00000000000000000E+00","2.0000000E+00"
SAN JOSE 95119 US               N++A^+R+F^+         Alok Alur","1.00000000000000000E+00","2.0000000E+00"
SAN JOSE 95119 US               NI+A^TTR+F^+        A Fanelli","1.00000000000000000E+00","2.0000000E+00"
SAN FRANCISCO 94108 US          NNNA^+TU^R+F^+      Rebecca Paley","1.00000000000000000E+00","2.0000000E+00"
SAN FRANCISCO 94115 US          NFD+A^BT,U^R+F^+    Kimberly E Hopkins","1.00000000000000000E+00","2.0000000E+00"
```

Figure 2-271   Create j04_INVCC_USPREP_INPUT_PATTERN_NCB 6/8


NCB_BCUSTOMER_ISOCODE_USPREP_NAME_DOMAIN.rpt - WordPad
File Edit View Insert Format Help

```
QSInvSample
,"NF+A^+TU^R+F^+       ","NF+A^+TU^R+F^+       BRANDON JENSEN                  625 HYDE ST APT 1               SAN FRANCISCO
,"NF+A^+T,U^R+F^+      ","NF+A^+T,U^R+F^+      CATHY PRESTON                   425 MARKET ST , STE 2200        SAN FRANCISCO
,"N++A^++MR+F^+        ","N++A^++MR+F^+        YESICA ANDERSON                 6177 PURPLE SAGE CT             SAN JOSE 9511
,"NF+A^+TTR+F^+        ","NF+A^+TTR+F^+        KELLY HOPKINS                   1482 RHODE ISLAND ST            SAN FRANCISCO
,"NFI+A^+TU^R+F^+      ","NFI+A^+TU^R+F^+      ANDREW I JENSEN                 44 MONTGOMERY ST STE 3705       SAN FRANCISCO
,"NFI+A^+TR+F^+        ","NFI+A^+TR+F^+        ALLISON R PRESTON               831 WEBSTER ST                  SAN FRANCISCO
,"NFI+A^++MR+F^+       ","NFI+A^++MR+F^+       BRUCE H ANDERSON                6177 PURPLE SAGE CT             SAN JOSE 9511
,"NFI+A+MR+F^+         ","NFI+A+MR+F^+         STEVEN C PRESTON                EMBARCADERO CTR                 SAN FRANCISCO
,"NFD+A^BT,U^R+F^+     ","NFD+A^BT,U^R+F^+     KIMBERLY E HOPKINS              2000 POST ST , APT 337          SAN FRANCISCO
,"NFD+A^>TMR+^+        ","NFD+A^>TMR+^+        MARTHA S PEET                   13055 23RD PL NE                SEATTLE 98125
,"NFD+A^>T,U^R+F^+     ","NFD+A^>T,U^R+F^+     ALAN S HOPKINS                  1200 15TH AVE , APT 1           SAN FRANCISCO
,"NFD+A^+TR+F^+        ","NFD+A^+TR+F^+        CHRISTINE E HOPKINS             407 PARIS ST                    SAN FRANCISCO
,"NF+A^S+MR+^+         ","NF+A^S+MR+^+         DANA SUTHERLAND                 8615 LA RIVIERA DR              SACRAMENTO 95
,"NF+A^>TR+F^+         ","NF+A^>TR+F^+         AARON JENSEN                    1363 14TH AVE                   SAN FRANCISCO
,"NFI+A^D+TR+^+        ","NFI+A^D+TR+^+        LORRAINE Y PETERSON             7269 N DEARING AVE              FRESNO 93720
,"NI+A^+MR+F^+         ","NI+A^+MR+F^+         A CARTER                        5528 MUIR DR                    SAN JOSE 9511
,"NI+A^+TR+F^+         ","NI+A^+TR+F^+         H WYMAN                         1445 OAK ST                     SAN FRANCISCO
,"NF+A^+TR+F^+         ","NF+A^+TR+F^+         AGNES WYMAN                     220 MONTGOMERY ST               SAN FRANCISCO
,"N++A^++R+F^+         ","N++A^++R+F^+         TORBEN ANDERSOM                 321 CURIE DRIVEE                SAN JOSE 9511
,"NF+A^+R+F^+          ","NF+A^+R+F^+          BARRY ROSEN                     560 GALEN                       SAN JOSE 9511
,"NF+A^++TR+F^+        ","NF+A^++TR+F^+        KURT MADI                       1603 BEL AIR AVENUE             SAN JOSE 9511
,"NF+A^++R+F^+         ","NF+A^++R+F^+         ALEXANDRA ANDERSON              321 CURIE DRIVEE                SAN JOSE 9511
,"NF+A^++MR+F^+        ","NF+A^++MR+F^+        CHRISTINA ANDERSON              6101 CAMINO VERDE DR            SAN JOSE 9511
,"NF+ABB^R+F^+         ","NF+ABB^R+F^+         ALLAN JENSEN                    PO BOX 7424                     SAN FRANCISCO
,"NF+A++R+F^+          ","NF+A++R+F^+          MARIA FANELLI                   BEL AIR                         SAN JOSE 9511
,"N+I+A^+TTU^R+F^+     ","N+I+A^+TTU^R+F^+     ANETTE A JENSEN                 77 GRAND VIEW AVE APT 202       SAN FRANCISCO
,"N+I&++A^+TR+F^+      ","N+I&++A^+TR+F^+      ANTON T & LARUE JENSEN          258 LISBON ST                   SAN FRANCISCO
,"N++A^ST,U^R+F^+      ","N++A^ST,U^R+F^+      ANDREAY PALEY                   555 CALIFORNIA ST , STE 2900    SAN FRANCISCO
,"N++A^>TU^R+F^+       ","N++A^>TU^R+F^+       BEATA HOPKINS                   2250 24TH ST APT 334            SAN FRANCISCO
,"N++A^+TU^R+F^+       ","N++A^+TU^R+F^+       BURR PRESTON                    3726 BRODERICK ST APT 8         SAN FRANCISCO
,"N++A^+TR+F^+         ","N++A^+TR+F^+         JASTINDERK KUMAR                3424 MAROUN PL                  SAN JOSE 9511
,"N++A^+R+F^+          ","N++A^+R+F^+          ALOK ALUR                       502 GAYLEN                      SAN JOSE 9511
,"NF+A^+TM^R+F^+       ","NF+A^+TM^R+F^+       REBECCA PALEY                   222 SUTTER ST FL 6              SAN FRANCISCO
,"NI+A^TTR+F^+         ","NI+A^TTR+F^+         A FANELLI                       2584 JUNCTION AVE               SAN JOSE 9511
,"NII+A^+TR+F^+        ","NII+A^+TR+F^+        C C HOPKINS                     2194 FOLSOM ST                  SAN FRANCISCO
```

Figure 2-272 Create j04_INVCC_USPREP_INPUT_PATTERN_NCB 7/8


NCB_BCUSTOMER_ISOCODE_USPREP_NAME_DOMAIN.rpt - WordPad
File Edit View Insert Format Help

```
      SAN FRANCISCO 94109 US          NNNA^+TU^R+F^+       Brandon Jensen","4.0000000000000000E+00","1.0256411E+01"
      SAN FRANCISCO 94105 US          NNNA^+T,U^R+F^+      Cathy Preston","2.0000000000000000E+00","5.1282053E+00"
      SAN JOSE 95119 US               N++A^++TR+F^+        Yesica Anderson","1.0000000000000000E+00","2.5641026E+00"
      SAN FRANCISCO 94107 US          NNNA^+TTR+F^+        Kelly Hopkins","1.0000000000000000E+00","2.5641026E+00"
      SAN FRANCISCO 94104 US          NFI+A^+TU^R+F^+      Andrew I Jensen","1.0000000000000000E+00","2.5641026E+00"
      SAN FRANCISCO 94117 US          NFI+AAAAR+F^+        Allison R Preston","1.0000000000000000E+00","2.5641026E+00"
      SAN JOSE 95119 US               NFI+A^++TR+F^+       Bruce H Anderson","1.0000000000000000E+00","2.5641026E+00"
      SAN FRANCISCO 94111 US          NFI+A+TR+F^+         Steven C Preston","1.0000000000000000E+00","2.5641026E+00"
      SAN FRANCISCO 94115 US          NFD+A^BT,U^R+F^+     Kimberly E Hopkins","1.0000000000000000E+00","2.5641026E+00"
      SEATTLE 98125 US                NFD+A^>TDR+^+        Martha S Peet","1.0000000000000000E+00","2.5641026E+00"
      SAN FRANCISCO 94122 US          NFD+A^>T,U^R+F^+     Alan S Hopkins","1.0000000000000000E+00","2.5641026E+00"
      SAN FRANCISCO 94112 US          NFD+AAAAR+F^+        Christine E Hopkins","1.0000000000000000E+00","2.5641026E+00"
      SACRAMENTO 95826 US             NNNA^S+TR+^+         Dana Sutherland","1.0000000000000000E+00","2.5641026E+00"
      SAN FRANCISCO 94122 US          NNNA^>TR+F^+         Aaron Jensen","1.0000000000000000E+00","2.5641026E+00"
      FRESNO 93720 US                 NFI+A^D+TR+^+        Lorraine Y Peterson","1.0000000000000000E+00","2.5641026E+00"
      SAN JOSE 95119 US               NI+AAAAR+F^+         A Carter","1.0000000000000000E+00","2.5641026E+00"
      SAN FRANCISCO 94117 US          NI+AAAAR+F^+         H Wyman","1.0000000000000000E+00","2.5641026E+00"
      SAN FRANCISCO 94104 US          NNNAAAAR+F^+         Agnes Wyman","1.0000000000000000E+00","2.5641026E+00"
      SAN JOSE 95119 US               N++A^++R+F^+         Torben Andersom","1.0000000000000000E+00","2.5641026E+00"
      SAN JOSE 95119 US               NNNA^+R+F^+          Barry Rosen","1.0000000000000000E+00","2.5641026E+00"
      SAN JOSE 95119 US               NNNA^++TR+F^+        Kurt Madi","1.0000000000000000E+00","2.5641026E+00"
      SAN JOSE 95119 US               NNNA^++R+F^+         Alexandra Anderson","1.0000000000000000E+00","2.5641026E+00"
      SAN JOSE 95119 US               NNNA^++TR+F^+        Christina Anderson","1.0000000000000000E+00","2.5641026E+00"
      SAN FRANCISCO 94120 US          NNNAAAAR+F^+         Allan Jensen","1.0000000000000000E+00","2.5641026E+00"
      SAN JOSE 95119 US               NNNAAAARRRRR         Maria Fanelli","1.0000000000000000E+00","2.5641026E+00"
      SAN FRANCISCO 94114 US          N+I+A^+TTU^R+F^+     Anette A Jensen","1.0000000000000000E+00","2.5641026E+00"
      SAN FRANCISCO 94112 US          N+I&++AAAAR+F^+      Anton T & Larue Jensen","1.0000000000000000E+00","2.5641026E+00"
00    SAN FRANCISCO 94104 US          N++A^ST,U^R+F^+      Andreay Paley","1.0000000000000000E+00","2.5641026E+00"
      SAN FRANCISCO 94107 US          N++A^>TU^R+F^+       Beata Hopkins","1.0000000000000000E+00","2.5641026E+00"
      SAN FRANCISCO 94123 US          N++A^+TU^R+F^+       Burr Preston","1.0000000000000000E+00","2.5641026E+00"
      SAN JOSE 95119 US               N++AAAAR+F^+         Jastinderk Kumar","1.0000000000000000E+00","2.5641026E+00"
      SAN JOSE 95119 US               N++A^+R+F^+          Alok Alur","1.0000000000000000E+00","2.5641026E+00"
      SAN FRANCISCO 94108 US          NNNA^+TU^R+F^+       Rebecca Paley","1.0000000000000000E+00","2.5641026E+00"
      SAN JOSE 95119 US               NI+A^TTR+F^+         A Fanelli","1.0000000000000000E+00","2.5641026E+00"
      SAN FRANCISCO 94110 US          NII+AAAAR+F^+        C C Hopkins","1.0000000000000000E+00","2.5641026E+00"
```

Figure 2-273 Create j04_INVCC_USPREP_INPUT_PATTERN_NCB 8/8

j05_CASS_USPREP_NCB

In this step, we use the CASS stage to validate, correct, and standardize the U.S. addresses in the Address domain. We also include a Transformer stage to add a second address line column to customer file because CASS requires two address lines as input for its processing.

Note: As mentioned earlier, CASS is a separately priced module that requires installation of the CASS module.

Figure 2-274 on page 743 shows the various stages that are used in this job. It includes the data sets that were created in “j03_STAN_USPREP_NCB” on page 729, a Transformer stage, a CASS stage, and a Data Set stage for each input data set. We modified the names of the stages as shown.

Because the configuration of this job is very similar to that described in "J05_CASS_USPREP" on page 219, we do not repeat it here. However, some of the configurations of interest are as follows:

- Figure 2-275 on page 744 shows the CASS stage configuration for the CUSTOMER file with details such as the assignment of columns to Address Line 1 and Address Line 2,and location of the output file.

  The configuration of the CASS stage for the BCUSTOMER file is similar, and we do not repeat it here.

- After saving, compiling, and running this job (Figure 2-276 on page 744), review the contents of the output of the CASS stage as shown in Figure 2-277 on page 745 through Figure 2-284 on page 752. It shows validation errors with some addresses as described in "J05_CASS_USPREP" on page 219.

Note: As mentioned earlier, generally, any data error should be presented for review by appropriate personnel as early as possible in the process. An A1 should be investigated and appropriate action taken.

Proceed now to "j06_INVCC_CASS_NCB" on page 752.

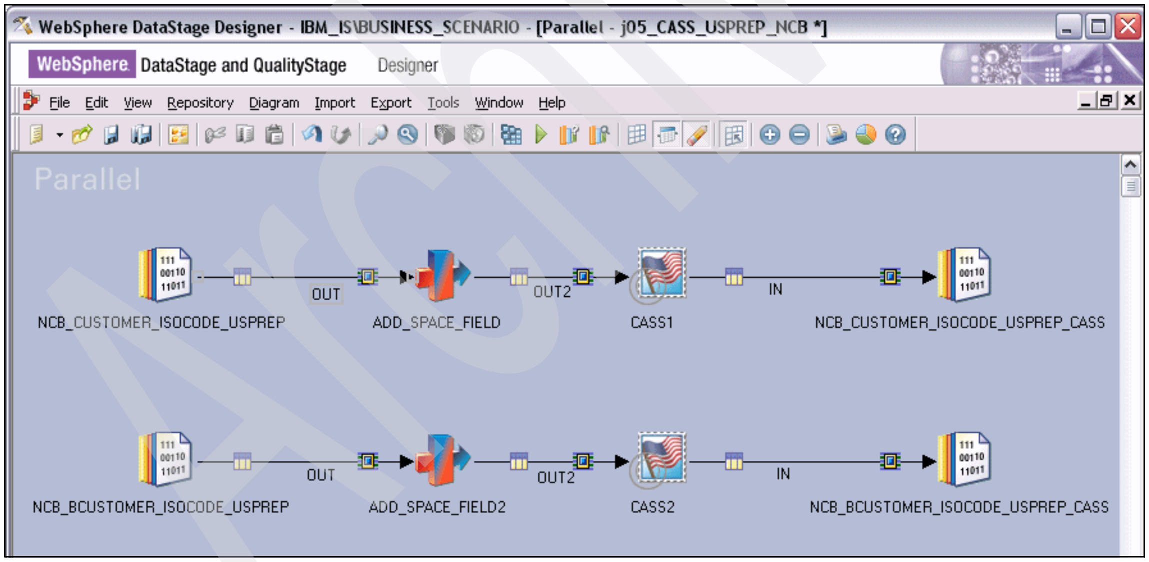
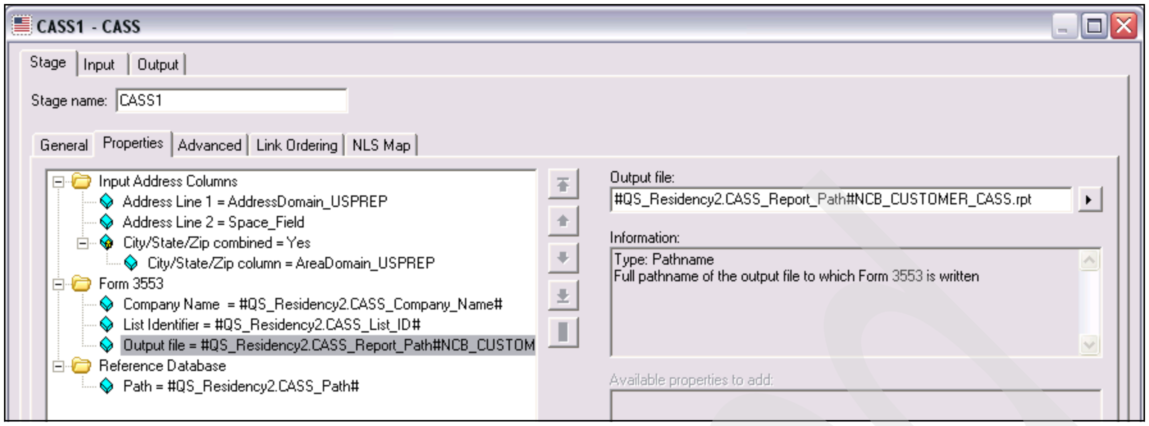

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j05_CASS_USPREP_NCB *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NCB_CUSTOMER_ISOCODE_USPREP
OUT
ADD_SPACE_FIELD
OUT2
CASS1
IN
NCB_CUSTOMER_ISOCODE_USPREP_CASS
NCB_BCUSTOMER_ISOCODE_USPREP
OUT
ADD_SPACE_FIELD2
OUT2
CASS2
IN
NCB_BCUSTOMER_ISOCODE_USPREP_CASS

Figure 2-274 Create j05_CASS_USPREP_NCB 1/11



CASS1 - CASS

Stage | Input | Output

Stage name: CASS1

General | Properties | Advanced | Link Ordering | NLS Map

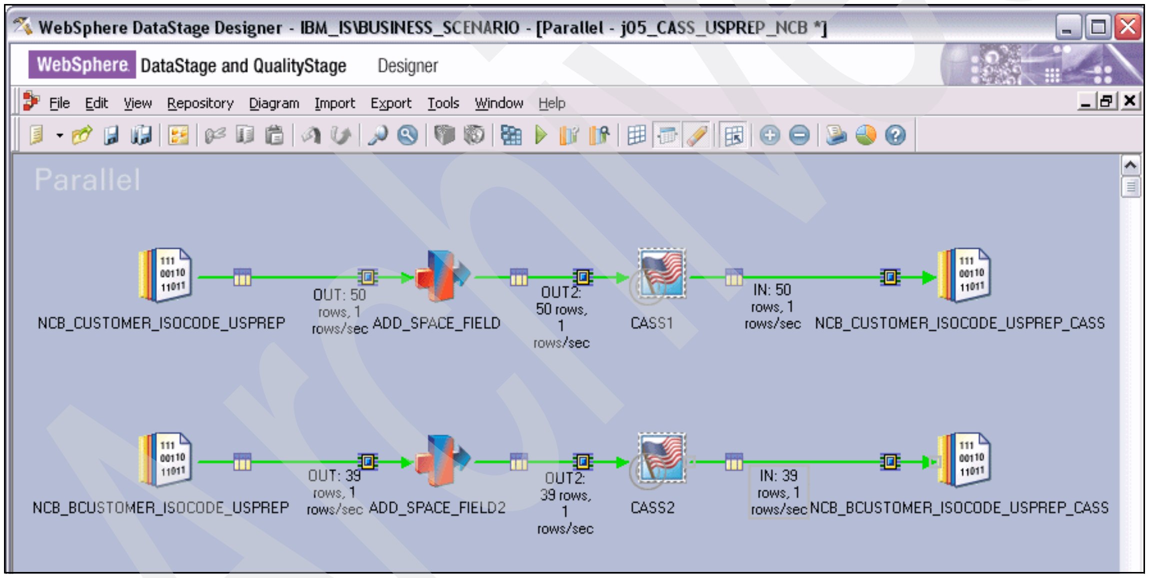
- Input Address Columns
  - Address Line 1 = AddressDomain_USPREP
  - Address Line 2 = Space_Field
  - City/State/Zip combined = Yes
    - City/State/Zip column = AreaDomain_USPREP
- Form 3553
  - Company Name = #QS_Residency2.CASS_Company_Name#
  - List Identifier = #QS_Residency2.CASS_List_ID#
  - Output file = #QS_Residency2.CASS_Report_Path#NCB_CUSTOM
- Reference Database
  - Path = #QS_Residency2.CASS_Path#

Output file: #QS_Residency2.CASS_Report_Path#NCB_CUSTOMER_CASS.rpt

Information:
Type: Pathname
Full pathname of the output file to which Form 3553 is written

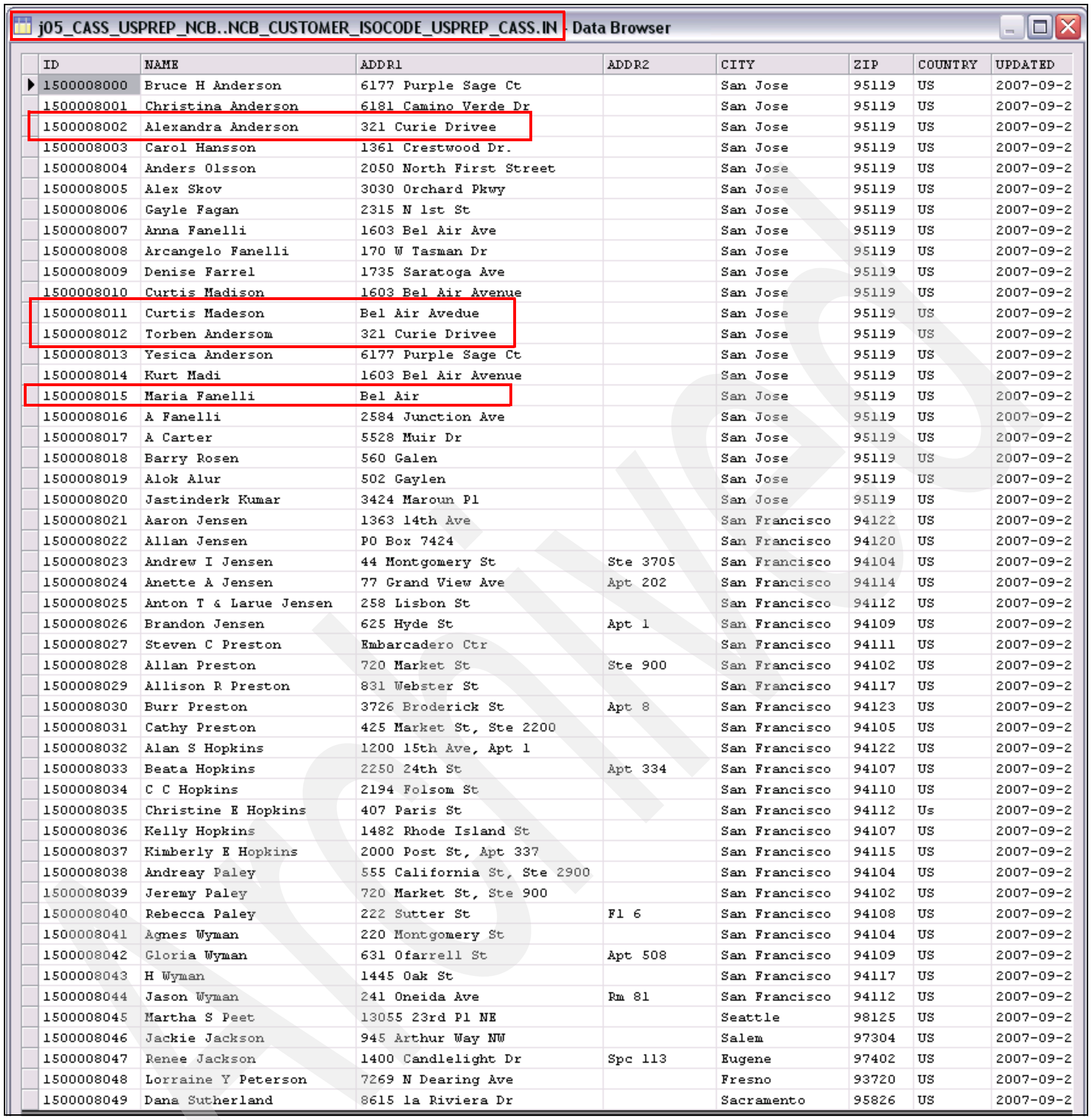
Available properties to add:

Figure 2-275 Create j05_CASS_USPREP_NCB 2/11



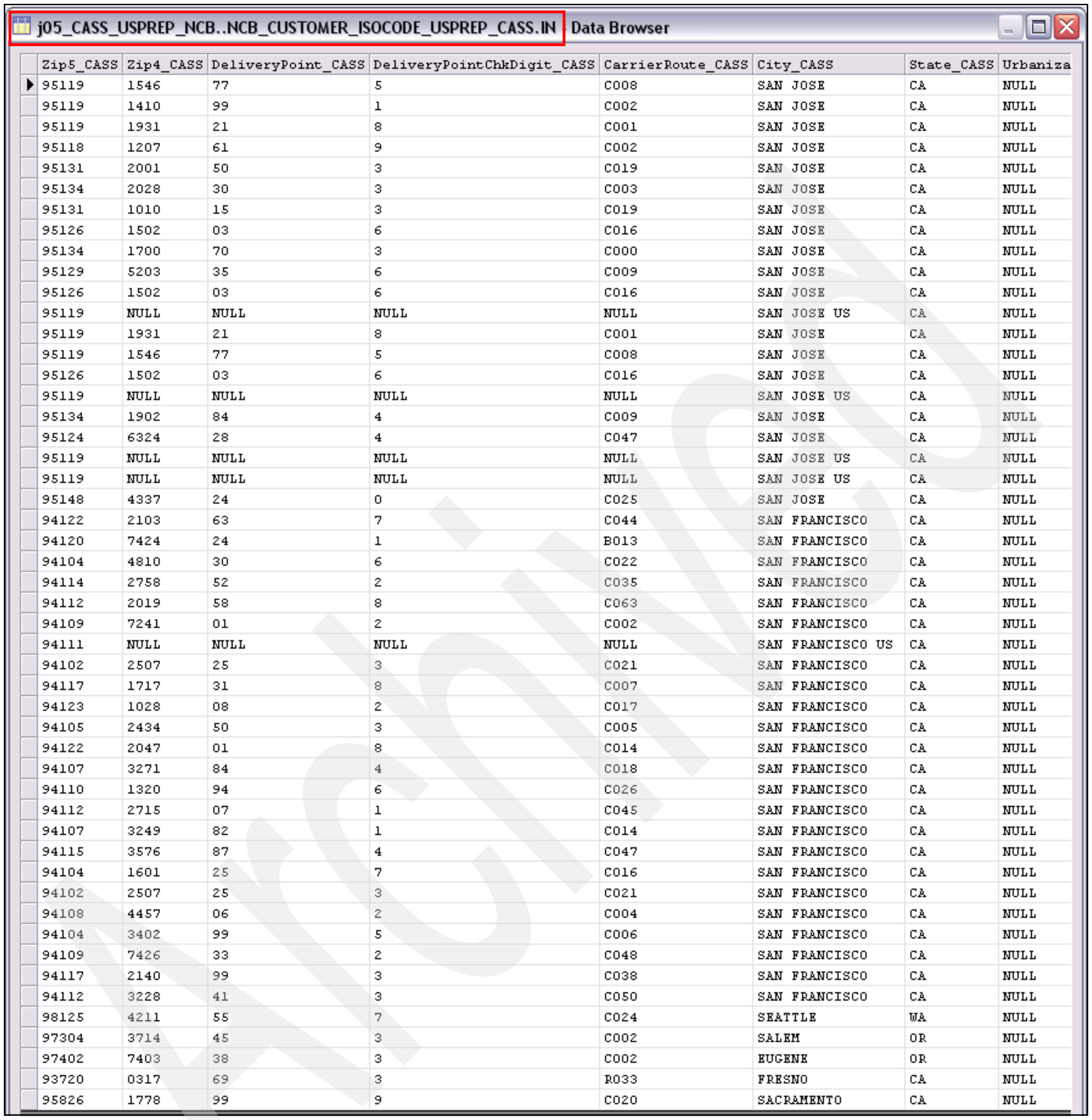
WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j05_CASS_USPREP_NCB *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NCB_CUSTOMER_ISOCODE_USPREP
OUT: 50 rows, 1 rows/sec
ADD_SPACE_FIELD
OUT2: 50 rows, 1 rows/sec
CASS1
IN: 50 rows, 1 rows/sec
NCB_CUSTOMER_ISOCODE_USPREP_CASS
NCB_BCUSTOMER_ISOCODE_USPREP
OUT: 39 rows, 1 rows/sec
ADD_SPACE_FIELD2
OUT2: 39 rows, 1 rows/sec
CASS2
IN: 39 rows, 1 rows/sec
NCB_BCUSTOMER_ISOCODE_USPREP_CASS

Figure 2-276 Create j05_CASS_USPREP_NCB 3/11


j05_CASS_USPREP_NCB..NCB_CUSTOMER_ISOCODE_USPREP_CASS.IN - Data Browser

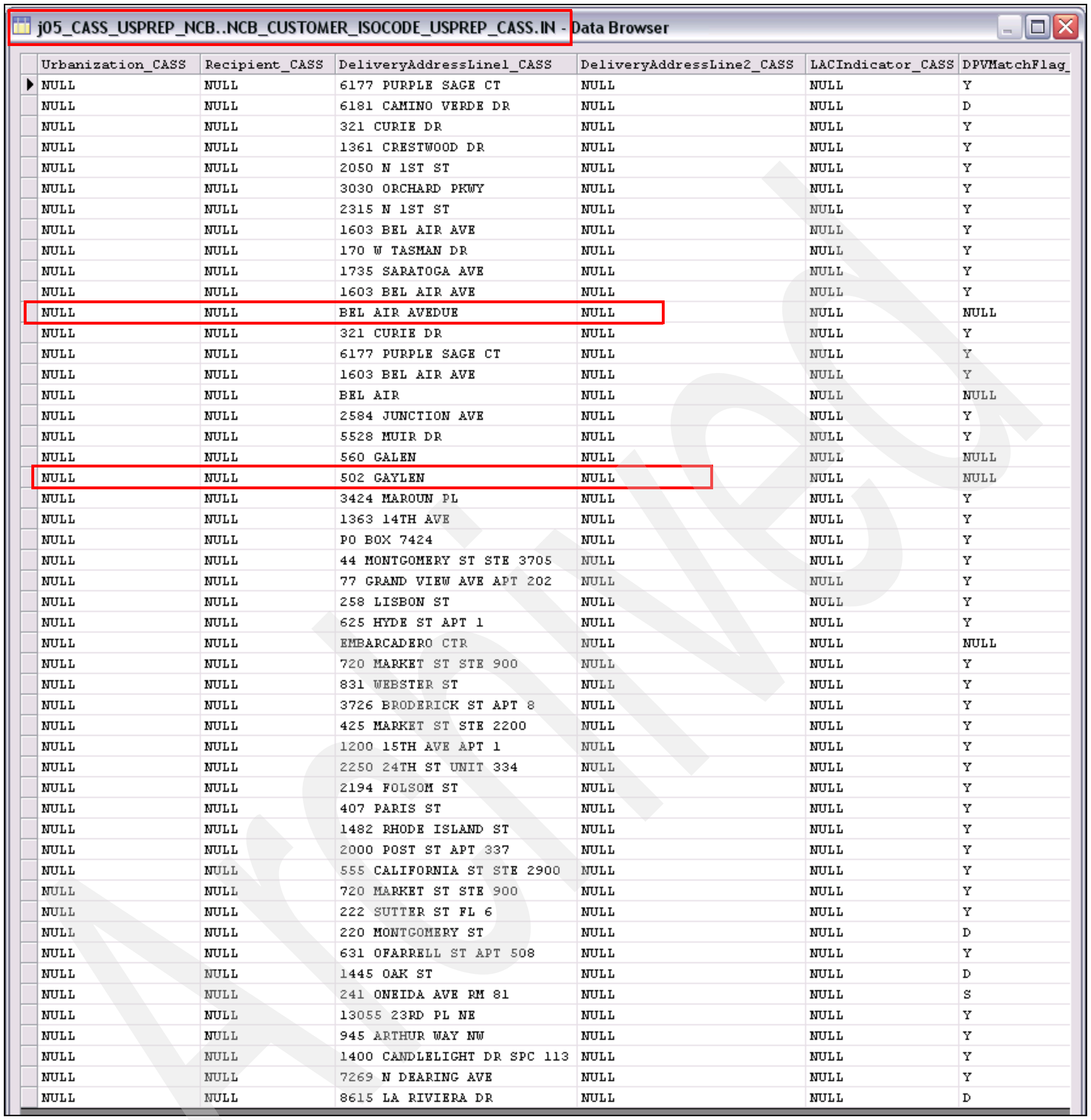
ID	NAME	ADDR1	ADDR2	CITY	ZIP	COUNTRY	UPDATED
1500008000	Bruce H Anderson	6177 Purple Sage Ct		San Jose	95119	US	2007-09-2
1500008001	Christina Anderson	6181 Camino Verde Dr		San Jose	95119	US	2007-09-2
1500008002	Alexandra Anderson	321 Curie Drivee		San Jose	95119	US	2007-09-2
1500008003	Carol Hansson	1361 Crestwood Dr.		San Jose	95119	US	2007-09-2
1500008004	Anders Olsson	2050 North First Street		San Jose	95119	US	2007-09-2
1500008005	Alex Skov	3030 Orchard Pkwy		San Jose	95119	US	2007-09-2
1500008006	Gayle Fagan	2315 N 1st St		San Jose	95119	US	2007-09-2
1500008007	Anna Fanelli	1603 Bel Air Ave		San Jose	95119	US	2007-09-2
1500008008	Arcangelo Fanelli	170 W Tasman Dr		San Jose	95119	US	2007-09-2
1500008009	Denise Farrel	1735 Saratoga Ave		San Jose	95119	US	2007-09-2
1500008010	Curtis Madison	1603 Bel Air Avenue		San Jose	95119	US	2007-09-2
1500008011	Curtis Madeson	Bel Air Avedue		San Jose	95119	US	2007-09-2
1500008012	Torben Andersom	321 Curie Drivee		San Jose	95119	US	2007-09-2
1500008013	Yesica Anderson	6177 Purple Sage Ct		San Jose	95119	US	2007-09-2
1500008014	Kurt Madi	1603 Bel Air Avenue		San Jose	95119	US	2007-09-2
1500008015	Maria Fanelli	Bel Air		San Jose	95119	US	2007-09-2
1500008016	A Fanelli	2584 Junction Ave		San Jose	95119	US	2007-09-2
1500008017	A Carter	5528 Muir Dr		San Jose	95119	US	2007-09-2
1500008018	Barry Rosen	560 Galen		San Jose	95119	US	2007-09-2
1500008019	Alok Alur	502 Gaylen		San Jose	95119	US	2007-09-2
1500008020	Jastinderk Kumar	3424 Maroun Pl		San Jose	95119	US	2007-09-2
1500008021	Aaron Jensen	1363 14th Ave		San Francisco	94122	US	2007-09-2
1500008022	Allan Jensen	PO Box 7424		San Francisco	94120	US	2007-09-2
1500008023	Andrew I Jensen	44 Montgomery St	Ste 3705	San Francisco	94104	US	2007-09-2
1500008024	Anette A Jensen	77 Grand View Ave	Apt 202	San Francisco	94114	US	2007-09-2
1500008025	Anton T & Larue Jensen	258 Lisbon St		San Francisco	94112	US	2007-09-2
1500008026	Brandon Jensen	625 Hyde St	Apt 1	San Francisco	94109	US	2007-09-2
1500008027	Steven C Preston	Embarcadero Ctr		San Francisco	94111	US	2007-09-2
1500008028	Allan Preston	720 Market St	Ste 900	San Francisco	94102	US	2007-09-2
1500008029	Allison R Preston	831 Webster St		San Francisco	94117	US	2007-09-2
1500008030	Burr Preston	3726 Broderick St	Apt 8	San Francisco	94123	US	2007-09-2
1500008031	Cathy Preston	425 Market St, Ste 2200		San Francisco	94105	US	2007-09-2
1500008032	Alan S Hopkins	1200 15th Ave, Apt 1		San Francisco	94122	US	2007-09-2
1500008033	Beata Hopkins	2250 24th St	Apt 334	San Francisco	94107	US	2007-09-2
1500008034	C C Hopkins	2194 Folsom St		San Francisco	94110	US	2007-09-2
1500008035	Christine E Hopkins	407 Paris St		San Francisco	94112	Us	2007-09-2
1500008036	Kelly Hopkins	1482 Rhode Island St		San Francisco	94107	US	2007-09-2
1500008037	Kimberly E Hopkins	2000 Post St, Apt 337		San Francisco	94115	US	2007-09-2
1500008038	Andreay Paley	555 California St, Ste 2900		San Francisco	94104	US	2007-09-2
1500008039	Jeremy Paley	720 Market St, Ste 900		San Francisco	94102	US	2007-09-2
1500008040	Rebecca Paley	222 Sutter St	Fl 6	San Francisco	94108	US	2007-09-2
1500008041	Agnes Wyman	220 Montgomery St		San Francisco	94104	US	2007-09-2
1500008042	Gloria Wyman	631 Ofarrell St	Apt 508	San Francisco	94109	US	2007-09-2
1500008043	H Wyman	1445 Oak St		San Francisco	94117	US	2007-09-2
1500008044	Jason Wyman	241 Oneida Ave	Rm 81	San Francisco	94112	US	2007-09-2
1500008045	Martha S Peet	13055 23rd Pl NE		Seattle	98125	US	2007-09-2
1500008046	Jackie Jackson	945 Arthur Way NW		Salem	97304	US	2007-09-2
1500008047	Renee Jackson	1400 Candlelight Dr	Spc 113	Eugene	97402	US	2007-09-2
1500008048	Lorraine Y Peterson	7269 N Dearing Ave		Fresno	93720	US	2007-09-2
1500008049	Dana Sutherland	8615 la Riviera Dr		Sacramento	95826	US	2007-09-2

Figure 2-277 Create j05_CASS_USPREP_NCB 4/11


j05_CASS_USPREP_NCB..NCB_CUSTOMER_ISOCODE_USPREP_CASS.IN - Data Browser

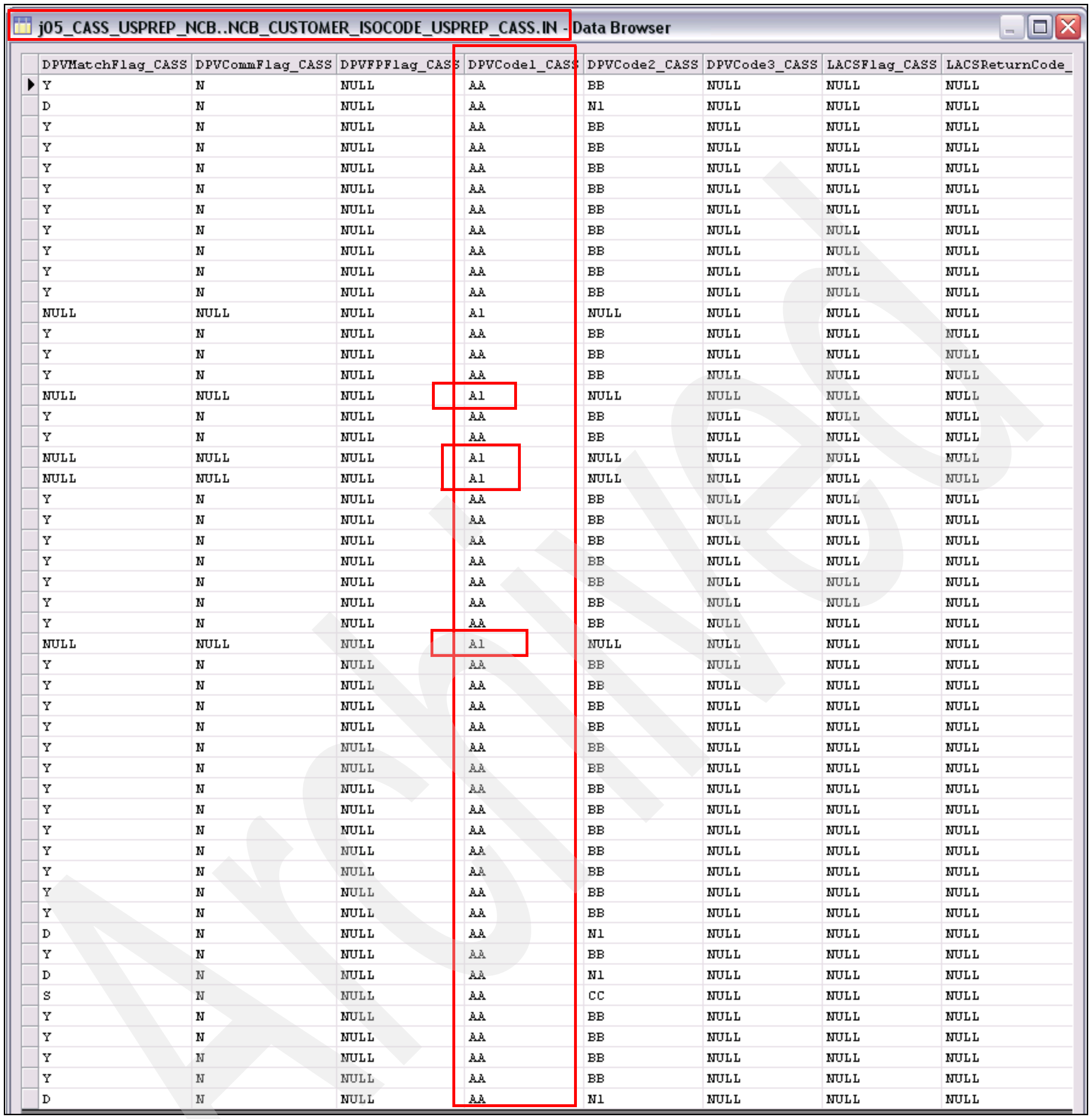
Zip5_CASS	Zip4_CASS	DeliveryPoint_CASS	DeliveryPointChkDigit_CASS	CarrierRoute_CASS	City_CASS	State_CASS	Urbaniza
95119	1546	77	5	C008	SAN JOSE	CA	NULL
95119	1410	99	1	C002	SAN JOSE	CA	NULL
95119	1931	21	8	C001	SAN JOSE	CA	NULL
95118	1207	61	9	C002	SAN JOSE	CA	NULL
95131	2001	50	3	C019	SAN JOSE	CA	NULL
95134	2028	30	3	C003	SAN JOSE	CA	NULL
95131	1010	15	3	C019	SAN JOSE	CA	NULL
95126	1502	03	6	C016	SAN JOSE	CA	NULL
95134	1700	70	3	C000	SAN JOSE	CA	NULL
95129	5203	35	6	C009	SAN JOSE	CA	NULL
95126	1502	03	6	C016	SAN JOSE	CA	NULL
95119	NULL	NULL	NULL	NULL	SAN JOSE US	CA	NULL
95119	1931	21	8	C001	SAN JOSE	CA	NULL
95119	1546	77	5	C008	SAN JOSE	CA	NULL
95126	1502	03	6	C016	SAN JOSE	CA	NULL
95119	NULL	NULL	NULL	NULL	SAN JOSE US	CA	NULL
95134	1902	84	4	C009	SAN JOSE	CA	NULL
95124	6324	28	4	C047	SAN JOSE	CA	NULL
95119	NULL	NULL	NULL	NULL	SAN JOSE US	CA	NULL
95119	NULL	NULL	NULL	NULL	SAN JOSE US	CA	NULL
95148	4337	24	0	C025	SAN JOSE	CA	NULL
94122	2103	63	7	C044	SAN FRANCISCO	CA	NULL
94120	7424	24	1	B013	SAN FRANCISCO	CA	NULL
94104	4810	30	6	C022	SAN FRANCISCO	CA	NULL
94114	2758	52	2	C035	SAN FRANCISCO	CA	NULL
94112	2019	58	8	C063	SAN FRANCISCO	CA	NULL
94109	7241	01	2	C002	SAN FRANCISCO	CA	NULL
94111	NULL	NULL	NULL	NULL	SAN FRANCISCO US	CA	NULL
94102	2507	25	3	C021	SAN FRANCISCO	CA	NULL
94117	1717	31	8	C007	SAN FRANCISCO	CA	NULL
94123	1028	08	2	C017	SAN FRANCISCO	CA	NULL
94105	2434	50	3	C005	SAN FRANCISCO	CA	NULL
94122	2047	01	8	C014	SAN FRANCISCO	CA	NULL
94107	3271	84	4	C018	SAN FRANCISCO	CA	NULL
94110	1320	94	6	C026	SAN FRANCISCO	CA	NULL
94112	2715	07	1	C045	SAN FRANCISCO	CA	NULL
94107	3249	82	1	C014	SAN FRANCISCO	CA	NULL
94115	3576	87	4	C047	SAN FRANCISCO	CA	NULL
94104	1601	25	7	C016	SAN FRANCISCO	CA	NULL
94102	2507	25	3	C021	SAN FRANCISCO	CA	NULL
94108	4457	06	2	C004	SAN FRANCISCO	CA	NULL
94104	3402	99	5	C006	SAN FRANCISCO	CA	NULL
94109	7426	33	2	C048	SAN FRANCISCO	CA	NULL
94117	2140	99	3	C038	SAN FRANCISCO	CA	NULL
94112	3228	41	3	C050	SAN FRANCISCO	CA	NULL
98125	4211	55	7	C024	SEATTLE	WA	NULL
97304	3714	45	3	C002	SALEM	OR	NULL
97402	7403	38	3	C002	EUGENE	OR	NULL
93720	0317	69	3	R033	FRESNO	CA	NULL
95826	1778	99	9	C020	SACRAMENTO	CA	NULL

Figure 2-278 Create j05_CASS_USPREP_NCB 5/11


j05_CASS_USPREP_NCB..NCB_CUSTOMER_ISOCODE_USPREP_CASS.IN - Data Browser

Urbanization_CASS	Recipient_CASS	DeliveryAddressLine1_CASS	DeliveryAddressLine2_CASS	LACIndicator_CASS	DPVMatchFlag_
NULL	NULL	6177 PURPLE SAGE CT	NULL	NULL	Y
NULL	NULL	6181 CAMINO VERDE DR	NULL	NULL	D
NULL	NULL	321 CURIE DR	NULL	NULL	Y
NULL	NULL	1361 CRESTWOOD DR	NULL	NULL	Y
NULL	NULL	2050 N 1ST ST	NULL	NULL	Y
NULL	NULL	3030 ORCHARD PKWY	NULL	NULL	Y
NULL	NULL	2315 N 1ST ST	NULL	NULL	Y
NULL	NULL	1603 BEL AIR AVE	NULL	NULL	Y
NULL	NULL	170 W TASMAN DR	NULL	NULL	Y
NULL	NULL	1735 SARATOGA AVE	NULL	NULL	Y
NULL	NULL	1603 BEL AIR AVE	NULL	NULL	Y
NULL	NULL	BEL AIR AVEDUE	NULL	NULL	NULL
NULL	NULL	321 CURIE DR	NULL	NULL	Y
NULL	NULL	6177 PURPLE SAGE CT	NULL	NULL	Y
NULL	NULL	1603 BEL AIR AVE	NULL	NULL	Y
NULL	NULL	BEL AIR	NULL	NULL	NULL
NULL	NULL	2584 JUNCTION AVE	NULL	NULL	Y
NULL	NULL	5528 MUIR DR	NULL	NULL	Y
NULL	NULL	560 GALEN	NULL	NULL	NULL
NULL	NULL	502 GAYLEN	NULL	NULL	NULL
NULL	NULL	3424 MAROUN PL	NULL	NULL	Y
NULL	NULL	1363 14TH AVE	NULL	NULL	Y
NULL	NULL	PO BOX 7424	NULL	NULL	Y
NULL	NULL	44 MONTGOMERY ST STE 3705	NULL	NULL	Y
NULL	NULL	77 GRAND VIEW AVE APT 202	NULL	NULL	Y
NULL	NULL	258 LISBON ST	NULL	NULL	Y
NULL	NULL	625 HYDE ST APT 1	NULL	NULL	Y
NULL	NULL	EMBARCADERO CTR	NULL	NULL	NULL
NULL	NULL	720 MARKET ST STE 900	NULL	NULL	Y
NULL	NULL	831 WEBSTER ST	NULL	NULL	Y
NULL	NULL	3726 BRODERICK ST APT 8	NULL	NULL	Y
NULL	NULL	425 MARKET ST STE 2200	NULL	NULL	Y
NULL	NULL	1200 15TH AVE APT 1	NULL	NULL	Y
NULL	NULL	2250 24TH ST UNIT 334	NULL	NULL	Y
NULL	NULL	2194 FOLSOM ST	NULL	NULL	Y
NULL	NULL	407 PARIS ST	NULL	NULL	Y
NULL	NULL	1482 RHODE ISLAND ST	NULL	NULL	Y
NULL	NULL	2000 POST ST APT 337	NULL	NULL	Y
NULL	NULL	555 CALIFORNIA ST STE 2900	NULL	NULL	Y
NULL	NULL	720 MARKET ST STE 900	NULL	NULL	Y
NULL	NULL	222 SUTTER ST FL 6	NULL	NULL	Y
NULL	NULL	220 MONTGOMERY ST	NULL	NULL	D
NULL	NULL	631 OFARRELL ST APT 508	NULL	NULL	Y
NULL	NULL	1445 OAK ST	NULL	NULL	D
NULL	NULL	241 ONEIDA AVE RM 81	NULL	NULL	S
NULL	NULL	13055 23RD PL NE	NULL	NULL	Y
NULL	NULL	945 ARTHUR WAY NW	NULL	NULL	Y
NULL	NULL	1400 CANDLELIGHT DR SPC 113	NULL	NULL	Y
NULL	NULL	7269 N DEARING AVE	NULL	NULL	Y
NULL	NULL	8615 LA RIVIERA DR	NULL	NULL	D

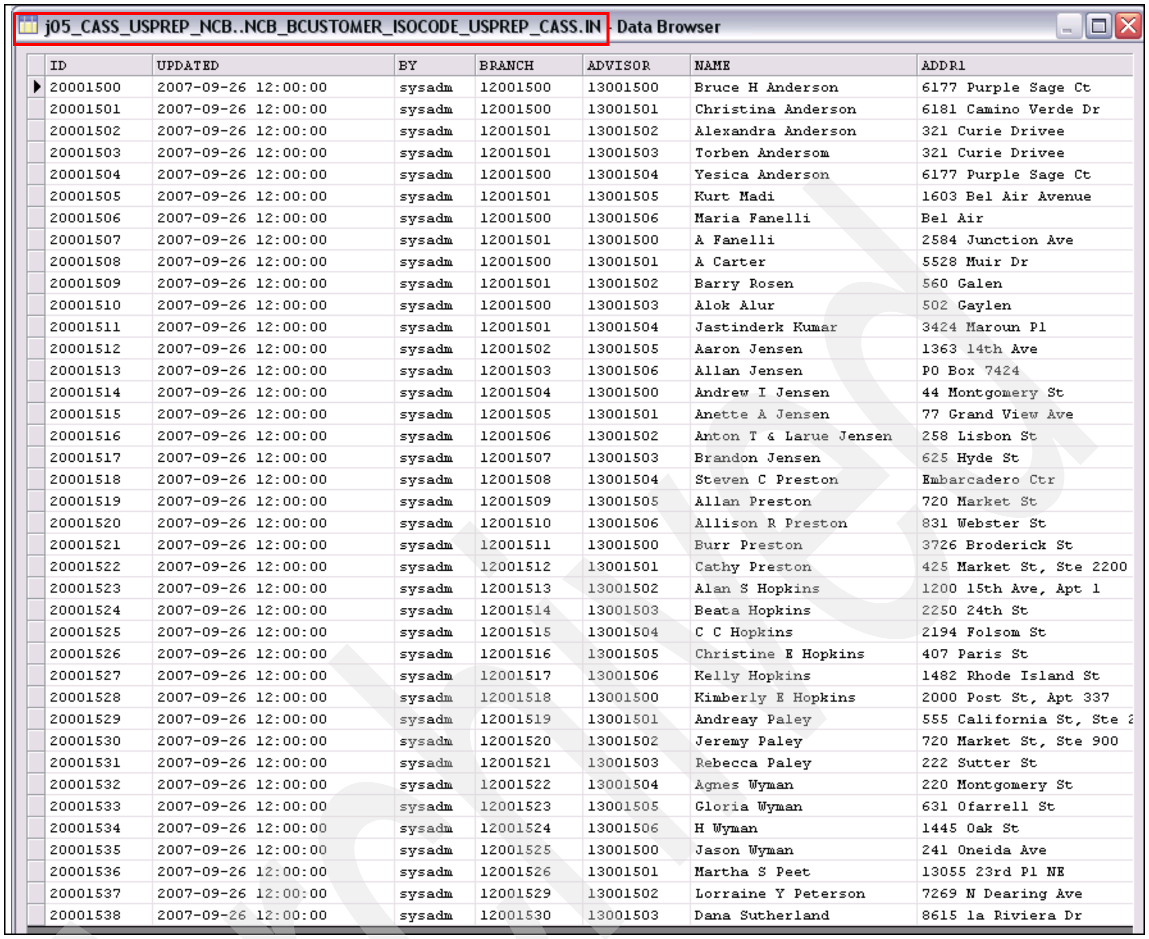
Figure 2-279   Create j05_CASS_USPREP_NCB 6/11



j05_CASS_USPREP_NCB..NCB_CUSTOMER_ISOCODE_USPREP_CASS.IN - Data Browser

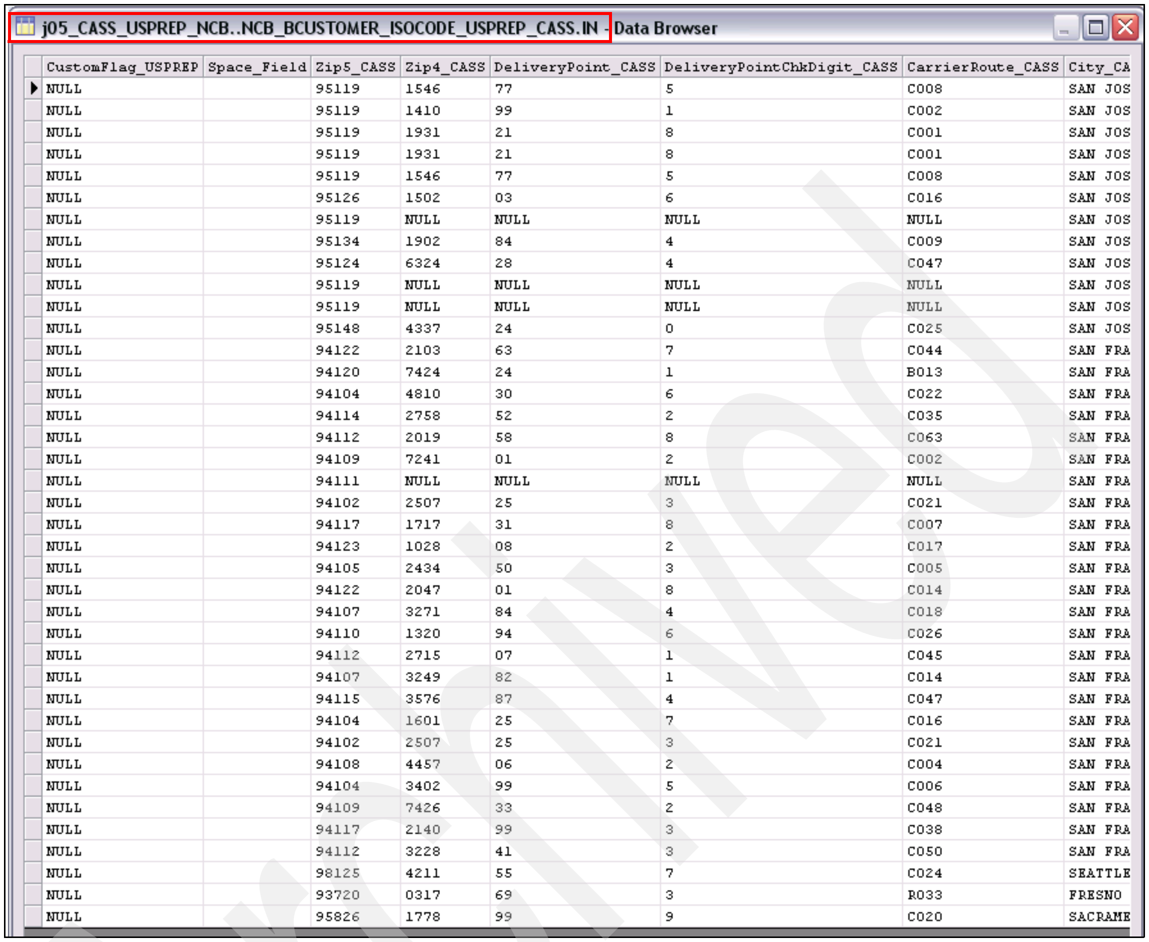
DPVMatchFlag_CASS	DPVCommFlag_CASS	DPVFPFlag_CASS	DPVCode1_CASS	DPVCode2_CASS	DPVCode3_CASS	LACSFlag_CASS	LACSReturnCode_
Y	N	NULL	AA	BB	NULL	NULL	NULL
D	N	NULL	AA	N1	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
NULL	NULL	NULL	A1	NULL	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
NULL	NULL	NULL	A1	NULL	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
NULL	NULL	NULL	A1	NULL	NULL	NULL	NULL
NULL	NULL	NULL	A1	NULL	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
NULL	NULL	NULL	A1	NULL	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
D	N	NULL	AA	N1	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
D	N	NULL	AA	N1	NULL	NULL	NULL
S	N	NULL	AA	CC	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
Y	N	NULL	AA	BB	NULL	NULL	NULL
D	N	NULL	AA	N1	NULL	NULL	NULL

Figure 2-280 Create j05_CASS_USPREP_NCB 7/11


j05_CASS_USPREP_NCB..NCB_BCUSTOMER_ISOCODE_USPREP_CASS.IN - Data Browser

ID	UPDATED	BY	BRANCH	ADVISOR	NAME	ADDR1
20001500	2007-09-26 12:00:00	sysadm	12001500	13001500	Bruce H Anderson	6177 Purple Sage Ct
20001501	2007-09-26 12:00:00	sysadm	12001500	13001501	Christina Anderson	6181 Camino Verde Dr
20001502	2007-09-26 12:00:00	sysadm	12001501	13001502	Alexandra Anderson	321 Curie Drivee
20001503	2007-09-26 12:00:00	sysadm	12001501	13001503	Torben Andersom	321 Curie Drivee
20001504	2007-09-26 12:00:00	sysadm	12001500	13001504	Yesica Anderson	6177 Purple Sage Ct
20001505	2007-09-26 12:00:00	sysadm	12001501	13001505	Kurt Madi	1603 Bel Air Avenue
20001506	2007-09-26 12:00:00	sysadm	12001500	13001506	Maria Fanelli	Bel Air
20001507	2007-09-26 12:00:00	sysadm	12001501	13001500	A Fanelli	2584 Junction Ave
20001508	2007-09-26 12:00:00	sysadm	12001500	13001501	A Carter	5528 Muir Dr
20001509	2007-09-26 12:00:00	sysadm	12001501	13001502	Barry Rosen	560 Galen
20001510	2007-09-26 12:00:00	sysadm	12001500	13001503	Alok Alur	502 Gaylen
20001511	2007-09-26 12:00:00	sysadm	12001501	13001504	Jastinderk Kumar	3424 Maroun Pl
20001512	2007-09-26 12:00:00	sysadm	12001502	13001505	Aaron Jensen	1363 14th Ave
20001513	2007-09-26 12:00:00	sysadm	12001503	13001506	Allan Jensen	PO Box 7424
20001514	2007-09-26 12:00:00	sysadm	12001504	13001500	Andrew I Jensen	44 Montgomery St
20001515	2007-09-26 12:00:00	sysadm	12001505	13001501	Anette A Jensen	77 Grand View Ave
20001516	2007-09-26 12:00:00	sysadm	12001506	13001502	Anton T & Larue Jensen	258 Lisbon St
20001517	2007-09-26 12:00:00	sysadm	12001507	13001503	Brandon Jensen	625 Hyde St
20001518	2007-09-26 12:00:00	sysadm	12001508	13001504	Steven C Preston	Embarcadero Ctr
20001519	2007-09-26 12:00:00	sysadm	12001509	13001505	Allan Preston	720 Market St
20001520	2007-09-26 12:00:00	sysadm	12001510	13001506	Allison R Preston	831 Webster St
20001521	2007-09-26 12:00:00	sysadm	12001511	13001500	Burr Preston	3726 Broderick St
20001522	2007-09-26 12:00:00	sysadm	12001512	13001501	Cathy Preston	425 Market St, Ste 2200
20001523	2007-09-26 12:00:00	sysadm	12001513	13001502	Alan S Hopkins	1200 15th Ave, Apt 1
20001524	2007-09-26 12:00:00	sysadm	12001514	13001503	Beata Hopkins	2250 24th St
20001525	2007-09-26 12:00:00	sysadm	12001515	13001504	C C Hopkins	2194 Folsom St
20001526	2007-09-26 12:00:00	sysadm	12001516	13001505	Christine E Hopkins	407 Paris St
20001527	2007-09-26 12:00:00	sysadm	12001517	13001506	Kelly Hopkins	1482 Rhode Island St
20001528	2007-09-26 12:00:00	sysadm	12001518	13001500	Kimberly E Hopkins	2000 Post St, Apt 337
20001529	2007-09-26 12:00:00	sysadm	12001519	13001501	Andreay Paley	555 California St, Ste 2
20001530	2007-09-26 12:00:00	sysadm	12001520	13001502	Jeremy Paley	720 Market St, Ste 900
20001531	2007-09-26 12:00:00	sysadm	12001521	13001503	Rebecca Paley	222 Sutter St
20001532	2007-09-26 12:00:00	sysadm	12001522	13001504	Agnes Wyman	220 Montgomery St
20001533	2007-09-26 12:00:00	sysadm	12001523	13001505	Gloria Wyman	631 Ofarrell St
20001534	2007-09-26 12:00:00	sysadm	12001524	13001506	H Wyman	1445 Oak St
20001535	2007-09-26 12:00:00	sysadm	12001525	13001500	Jason Wyman	241 Oneida Ave
20001536	2007-09-26 12:00:00	sysadm	12001526	13001501	Martha S Peet	13055 23rd Pl NE
20001537	2007-09-26 12:00:00	sysadm	12001529	13001502	Lorraine Y Peterson	7269 N Dearing Ave
20001538	2007-09-26 12:00:00	sysadm	12001530	13001503	Dana Sutherland	8615 la Riviera Dr

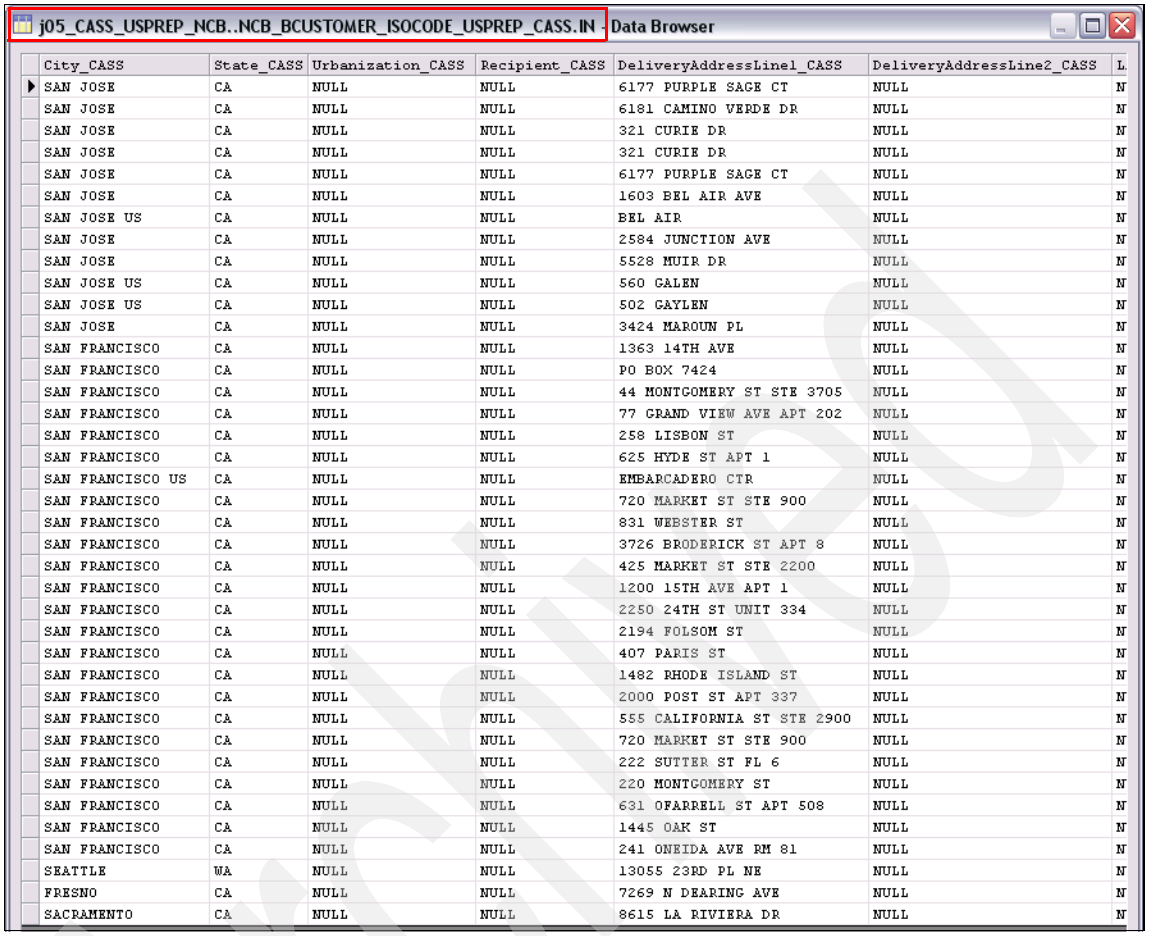
Figure 2-281 Create j05_CASS_USPREP_NCB 8/11



j05_CASS_USPREP_NCB..NCB_BCUSTOMER_ISOCODE_USPREP_CASS.IN - Data Browser

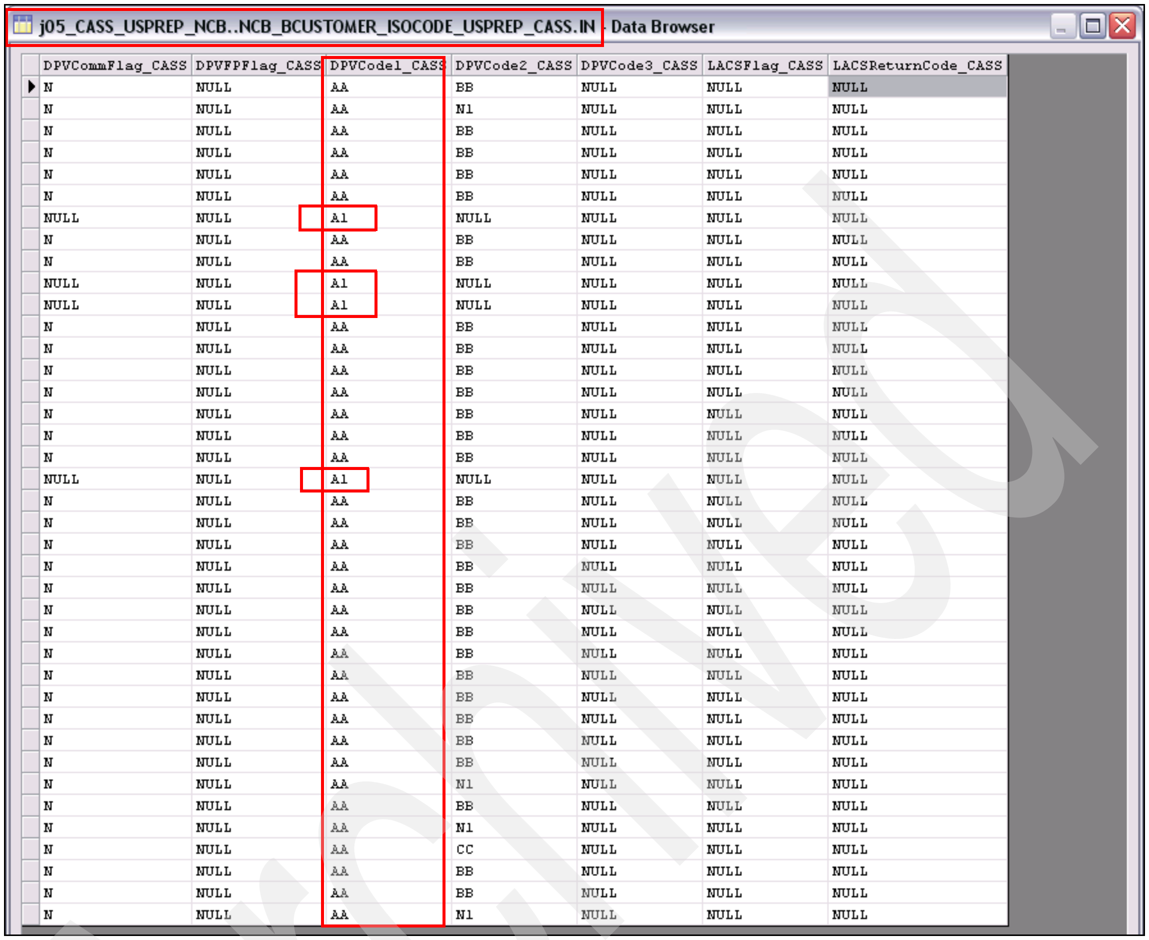
CustomFlag_USPREP	Space_Field	Zip5_CASS	Zip4_CASS	DeliveryPoint_CASS	DeliveryPointChkDigit_CASS	CarrierRoute_CASS	City_CA
NULL		95119	1546	77	5	C008	SAN JOS
NULL		95119	1410	99	1	C002	SAN JOS
NULL		95119	1931	21	8	C001	SAN JOS
NULL		95119	1931	21	8	C001	SAN JOS
NULL		95119	1546	77	5	C008	SAN JOS
NULL		95126	1502	03	6	C016	SAN JOS
NULL		95119	NULL	NULL	NULL	NULL	SAN JOS
NULL		95134	1902	84	4	C009	SAN JOS
NULL		95124	6324	28	4	C047	SAN JOS
NULL		95119	NULL	NULL	NULL	NULL	SAN JOS
NULL		95119	NULL	NULL	NULL	NULL	SAN JOS
NULL		95148	4337	24	0	C025	SAN JOS
NULL		94122	2103	63	7	C044	SAN FRA
NULL		94120	7424	24	1	B013	SAN FRA
NULL		94104	4810	30	6	C022	SAN FRA
NULL		94114	2758	52	2	C035	SAN FRA
NULL		94112	2019	58	8	C063	SAN FRA
NULL		94109	7241	01	2	C002	SAN FRA
NULL		94111	NULL	NULL	NULL	NULL	SAN FRA
NULL		94102	2507	25	3	C021	SAN FRA
NULL		94117	1717	31	8	C007	SAN FRA
NULL		94123	1028	08	2	C017	SAN FRA
NULL		94105	2434	50	3	C005	SAN FRA
NULL		94122	2047	01	8	C014	SAN FRA
NULL		94107	3271	84	4	C018	SAN FRA
NULL		94110	1320	94	6	C026	SAN FRA
NULL		94112	2715	07	1	C045	SAN FRA
NULL		94107	3249	82	1	C014	SAN FRA
NULL		94115	3576	87	4	C047	SAN FRA
NULL		94104	1601	25	7	C016	SAN FRA
NULL		94102	2507	25	3	C021	SAN FRA
NULL		94108	4457	06	2	C004	SAN FRA
NULL		94104	3402	99	5	C006	SAN FRA
NULL		94109	7426	33	2	C048	SAN FRA
NULL		94117	2140	99	3	C038	SAN FRA
NULL		94112	3228	41	3	C050	SAN FRA
NULL		98125	4211	55	7	C024	SEATTLE
NULL		93720	0317	69	3	R033	FRESNO
NULL		95826	1778	99	9	C020	SACRAME

Figure 2-282 Create j05_CASS_USPREP_NCB 9/11


j05_CASS_USPREP_NCB..NCB_BCUSTOMER_ISOCODE_USPREP_CASS.IN - Data Browser

City_CASS	State_CASS	Urbanization_CASS	Recipient_CASS	DeliveryAddressLine1_CASS	DeliveryAddressLine2_CASS	L
SAN JOSE	CA	NULL	NULL	6177 PURPLE SAGE CT	NULL	N
SAN JOSE	CA	NULL	NULL	6181 CAMINO VERDE DR	NULL	N
SAN JOSE	CA	NULL	NULL	321 CURIE DR	NULL	N
SAN JOSE	CA	NULL	NULL	321 CURIE DR	NULL	N
SAN JOSE	CA	NULL	NULL	6177 PURPLE SAGE CT	NULL	N
SAN JOSE	CA	NULL	NULL	1603 BEL AIR AVE	NULL	N
SAN JOSE US	CA	NULL	NULL	BEL AIR	NULL	N
SAN JOSE	CA	NULL	NULL	2584 JUNCTION AVE	NULL	N
SAN JOSE	CA	NULL	NULL	5528 MUIR DR	NULL	N
SAN JOSE US	CA	NULL	NULL	560 GALEN	NULL	N
SAN JOSE US	CA	NULL	NULL	502 GAYLEN	NULL	N
SAN JOSE	CA	NULL	NULL	3424 MAROUN PL	NULL	N
SAN FRANCISCO	CA	NULL	NULL	1363 14TH AVE	NULL	N
SAN FRANCISCO	CA	NULL	NULL	PO BOX 7424	NULL	N
SAN FRANCISCO	CA	NULL	NULL	44 MONTGOMERY ST STE 3705	NULL	N
SAN FRANCISCO	CA	NULL	NULL	77 GRAND VIEW AVE APT 202	NULL	N
SAN FRANCISCO	CA	NULL	NULL	258 LISBON ST	NULL	N
SAN FRANCISCO	CA	NULL	NULL	625 HYDE ST APT 1	NULL	N
SAN FRANCISCO US	CA	NULL	NULL	EMBARCADERO CTR	NULL	N
SAN FRANCISCO	CA	NULL	NULL	720 MARKET ST STE 900	NULL	N
SAN FRANCISCO	CA	NULL	NULL	831 WEBSTER ST	NULL	N
SAN FRANCISCO	CA	NULL	NULL	3726 BRODERICK ST APT 8	NULL	N
SAN FRANCISCO	CA	NULL	NULL	425 MARKET ST STE 2200	NULL	N
SAN FRANCISCO	CA	NULL	NULL	1200 15TH AVE APT 1	NULL	N
SAN FRANCISCO	CA	NULL	NULL	2250 24TH ST UNIT 334	NULL	N
SAN FRANCISCO	CA	NULL	NULL	2194 FOLSOM ST	NULL	N
SAN FRANCISCO	CA	NULL	NULL	407 PARIS ST	NULL	N
SAN FRANCISCO	CA	NULL	NULL	1482 RHODE ISLAND ST	NULL	N
SAN FRANCISCO	CA	NULL	NULL	2000 POST ST APT 337	NULL	N
SAN FRANCISCO	CA	NULL	NULL	555 CALIFORNIA ST STE 2900	NULL	N
SAN FRANCISCO	CA	NULL	NULL	720 MARKET ST STE 900	NULL	N
SAN FRANCISCO	CA	NULL	NULL	222 SUTTER ST FL 6	NULL	N
SAN FRANCISCO	CA	NULL	NULL	220 MONTGOMERY ST	NULL	N
SAN FRANCISCO	CA	NULL	NULL	631 OFARRELL ST APT 508	NULL	N
SAN FRANCISCO	CA	NULL	NULL	1445 OAK ST	NULL	N
SAN FRANCISCO	CA	NULL	NULL	241 ONEIDA AVE RM 81	NULL	N
SEATTLE	WA	NULL	NULL	13055 23RD PL NE	NULL	N
FRESNO	CA	NULL	NULL	7269 N DEARING AVE	NULL	N
SACRAMENTO	CA	NULL	NULL	8615 LA RIVIERA DR	NULL	N

Figure 2-283 Create j05_CASS_USPREP_NCB 10/11



j05_CASS_USPREP_NCB..NCB_BCUSTOMER_ISOCODE_USPREP_CASS.IN - Data Browser

DPVCommFlag_CASS	DPVFPFlag_CASS	DPVCode1_CASS	DPVCode2_CASS	DPVCode3_CASS	LACSFlag_CASS	LACSReturnCode_CASS
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	N1	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
NULL	NULL	A1	NULL	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
NULL	NULL	A1	NULL	NULL	NULL	NULL
NULL	NULL	A1	NULL	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
NULL	NULL	A1	NULL	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	N1	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	N1	NULL	NULL	NULL
N	NULL	AA	CC	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	BB	NULL	NULL	NULL
N	NULL	AA	N1	NULL	NULL	NULL

Figure 2-284 Create j05_CASS_USPREP_NCB 11/11

j06_INVCC_CASS_NCB

We then run the Investigate stage with character concatenate option on the results of job "j05_CASS_USPREP_NCB" on page 742 step to determine addresses that are not recognized by CASS (delivery point verification or DPV).

Note: As mentioned earlier, due to a bug with handling nulls, we introduced a Transformer stage to convert nulls to a space using column derivation.

We investigate using character concatenate (on CASS generated columns DPVMATCHFLAG_CASS and DPVCODE1_CASS) using a *C* mask and *X* masks on the DeliveryAddressLine1_CASS, City_CASS, State_CASS, and

Zip5_CASS columns. A value of `A1` in the DPVCODE1_CASS field indicates a potential problem.

Figure 2-285 shows the various stages that are used in this job. It includes the data sets that were created in “j05_CASS_USPREP_NCB” on page 742, a Transformer stage for handling nulls, an Investigate stage, and an output Sequential File stage for each input data set. We modified the names of the stages as shown.

Because the configuration of this job is very similar to that described in “J06_INVCC_CASS” on page 228, we do not repeat it here. However, some of the configurations of interest are as follows:

- Figure 2-286 on page 754 shows the configuration of the INV_DPV_INFO1 Investigate stage window for the CUSTOMER file with columns DPVMatchFlag_CASS and DPVCode1_CASS selected with the *C* masks for each character. The *X* masks are used for columns Delivery_AddressLine1_CASS,City_CASS, State_CASS, and Zip5_CASS.

  The configuration of the INV_DPV_INFO2 Investigate stage window for the BCUSTOMER file is similar, and we do not repeat it here.

- After saving, compiling, and running this job (Figure 2-287 on page 754), the output of the Investigate stages are shown in Figure 2-288 on page 755 and Figure 2-289 on page 755. It shows a a few records with values `A1` in the DPVMatchFlag_CASS and DPVCode1_CASS character concatenated columns, indicating a potential problem that required further investigation.

Proceed now to “j07_PREP_CASS_NCB” on page 755.

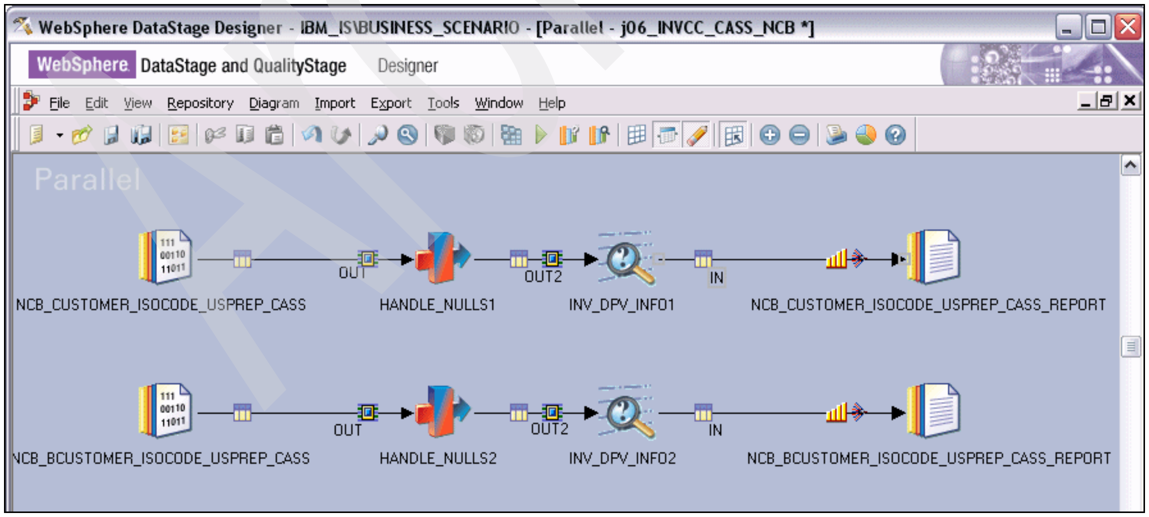
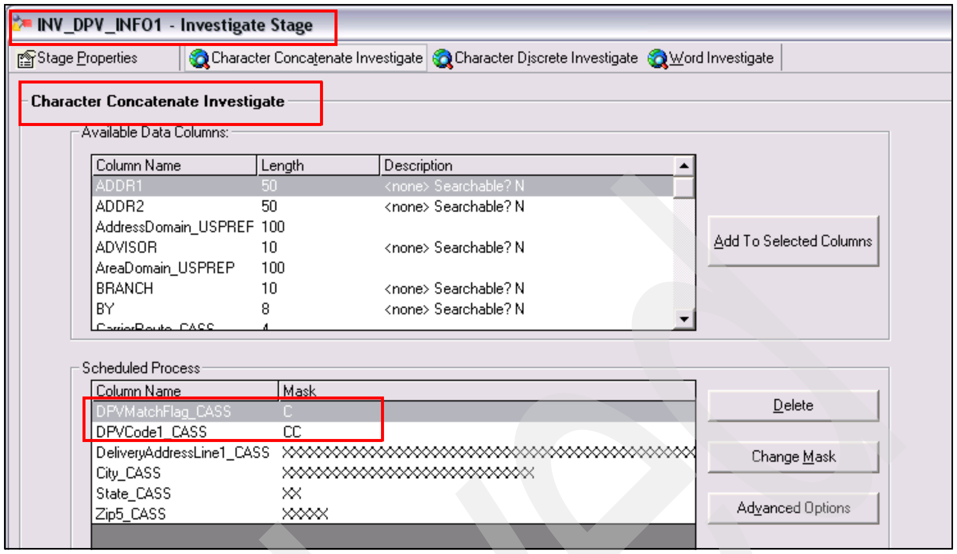

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j06_INVCC_CASS_NCB *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NCB_CUSTOMER_ISOCODE_USPREP_CASS
OUT
HANDLE_NULLS1
OUT2
INV_DPV_INFO1
IN
NCB_CUSTOMER_ISOCODE_USPREP_CASS_REPORT
NCB_BCUSTOMER_ISOCODE_USPREP_CASS
OUT
HANDLE_NULLS2
OUT2
INV_DPV_INFO2
IN
NCB_BCUSTOMER_ISOCODE_USPREP_CASS_REPORT

Figure 2-285   Create j06_INVCC_CASS_NCB 1/5



INV_DPV_INFO1 - Investigate Stage

Stage Properties | Character Concatenate Investigate | Character Discrete Investigate | Word Investigate

Character Concatenate Investigate

Available Data Columns:

Column Name	Length	Description
ADDR1	50	<none> Searchable? N
ADDR2	50	<none> Searchable? N
AddressDomain_USPREF	100	
ADVISOR	10	<none> Searchable? N
AreaDomain_USPREP	100	
BRANCH	10	<none> Searchable? N
BY	8	<none> Searchable? N
CarrierRoute_CASS	4	

Add To Selected Columns

Scheduled Process

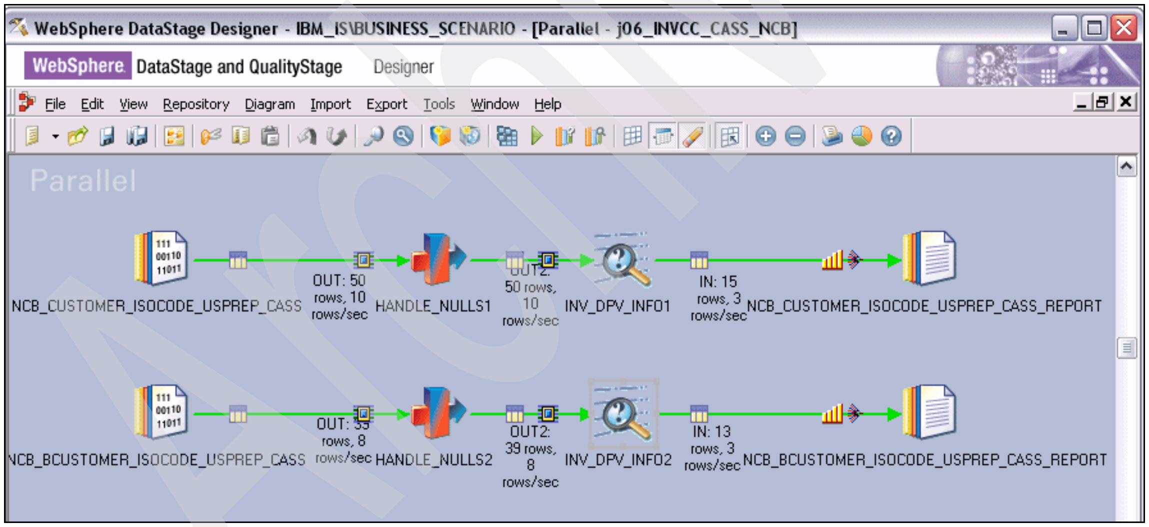
Column Name	Mask
DPVMatchFlag_CASS	C
DPVCode1_CASS	CC
DeliveryAddressLine1_CASS	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
City_CASS	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
State_CASS	XX
Zip5_CASS	XXXXX

Delete

Change Mask

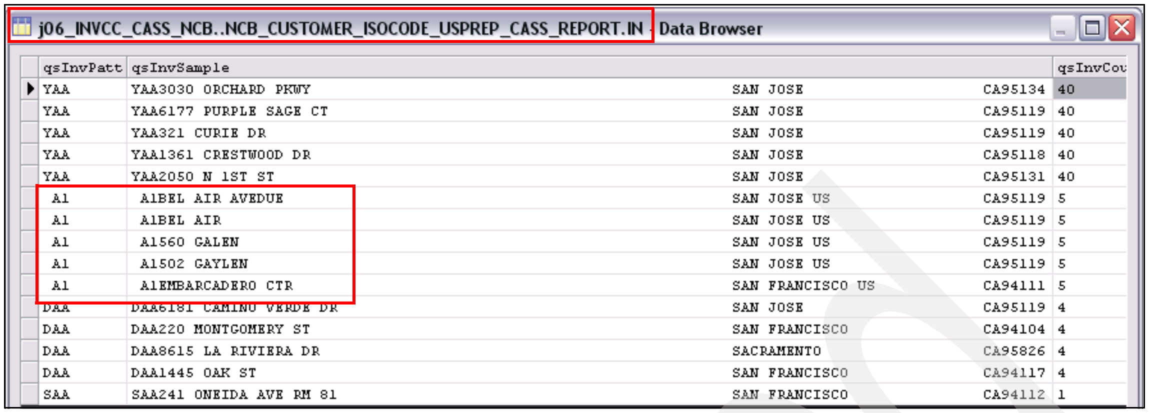
Advanced Options

Figure 2-286 Create j06_INVCC_CASS_NCB 2/5


WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j06_INVCC_CASS_NCB]

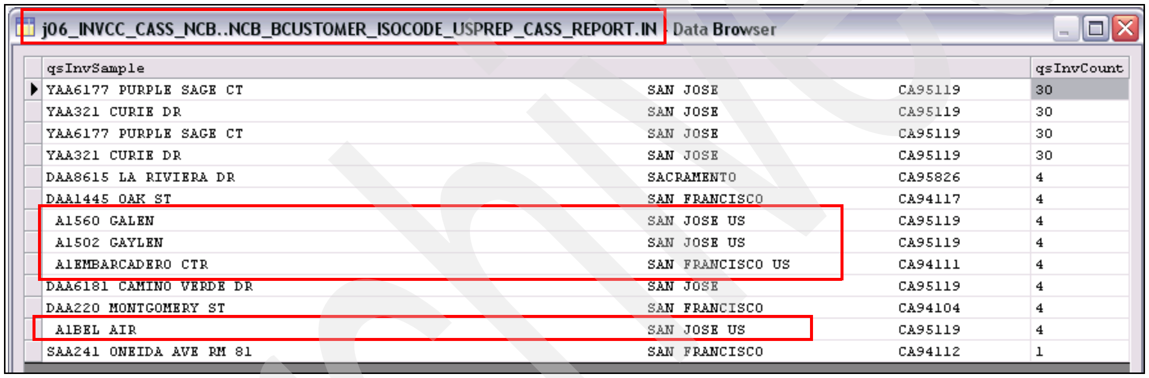
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NCB_CUSTOMER_ISOCODE_USPREP_CASS
OUT: 50 rows, 10 rows/sec
HANDLE_NULLS1
OUT2: 50 rows, 10 rows/sec
INV_DPV_INFO1
IN: 15 rows, 3 rows/sec
NCB_CUSTOMER_ISOCODE_USPREP_CASS_REPORT
NCB_BCUSTOMER_ISOCODE_USPREP_CASS
OUT: 39 rows, 8 rows/sec
HANDLE_NULLS2
OUT2: 39 rows, 8 rows/sec
INV_DPV_INFO2
IN: 13 rows, 3 rows/sec
NCB_BCUSTOMER_ISOCODE_USPREP_CASS_REPORT

Figure 2-287 Create j06_INVCC_CASS_NCB 3/5


j06_INVCC_CASS_NCB..NCB_CUSTOMER_ISOCODE_USPREP_CASS_REPORT.IN - Data Browser

qsInvPatt	qsInvSample			qsInvCou
YAA	YAA3030 ORCHARD PKWY	SAN JOSE	CA95134	40
YAA	YAA6177 PURPLE SAGE CT	SAN JOSE	CA95119	40
YAA	YAA321 CURIE DR	SAN JOSE	CA95119	40
YAA	YAA1361 CRESTWOOD DR	SAN JOSE	CA95118	40
YAA	YAA2050 N 1ST ST	SAN JOSE	CA95131	40
A1	A1BEL AIR AVEDUE	SAN JOSE US	CA95119	5
A1	A1BEL AIR	SAN JOSE US	CA95119	5
A1	A1560 GALEN	SAN JOSE US	CA95119	5
A1	A1502 GAYLEN	SAN JOSE US	CA95119	5
A1	A1EMBARCADERO CTR	SAN FRANCISCO US	CA94111	5
DAA	DAA6181 CAMINO VERDE DR	SAN JOSE	CA95119	4
DAA	DAA220 MONTGOMERY ST	SAN FRANCISCO	CA94104	4
DAA	DAA8615 LA RIVIERA DR	SACRAMENTO	CA95826	4
DAA	DAA1445 OAK ST	SAN FRANCISCO	CA94117	4
SAA	SAA241 ONEIDA AVE RM 81	SAN FRANCISCO	CA94112	1

Figure 2-288 Create j06_INVCC_CASS_NCB 4/5


j06_INVCC_CASS_NCB..NCB_BCUSTOMER_ISOCODE_USPREP_CASS_REPORT.IN - Data Browser

qsInvSample			qsInvCount
YAA6177 PURPLE SAGE CT	SAN JOSE	CA95119	30
YAA321 CURIE DR	SAN JOSE	CA95119	30
YAA6177 PURPLE SAGE CT	SAN JOSE	CA95119	30
YAA321 CURIE DR	SAN JOSE	CA95119	30
DAA8615 LA RIVIERA DR	SACRAMENTO	CA95826	4
DAA1445 OAK ST	SAN FRANCISCO	CA94117	4
A1560 GALEN	SAN JOSE US	CA95119	4
A1502 GAYLEN	SAN JOSE US	CA95119	4
A1EMBARCADERO CTR	SAN FRANCISCO US	CA94111	4
DAA6181 CAMINO VERDE DR	SAN JOSE	CA95119	4
DAA220 MONTGOMERY ST	SAN FRANCISCO	CA94104	4
A1BEL AIR	SAN JOSE US	CA95119	4
SAA241 ONEIDA AVE RM 81	SAN FRANCISCO	CA94112	1

Figure 2-289 Create j06_INVCC_CASS_NCB 5/5

j07_PREP_CASS_NCB

In this step, we standardize the address contents of the output of job "j05_CASS_USPREP_NCB" on page 742 using the domain-preprocessor USPREP rule set. This parsed and moved the address, as well as area content from selected input columns to the AddressDomain_USPREP and AreaDomain_USPREP columns. We also add a column to the output that only had the first three characters of the ZIP code using a Transformer stage. This new column (ZIP3) is used as a blocking variable in an upcoming matching stage.

After CASS validates and corrects the address fields, we run the Standardize stage again with the domain-preprocessor rule set USPREP on the CASS corrected address fields using the literals[7] ZQPUTRZQ and ZQPUTAZQ.

Figure 2-290 on page 757 shows the various stages that are used in this job. It includes the data sets that were created in “j05_CASS_USPREP_NCB” on page 742, a Standardize stage, a Transformer stage to add a column, and an output Data Set stage for each of the input data sets. We modified the names of the stages as shown.

Because the configuration of this job is very similar to that described in “J07_STAN_CUSTOMER_Domain_Preprocessor” on page 234, we do not repeat it here. However, some of the configurations of interest are as follows:

- After saving, compiling, and running this job (Figure 2-290 on page 757), you can view the content of output data set as shown in Figure 2-291 on page 758 through Figure 2-296 on page 763.

  This report shows the following:

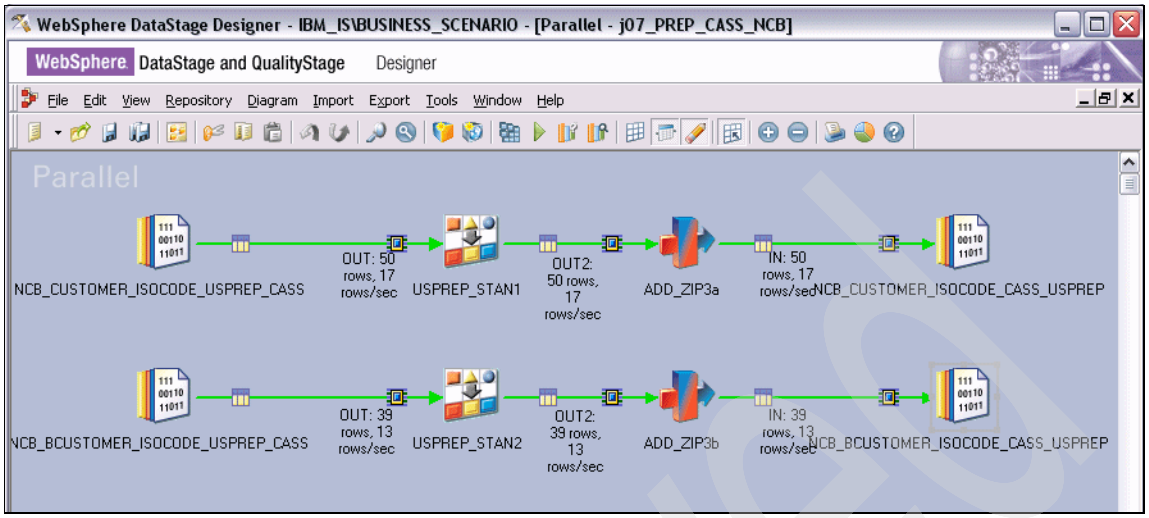
  – Columns AddressDomain_USPREP (contains apartment, street name, and street type tokens), and AreaDomain_USPREP (contains state and ZIP code tokens) that were parsed from the input columns.

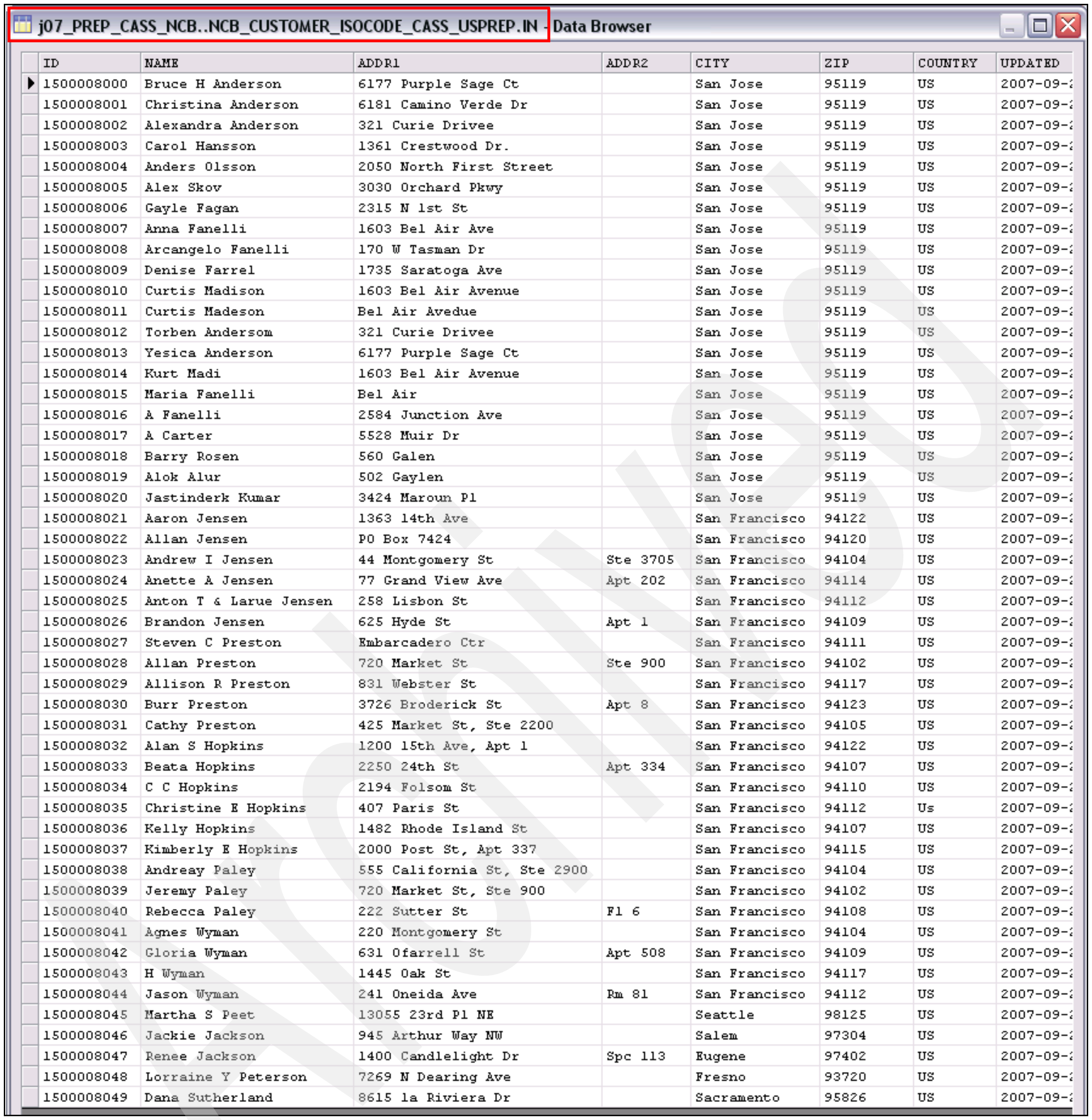
  – InputPattern_USPREP and OutboundPattern_USPREP columns that contain the patterns generated after processing the address columns in the input file.

A visual analysis of the report shows correct parsing of addresses into the AddressDomain_USPREP columns and Area_Domain_USPREP columns of the CUSTOMER and BCUSTOMER files. In the real-world, the volume of data would be too large to attempt a visual analysis of the report.

Proceed now to “j08_STAN_CASS_NCB” on page 763.

[7] The literal ZQPUTAZQ automatically defaults the entire field to the Address Domain, while ZQPUTRZQ automatically defaults the entire field to the Area Domain. Example 1-1 on page 33 has a brief overview of the available literals.

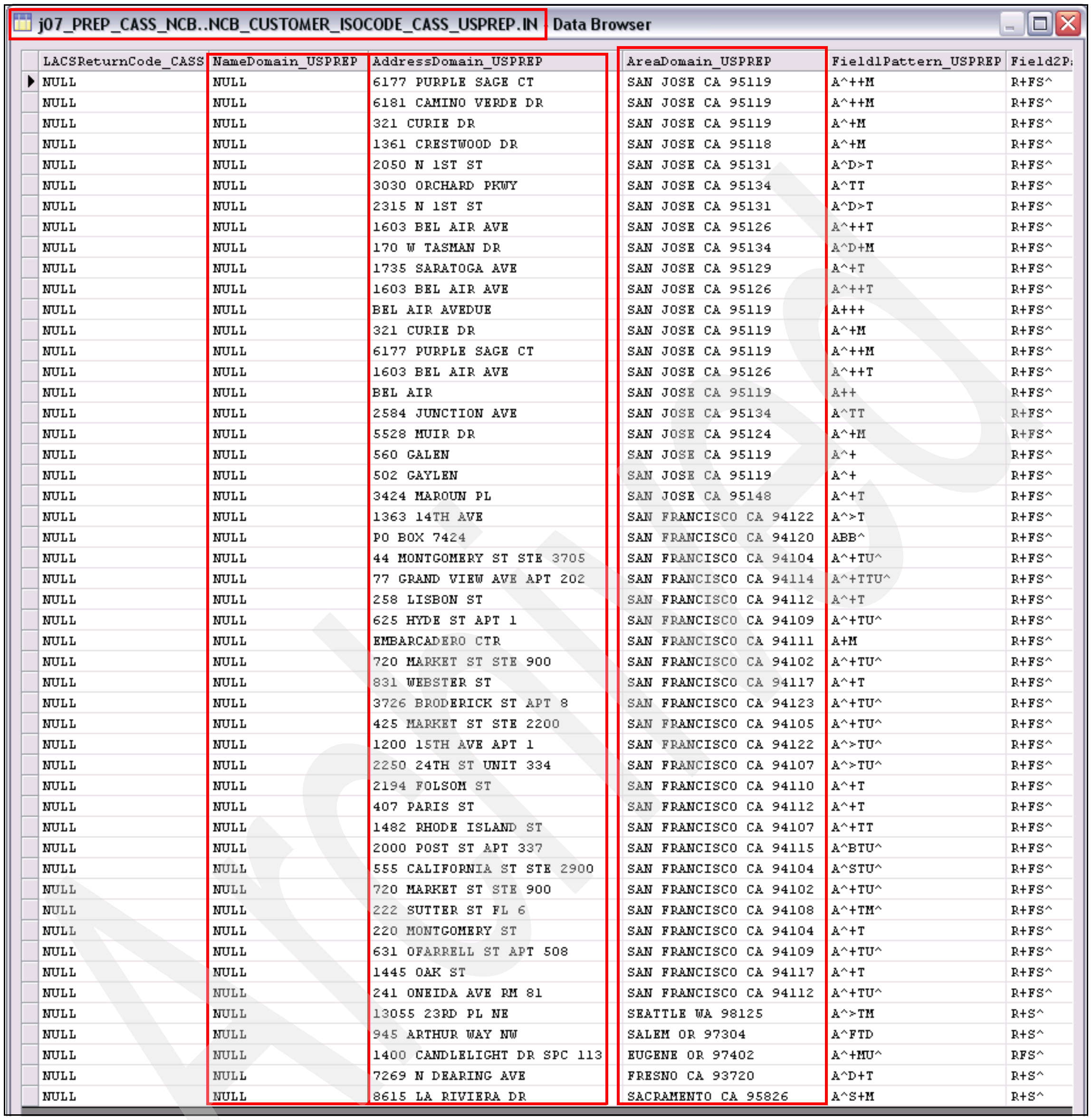

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j07_PREP_CASS_NCB]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NCB_CUSTOMER_ISOCODE_USPREP_CASS
OUT: 50 rows, 17 rows/sec
USPREP_STAN1
OUT2: 50 rows, 17 rows/sec
ADD_ZIP3a
IN: 50 rows, 17 rows/sec
NCB_CUSTOMER_ISOCODE_CASS_USPREP
NCB_BCUSTOMER_ISOCODE_USPREP_CASS
OUT: 39 rows, 13 rows/sec
USPREP_STAN2
OUT2: 39 rows, 13 rows/sec
ADD_ZIP3b
IN: 39 rows, 13 rows/sec
NCB_BCUSTOMER_ISOCODE_CASS_USPREP
Figure 2-290 Create j07_PREP_CASS_NCB 1/7



j07_PREP_CASS_NCB..NCB_CUSTOMER_ISOCODE_CASS_USPREP.IN - Data Browser

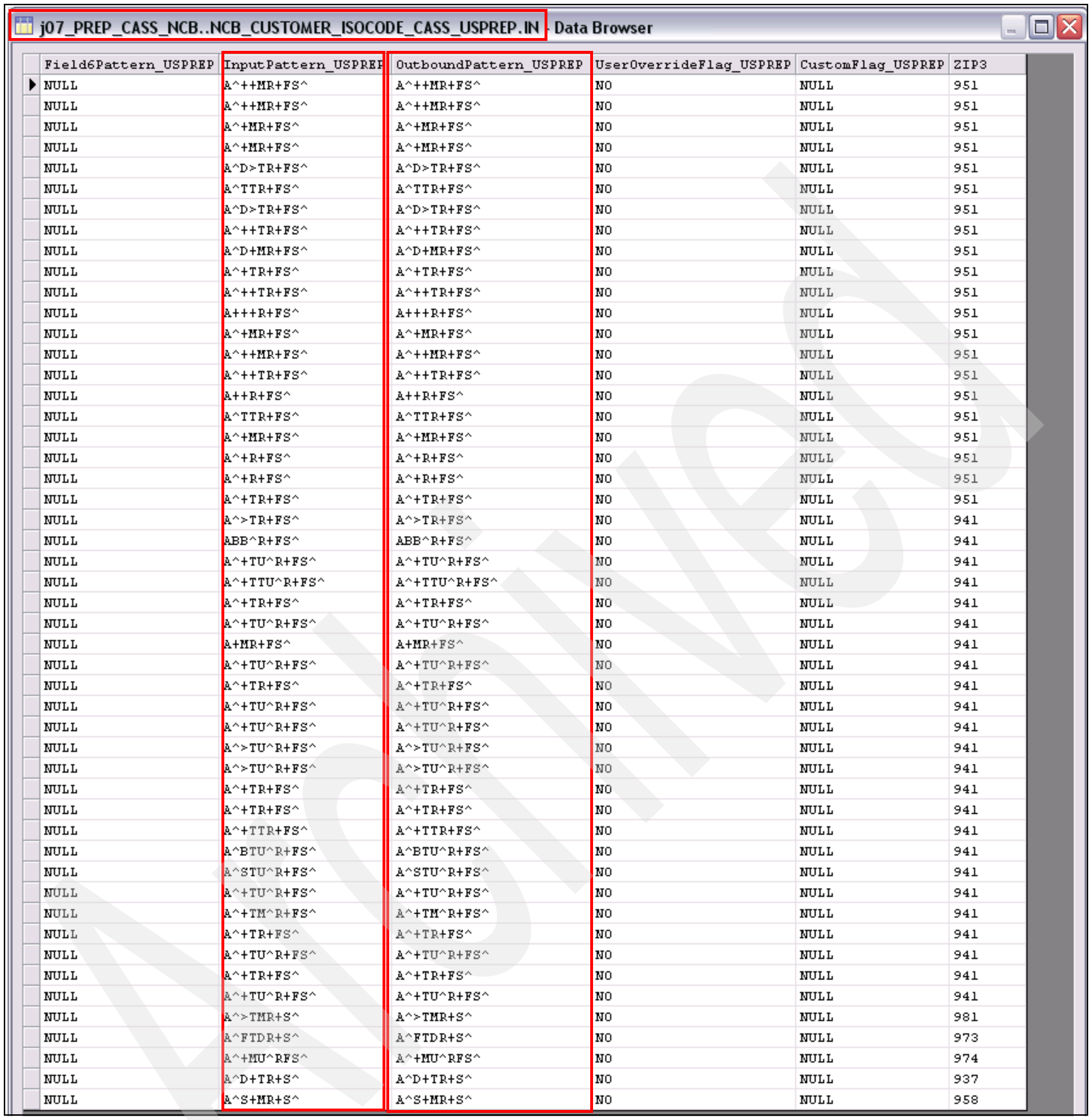
ID	NAME	ADDR1	ADDR2	CITY	ZIP	COUNTRY	UPDATED
1500008000	Bruce H Anderson	6177 Purple Sage Ct		San Jose	95119	US	2007-09-
1500008001	Christina Anderson	6181 Camino Verde Dr		San Jose	95119	US	2007-09-
1500008002	Alexandra Anderson	321 Curie Drivee		San Jose	95119	US	2007-09-
1500008003	Carol Hansson	1361 Crestwood Dr.		San Jose	95119	US	2007-09-
1500008004	Anders Olsson	2050 North First Street		San Jose	95119	US	2007-09-
1500008005	Alex Skov	3030 Orchard Pkwy		San Jose	95119	US	2007-09-
1500008006	Gayle Fagan	2315 N 1st St		San Jose	95119	US	2007-09-
1500008007	Anna Fanelli	1603 Bel Air Ave		San Jose	95119	US	2007-09-
1500008008	Arcangelo Fanelli	170 W Tasman Dr		San Jose	95119	US	2007-09-
1500008009	Denise Farrel	1735 Saratoga Ave		San Jose	95119	US	2007-09-
1500008010	Curtis Madison	1603 Bel Air Avenue		San Jose	95119	US	2007-09-
1500008011	Curtis Madeson	Bel Air Avedue		San Jose	95119	US	2007-09-
1500008012	Torben Andersom	321 Curie Drivee		San Jose	95119	US	2007-09-
1500008013	Yesica Anderson	6177 Purple Sage Ct		San Jose	95119	US	2007-09-
1500008014	Kurt Madi	1603 Bel Air Avenue		San Jose	95119	US	2007-09-
1500008015	Maria Fanelli	Bel Air		San Jose	95119	US	2007-09-
1500008016	A Fanelli	2584 Junction Ave		San Jose	95119	US	2007-09-
1500008017	A Carter	5528 Muir Dr		San Jose	95119	US	2007-09-
1500008018	Barry Rosen	560 Galen		San Jose	95119	US	2007-09-
1500008019	Alok Alur	502 Gaylen		San Jose	95119	US	2007-09-
1500008020	Jastinderk Kumar	3424 Maroun Pl		San Jose	95119	US	2007-09-
1500008021	Aaron Jensen	1363 14th Ave		San Francisco	94122	US	2007-09-
1500008022	Allan Jensen	PO Box 7424		San Francisco	94120	US	2007-09-
1500008023	Andrew I Jensen	44 Montgomery St	Ste 3705	San Francisco	94104	US	2007-09-
1500008024	Anette A Jensen	77 Grand View Ave	Apt 202	San Francisco	94114	US	2007-09-
1500008025	Anton T & Larue Jensen	258 Lisbon St		San Francisco	94112	US	2007-09-
1500008026	Brandon Jensen	625 Hyde St	Apt 1	San Francisco	94109	US	2007-09-
1500008027	Steven C Preston	Embarcadero Ctr		San Francisco	94111	US	2007-09-
1500008028	Allan Preston	720 Market St	Ste 900	San Francisco	94102	US	2007-09-
1500008029	Allison R Preston	831 Webster St		San Francisco	94117	US	2007-09-
1500008030	Burr Preston	3726 Broderick St	Apt 8	San Francisco	94123	US	2007-09-
1500008031	Cathy Preston	425 Market St, Ste 2200		San Francisco	94105	US	2007-09-
1500008032	Alan S Hopkins	1200 15th Ave, Apt 1		San Francisco	94122	US	2007-09-
1500008033	Beata Hopkins	2250 24th St	Apt 334	San Francisco	94107	US	2007-09-
1500008034	C C Hopkins	2194 Folsom St		San Francisco	94110	US	2007-09-
1500008035	Christine E Hopkins	407 Paris St		San Francisco	94112	Us	2007-09-
1500008036	Kelly Hopkins	1482 Rhode Island St		San Francisco	94107	US	2007-09-
1500008037	Kimberly E Hopkins	2000 Post St, Apt 337		San Francisco	94115	US	2007-09-
1500008038	Andreay Paley	555 California St, Ste 2900		San Francisco	94104	US	2007-09-
1500008039	Jeremy Paley	720 Market St, Ste 900		San Francisco	94102	US	2007-09-
1500008040	Rebecca Paley	222 Sutter St	Fl 6	San Francisco	94108	US	2007-09-
1500008041	Agnes Wyman	220 Montgomery St		San Francisco	94104	US	2007-09-
1500008042	Gloria Wyman	631 Ofarrell St	Apt 508	San Francisco	94109	US	2007-09-
1500008043	H Wyman	1445 Oak St		San Francisco	94117	US	2007-09-
1500008044	Jason Wyman	241 Oneida Ave	Rm 81	San Francisco	94112	US	2007-09-
1500008045	Martha S Peet	13055 23rd Pl NE		Seattle	98125	US	2007-09-
1500008046	Jackie Jackson	945 Arthur Way NW		Salem	97304	US	2007-09-
1500008047	Renee Jackson	1400 Candlelight Dr	Spc 113	Eugene	97402	US	2007-09-
1500008048	Lorraine Y Peterson	7269 N Dearing Ave		Fresno	93720	US	2007-09-
1500008049	Dana Sutherland	8615 la Riviera Dr		Sacramento	95826	US	2007-09-

Figure 2-291 Create j07_PREP_CASS_NCB 2/7


j07_PREP_CASS_NCB..NCB_CUSTOMER_ISOCODE_CASS_USPREP.IN Data Browser

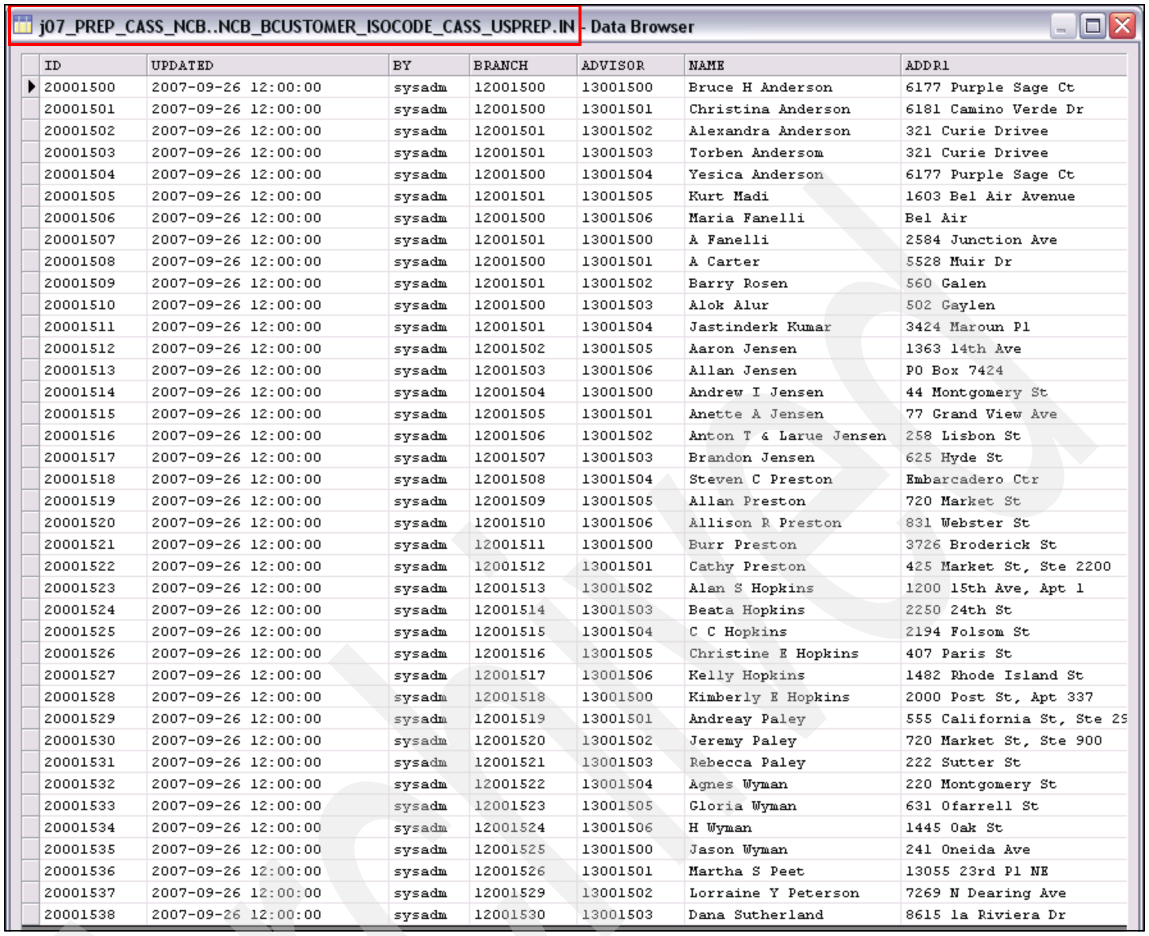
LACSReturnCode_CASS	NameDomain_USPREP	AddressDomain_USPREP	AreaDomain_USPREP	Field1Pattern_USPREP	Field2P
NULL	NULL	6177 PURPLE SAGE CT	SAN JOSE CA 95119	A^++M	R+FS^
NULL	NULL	6181 CAMINO VERDE DR	SAN JOSE CA 95119	A^++M	R+FS^
NULL	NULL	321 CURIE DR	SAN JOSE CA 95119	A^+M	R+FS^
NULL	NULL	1361 CRESTWOOD DR	SAN JOSE CA 95118	A^+M	R+FS^
NULL	NULL	2050 N 1ST ST	SAN JOSE CA 95131	A^D>T	R+FS^
NULL	NULL	3030 ORCHARD PKWY	SAN JOSE CA 95134	A^TT	R+FS^
NULL	NULL	2315 N 1ST ST	SAN JOSE CA 95131	A^D>T	R+FS^
NULL	NULL	1603 BEL AIR AVE	SAN JOSE CA 95126	A^++T	R+FS^
NULL	NULL	170 W TASMAN DR	SAN JOSE CA 95134	A^D+M	R+FS^
NULL	NULL	1735 SARATOGA AVE	SAN JOSE CA 95129	A^+T	R+FS^
NULL	NULL	1603 BEL AIR AVE	SAN JOSE CA 95126	A^++T	R+FS^
NULL	NULL	BEL AIR AVEDUE	SAN JOSE CA 95119	A+++	R+FS^
NULL	NULL	321 CURIE DR	SAN JOSE CA 95119	A^+M	R+FS^
NULL	NULL	6177 PURPLE SAGE CT	SAN JOSE CA 95119	A^++M	R+FS^
NULL	NULL	1603 BEL AIR AVE	SAN JOSE CA 95126	A^++T	R+FS^
NULL	NULL	BEL AIR	SAN JOSE CA 95119	A++	R+FS^
NULL	NULL	2584 JUNCTION AVE	SAN JOSE CA 95134	A^TT	R+FS^
NULL	NULL	5528 MUIR DR	SAN JOSE CA 95124	A^+M	R+FS^
NULL	NULL	560 GALEN	SAN JOSE CA 95119	A^+	R+FS^
NULL	NULL	502 GAYLEN	SAN JOSE CA 95119	A^+	R+FS^
NULL	NULL	3424 MAROUN PL	SAN JOSE CA 95148	A^+T	R+FS^
NULL	NULL	1363 14TH AVE	SAN FRANCISCO CA 94122	A^>T	R+FS^
NULL	NULL	PO BOX 7424	SAN FRANCISCO CA 94120	ABB^	R+FS^
NULL	NULL	44 MONTGOMERY ST STE 3705	SAN FRANCISCO CA 94104	A^+TU^	R+FS^
NULL	NULL	77 GRAND VIEW AVE APT 202	SAN FRANCISCO CA 94114	A^+TTU^	R+FS^
NULL	NULL	258 LISBON ST	SAN FRANCISCO CA 94112	A^+T	R+FS^
NULL	NULL	625 HYDE ST APT 1	SAN FRANCISCO CA 94109	A^+TU^	R+FS^
NULL	NULL	EMBARCADERO CTR	SAN FRANCISCO CA 94111	A+M	R+FS^
NULL	NULL	720 MARKET ST STE 900	SAN FRANCISCO CA 94102	A^+TU^	R+FS^
NULL	NULL	831 WEBSTER ST	SAN FRANCISCO CA 94117	A^+T	R+FS^
NULL	NULL	3726 BRODERICK ST APT 8	SAN FRANCISCO CA 94123	A^+TU^	R+FS^
NULL	NULL	425 MARKET ST STE 2200	SAN FRANCISCO CA 94105	A^+TU^	R+FS^
NULL	NULL	1200 15TH AVE APT 1	SAN FRANCISCO CA 94122	A^>TU^	R+FS^
NULL	NULL	2250 24TH ST UNIT 334	SAN FRANCISCO CA 94107	A^>TU^	R+FS^
NULL	NULL	2194 FOLSOM ST	SAN FRANCISCO CA 94110	A^+T	R+FS^
NULL	NULL	407 PARIS ST	SAN FRANCISCO CA 94112	A^+T	R+FS^
NULL	NULL	1482 RHODE ISLAND ST	SAN FRANCISCO CA 94107	A^+TT	R+FS^
NULL	NULL	2000 POST ST APT 337	SAN FRANCISCO CA 94115	A^BTU^	R+FS^
NULL	NULL	555 CALIFORNIA ST STE 2900	SAN FRANCISCO CA 94104	A^STU^	R+FS^
NULL	NULL	720 MARKET ST STE 900	SAN FRANCISCO CA 94102	A^+TU^	R+FS^
NULL	NULL	222 SUTTER ST FL 6	SAN FRANCISCO CA 94108	A^+TM^	R+FS^
NULL	NULL	220 MONTGOMERY ST	SAN FRANCISCO CA 94104	A^+T	R+FS^
NULL	NULL	631 OFARRELL ST APT 508	SAN FRANCISCO CA 94109	A^+TU^	R+FS^
NULL	NULL	1445 OAK ST	SAN FRANCISCO CA 94117	A^+T	R+FS^
NULL	NULL	241 ONEIDA AVE RM 81	SAN FRANCISCO CA 94112	A^+TU^	R+FS^
NULL	NULL	13055 23RD PL NE	SEATTLE WA 98125	A^>TM	R+S^
NULL	NULL	945 ARTHUR WAY NW	SALEM OR 97304	A^FTD	R+S^
NULL	NULL	1400 CANDLELIGHT DR SPC 113	EUGENE OR 97402	A^+MU^	RFS^
NULL	NULL	7269 N DEARING AVE	FRESNO CA 93720	A^D+T	R+S^
NULL	NULL	8615 LA RIVIERA DR	SACRAMENTO CA 95826	A^S+M	R+S^

Figure 2-292 Create j07_PREP_CASS_NCB 3/7


j07_PREP_CASS_NCB..NCB_CUSTOMER_ISOCODE_CASS_USPREP.IN - Data Browser

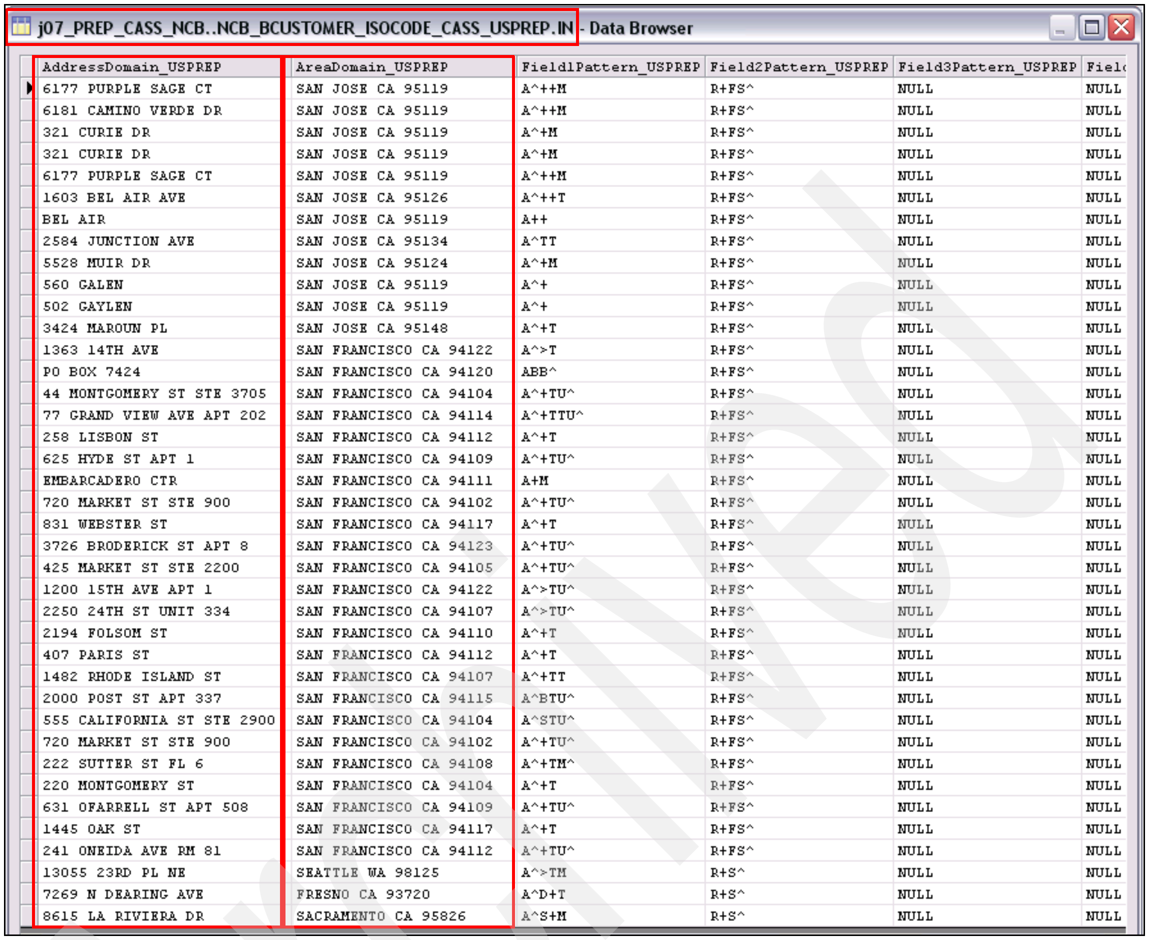
Field6Pattern_USPREP	InputPattern_USPREP	OutboundPattern_USPREP	UserOverrideFlag_USPREP	CustomFlag_USPREP	ZIP3
NULL	A^++MR+FS^	A^++MR+FS^	NO	NULL	951
NULL	A^++MR+FS^	A^++MR+FS^	NO	NULL	951
NULL	A^+MR+FS^	A^+MR+FS^	NO	NULL	951
NULL	A^+MR+FS^	A^+MR+FS^	NO	NULL	951
NULL	A^D>TR+FS^	A^D>TR+FS^	NO	NULL	951
NULL	A^TTR+FS^	A^TTR+FS^	NO	NULL	951
NULL	A^D>TR+FS^	A^D>TR+FS^	NO	NULL	951
NULL	A^++TR+FS^	A^++TR+FS^	NO	NULL	951
NULL	A^D+MR+FS^	A^D+MR+FS^	NO	NULL	951
NULL	A^+TR+FS^	A^+TR+FS^	NO	NULL	951
NULL	A^++TR+FS^	A^++TR+FS^	NO	NULL	951
NULL	A+++R+FS^	A+++R+FS^	NO	NULL	951
NULL	A^+MR+FS^	A^+MR+FS^	NO	NULL	951
NULL	A^++MR+FS^	A^++MR+FS^	NO	NULL	951
NULL	A^++TR+FS^	A^++TR+FS^	NO	NULL	951
NULL	A++R+FS^	A++R+FS^	NO	NULL	951
NULL	A^TTR+FS^	A^TTR+FS^	NO	NULL	951
NULL	A^+MR+FS^	A^+MR+FS^	NO	NULL	951
NULL	A^+R+FS^	A^+R+FS^	NO	NULL	951
NULL	A^+R+FS^	A^+R+FS^	NO	NULL	951
NULL	A^+TR+FS^	A^+TR+FS^	NO	NULL	951
NULL	A^>TR+FS^	A^>TR+FS^	NO	NULL	941
NULL	ABB^R+FS^	ABB^R+FS^	NO	NULL	941
NULL	A^+TU^R+FS^	A^+TU^R+FS^	NO	NULL	941
NULL	A^+TTU^R+FS^	A^+TTU^R+FS^	NO	NULL	941
NULL	A^+TR+FS^	A^+TR+FS^	NO	NULL	941
NULL	A^+TU^R+FS^	A^+TU^R+FS^	NO	NULL	941
NULL	A+MR+FS^	A+MR+FS^	NO	NULL	941
NULL	A^+TU^R+FS^	A^+TU^R+FS^	NO	NULL	941
NULL	A^+TR+FS^	A^+TR+FS^	NO	NULL	941
NULL	A^+TU^R+FS^	A^+TU^R+FS^	NO	NULL	941
NULL	A^+TU^R+FS^	A^+TU^R+FS^	NO	NULL	941
NULL	A^>TU^R+FS^	A^>TU^R+FS^	NO	NULL	941
NULL	A^>TU^R+FS^	A^>TU^R+FS^	NO	NULL	941
NULL	A^+TR+FS^	A^+TR+FS^	NO	NULL	941
NULL	A^+TR+FS^	A^+TR+FS^	NO	NULL	941
NULL	A^+TTR+FS^	A^+TTR+FS^	NO	NULL	941
NULL	A^BTU^R+FS^	A^BTU^R+FS^	NO	NULL	941
NULL	A^STU^R+FS^	A^STU^R+FS^	NO	NULL	941
NULL	A^+TU^R+FS^	A^+TU^R+FS^	NO	NULL	941
NULL	A^+TM^R+FS^	A^+TM^R+FS^	NO	NULL	941
NULL	A^+TR+FS^	A^+TR+FS^	NO	NULL	941
NULL	A^+TU^R+FS^	A^+TU^R+FS^	NO	NULL	941
NULL	A^+TR+FS^	A^+TR+FS^	NO	NULL	941
NULL	A^+TU^R+FS^	A^+TU^R+FS^	NO	NULL	941
NULL	A^>TMR+S^	A^>TMR+S^	NO	NULL	981
NULL	A^FTDR+S^	A^FTDR+S^	NO	NULL	973
NULL	A^+MU^RFS^	A^+MU^RFS^	NO	NULL	974
NULL	A^D+TR+S^	A^D+TR+S^	NO	NULL	937
NULL	A^S+MR+S^	A^S+MR+S^	NO	NULL	958

Figure 2-293 Create j07_PREP_CASS_NCB 4/7


j07_PREP_CASS_NCB..NCB_BCUSTOMER_ISOCODE_CASS_USPREP.IN - Data Browser

ID	UPDATED	BY	BRANCH	ADVISOR	NAME	ADDR1
20001500	2007-09-26 12:00:00	sysadm	12001500	13001500	Bruce H Anderson	6177 Purple Sage Ct
20001501	2007-09-26 12:00:00	sysadm	12001500	13001501	Christina Anderson	6181 Camino Verde Dr
20001502	2007-09-26 12:00:00	sysadm	12001501	13001502	Alexandra Anderson	321 Curie Drivee
20001503	2007-09-26 12:00:00	sysadm	12001501	13001503	Torben Andersom	321 Curie Drivee
20001504	2007-09-26 12:00:00	sysadm	12001500	13001504	Yesica Anderson	6177 Purple Sage Ct
20001505	2007-09-26 12:00:00	sysadm	12001501	13001505	Kurt Madi	1603 Bel Air Avenue
20001506	2007-09-26 12:00:00	sysadm	12001500	13001506	Maria Fanelli	Bel Air
20001507	2007-09-26 12:00:00	sysadm	12001501	13001500	A Fanelli	2584 Junction Ave
20001508	2007-09-26 12:00:00	sysadm	12001500	13001501	A Carter	5528 Muir Dr
20001509	2007-09-26 12:00:00	sysadm	12001501	13001502	Barry Rosen	560 Galen
20001510	2007-09-26 12:00:00	sysadm	12001500	13001503	Alok Alur	502 Gaylen
20001511	2007-09-26 12:00:00	sysadm	12001501	13001504	Jastinderk Kumar	3424 Maroun Pl
20001512	2007-09-26 12:00:00	sysadm	12001502	13001505	Aaron Jensen	1363 14th Ave
20001513	2007-09-26 12:00:00	sysadm	12001503	13001506	Allan Jensen	PO Box 7424
20001514	2007-09-26 12:00:00	sysadm	12001504	13001500	Andrew I Jensen	44 Montgomery St
20001515	2007-09-26 12:00:00	sysadm	12001505	13001501	Anette A Jensen	77 Grand View Ave
20001516	2007-09-26 12:00:00	sysadm	12001506	13001502	Anton T & Larue Jensen	258 Lisbon St
20001517	2007-09-26 12:00:00	sysadm	12001507	13001503	Brandon Jensen	625 Hyde St
20001518	2007-09-26 12:00:00	sysadm	12001508	13001504	Steven C Preston	Embarcadero Ctr
20001519	2007-09-26 12:00:00	sysadm	12001509	13001505	Allan Preston	720 Market St
20001520	2007-09-26 12:00:00	sysadm	12001510	13001506	Allison R Preston	831 Webster St
20001521	2007-09-26 12:00:00	sysadm	12001511	13001500	Burr Preston	3726 Broderick St
20001522	2007-09-26 12:00:00	sysadm	12001512	13001501	Cathy Preston	425 Market St, Ste 2200
20001523	2007-09-26 12:00:00	sysadm	12001513	13001502	Alan S Hopkins	1200 15th Ave, Apt 1
20001524	2007-09-26 12:00:00	sysadm	12001514	13001503	Beata Hopkins	2250 24th St
20001525	2007-09-26 12:00:00	sysadm	12001515	13001504	C C Hopkins	2194 Folsom St
20001526	2007-09-26 12:00:00	sysadm	12001516	13001505	Christine E Hopkins	407 Paris St
20001527	2007-09-26 12:00:00	sysadm	12001517	13001506	Kelly Hopkins	1482 Rhode Island St
20001528	2007-09-26 12:00:00	sysadm	12001518	13001500	Kimberly E Hopkins	2000 Post St, Apt 337
20001529	2007-09-26 12:00:00	sysadm	12001519	13001501	Andreay Paley	555 California St, Ste 29
20001530	2007-09-26 12:00:00	sysadm	12001520	13001502	Jeremy Paley	720 Market St, Ste 900
20001531	2007-09-26 12:00:00	sysadm	12001521	13001503	Rebecca Paley	222 Sutter St
20001532	2007-09-26 12:00:00	sysadm	12001522	13001504	Agnes Wyman	220 Montgomery St
20001533	2007-09-26 12:00:00	sysadm	12001523	13001505	Gloria Wyman	631 Ofarrell St
20001534	2007-09-26 12:00:00	sysadm	12001524	13001506	H Wyman	1445 Oak St
20001535	2007-09-26 12:00:00	sysadm	12001525	13001500	Jason Wyman	241 Oneida Ave
20001536	2007-09-26 12:00:00	sysadm	12001526	13001501	Martha S Peet	13055 23rd Pl NE
20001537	2007-09-26 12:00:00	sysadm	12001529	13001502	Lorraine Y Peterson	7269 N Dearing Ave
20001538	2007-09-26 12:00:00	sysadm	12001530	13001503	Dana Sutherland	8615 la Riviera Dr

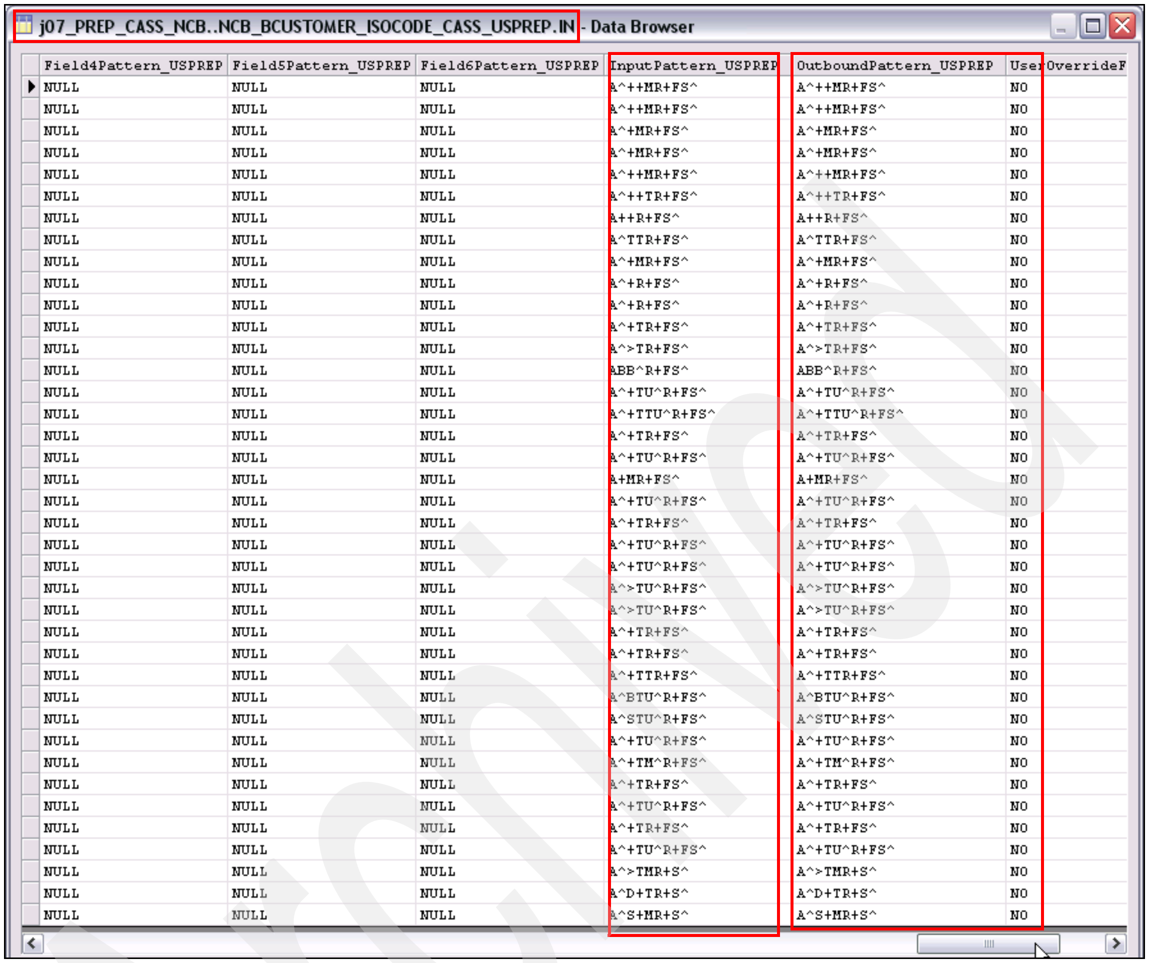
Figure 2-294 Create j07_PREP_CASS_NCB 5/7



j07_PREP_CASS_NCB..NCB_BCUSTOMER_ISOCODE_CASS_USPREP.IN - Data Browser

AddressDomain_USPREP	AreaDomain_USPREP	Field1Pattern_USPREP	Field2Pattern_USPREP	Field3Pattern_USPREP	Fiel
6177 PURPLE SAGE CT	SAN JOSE CA 95119	A^++M	R+FS^	NULL	NULL
6181 CAMINO VERDE DR	SAN JOSE CA 95119	A^++M	R+FS^	NULL	NULL
321 CURIE DR	SAN JOSE CA 95119	A^+M	R+FS^	NULL	NULL
321 CURIE DR	SAN JOSE CA 95119	A^+M	R+FS^	NULL	NULL
6177 PURPLE SAGE CT	SAN JOSE CA 95119	A^++M	R+FS^	NULL	NULL
1603 BEL AIR AVE	SAN JOSE CA 95126	A^++T	R+FS^	NULL	NULL
BEL AIR	SAN JOSE CA 95119	A++	R+FS^	NULL	NULL
2584 JUNCTION AVE	SAN JOSE CA 95134	A^TT	R+FS^	NULL	NULL
5528 MUIR DR	SAN JOSE CA 95124	A^+M	R+FS^	NULL	NULL
560 GALEN	SAN JOSE CA 95119	A^+	R+FS^	NULL	NULL
502 GAYLEN	SAN JOSE CA 95119	A^+	R+FS^	NULL	NULL
3424 MAROUN PL	SAN JOSE CA 95148	A^+T	R+FS^	NULL	NULL
1363 14TH AVE	SAN FRANCISCO CA 94122	A^>T	R+FS^	NULL	NULL
PO BOX 7424	SAN FRANCISCO CA 94120	ABB^	R+FS^	NULL	NULL
44 MONTGOMERY ST STE 3705	SAN FRANCISCO CA 94104	A^+TU^	R+FS^	NULL	NULL
77 GRAND VIEW AVE APT 202	SAN FRANCISCO CA 94114	A^+TTU^	R+FS^	NULL	NULL
258 LISBON ST	SAN FRANCISCO CA 94112	A^+T	R+FS^	NULL	NULL
625 HYDE ST APT 1	SAN FRANCISCO CA 94109	A^+TU^	R+FS^	NULL	NULL
EMBARCADERO CTR	SAN FRANCISCO CA 94111	A+M	R+FS^	NULL	NULL
720 MARKET ST STE 900	SAN FRANCISCO CA 94102	A^+TU^	R+FS^	NULL	NULL
831 WEBSTER ST	SAN FRANCISCO CA 94117	A^+T	R+FS^	NULL	NULL
3726 BRODERICK ST APT 8	SAN FRANCISCO CA 94123	A^+TU^	R+FS^	NULL	NULL
425 MARKET ST STE 2200	SAN FRANCISCO CA 94105	A^+TU^	R+FS^	NULL	NULL
1200 15TH AVE APT 1	SAN FRANCISCO CA 94122	A^>TU^	R+FS^	NULL	NULL
2250 24TH ST UNIT 334	SAN FRANCISCO CA 94107	A^>TU^	R+FS^	NULL	NULL
2194 FOLSOM ST	SAN FRANCISCO CA 94110	A^+T	R+FS^	NULL	NULL
407 PARIS ST	SAN FRANCISCO CA 94112	A^+T	R+FS^	NULL	NULL
1482 RHODE ISLAND ST	SAN FRANCISCO CA 94107	A^+TT	R+FS^	NULL	NULL
2000 POST ST APT 337	SAN FRANCISCO CA 94115	A^BTU^	R+FS^	NULL	NULL
555 CALIFORNIA ST STE 2900	SAN FRANCISCO CA 94104	A^STU^	R+FS^	NULL	NULL
720 MARKET ST STE 900	SAN FRANCISCO CA 94102	A^+TU^	R+FS^	NULL	NULL
222 SUTTER ST FL 6	SAN FRANCISCO CA 94108	A^+TM^	R+FS^	NULL	NULL
220 MONTGOMERY ST	SAN FRANCISCO CA 94104	A^+T	R+FS^	NULL	NULL
631 OFARRELL ST APT 508	SAN FRANCISCO CA 94109	A^+TU^	R+FS^	NULL	NULL
1445 OAK ST	SAN FRANCISCO CA 94117	A^+T	R+FS^	NULL	NULL
241 ONEIDA AVE RM 81	SAN FRANCISCO CA 94112	A^+TU^	R+FS^	NULL	NULL
13055 23RD PL NE	SEATTLE WA 98125	A^>TM	R+S^	NULL	NULL
7269 N DEARING AVE	FRESNO CA 93720	A^D+T	R+S^	NULL	NULL
8615 LA RIVIERA DR	SACRAMENTO CA 95826	A^S+M	R+S^	NULL	NULL

Figure 2-295 Create j07_PREP_CASS_NCB 6/7



j07_PREP_CASS_NCB..NCB_BCUSTOMER_ISOCODE_CASS_USPREP.IN - Data Browser

Field4Pattern_USPREP	Field5Pattern_USPREP	Field6Pattern_USPREP	InputPattern_USPREP	OutboundPattern_USPREP	UserOverrideF
NULL	NULL	NULL	A^++MR+FS^	A^++MR+FS^	NO
NULL	NULL	NULL	A^++MR+FS^	A^++MR+FS^	NO
NULL	NULL	NULL	A^+MR+FS^	A^+MR+FS^	NO
NULL	NULL	NULL	A^+MR+FS^	A^+MR+FS^	NO
NULL	NULL	NULL	A^++MR+FS^	A^++MR+FS^	NO
NULL	NULL	NULL	A^++TR+FS^	A^++TR+FS^	NO
NULL	NULL	NULL	A++R+FS^	A++R+FS^	NO
NULL	NULL	NULL	A^TTR+FS^	A^TTR+FS^	NO
NULL	NULL	NULL	A^+MR+FS^	A^+MR+FS^	NO
NULL	NULL	NULL	A^+R+FS^	A^+R+FS^	NO
NULL	NULL	NULL	A^+R+FS^	A^+R+FS^	NO
NULL	NULL	NULL	A^+TR+FS^	A^+TR+FS^	NO
NULL	NULL	NULL	A^>TR+FS^	A^>TR+FS^	NO
NULL	NULL	NULL	ABB^R+FS^	ABB^R+FS^	NO
NULL	NULL	NULL	A^+TU^R+FS^	A^+TU^R+FS^	NO
NULL	NULL	NULL	A^+TTU^R+FS^	A^+TTU^R+FS^	NO
NULL	NULL	NULL	A^+TR+FS^	A^+TR+FS^	NO
NULL	NULL	NULL	A^+TU^R+FS^	A^+TU^R+FS^	NO
NULL	NULL	NULL	A+MR+FS^	A+MR+FS^	NO
NULL	NULL	NULL	A^+TU^R+FS^	A^+TU^R+FS^	NO
NULL	NULL	NULL	A^+TR+FS^	A^+TR+FS^	NO
NULL	NULL	NULL	A^+TU^R+FS^	A^+TU^R+FS^	NO
NULL	NULL	NULL	A^+TU^R+FS^	A^+TU^R+FS^	NO
NULL	NULL	NULL	A^>TU^R+FS^	A^>TU^R+FS^	NO
NULL	NULL	NULL	A^>TU^R+FS^	A^>TU^R+FS^	NO
NULL	NULL	NULL	A^+TR+FS^	A^+TR+FS^	NO
NULL	NULL	NULL	A^+TR+FS^	A^+TR+FS^	NO
NULL	NULL	NULL	A^+TTR+FS^	A^+TTR+FS^	NO
NULL	NULL	NULL	A^BTU^R+FS^	A^BTU^R+FS^	NO
NULL	NULL	NULL	A^STU^R+FS^	A^STU^R+FS^	NO
NULL	NULL	NULL	A^+TU^R+FS^	A^+TU^R+FS^	NO
NULL	NULL	NULL	A^+TM^R+FS^	A^+TM^R+FS^	NO
NULL	NULL	NULL	A^+TR+FS^	A^+TR+FS^	NO
NULL	NULL	NULL	A^+TU^R+FS^	A^+TU^R+FS^	NO
NULL	NULL	NULL	A^+TR+FS^	A^+TR+FS^	NO
NULL	NULL	NULL	A^+TU^R+FS^	A^+TU^R+FS^	NO
NULL	NULL	NULL	A^>TMR+S^	A^>TMR+S^	NO
NULL	NULL	NULL	A^D+TR+S^	A^D+TR+S^	NO
NULL	NULL	NULL	A^S+MR+S^	A^S+MR+S^	NO

Figure 2-296 Create j07_PREP_CASS_NCB 7/7

j08_STAN_CASS_NCB

In this step, we standardize the name and address contents of the output of job "j07_PREP_CASS_NCB" on page 755 using the domain-specific rule sets USNAME (with column NameDomain_USPREP), USADDR (with column AddressDomain_USPREP), and USAREA (with column AreaDomain_USPREP). Separate processes are defined—one for each rule set.

Note: Here again, we had to introduce a null handling Transformer stage to circumvent a bug.

Figure 2-297 shows the various stages that are used in this job. It includes the data sets that were created in “j07_PREP_CASS_NCB” on page 755, a Standardize stage, a Transformer stage to handle nulls, and an output Data Set stage for each of the input data sets. We modified the names of the stages as shown.

Because the configuration of this job is very similar to that described in “J08_STAN_CUSTOMER_Domain_Specific” on page 241, we do not repeat it here.

After saving, compiling, and running this job (Figure 2-297 on page 764), the standardized output is written to new columns such as:

- FirstName_USNAME and PrimaryName_USNAME
- InputPattern_USNAME
- UnhandledPattern_USNAME
- UnhandledData_USNAME
- InputPattern_USNAME
- UnhandledPattern_USADDR
- UnhandledData_USADDR
- InputPattern_USADDR
- UnhandledPattern_USAREA
- d UnhandledData_USAREA

The next step identifies any unhandled patterns and classifications by performing an investigate as described in “j09a_INVCC_CUSTOMER_STAN_CASS_NCB” on page 765 through “j09b_INVCC_BCUSTOMER_STAN_CASS_NCB” on page 771.

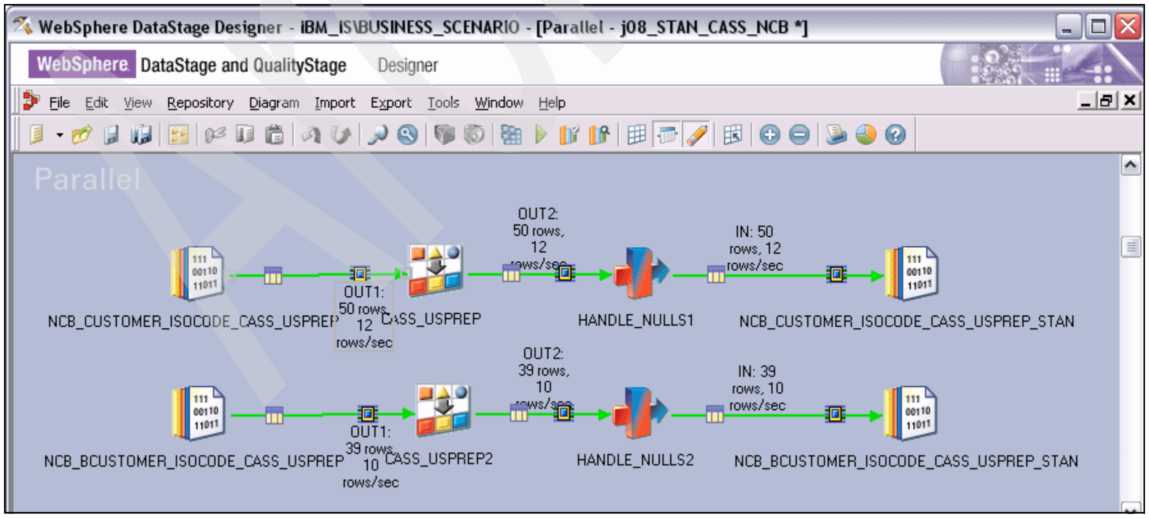

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j08_STAN_CASS_NCB *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NCB_CUSTOMER_ISOCODE_CASS_USPREP
OUT1: 50 rows, 12 rows/sec
CASS_USPREP
OUT2: 50 rows, 12 rows/sec
HANDLE_NULLS1
IN: 50 rows, 12 rows/sec
NCB_CUSTOMER_ISOCODE_CASS_USPREP_STAN
NCB_BCUSTOMER_ISOCODE_CASS_USPREP
OUT1: 39 rows, 10 rows/sec
CASS_USPREP2
OUT2: 39 rows, 10 rows/sec
HANDLE_NULLS2
IN: 39 rows, 10 rows/sec
NCB_BCUSTOMER_ISOCODE_CASS_USPREP_STAN

Figure 2-297 Create j08_STAN_CASS_NCB

j09a_INVCC_CUSTOMER_STAN_CASS_NCB

In this step, we identify unhandled patterns and classifications in the previous step in the CUSTOMER file. We run a series of Investigate stage with the character concatenate option using the *C* mask on the unhandled pattern column from the results of "j08_STAN_CASS_NCB" on page 763. The columns that we investigate corresponded to the name, address, and area domains. Unhandled patterns are reviewed and classification and input pattern overrides are generated to rectify the problem. We then rerun the Standardize stage with the overrides in place and verify using Investigate that all the unhandled patterns are resolved.

Figure 2-298 on page 766 shows the various stages that are used in this job. It includes the data set that was created in "j08_STAN_CASS_NCB" on page 763, an Investigate stage with the character concatenate option, and a sequential file for the column frequency report. We modified the names of the stages as shown.

Because the configuration of this job is very similar to that described in "J09_INVCC_STAN_CUSTOMER" on page 253, we do not repeat it here.

After saving, compiling, and running this job (Figure 2-298 on page 766) on the CUSTOMER file, the contents of the output of the investigate stage shows unhandled patterns as follows:

- Unhandled address patterns in Figure 2-300 on page 766 and Figure 2-301 on page 767.

  An Input Pattern override was generated for the unhandled pattern as shown in Figure 2-303 on page 767 and Figure 2-304 on page 768. We then rerun the "j08_STAN_CASS_NCB" on page 763 and "j09a_INVCC_CUSTOMER_STAN_CASS_NCB" on page 765 jobs. Figure 2-305 on page 768 shows no more unhandled patterns relating to the address domain.

- Unhandled (or incorrectly handled) name patterns in Figure 2-306 on page 769 where PrimaryName_USNAME has some incorrect data (such as ANDERS OLSSON and ANDREAY PALEY).

  A Classification override was generated for the unclassified firstname tokens as shown in Figure 2-307 on page 770. We then rerun the "j08_STAN_CASS_NCB" on page 763 and "j09a_INVCC_CUSTOMER_STAN_CASS_NCB" on page 765 jobs. Figure 2-308 on page 771 shows that everything is handled correctly.

Proceed now to unhandled patterns in the address domain in "j09b_INVCC_BCUSTOMER_STAN_CASS_NCB" on page 771.

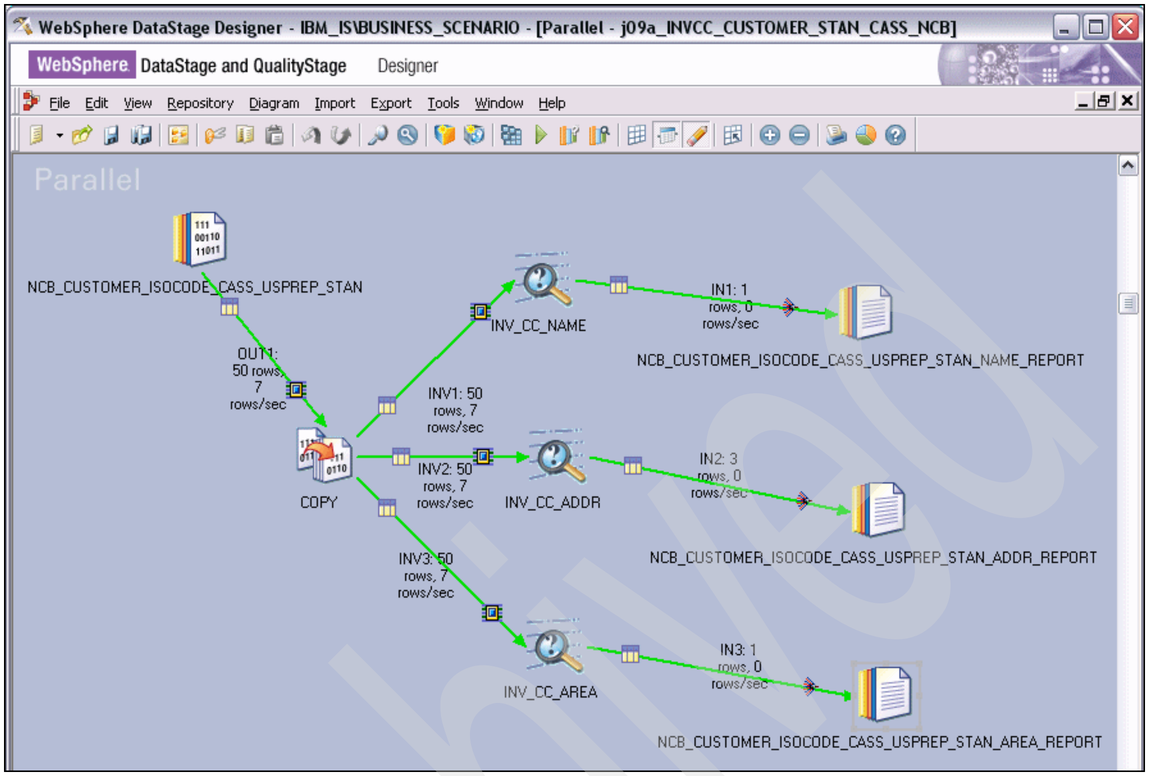
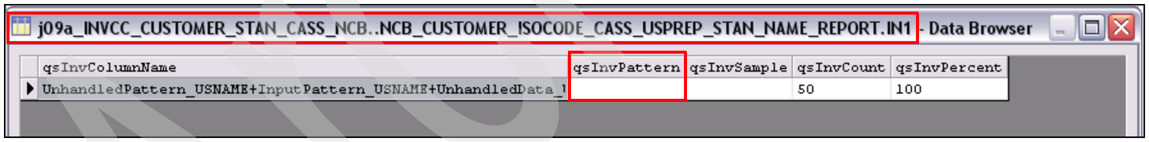
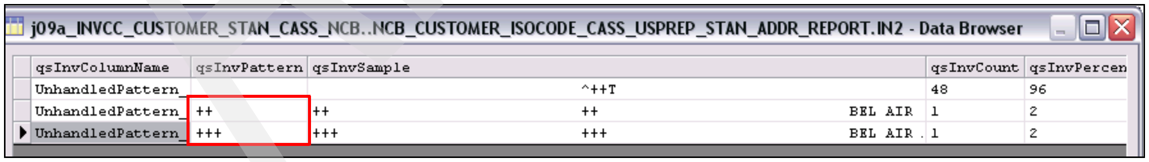

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j09a_INVCC_CUSTOMER_STAN_CASS_NCB]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NCB_CUSTOMER_ISOCODE_CASS_USPREP_STAN
OUT1: 50 rows, 7 rows/sec
COPY
INV1: 50 rows, 7 rows/sec
INV2: 50 rows, 7 rows/sec
INV3: 50 rows, 7 rows/sec
INV_CC_NAME
INV_CC_ADDR
INV_CC_AREA
IN1: 1 rows, 0 rows/sec
IN2: 3 rows, 0 rows/sec
IN3: 1 rows, 0 rows/sec
NCB_CUSTOMER_ISOCODE_CASS_USPREP_STAN_NAME_REPORT
NCB_CUSTOMER_ISOCODE_CASS_USPREP_STAN_ADDR_REPORT
NCB_CUSTOMER_ISOCODE_CASS_USPREP_STAN_AREA_REPORT

Figure 2-298 Create j09a_INVCC_CUSTOMER_STAN_CASS_NCB 1/11


j09a_INVCC_CUSTOMER_STAN_CASS_NCB..NCB_CUSTOMER_ISOCODE_CASS_USPREP_STAN_NAME_REPORT.IN1 - Data Browser

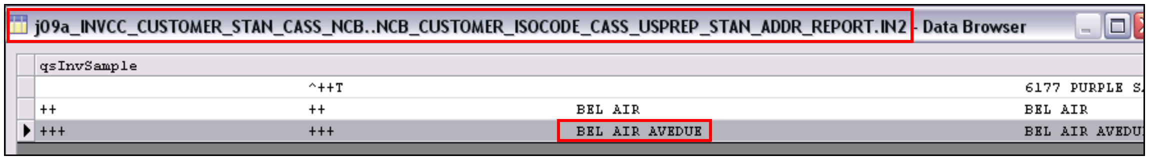
qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_1			50	100

Figure 2-299 Create j09a_INVCC_CUSTOMER_STAN_CASS_NCB 2/11


j09a_INVCC_CUSTOMER_STAN_CASS_NCB..NCB_CUSTOMER_ISOCODE_CASS_USPREP_STAN_ADDR_REPORT.IN2 - Data Browser

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercen
UnhandledPattern_		^++T	48	96
UnhandledPattern_	++	++ ++ BEL AIR	1	2
UnhandledPattern_	+++	+++ +++ BEL AIR .	1	2

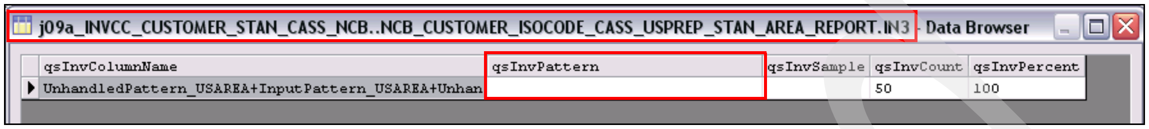
Figure 2-300 Create j09a_INVCC_CUSTOMER_STAN_CASS_NCB 3/11



j09a_INVCC_CUSTOMER_STAN_CASS_NCB..NCB_CUSTOMER_ISOCODE_CASS_USPREP_STAN_ADDR_REPORT.IN2 - Data Browser

```
qsInvSample
                              ^++T                                                      6177 PURPLE S
++                            ++                            BEL AIR                     BEL AIR
+++                           +++                           BEL AIR AVEDUE              BEL AIR AVEDU
```

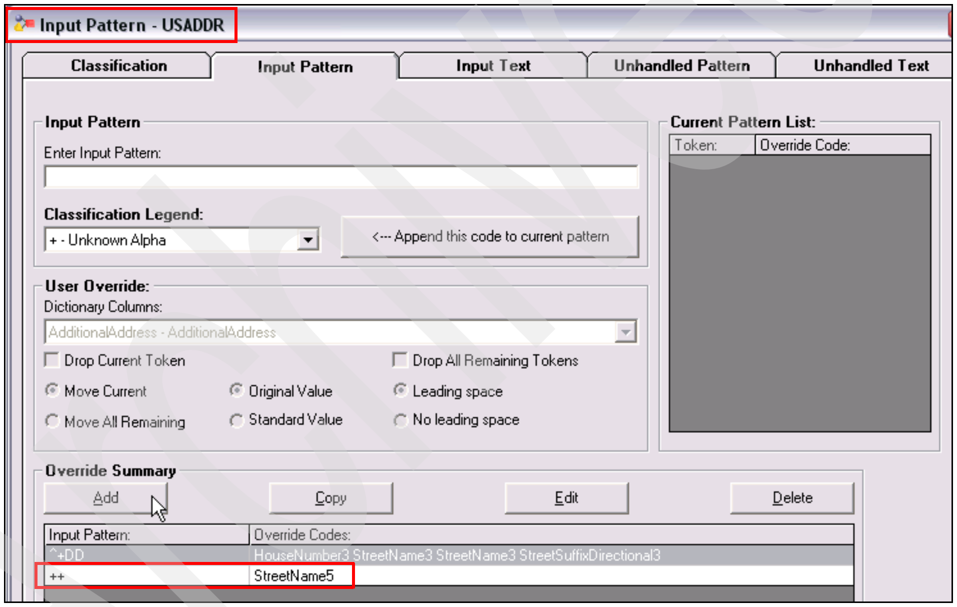
Figure 2-301 Create j09a_INVCC_CUSTOMER_STAN_CASS_NCB 4/11



j09a_INVCC_CUSTOMER_STAN_CASS_NCB..NCB_CUSTOMER_ISOCODE_CASS_USPREP_STAN_AREA_REPORT.IN3 - Data Browser

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
UnhandledPattern_USAREA+InputPattern_USAREA+Unhan			50	100

Figure 2-302 Create j09a_INVCC_CUSTOMER_STAN_CASS_NCB 5/11



Input Pattern - USADDR

Input Pattern

Enter Input Pattern:

Classification Legend: + - Unknown Alpha

User Override:

Dictionary Columns: AdditionalAddress - AdditionalAddress

Drop Current Token [ ] Drop All Remaining Tokens [ ]

[✓ Move Current] Move All Remaining

[✓ Original Value] Standard Value

[✓ Leading space] No leading space

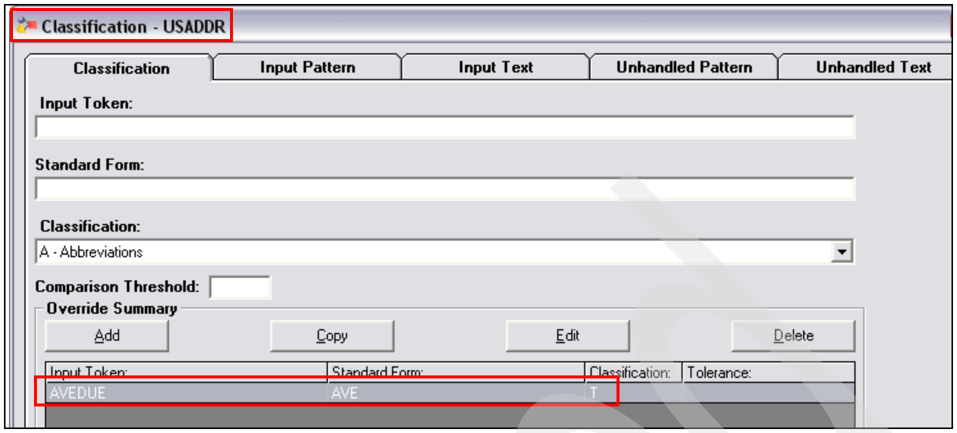
Current Pattern List:

Token:	Override Code:

Override Summary

Input Pattern:	Override Codes:
^+DD	HouseNumber3 StreetName3 StreetName3 StreetSuffixDirectional3
++	StreetName5

Figure 2-303 Create j09a_INVCC_CUSTOMER_STAN_CASS_NCB 6/11


Classification - USADDR

Classification | Input Pattern | Input Text | Unhandled Pattern | Unhandled Text

Input Token:

Standard Form:

Classification: A - Abbreviations

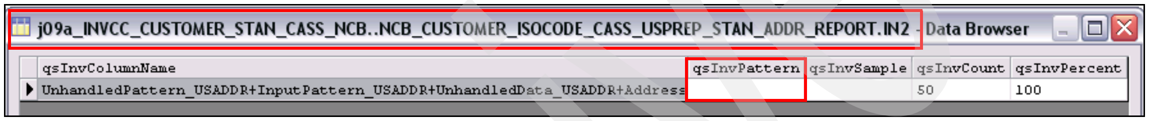
Comparison Threshold:

Override Summary

Add | Copy | Edit | Delete

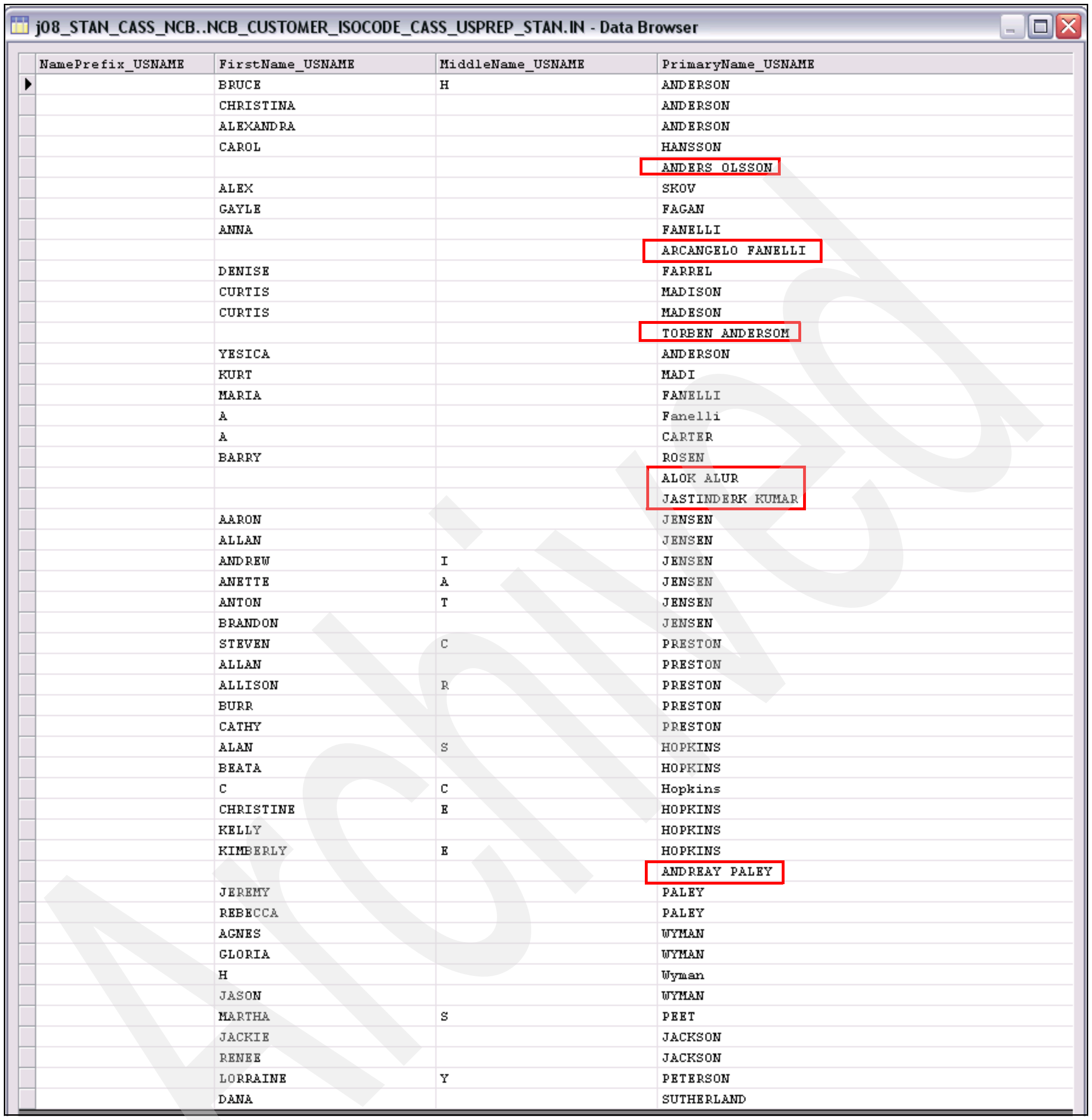
Input Token:	Standard Form:	Classification:	Tolerance:
AVEDUE	AVE	T	

Figure 2-304   Create j09a_INVCC_CUSTOMER_STAN_CASS_NCB 7/11


j09a_INVCC_CUSTOMER_STAN_CASS_NCB..NCB_CUSTOMER_ISOCODE_CASS_USPREP_STAN_ADDR_REPORT.IN2 - Data Browser

qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+Address			50	100

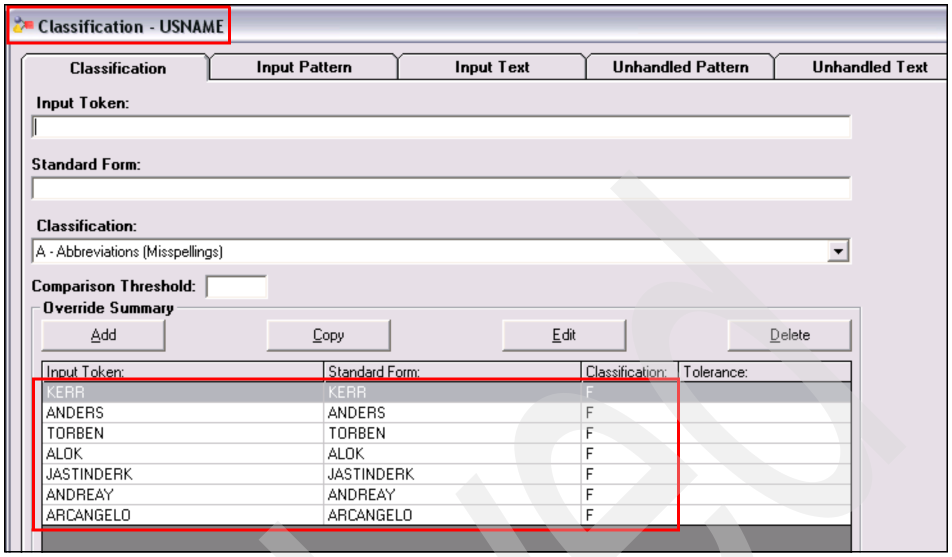
Figure 2-305   Create j09a_INVCC_CUSTOMER_STAN_CASS_NCB 8/11



j08_STAN_CASS_NCB..NCB_CUSTOMER_ISOCODE_CASS_USPREP_STAN.IN - Data Browser

NamePrefix_USNAME	FirstName_USNAME	MiddleName_USNAME	PrimaryName_USNAME
	BRUCE	H	ANDERSON
	CHRISTINA		ANDERSON
	ALEXANDRA		ANDERSON
	CAROL		HANSSON
			ANDERS OLSSON
	ALEX		SKOV
	GAYLE		FAGAN
	ANNA		FANELLI
			ARCANGELO FANELLI
	DENISE		FARREL
	CURTIS		MADISON
	CURTIS		MADESON
			TORBEN ANDERSOM
	YESICA		ANDERSON
	KURT		MADI
	MARIA		FANELLI
	A		Fanelli
	A		CARTER
	BARRY		ROSEN
			ALOK ALUR
			JASTINDERK KUMAR
	AARON		JENSEN
	ALLAN		JENSEN
	ANDREW	I	JENSEN
	ANETTE	A	JENSEN
	ANTON	T	JENSEN
	BRANDON		JENSEN
	STEVEN	C	PRESTON
	ALLAN		PRESTON
	ALLISON	R	PRESTON
	BURR		PRESTON
	CATHY		PRESTON
	ALAN	S	HOPKINS
	BEATA		HOPKINS
	C	C	Hopkins
	CHRISTINE	E	HOPKINS
	KELLY		HOPKINS
	KIMBERLY	E	HOPKINS
			ANDREAY PALEY
	JEREMY		PALEY
	REBECCA		PALEY
	AGNES		WYMAN
	GLORIA		WYMAN
	H		Wyman
	JASON		WYMAN
	MARTHA	S	PEET
	JACKIE		JACKSON
	RENEE		JACKSON
	LORRAINE	Y	PETERSON
	DANA		SUTHERLAND

Figure 2-306 Create j09a_INVCC_CUSTOMER_STAN_CASS_NCB 9/11



Input Token:

Standard Form:

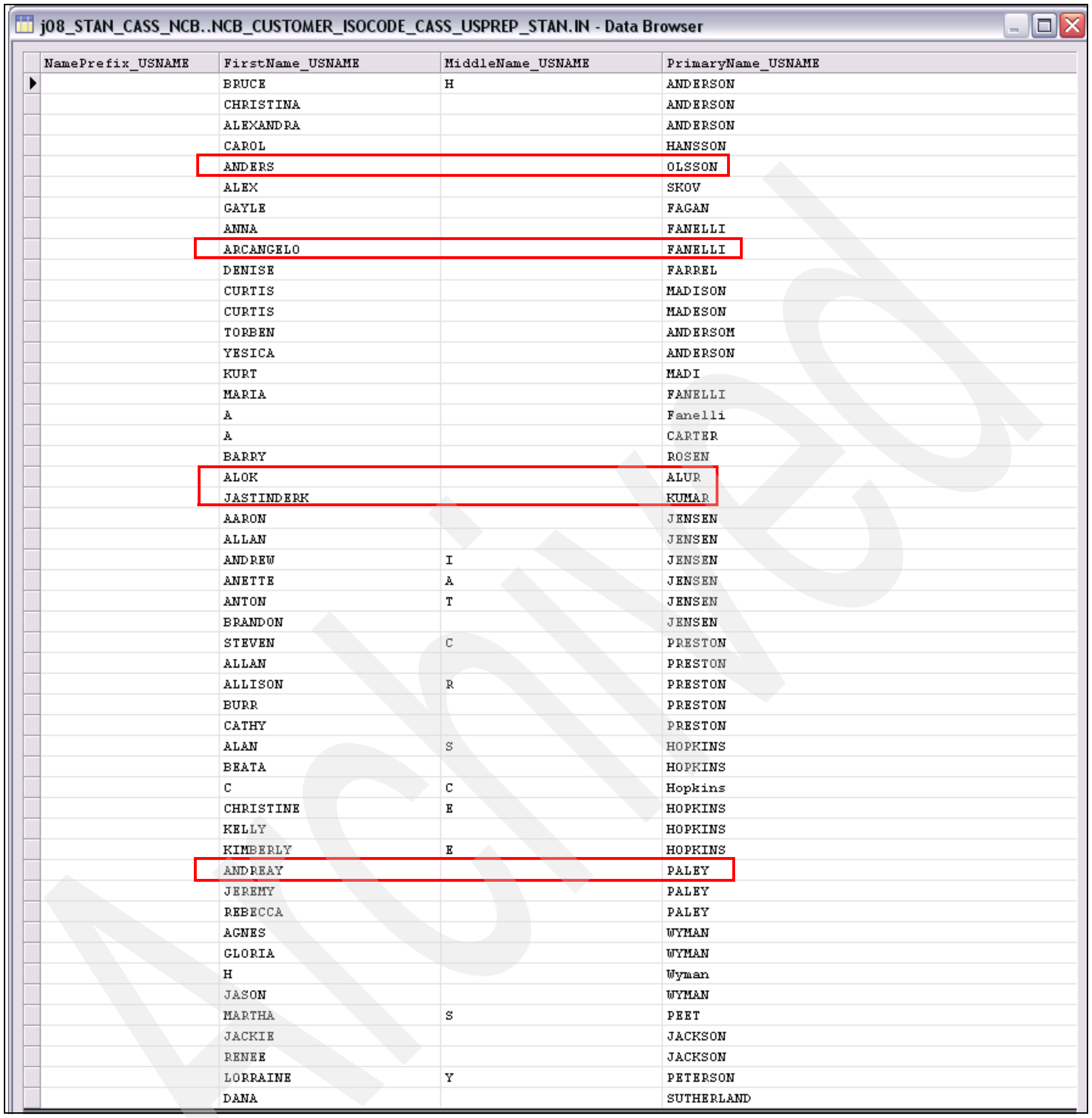
Classification: A - Abbreviations (Misspellings)

Comparison Threshold:

Override Summary

Input Token:	Standard Form:	Classification:	Tolerance:
KERR	KERR	F	
ANDERS	ANDERS	F	
TORBEN	TORBEN	F	
ALOK	ALOK	F	
JASTINDERK	JASTINDERK	F	
ANDREAY	ANDREAY	F	
ARCANGELO	ARCANGELO	F	

Figure 2-307 Create j09a_INVCC_CUSTOMER_STAN_CASS_NCB 10/11


j08_STAN_CASS_NCB..NCB_CUSTOMER_ISOCODE_CASS_USPREP_STAN.IN - Data Browser

NamePrefix_USNAME	FirstName_USNAME	MiddleName_USNAME	PrimaryName_USNAME
	BRUCE	H	ANDERSON
	CHRISTINA		ANDERSON
	ALEXANDRA		ANDERSON
	CAROL		HANSSON
	ANDERS		OLSSON
	ALEX		SKOV
	GAYLE		FAGAN
	ANNA		FANELLI
	ARCANGELO		FANELLI
	DENISE		FARREL
	CURTIS		MADISON
	CURTIS		MADESON
	TORBEN		ANDERSOM
	YESICA		ANDERSON
	KURT		MADI
	MARIA		FANELLI
	A		Fanelli
	A		CARTER
	BARRY		ROSEN
	ALOK		ALUR
	JASTINDERK		KUMAR
	AARON		JENSEN
	ALLAN		JENSEN
	ANDREW	I	JENSEN
	ANETTE	A	JENSEN
	ANTON	T	JENSEN
	BRANDON		JENSEN
	STEVEN	C	PRESTON
	ALLAN		PRESTON
	ALLISON	R	PRESTON
	BURR		PRESTON
	CATHY		PRESTON
	ALAN	S	HOPKINS
	BEATA		HOPKINS
	C	C	Hopkins
	CHRISTINE	E	HOPKINS
	KELLY		HOPKINS
	KIMBERLY	E	HOPKINS
	ANDREAY		PALEY
	JEREMY		PALEY
	REBECCA		PALEY
	AGNES		WYMAN
	GLORIA		WYMAN
	H		Wyman
	JASON		WYMAN
	MARTHA	S	PEET
	JACKIE		JACKSON
	RENEE		JACKSON
	LORRAINE	Y	PETERSON
	DANA		SUTHERLAND

Figure 2-308 Create j09a_INVCC_CUSTOMER_STAN_CASS_NCB 11/11

j09b_INVCC_BCUSTOMER_STAN_CASS_NCB

In this step, we identify unhandled patterns and classifications in the previous step in the BCUSTOMER file. We run a series of Investigate stage with the character concatenate option using the *C* mask on the unhandled pattern column from the results of "j08_STAN_CASS_NCB" on page 763. The columns that we investigated correspond to the name, address, and area domains. No unhandled patterns are detected.

Figure 2-309 shows the various stages that are used in this job. It includes the data set that was created in “j08_STAN_CASS_NCB” on page 763, an Investigate stage with the character concatenate option, and a sequential file for the column frequency report. We modified the names of the stages as shown.

Because the configuration of this job is very similar to that described in “J09_INVCC_STAN_CUSTOMER” on page 253, we do not repeat it here.

After saving, compiling, and running this job (Figure 2-309 on page 772) on the BCUSTOMER file, the contents of the output of the investigate stage show no unhandled patterns as shown in Figure 2-310 on page 773 through Figure 2-312 on page 773.

Proceed now to unhandled patterns in the address domain in “j09b_INVCC_BCUSTOMER_STAN_CASS_NCB” on page 771.

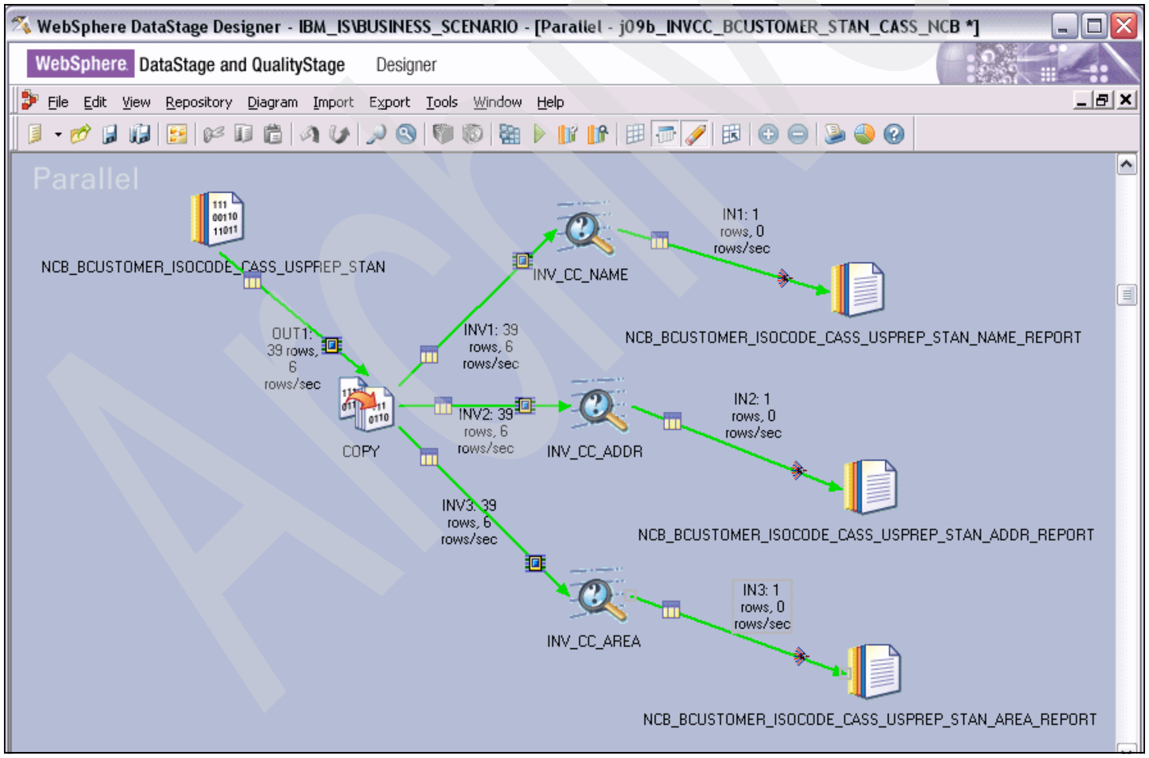
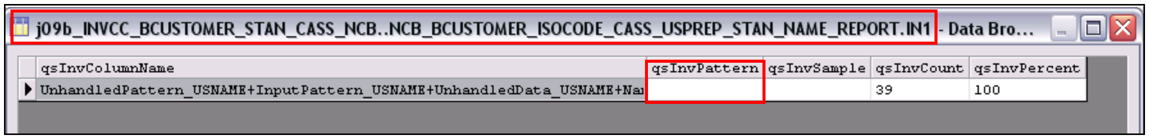

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j09b_INVCC_BCUSTOMER_STAN_CASS_NCB *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NCB_BCUSTOMER_ISOCODE_CASS_USPREP_STAN
OUT1: 39 rows, 6 rows/sec
COPY
INV1: 39 rows, 6 rows/sec
INV2: 39 rows, 6 rows/sec
INV3: 39 rows, 6 rows/sec
INV_CC_NAME
INV_CC_ADDR
INV_CC_AREA
IN1: 1 rows, 0 rows/sec
IN2: 1 rows, 0 rows/sec
IN3: 1 rows, 0 rows/sec
NCB_BCUSTOMER_ISOCODE_CASS_USPREP_STAN_NAME_REPORT
NCB_BCUSTOMER_ISOCODE_CASS_USPREP_STAN_ADDR_REPORT
NCB_BCUSTOMER_ISOCODE_CASS_USPREP_STAN_AREA_REPORT

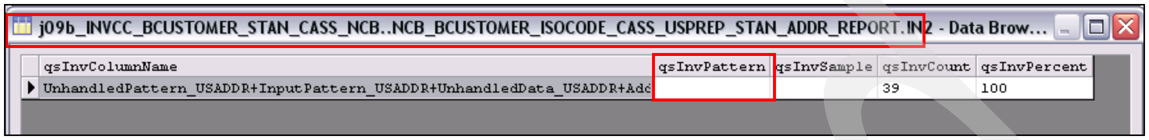
Figure 2-309 Create j09b_INVCC_BCUSTOMER_STAN_CASS_NCB 1/4



j09b_INVCC_BCUSTOMER_STAN_CASS_NCB..NCB_BCUSTOMER_ISOCODE_CASS_USPREP_STAN_NAME_REPORT.IN1 - Data Bro...

	qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
▶	UnhandledPattern_USNAME+InputPattern_USNAME+UnhandledData_USNAME+Na			39	100

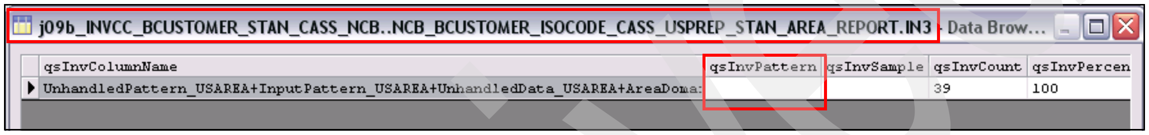
Figure 2-310   Create j09b_INVCC_BCUSTOMER_STAN_CASS_NCB 2/4



j09b_INVCC_BCUSTOMER_STAN_CASS_NCB..NCB_BCUSTOMER_ISOCODE_CASS_USPREP_STAN_ADDR_REPORT.IN2 - Data Brow...

	qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercent
▶	UnhandledPattern_USADDR+InputPattern_USADDR+UnhandledData_USADDR+Add			39	100

Figure 2-311   Create j09b_INVCC_BCUSTOMER_STAN_CASS_NCB 3/4



j09b_INVCC_BCUSTOMER_STAN_CASS_NCB..NCB_BCUSTOMER_ISOCODE_CASS_USPREP_STAN_AREA_REPORT.IN3 - Data Brow...

	qsInvColumnName	qsInvPattern	qsInvSample	qsInvCount	qsInvPercen
▶	UnhandledPattern_USAREA+InputPattern_USAREA+UnhandledData_USAREA+AreaDoma:			39	100

Figure 2-312   Create j09b_INVCC_BCUSTOMER_STAN_CASS_NCB 4/4

j10_PREPARE_NCB_DATA_FOR_FUNNEL

After the unhandled patterns are handled, we need to merge the standardized name and address file of the CUSTOMER and BCUSTOMER in order to identify duplicates (with data from the North American Bank's standardized name and addresses from their core and non-core services as described in 2.6.1, "Cleansing North American Bank's core and non-core services" on page 508), and to survive the "best" data into the CRM system as described in 2.6.3, "Matching and surviving Northern California Bank and Northern California Bank information" on page 783.

Prior to merging the two files that were created in "j08_STAN_CASS_NCB" on page 763, we need to prepare the two files for merging in the Funnel stage by ensuring that all the files have the same number of columns and same names of columns using a Transformer stage.

Figure 2-313 on page 774 shows the various stages that are used in this job. It includes the data sets that were created in "j08_STAN_CASS_NCB" on page 763, a Transformer stage, and an output Data Set stage for each Transformer stage. We modified the names of the stages as shown.

Figure 2-314 on page 775 through Figure 2-316 on page 777 show the PICK_COLUMNS1 - Transformer windows that pick the appropriate columns from the CUSTOMER file that was created in "j08_STAN_CASS_NCB" on page 763. Certain columns are added (that exist in the other files to be merged with such as BANKID, NCB_CORE_ID and NCB_NON_CORE_ID) with an empty string or other value as highlighted.

Figure 2-317 on page 778 through Figure 2-320 on page 781 show the corresponding PICK_COLUMNS2 - Transformer windows that pick the appropriate columns from the BCUSTOMER file that was created in "j08_STAN_CASS_NCB" on page 763. Here again, certain columns are added (such as HOMEPHONE and CELLPHONE) with an empty string or other value as highlighted.

After compiling and running this job (not shown here), proceed to the "j11_FUNNEL_NCB_DATA" on page 782 step.

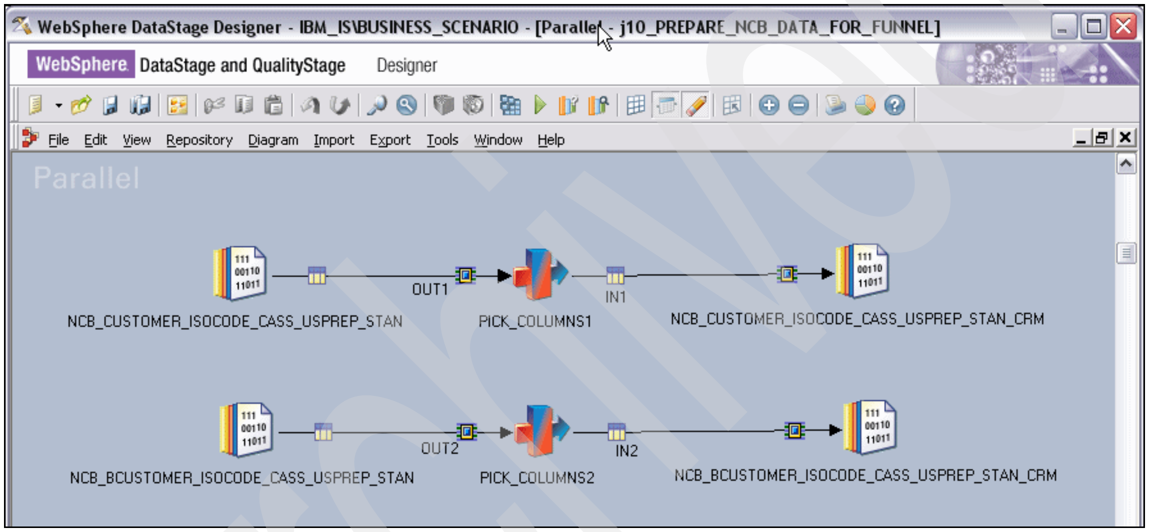
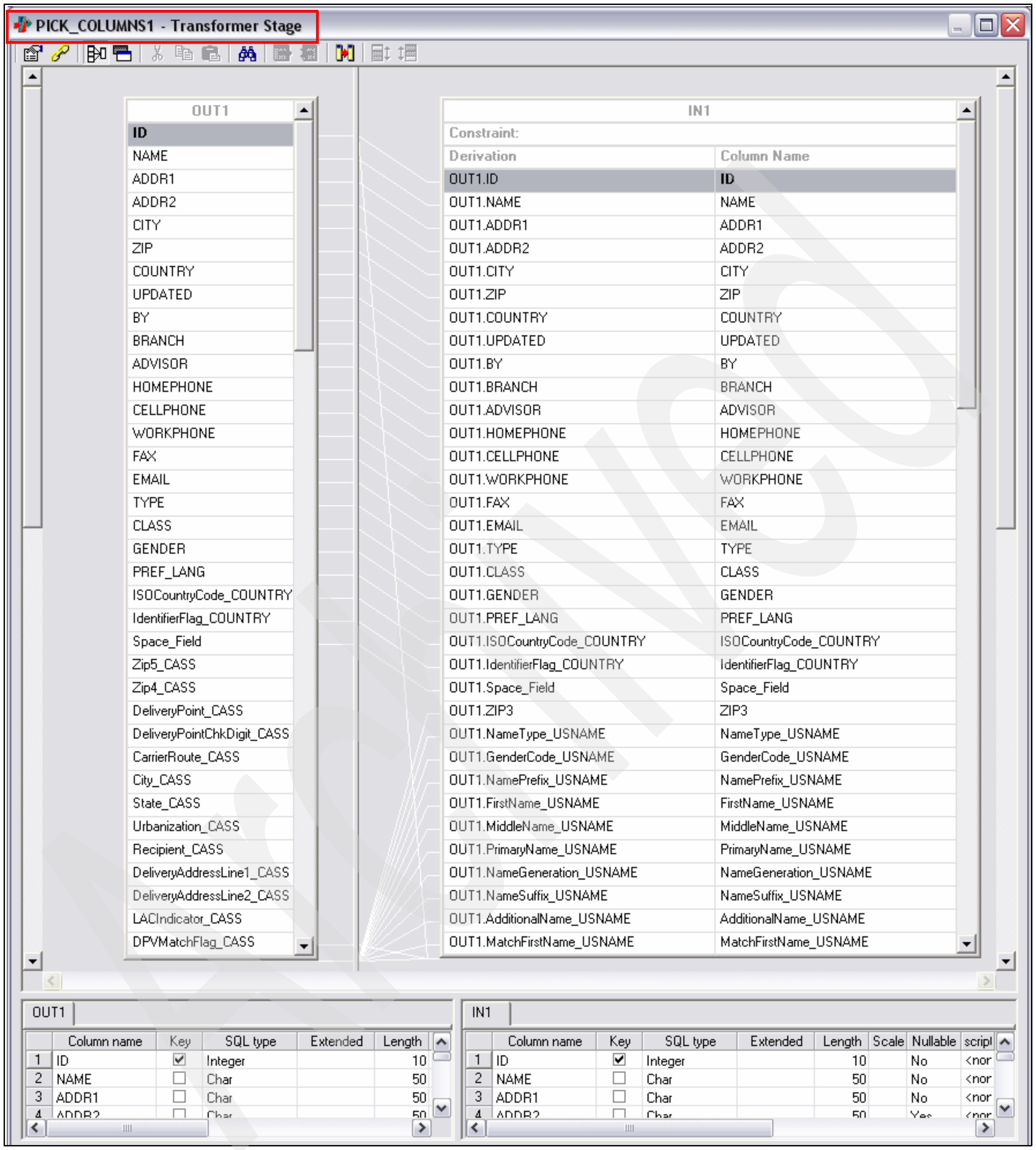

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j10_PREPARE_NCB_DATA_FOR_FUNNEL]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NCB_CUSTOMER_ISOCODE_CASS_USPREP_STAN
OUT1
PICK_COLUMNS1
IN1
NCB_CUSTOMER_ISOCODE_CASS_USPREP_STAN_CRM
NCB_BCUSTOMER_ISOCODE_CASS_USPREP_STAN
OUT2
PICK_COLUMNS2
IN2
NCB_BCUSTOMER_ISOCODE_CASS_USPREP_STAN_CRM

Figure 2-313 Create j10_PREPARE_NCB_DATA_FOR_FUNNEL 1/8


PICK_COLUMNS1 - Transformer Stage

OUT1
ID
NAME
ADDR1
ADDR2
CITY
ZIP
COUNTRY
UPDATED
BY
BRANCH
ADVISOR
HOMEPHONE
CELLPHONE
WORKPHONE
FAX
EMAIL
TYPE
CLASS
GENDER
PREF_LANG
ISOCountryCode_COUNTRY
IdentifierFlag_COUNTRY
Space_Field
Zip5_CASS
Zip4_CASS
DeliveryPoint_CASS
DeliveryPointChkDigit_CASS
CarrierRoute_CASS
City_CASS
State_CASS
Urbanization_CASS
Recipient_CASS
DeliveryAddressLine1_CASS
DeliveryAddressLine2_CASS
LACIndicator_CASS
DPVMatchFlag_CASS

IN1

Constraint:

Derivation	Column Name
OUT1.ID	ID
OUT1.NAME	NAME
OUT1.ADDR1	ADDR1
OUT1.ADDR2	ADDR2
OUT1.CITY	CITY
OUT1.ZIP	ZIP
OUT1.COUNTRY	COUNTRY
OUT1.UPDATED	UPDATED
OUT1.BY	BY
OUT1.BRANCH	BRANCH
OUT1.ADVISOR	ADVISOR
OUT1.HOMEPHONE	HOMEPHONE
OUT1.CELLPHONE	CELLPHONE
OUT1.WORKPHONE	WORKPHONE
OUT1.FAX	FAX
OUT1.EMAIL	EMAIL
OUT1.TYPE	TYPE
OUT1.CLASS	CLASS
OUT1.GENDER	GENDER
OUT1.PREF_LANG	PREF_LANG
OUT1.ISOCountryCode_COUNTRY	ISOCountryCode_COUNTRY
OUT1.IdentifierFlag_COUNTRY	IdentifierFlag_COUNTRY
OUT1.Space_Field	Space_Field
OUT1.ZIP3	ZIP3
OUT1.NameType_USNAME	NameType_USNAME
OUT1.GenderCode_USNAME	GenderCode_USNAME
OUT1.NamePrefix_USNAME	NamePrefix_USNAME
OUT1.FirstName_USNAME	FirstName_USNAME
OUT1.MiddleName_USNAME	MiddleName_USNAME
OUT1.PrimaryName_USNAME	PrimaryName_USNAME
OUT1.NameGeneration_USNAME	NameGeneration_USNAME
OUT1.NameSuffix_USNAME	NameSuffix_USNAME
OUT1.AdditionalName_USNAME	AdditionalName_USNAME
OUT1.MatchFirstName_USNAME	MatchFirstName_USNAME

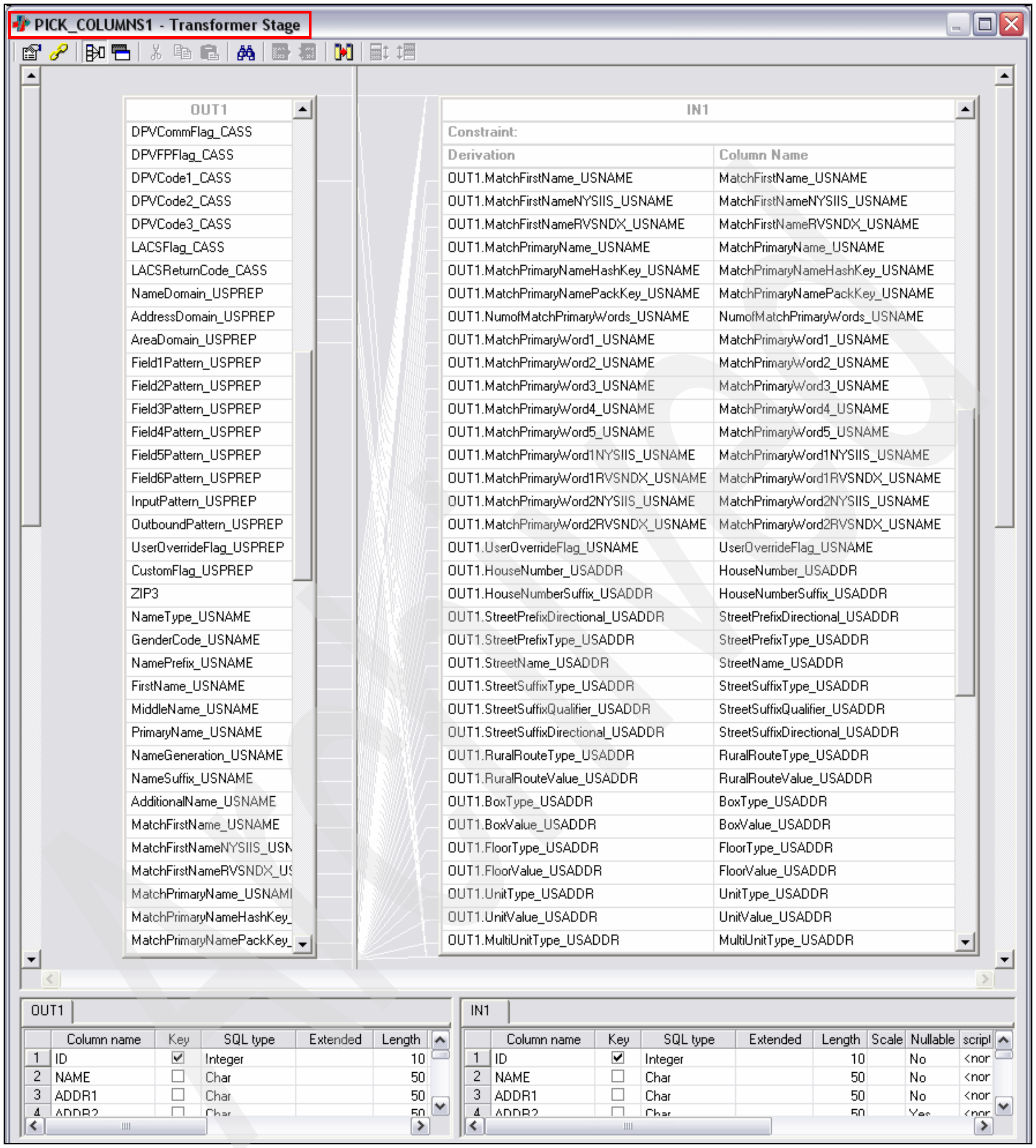
OUT1

	Column name	Key	SQL type	Extended	Length
1	ID	✓	Integer		10
2	NAME	☐	Char		50
3	ADDR1	☐	Char		50
4	ADDR2	☐	Char		50

IN1

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	script
1	ID	✓	Integer		10		No	<nor
2	NAME	☐	Char		50		No	<nor
3	ADDR1	☐	Char		50		No	<nor
4	ADDR2	☐	Char		50		Yes	<nor

Figure 2-314 Create j10_PREPARE_NCB_DATA_FOR_FUNNEL 2/8


PICK_COLUMNS1 - Transformer Stage

OUT1
DPVCommFlag_CASS
DPVFPFlag_CASS
DPVCode1_CASS
DPVCode2_CASS
DPVCode3_CASS
LACSFlag_CASS
LACSReturnCode_CASS
NameDomain_USPREP
AddressDomain_USPREP
AreaDomain_USPREP
Field1Pattern_USPREP
Field2Pattern_USPREP
Field3Pattern_USPREP
Field4Pattern_USPREP
Field5Pattern_USPREP
Field6Pattern_USPREP
InputPattern_USPREP
OutboundPattern_USPREP
UserOverrideFlag_USPREP
CustomFlag_USPREP
ZIP3
NameType_USNAME
GenderCode_USNAME
NamePrefix_USNAME
FirstName_USNAME
MiddleName_USNAME
PrimaryName_USNAME
NameGeneration_USNAME
NameSuffix_USNAME
AdditionalName_USNAME
MatchFirstName_USNAME
MatchFirstNameNYSIIS_USN
MatchFirstNameRVSNDX_US
MatchPrimaryName_USNAMI
MatchPrimaryNameHashKey_
MatchPrimaryNamePackKey_

IN1

Constraint:

Derivation	Column Name
OUT1.MatchFirstName_USNAME	MatchFirstName_USNAME
OUT1.MatchFirstNameNYSIIS_USNAME	MatchFirstNameNYSIIS_USNAME
OUT1.MatchFirstNameRVSNDX_USNAME	MatchFirstNameRVSNDX_USNAME
OUT1.MatchPrimaryName_USNAME	MatchPrimaryName_USNAME
OUT1.MatchPrimaryNameHashKey_USNAME	MatchPrimaryNameHashKey_USNAME
OUT1.MatchPrimaryNamePackKey_USNAME	MatchPrimaryNamePackKey_USNAME
OUT1.NumofMatchPrimaryWords_USNAME	NumofMatchPrimaryWords_USNAME
OUT1.MatchPrimaryWord1_USNAME	MatchPrimaryWord1_USNAME
OUT1.MatchPrimaryWord2_USNAME	MatchPrimaryWord2_USNAME
OUT1.MatchPrimaryWord3_USNAME	MatchPrimaryWord3_USNAME
OUT1.MatchPrimaryWord4_USNAME	MatchPrimaryWord4_USNAME
OUT1.MatchPrimaryWord5_USNAME	MatchPrimaryWord5_USNAME
OUT1.MatchPrimaryWord1NYSIIS_USNAME	MatchPrimaryWord1NYSIIS_USNAME
OUT1.MatchPrimaryWord1RVSNDX_USNAME	MatchPrimaryWord1RVSNDX_USNAME
OUT1.MatchPrimaryWord2NYSIIS_USNAME	MatchPrimaryWord2NYSIIS_USNAME
OUT1.MatchPrimaryWord2RVSNDX_USNAME	MatchPrimaryWord2RVSNDX_USNAME
OUT1.UserOverrideFlag_USNAME	UserOverrideFlag_USNAME
OUT1.HouseNumber_USADDR	HouseNumber_USADDR
OUT1.HouseNumberSuffix_USADDR	HouseNumberSuffix_USADDR
OUT1.StreetPrefixDirectional_USADDR	StreetPrefixDirectional_USADDR
OUT1.StreetPrefixType_USADDR	StreetPrefixType_USADDR
OUT1.StreetName_USADDR	StreetName_USADDR
OUT1.StreetSuffixType_USADDR	StreetSuffixType_USADDR
OUT1.StreetSuffixQualifier_USADDR	StreetSuffixQualifier_USADDR
OUT1.StreetSuffixDirectional_USADDR	StreetSuffixDirectional_USADDR
OUT1.RuralRouteType_USADDR	RuralRouteType_USADDR
OUT1.RuralRouteValue_USADDR	RuralRouteValue_USADDR
OUT1.BoxType_USADDR	BoxType_USADDR
OUT1.BoxValue_USADDR	BoxValue_USADDR
OUT1.FloorType_USADDR	FloorType_USADDR
OUT1.FloorValue_USADDR	FloorValue_USADDR
OUT1.UnitType_USADDR	UnitType_USADDR
OUT1.UnitValue_USADDR	UnitValue_USADDR
OUT1.MultiUnitType_USADDR	MultiUnitType_USADDR

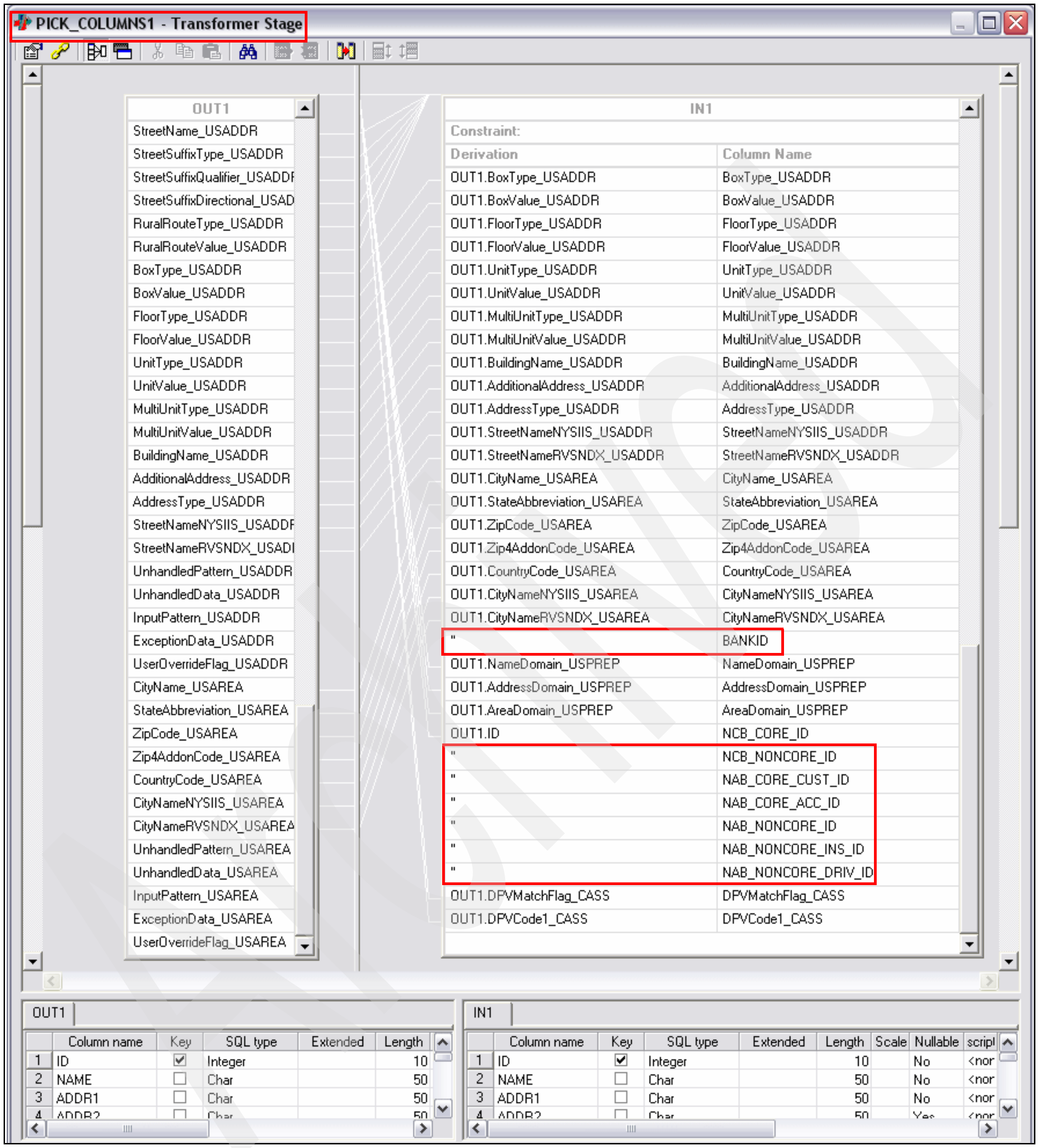
OUT1

	Column name	Key	SQL type	Extended	Length
1	ID	✓	Integer		10
2	NAME	☐	Char		50
3	ADDR1	☐	Char		50
4	ADDR2	☐	Char		50

IN1

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripl
1	ID	✓	Integer		10		No	<nor
2	NAME	☐	Char		50		No	<nor
3	ADDR1	☐	Char		50		No	<nor
4	ADDR2	☐	Char		50		Yes	<nor

Figure 2-315 Create j10_PREPARE_NCB_DATA_FOR_FUNNEL 3/8


PICK_COLUMNS1 - Transformer Stage

OUT1
StreetName_USADDR
StreetSuffixType_USADDR
StreetSuffixQualifier_USADDF
StreetSuffixDirectional_USAD
RuralRouteType_USADDR
RuralRouteValue_USADDR
BoxType_USADDR
BoxValue_USADDR
FloorType_USADDR
FloorValue_USADDR
UnitType_USADDR
UnitValue_USADDR
MultiUnitType_USADDR
MultiUnitValue_USADDR
BuildingName_USADDR
AdditionalAddress_USADDR
AddressType_USADDR
StreetNameNYSIIS_USADDF
StreetNameRVSNDX_USADI
UnhandledPattern_USADDR
UnhandledData_USADDR
InputPattern_USADDR
ExceptionData_USADDR
UserOverrideFlag_USADDR
CityName_USAREA
StateAbbreviation_USAREA
ZipCode_USAREA
Zip4AddonCode_USAREA
CountryCode_USAREA
CityNameNYSIIS_USAREA
CityNameRVSNDX_USAREA
UnhandledPattern_USAREA
UnhandledData_USAREA
InputPattern_USAREA
ExceptionData_USAREA
UserOverrideFlag_USAREA

IN1

Constraint:

Derivation	Column Name
OUT1.BoxType_USADDR	BoxType_USADDR
OUT1.BoxValue_USADDR	BoxValue_USADDR
OUT1.FloorType_USADDR	FloorType_USADDR
OUT1.FloorValue_USADDR	FloorValue_USADDR
OUT1.UnitType_USADDR	UnitType_USADDR
OUT1.UnitValue_USADDR	UnitValue_USADDR
OUT1.MultiUnitType_USADDR	MultiUnitType_USADDR
OUT1.MultiUnitValue_USADDR	MultiUnitValue_USADDR
OUT1.BuildingName_USADDR	BuildingName_USADDR
OUT1.AdditionalAddress_USADDR	AdditionalAddress_USADDR
OUT1.AddressType_USADDR	AddressType_USADDR
OUT1.StreetNameNYSIIS_USADDR	StreetNameNYSIIS_USADDR
OUT1.StreetNameRVSNDX_USADDR	StreetNameRVSNDX_USADDR
OUT1.CityName_USAREA	CityName_USAREA
OUT1.StateAbbreviation_USAREA	StateAbbreviation_USAREA
OUT1.ZipCode_USAREA	ZipCode_USAREA
OUT1.Zip4AddonCode_USAREA	Zip4AddonCode_USAREA
OUT1.CountryCode_USAREA	CountryCode_USAREA
OUT1.CityNameNYSIIS_USAREA	CityNameNYSIIS_USAREA
OUT1.CityNameRVSNDX_USAREA	CityNameRVSNDX_USAREA
"	BANKID
OUT1.NameDomain_USPREP	NameDomain_USPREP
OUT1.AddressDomain_USPREP	AddressDomain_USPREP
OUT1.AreaDomain_USPREP	AreaDomain_USPREP
OUT1.ID	NCB_CORE_ID
"	NCB_NONCORE_ID
"	NAB_CORE_CUST_ID
"	NAB_CORE_ACC_ID
"	NAB_NONCORE_ID
"	NAB_NONCORE_INS_ID
"	NAB_NONCORE_DRIV_ID
OUT1.DPVMatchFlag_CASS	DPVMatchFlag_CASS
OUT1.DPVCode1_CASS	DPVCode1_CASS

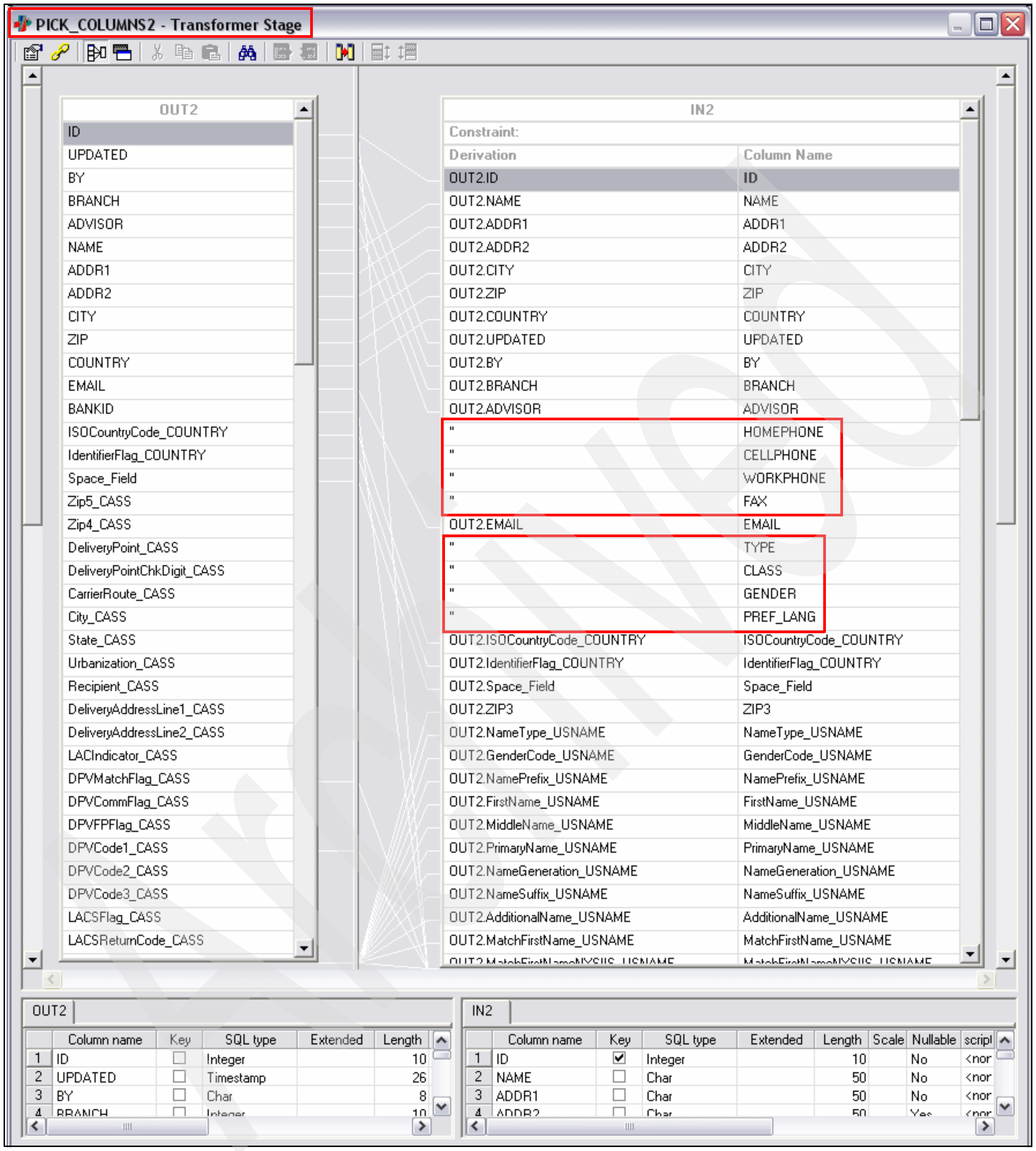
OUT1

	Column name	Key	SQL type	Extended	Length
1	ID	✓	Integer		10
2	NAME		Char		50
3	ADDR1		Char		50
4	ADDR2		Char		50

IN1

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scrip
1	ID	✓	Integer		10		No	<nor
2	NAME		Char		50		No	<nor
3	ADDR1		Char		50		No	<nor
4	ADDR2		Char		50		Yes	<nor

Figure 2-316 Create j10_PREPARE_NCB_DATA_FOR_FUNNEL 4/8



PICK_COLUMNS2 - Transformer Stage

OUT2
ID
UPDATED
BY
BRANCH
ADVISOR
NAME
ADDR1
ADDR2
CITY
ZIP
COUNTRY
EMAIL
BANKID
ISOCountryCode_COUNTRY
IdentifierFlag_COUNTRY
Space_Field
Zip5_CASS
Zip4_CASS
DeliveryPoint_CASS
DeliveryPointChkDigit_CASS
CarrierRoute_CASS
City_CASS
State_CASS
Urbanization_CASS
Recipient_CASS
DeliveryAddressLine1_CASS
DeliveryAddressLine2_CASS
LACIndicator_CASS
DPVMatchFlag_CASS
DPVCommFlag_CASS
DPVFPFlag_CASS
DPVCode1_CASS
DPVCode2_CASS
DPVCode3_CASS
LACSFlag_CASS
LACSReturnCode_CASS

IN2

Constraint:

Derivation	Column Name
OUT2.ID	ID
OUT2.NAME	NAME
OUT2.ADDR1	ADDR1
OUT2.ADDR2	ADDR2
OUT2.CITY	CITY
OUT2.ZIP	ZIP
OUT2.COUNTRY	COUNTRY
OUT2.UPDATED	UPDATED
OUT2.BY	BY
OUT2.BRANCH	BRANCH
OUT2.ADVISOR	ADVISOR
"	HOMEPHONE
"	CELLPHONE
"	WORKPHONE
"	FAX
OUT2.EMAIL	EMAIL
"	TYPE
"	CLASS
"	GENDER
"	PREF_LANG
OUT2.ISOCountryCode_COUNTRY	ISOCountryCode_COUNTRY
OUT2.IdentifierFlag_COUNTRY	IdentifierFlag_COUNTRY
OUT2.Space_Field	Space_Field
OUT2.ZIP3	ZIP3
OUT2.NameType_USNAME	NameType_USNAME
OUT2.GenderCode_USNAME	GenderCode_USNAME
OUT2.NamePrefix_USNAME	NamePrefix_USNAME
OUT2.FirstName_USNAME	FirstName_USNAME
OUT2.MiddleName_USNAME	MiddleName_USNAME
OUT2.PrimaryName_USNAME	PrimaryName_USNAME
OUT2.NameGeneration_USNAME	NameGeneration_USNAME
OUT2.NameSuffix_USNAME	NameSuffix_USNAME
OUT2.AdditionalName_USNAME	AdditionalName_USNAME
OUT2.MatchFirstName_USNAME	MatchFirstName_USNAME
OUT2.MatchFirstNameNYSIIS_USNAME	MatchFirstNameNYSIIS_USNAME

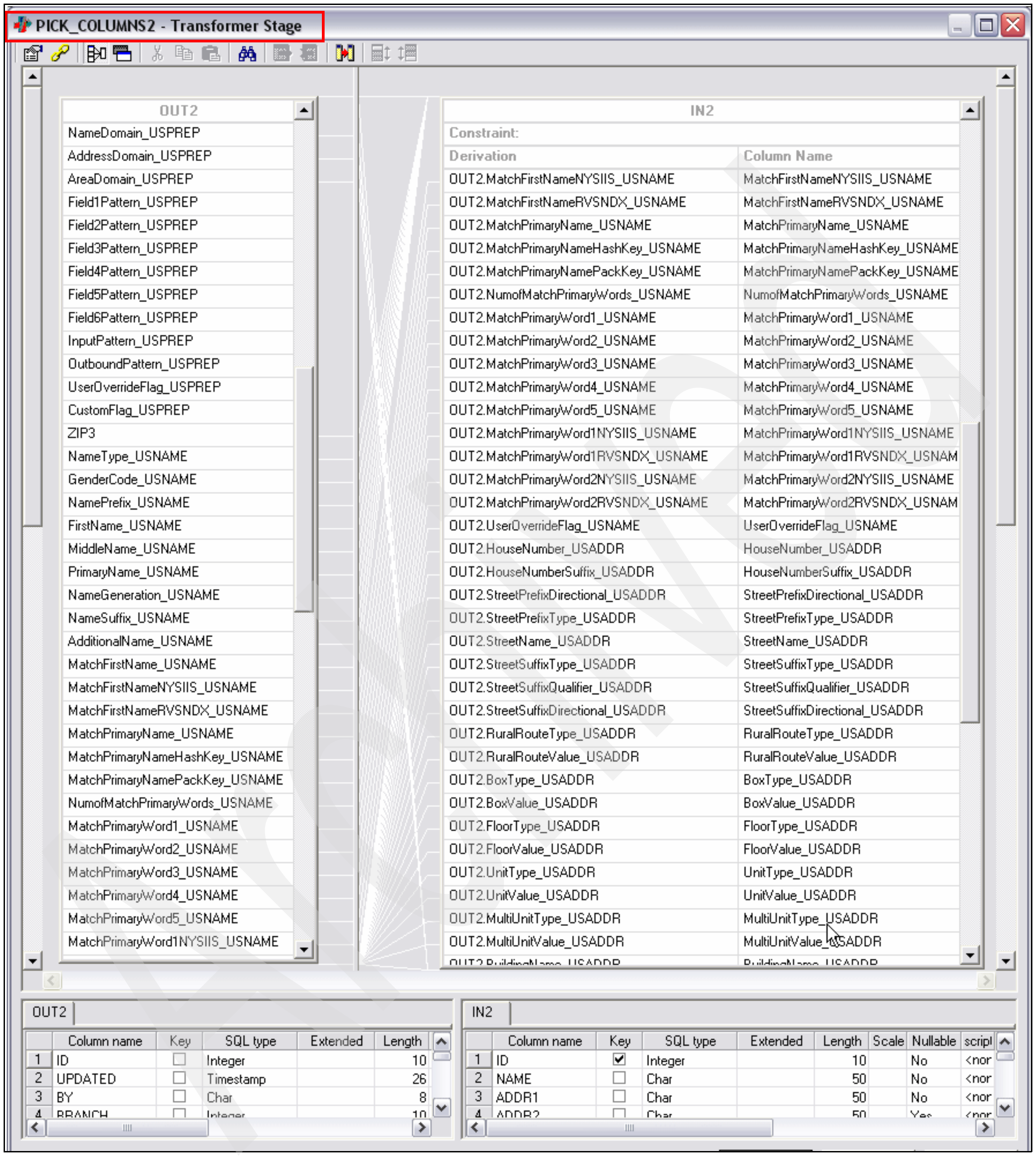
OUT2

	Column name	Key	SQL type	Extended	Length
1	ID	☐	Integer		10
2	UPDATED	☐	Timestamp		26
3	BY	☐	Char		8
4	BRANCH	☐	Integer		10

IN2

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripl
1	ID	☑	Integer		10		No	<nor
2	NAME	☐	Char		50		No	<nor
3	ADDR1	☐	Char		50		No	<nor
4	ADDR2	☐	Char		50		Yes	<nor

Figure 2-317   Create j10_PREPARE_NCB_DATA_FOR_FUNNEL 5/8



PICK_COLUMNS2 - Transformer Stage

OUT2
NameDomain_USPREP
AddressDomain_USPREP
AreaDomain_USPREP
Field1Pattern_USPREP
Field2Pattern_USPREP
Field3Pattern_USPREP
Field4Pattern_USPREP
Field5Pattern_USPREP
Field6Pattern_USPREP
InputPattern_USPREP
OutboundPattern_USPREP
UserOverrideFlag_USPREP
CustomFlag_USPREP
ZIP3
NameType_USNAME
GenderCode_USNAME
NamePrefix_USNAME
FirstName_USNAME
MiddleName_USNAME
PrimaryName_USNAME
NameGeneration_USNAME
NameSuffix_USNAME
AdditionalName_USNAME
MatchFirstName_USNAME
MatchFirstNameNYSIIS_USNAME
MatchFirstNameRVSNDX_USNAME
MatchPrimaryName_USNAME
MatchPrimaryNameHashKey_USNAME
MatchPrimaryNamePackKey_USNAME
NumofMatchPrimaryWords_USNAME
MatchPrimaryWord1_USNAME
MatchPrimaryWord2_USNAME
MatchPrimaryWord3_USNAME
MatchPrimaryWord4_USNAME
MatchPrimaryWord5_USNAME
MatchPrimaryWord1NYSIIS_USNAME

IN2

Constraint:

Derivation	Column Name
OUT2.MatchFirstNameNYSIIS_USNAME	MatchFirstNameNYSIIS_USNAME
OUT2.MatchFirstNameRVSNDX_USNAME	MatchFirstNameRVSNDX_USNAME
OUT2.MatchPrimaryName_USNAME	MatchPrimaryName_USNAME
OUT2.MatchPrimaryNameHashKey_USNAME	MatchPrimaryNameHashKey_USNAME
OUT2.MatchPrimaryNamePackKey_USNAME	MatchPrimaryNamePackKey_USNAME
OUT2.NumofMatchPrimaryWords_USNAME	NumofMatchPrimaryWords_USNAME
OUT2.MatchPrimaryWord1_USNAME	MatchPrimaryWord1_USNAME
OUT2.MatchPrimaryWord2_USNAME	MatchPrimaryWord2_USNAME
OUT2.MatchPrimaryWord3_USNAME	MatchPrimaryWord3_USNAME
OUT2.MatchPrimaryWord4_USNAME	MatchPrimaryWord4_USNAME
OUT2.MatchPrimaryWord5_USNAME	MatchPrimaryWord5_USNAME
OUT2.MatchPrimaryWord1NYSIIS_USNAME	MatchPrimaryWord1NYSIIS_USNAME
OUT2.MatchPrimaryWord1RVSNDX_USNAME	MatchPrimaryWord1RVSNDX_USNAM
OUT2.MatchPrimaryWord2NYSIIS_USNAME	MatchPrimaryWord2NYSIIS_USNAME
OUT2.MatchPrimaryWord2RVSNDX_USNAME	MatchPrimaryWord2RVSNDX_USNAM
OUT2.UserOverrideFlag_USNAME	UserOverrideFlag_USNAME
OUT2.HouseNumber_USADDR	HouseNumber_USADDR
OUT2.HouseNumberSuffix_USADDR	HouseNumberSuffix_USADDR
OUT2.StreetPrefixDirectional_USADDR	StreetPrefixDirectional_USADDR
OUT2.StreetPrefixType_USADDR	StreetPrefixType_USADDR
OUT2.StreetName_USADDR	StreetName_USADDR
OUT2.StreetSuffixType_USADDR	StreetSuffixType_USADDR
OUT2.StreetSuffixQualifier_USADDR	StreetSuffixQualifier_USADDR
OUT2.StreetSuffixDirectional_USADDR	StreetSuffixDirectional_USADDR
OUT2.RuralRouteType_USADDR	RuralRouteType_USADDR
OUT2.RuralRouteValue_USADDR	RuralRouteValue_USADDR
OUT2.BoxType_USADDR	BoxType_USADDR
OUT2.BoxValue_USADDR	BoxValue_USADDR
OUT2.FloorType_USADDR	FloorType_USADDR
OUT2.FloorValue_USADDR	FloorValue_USADDR
OUT2.UnitType_USADDR	UnitType_USADDR
OUT2.UnitValue_USADDR	UnitValue_USADDR
OUT2.MultiUnitType_USADDR	MultiUnitType_USADDR
OUT2.MultiUnitValue_USADDR	MultiUnitValue_USADDR

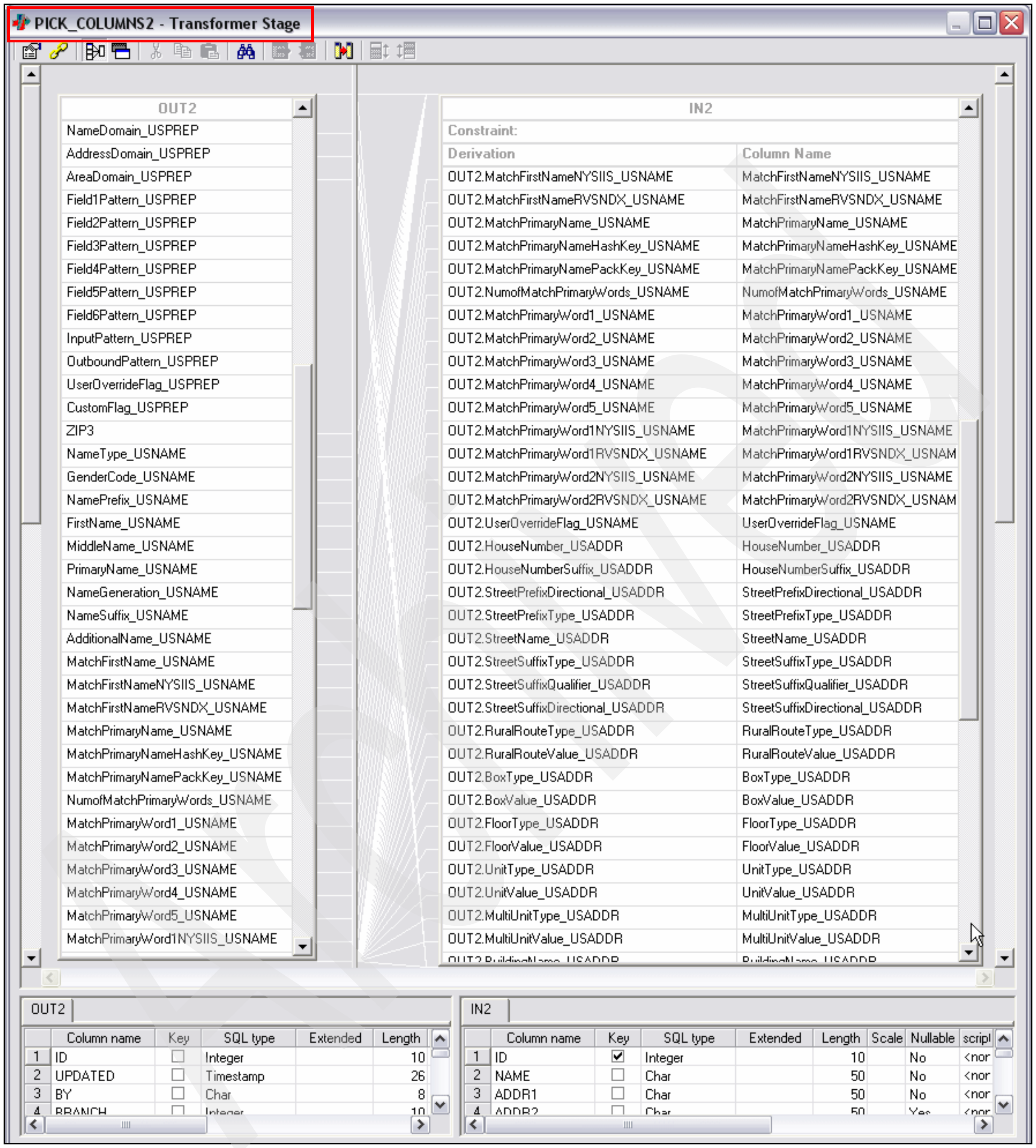
OUT2

	Column name	Key	SQL type	Extended	Length
1	ID	☐	Integer		10
2	UPDATED	☐	Timestamp		26
3	BY	☐	Char		8
4	BRANCH	☐	Integer		10

IN2

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripl
1	ID	✓	Integer		10		No	<nor
2	NAME	☐	Char		50		No	<nor
3	ADDR1	☐	Char		50		No	<nor
4	ADDR2	☐	Char		50		Yes	<nor

Figure 2-318 Create j10_PREPARE_NCB_DATA_FOR_FUNNEL 6/8


PICK_COLUMNS2 - Transformer Stage

OUT2
NameDomain_USPREP
AddressDomain_USPREP
AreaDomain_USPREP
Field1Pattern_USPREP
Field2Pattern_USPREP
Field3Pattern_USPREP
Field4Pattern_USPREP
Field5Pattern_USPREP
Field6Pattern_USPREP
InputPattern_USPREP
OutboundPattern_USPREP
UserOverrideFlag_USPREP
CustomFlag_USPREP
ZIP3
NameType_USNAME
GenderCode_USNAME
NamePrefix_USNAME
FirstName_USNAME
MiddleName_USNAME
PrimaryName_USNAME
NameGeneration_USNAME
NameSuffix_USNAME
AdditionalName_USNAME
MatchFirstName_USNAME
MatchFirstNameNYSIIS_USNAME
MatchFirstNameRVSNDX_USNAME
MatchPrimaryName_USNAME
MatchPrimaryNameHashKey_USNAME
MatchPrimaryNamePackKey_USNAME
NumofMatchPrimaryWords_USNAME
MatchPrimaryWord1_USNAME
MatchPrimaryWord2_USNAME
MatchPrimaryWord3_USNAME
MatchPrimaryWord4_USNAME
MatchPrimaryWord5_USNAME
MatchPrimaryWord1NYSIIS_USNAME

IN2

Constraint:

Derivation	Column Name
OUT2.MatchFirstNameNYSIIS_USNAME	MatchFirstNameNYSIIS_USNAME
OUT2.MatchFirstNameRVSNDX_USNAME	MatchFirstNameRVSNDX_USNAME
OUT2.MatchPrimaryName_USNAME	MatchPrimaryName_USNAME
OUT2.MatchPrimaryNameHashKey_USNAME	MatchPrimaryNameHashKey_USNAME
OUT2.MatchPrimaryNamePackKey_USNAME	MatchPrimaryNamePackKey_USNAME
OUT2.NumofMatchPrimaryWords_USNAME	NumofMatchPrimaryWords_USNAME
OUT2.MatchPrimaryWord1_USNAME	MatchPrimaryWord1_USNAME
OUT2.MatchPrimaryWord2_USNAME	MatchPrimaryWord2_USNAME
OUT2.MatchPrimaryWord3_USNAME	MatchPrimaryWord3_USNAME
OUT2.MatchPrimaryWord4_USNAME	MatchPrimaryWord4_USNAME
OUT2.MatchPrimaryWord5_USNAME	MatchPrimaryWord5_USNAME
OUT2.MatchPrimaryWord1NYSIIS_USNAME	MatchPrimaryWord1NYSIIS_USNAME
OUT2.MatchPrimaryWord1RVSNDX_USNAME	MatchPrimaryWord1RVSNDX_USNAM
OUT2.MatchPrimaryWord2NYSIIS_USNAME	MatchPrimaryWord2NYSIIS_USNAME
OUT2.MatchPrimaryWord2RVSNDX_USNAME	MatchPrimaryWord2RVSNDX_USNAM
OUT2.UserOverrideFlag_USNAME	UserOverrideFlag_USNAME
OUT2.HouseNumber_USADDR	HouseNumber_USADDR
OUT2.HouseNumberSuffix_USADDR	HouseNumberSuffix_USADDR
OUT2.StreetPrefixDirectional_USADDR	StreetPrefixDirectional_USADDR
OUT2.StreetPrefixType_USADDR	StreetPrefixType_USADDR
OUT2.StreetName_USADDR	StreetName_USADDR
OUT2.StreetSuffixType_USADDR	StreetSuffixType_USADDR
OUT2.StreetSuffixQualifier_USADDR	StreetSuffixQualifier_USADDR
OUT2.StreetSuffixDirectional_USADDR	StreetSuffixDirectional_USADDR
OUT2.RuralRouteType_USADDR	RuralRouteType_USADDR
OUT2.RuralRouteValue_USADDR	RuralRouteValue_USADDR
OUT2.BoxType_USADDR	BoxType_USADDR
OUT2.BoxValue_USADDR	BoxValue_USADDR
OUT2.FloorType_USADDR	FloorType_USADDR
OUT2.FloorValue_USADDR	FloorValue_USADDR
OUT2.UnitType_USADDR	UnitType_USADDR
OUT2.UnitValue_USADDR	UnitValue_USADDR
OUT2.MultiUnitType_USADDR	MultiUnitType_USADDR
OUT2.MultiUnitValue_USADDR	MultiUnitValue_USADDR
OUT2.BuildingName_USADDR	BuildingName_USADDR

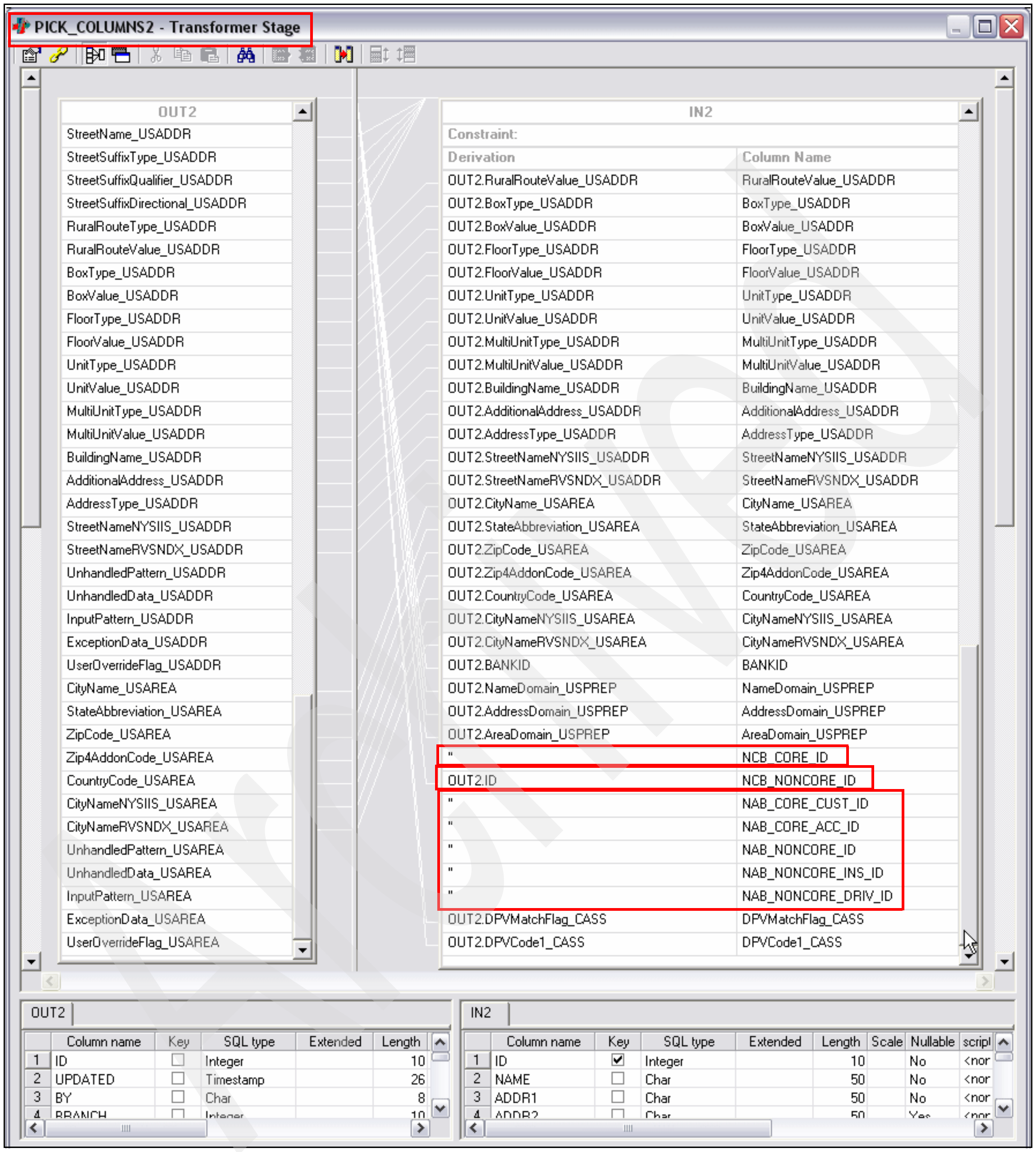
OUT2

	Column name	Key	SQL type	Extended	Length
1	ID	☐	Integer		10
2	UPDATED	☐	Timestamp		26
3	BY	☐	Char		8
4	BRANCH	☐	Integer		10

IN2

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripl
1	ID	☑	Integer		10		No	<nor
2	NAME	☐	Char		50		No	<nor
3	ADDR1	☐	Char		50		No	<nor
4	ADDR2	☐	Char		50		Yes	<nor

Figure 2-319 Create j10_PREPARE_NCB_DATA_FOR_FUNNEL 7/8


PICK_COLUMNS2 - Transformer Stage

OUT2
StreetName_USADDR
StreetSuffixType_USADDR
StreetSuffixQualifier_USADDR
StreetSuffixDirectional_USADDR
RuralRouteType_USADDR
RuralRouteValue_USADDR
BoxType_USADDR
BoxValue_USADDR
FloorType_USADDR
FloorValue_USADDR
UnitType_USADDR
UnitValue_USADDR
MultiUnitType_USADDR
MultiUnitValue_USADDR
BuildingName_USADDR
AdditionalAddress_USADDR
AddressType_USADDR
StreetNameNYSIIS_USADDR
StreetNameRVSNDX_USADDR
UnhandledPattern_USADDR
UnhandledData_USADDR
InputPattern_USADDR
ExceptionData_USADDR
UserOverrideFlag_USADDR
CityName_USAREA
StateAbbreviation_USAREA
ZipCode_USAREA
Zip4AddonCode_USAREA
CountryCode_USAREA
CityNameNYSIIS_USAREA
CityNameRVSNDX_USAREA
UnhandledPattern_USAREA
UnhandledData_USAREA
InputPattern_USAREA
ExceptionData_USAREA
UserOverrideFlag_USAREA

IN2

Constraint:

Derivation	Column Name
OUT2.RuralRouteValue_USADDR	RuralRouteValue_USADDR
OUT2.BoxType_USADDR	BoxType_USADDR
OUT2.BoxValue_USADDR	BoxValue_USADDR
OUT2.FloorType_USADDR	FloorType_USADDR
OUT2.FloorValue_USADDR	FloorValue_USADDR
OUT2.UnitType_USADDR	UnitType_USADDR
OUT2.UnitValue_USADDR	UnitValue_USADDR
OUT2.MultiUnitType_USADDR	MultiUnitType_USADDR
OUT2.MultiUnitValue_USADDR	MultiUnitValue_USADDR
OUT2.BuildingName_USADDR	BuildingName_USADDR
OUT2.AdditionalAddress_USADDR	AdditionalAddress_USADDR
OUT2.AddressType_USADDR	AddressType_USADDR
OUT2.StreetNameNYSIIS_USADDR	StreetNameNYSIIS_USADDR
OUT2.StreetNameRVSNDX_USADDR	StreetNameRVSNDX_USADDR
OUT2.CityName_USAREA	CityName_USAREA
OUT2.StateAbbreviation_USAREA	StateAbbreviation_USAREA
OUT2.ZipCode_USAREA	ZipCode_USAREA
OUT2.Zip4AddonCode_USAREA	Zip4AddonCode_USAREA
OUT2.CountryCode_USAREA	CountryCode_USAREA
OUT2.CityNameNYSIIS_USAREA	CityNameNYSIIS_USAREA
OUT2.CityNameRVSNDX_USAREA	CityNameRVSNDX_USAREA
OUT2.BANKID	BANKID
OUT2.NameDomain_USPREP	NameDomain_USPREP
OUT2.AddressDomain_USPREP	AddressDomain_USPREP
OUT2.AreaDomain_USPREP	AreaDomain_USPREP
"	NCB_CORE_ID
OUT2.ID	NCB_NONCORE_ID
"	NAB_CORE_CUST_ID
"	NAB_CORE_ACC_ID
"	NAB_NONCORE_ID
"	NAB_NONCORE_INS_ID
"	NAB_NONCORE_DRIV_ID
OUT2.DPVMatchFlag_CASS	DPVMatchFlag_CASS
OUT2.DPVCode1_CASS	DPVCode1_CASS

OUT2

	Column name	Key	SQL type	Extended	Length
1	ID	☐	Integer		10
2	UPDATED	☐	Timestamp		26
3	BY	☐	Char		8
4	BRANCH	☐	Integer		10

IN2

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripl
1	ID	✓	Integer		10		No	<nor
2	NAME	☐	Char		50		No	<nor
3	ADDR1	☐	Char		50		No	<nor
4	ADDR2	☐	Char		50		Yes	<nor

Figure 2-320 Create j10_PREPARE_NCB_DATA_FOR_FUNNEL 8/8

j11_FUNNEL_NCB_DATA

In this step, we merge the two files that were created in "j10_PREPARE_NCB_DATA_FOR_FUNNEL" on page 773 using the Funnel stage. The output of this step is then included for matching and surviving the "best" data as described in 2.6.3, "Matching and surviving Northern California Bank and Northern California Bank information" on page 783.

Figure 2-321 on page 782 shows the various stages that are used in this job. It includes the data sets that were created in "j10_PREPARE_NCB_DATA_FOR_FUNNEL" on page 773, a Funnel stage, and an output Data Set stage. We modified the stage names as shown.

Because we have described these configurations previously, we do not repeat them here.

Figure 2-321 shows the results after compiling and running this job.

Proceed now to 2.6.3, "Matching and surviving Northern California Bank and Northern California Bank information" on page 783.

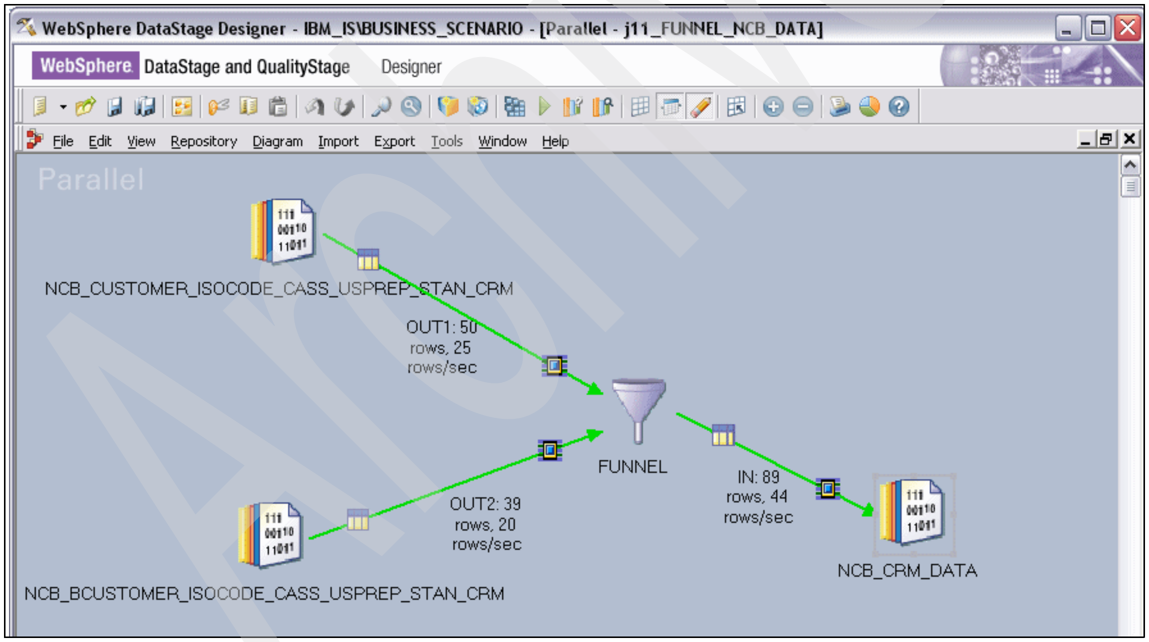

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j11_FUNNEL_NCB_DATA]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NCB_CUSTOMER_ISOCODE_CASS_USPREP_STAN_CRM
OUT1: 50 rows, 25 rows/sec
FUNNEL
IN: 89 rows, 44 rows/sec
OUT2: 39 rows, 20 rows/sec
NCB_CRM_DATA
NCB_BCUSTOMER_ISOCODE_CASS_USPREP_STAN_CRM

Figure 2-321 Create j11_FUNNEL_NCB_DATA

2.6.3 Matching and surviving Northern California Bank and Northern California Bank information

Figure 2-8 on page 500 shows the processing flow and jobs that are used for matching and surviving name and addresses from Northern California Bank's core and non-core services and North American Bank's core and non-core services into the CRM system.

We describe the steps briefly here:

1. In order to survive the "best" name and address data from the Northern California Bank and North American Bank's systems, we needed to merge the standardized name and address files created in steps "j14_FUNNEL_NAB_DATA_FOR_CRM" on page 705 and "j11_FUNNEL_NCB_DATA" on page 782 as described in 2.6.1, "Cleansing North American Bank's core and non-core services" on page 508 and 2.6.2, "Cleansing Northern California Bank's core and non-core services" on page 706 of the CUSTOMER respectively.

   Prior to merging these two files, we needed to prepare the two files for merging in the Funnel stage by ensuring that all the files have the same number of columns and same names of columns using a Transformer stage.

   Job "j12_PREPARE_NCB_NAB_DATA_FOR_FUNNEL" on page 785 performs this step.
2. In this step, we merges the two files that were created in the "j12_PREPARE_NCB_NAB_DATA_FOR_FUNNEL" on page 785 using the Funnel stage.

   Job "j13_FUNNEL_NCB_NAB_CRM_DATA" on page 797 performs this step.
3. Prior to matching, we generated the frequency distribution on all the columns in the merged file of the previous step "j13_FUNNEL_NCB_NAB_CRM_DATA" on page 797 using the Match Frequency stage. Here again, the idea was to generate match frequency for all the columns so that it can be used with any match specification.

   Job "j14_CRM_FREQUENCY" on page 798 performs this step.
4. Next, we generated a match specification for an Unduplicate match stage using as input the match frequency data created in the "j14_CRM_FREQUENCY" on page 798 job. The specification included a single pass that blocked on two phonetic encoding (NYSIIS) columns MatchFirstNameNYSIIS_USNAME, MatchPrimaryWord1NYSIIS_USNAME, and the 3-digit ZIP code column ZIP3_HOME corresponding to the home address.

Note: The work address only appears in the North American Bank's core services and, therefore, requires no matching and survive. Therefore, the ZIP code of the work address is not relevant here.

Job "j14a_MATCHSPEC" on page 801 performs this step.

5. Next, we determined whether there were duplicates in the merged records of the North American Bank and Northern California Bank's core and non-core services customers, using the Unduplicate stage with the match specification and match frequency information that is created in steps "j14a_MATCHSPEC" on page 801 and "j14_CRM_FREQUENCY" on page 798 respectively. As before, the output was matched records (merge of master and duplicates using a Funnel stage), records for clerical review, and residuals (records that do not match).

   Job "j15_UNDUP_CRM" on page 809 performs this step.

6. The next step was to survive the "best" information from the set of matched records created in the "j15_UNDUP_CRM" on page 809 step using the SURVIVE stage. We introduced a Transformer stage to add a column DPVCode_num that contained the recoded value of the contents of the DPVCode1_CASS column—if the column contained `AA`, to put a `1` in the DPVCode_num column, otherwise put a `0`. This was done to circumvent a bug in the Survive Rule Builder that had problems comparing string content but that is not numeric content.

   Job "j16_SURVIVE_CRM" on page 814 performs this step.

7. Next, we merged the survived records from the "j16_SURVIVE_CRM" on page 814 step with the residual records that were created in the "j15_UNDUP_CRM" on page 809 step using a Funnel stage. The information from these merged files was required to update the CRM system.

   Job "j17_FUNNEL_UNDUP_RES_DATA" on page 829 performs this step.

8. The output of the "j17_FUNNEL_UNDUP_RES_DATA" on page 829 step needed to be transformed prior to using it for updating the relevant address columns in the CRM system. Two Transformer stages performed these functions. The output of this step contained data to update the CRM system with the "best" information from the North American Bank and Northern California Bank's core and non-core systems.

   Job "j18_CRM_DATA_TRANSFORM" on page 833 performs this step.

Note: We do not show the actual update of the CRM system tables using this file. The IBM WebSphere DataStage jobs that are used to perform this update are described in *IBM InfoSphere DataStage Data Flow and Job Design*, SG24-7576.

We describe these jobs in more detail in the following sections.

j12_PREPARE_NCB_NAB_DATA_FOR_FUNNEL

To survive the "best" name and address data from the Northern California Bank and North American Bank's systems, we need to merge the standardized name and address files that were created in "j14_FUNNEL_NAB_DATA_FOR_CRM" on page 705 and "j11_FUNNEL_NCB_DATA" on page 782 (as described in 2.6.1, "Cleansing North American Bank's core and non-core services" on page 508 and 2.6.2, "Cleansing Northern California Bank's core and non-core services" on page 706).

Prior to merging these two files, we need to prepare the two files for merging in the Funnel stage by ensuring that all the files have the same number of columns and same names of columns using a Transformer stage.

Figure 2-322 on page 786 shows the various stages that are used in this job. It includes the data sets that were created in "j14_FUNNEL_NAB_DATA_FOR_CRM" on page 705 and "j11_FUNNEL_NCB_DATA" on page 782, a Transformer stage, and an output Data Set stage for each Transformer stage. We modified the names of the stages as shown.

Figure 2-323 on page 787 through Figure 2-327 on page 791 show the PICK_COLUMNS1 - Transformer windows that pick the appropriate columns from the Northern California Bank name and address file that was created in "j11_FUNNEL_NCB_DATA" on page 782. Certain columns are renamed and others are added with an empty string as highlighted.

Figure 2-328 on page 792 through Figure 2-332 on page 796 show the corresponding PICK_COLUMNS2 - Transformer windows that pick the appropriate columns from the North American Bank name and addresses file created in the "j14_FUNNEL_NAB_DATA_FOR_CRM" on page 705 step. Here again, certain columns are renamed and others are added with an empty string as highlighted.

After compiling and running this job (Figure 2-333 on page 797), proceed to "j13_FUNNEL_NCB_NAB_CRM_DATA" on page 797.

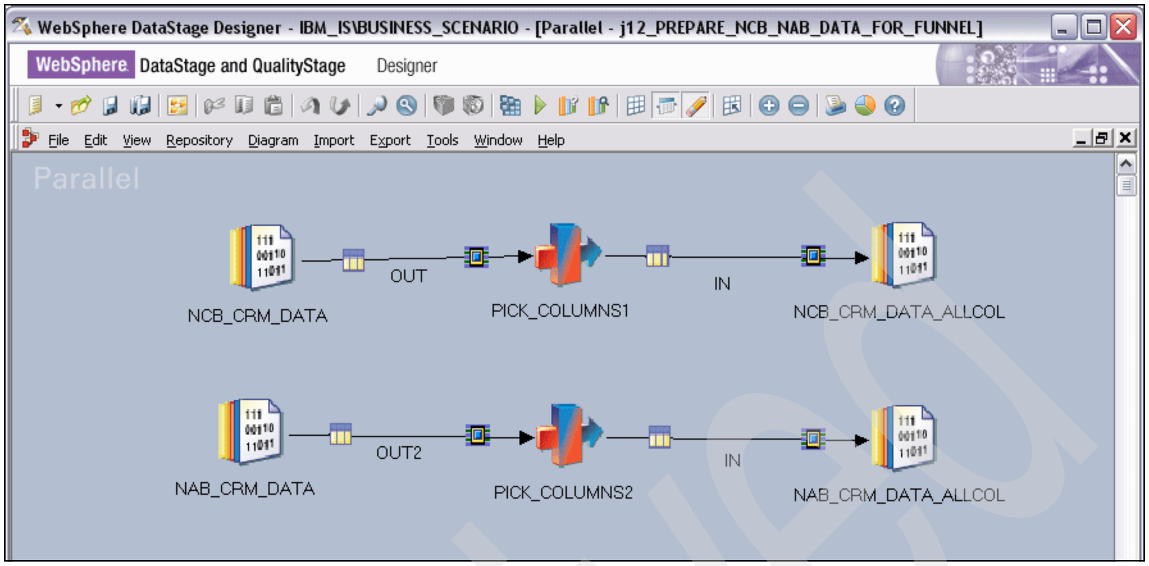
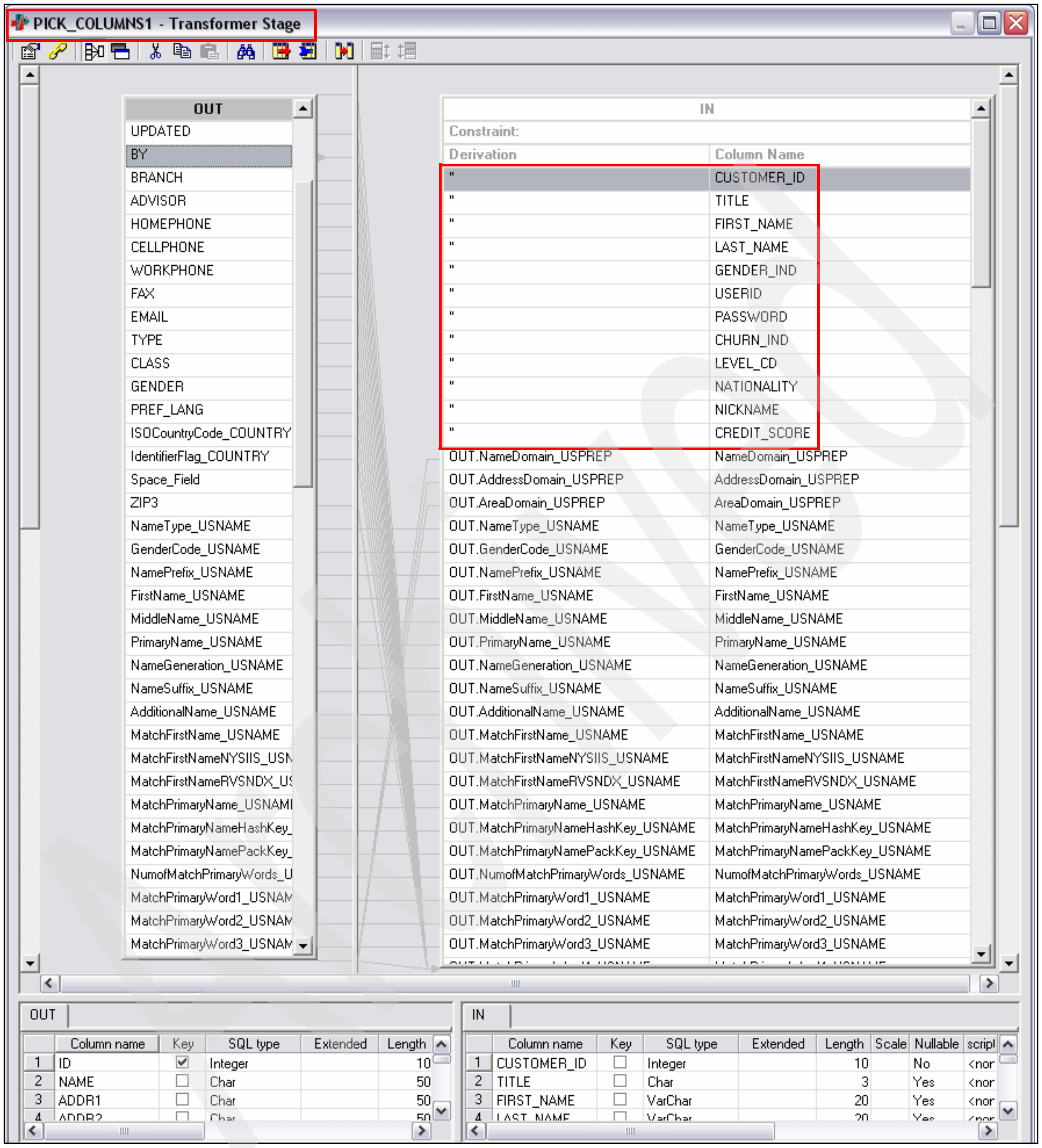

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j12_PREPARE_NCB_NAB_DATA_FOR_FUNNEL]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NCB_CRM_DATA
OUT
PICK_COLUMNS1
IN
NCB_CRM_DATA_ALLCOL
NAB_CRM_DATA
OUT2
PICK_COLUMNS2
IN
NAB_CRM_DATA_ALLCOL

Figure 2-322 Create j12_PREPARE_NCB_NAB_DATA_FOR_FUNNEL 1/12



PICK_COLUMNS1 - Transformer Stage

OUT
UPDATED
BY
BRANCH
ADVISOR
HOMEPHONE
CELLPHONE
WORKPHONE
FAX
EMAIL
TYPE
CLASS
GENDER
PREF_LANG
ISOCountryCode_COUNTRY
IdentifierFlag_COUNTRY
Space_Field
ZIP3
NameType_USNAME
GenderCode_USNAME
NamePrefix_USNAME
FirstName_USNAME
MiddleName_USNAME
PrimaryName_USNAME
NameGeneration_USNAME
NameSuffix_USNAME
AdditionalName_USNAME
MatchFirstName_USNAME
MatchFirstNameNYSIIS_USN
MatchFirstNameRVSNDX_US
MatchPrimaryName_USNAMI
MatchPrimaryNameHashKey_
MatchPrimaryNamePackKey_
NumofMatchPrimaryWords_U
MatchPrimaryWord1_USNAM
MatchPrimaryWord2_USNAM
MatchPrimaryWord3_USNAM

IN

Constraint:

Derivation	Column Name
"	CUSTOMER_ID
"	TITLE
"	FIRST_NAME
"	LAST_NAME
"	GENDER_IND
"	USERID
"	PASSWORD
"	CHURN_IND
"	LEVEL_CD
"	NATIONALITY
"	NICKNAME
"	CREDIT_SCORE
OUT.NameDomain_USPREP	NameDomain_USPREP
OUT.AddressDomain_USPREP	AddressDomain_USPREP
OUT.AreaDomain_USPREP	AreaDomain_USPREP
OUT.NameType_USNAME	NameType_USNAME
OUT.GenderCode_USNAME	GenderCode_USNAME
OUT.NamePrefix_USNAME	NamePrefix_USNAME
OUT.FirstName_USNAME	FirstName_USNAME
OUT.MiddleName_USNAME	MiddleName_USNAME
OUT.PrimaryName_USNAME	PrimaryName_USNAME
OUT.NameGeneration_USNAME	NameGeneration_USNAME
OUT.NameSuffix_USNAME	NameSuffix_USNAME
OUT.AdditionalName_USNAME	AdditionalName_USNAME
OUT.MatchFirstName_USNAME	MatchFirstName_USNAME
OUT.MatchFirstNameNYSIIS_USNAME	MatchFirstNameNYSIIS_USNAME
OUT.MatchFirstNameRVSNDX_USNAME	MatchFirstNameRVSNDX_USNAME
OUT.MatchPrimaryName_USNAME	MatchPrimaryName_USNAME
OUT.MatchPrimaryNameHashKey_USNAME	MatchPrimaryNameHashKey_USNAME
OUT.MatchPrimaryNamePackKey_USNAME	MatchPrimaryNamePackKey_USNAME
OUT.NumofMatchPrimaryWords_USNAME	NumofMatchPrimaryWords_USNAME
OUT.MatchPrimaryWord1_USNAME	MatchPrimaryWord1_USNAME
OUT.MatchPrimaryWord2_USNAME	MatchPrimaryWord2_USNAME
OUT.MatchPrimaryWord3_USNAME	MatchPrimaryWord3_USNAME

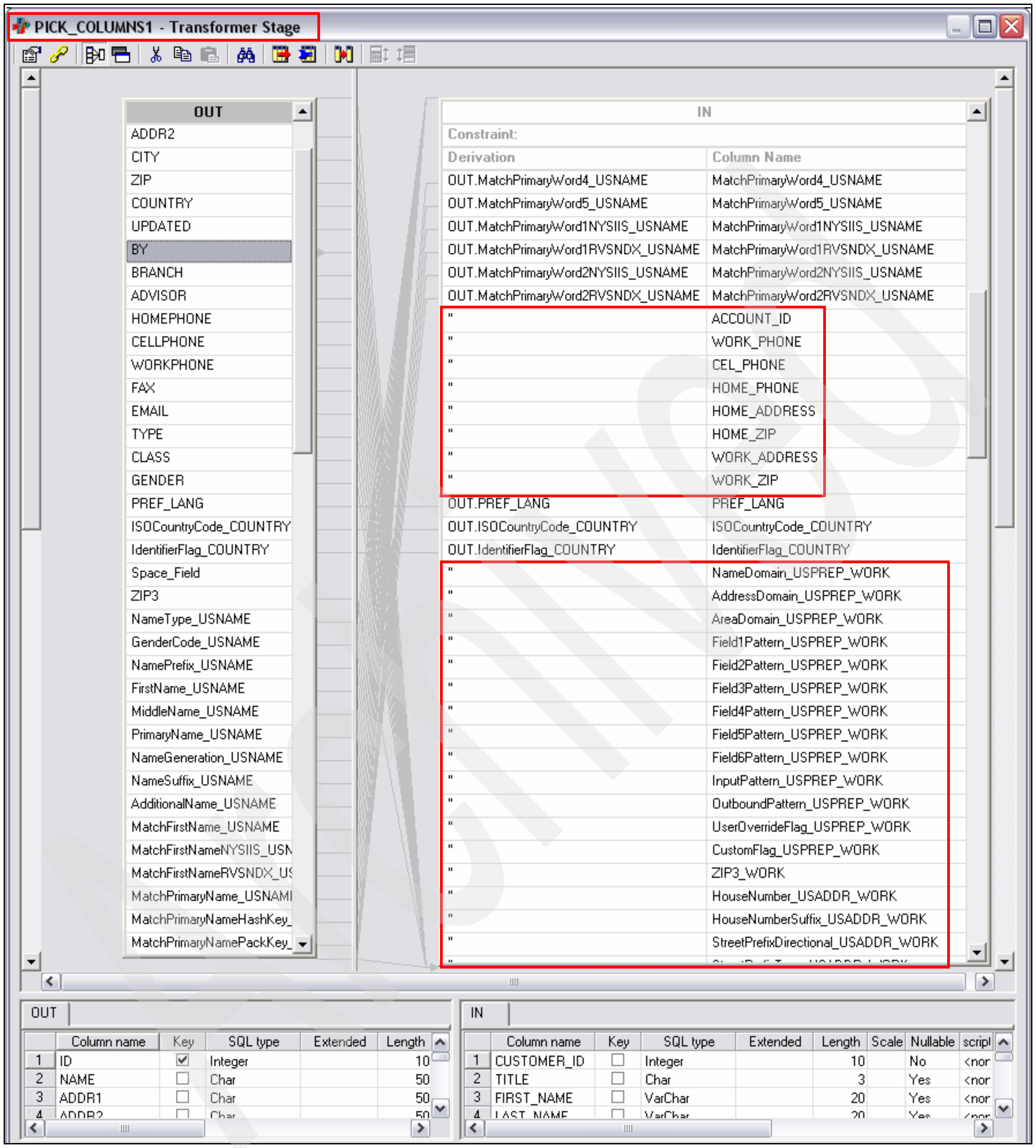
OUT

	Column name	Key	SQL type	Extended	Length
1	ID	✓	Integer		10
2	NAME	☐	Char		50
3	ADDR1	☐	Char		50
4	ADDR2	☐	Char		50

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripl
1	CUSTOMER_ID	☐	Integer		10		No	<nor
2	TITLE	☐	Char		3		Yes	<nor
3	FIRST_NAME	☐	VarChar		20		Yes	<nor
4	LAST_NAME	☐	VarChar		20		Yes	<nor

Figure 2-323 Create j12_PREPARE_NCB_NAB_DATA_FOR_FUNNEL 2/12



PICK_COLUMNS1 - Transformer Stage

OUT
ADDR2
CITY
ZIP
COUNTRY
UPDATED
BY
BRANCH
ADVISOR
HOMEPHONE
CELLPHONE
WORKPHONE
FAX
EMAIL
TYPE
CLASS
GENDER
PREF_LANG
ISOCountryCode_COUNTRY
IdentifierFlag_COUNTRY
Space_Field
ZIP3
NameType_USNAME
GenderCode_USNAME
NamePrefix_USNAME
FirstName_USNAME
MiddleName_USNAME
PrimaryName_USNAME
NameGeneration_USNAME
NameSuffix_USNAME
AdditionalName_USNAME
MatchFirstName_USNAME
MatchFirstNameNYSIIS_USN
MatchFirstNameRVSNDX_US
MatchPrimaryName_USNAMI
MatchPrimaryNameHashKey_
MatchPrimaryNamePackKey_

IN

Constraint:

Derivation	Column Name
OUT.MatchPrimaryWord4_USNAME	MatchPrimaryWord4_USNAME
OUT.MatchPrimaryWord5_USNAME	MatchPrimaryWord5_USNAME
OUT.MatchPrimaryWord1NYSIIS_USNAME	MatchPrimaryWord1NYSIIS_USNAME
OUT.MatchPrimaryWord1RVSNDX_USNAME	MatchPrimaryWord1RVSNDX_USNAME
OUT.MatchPrimaryWord2NYSIIS_USNAME	MatchPrimaryWord2NYSIIS_USNAME
OUT.MatchPrimaryWord2RVSNDX_USNAME	MatchPrimaryWord2RVSNDX_USNAME
""	ACCOUNT_ID
""	WORK_PHONE
""	CEL_PHONE
""	HOME_PHONE
""	HOME_ADDRESS
""	HOME_ZIP
""	WORK_ADDRESS
""	WORK_ZIP
OUT.PREF_LANG	PREF_LANG
OUT.ISOCountryCode_COUNTRY	ISOCountryCode_COUNTRY
OUT.IdentifierFlag_COUNTRY	IdentifierFlag_COUNTRY
""	NameDomain_USPREP_WORK
""	AddressDomain_USPREP_WORK
""	AreaDomain_USPREP_WORK
""	Field1Pattern_USPREP_WORK
""	Field2Pattern_USPREP_WORK
""	Field3Pattern_USPREP_WORK
""	Field4Pattern_USPREP_WORK
""	Field5Pattern_USPREP_WORK
""	Field6Pattern_USPREP_WORK
""	InputPattern_USPREP_WORK
""	OutboundPattern_USPREP_WORK
""	UserOverrideFlag_USPREP_WORK
""	CustomFlag_USPREP_WORK
""	ZIP3_WORK
""	HouseNumber_USADDR_WORK
""	HouseNumberSuffix_USADDR_WORK
""	StreetPrefixDirectional_USADDR_WORK

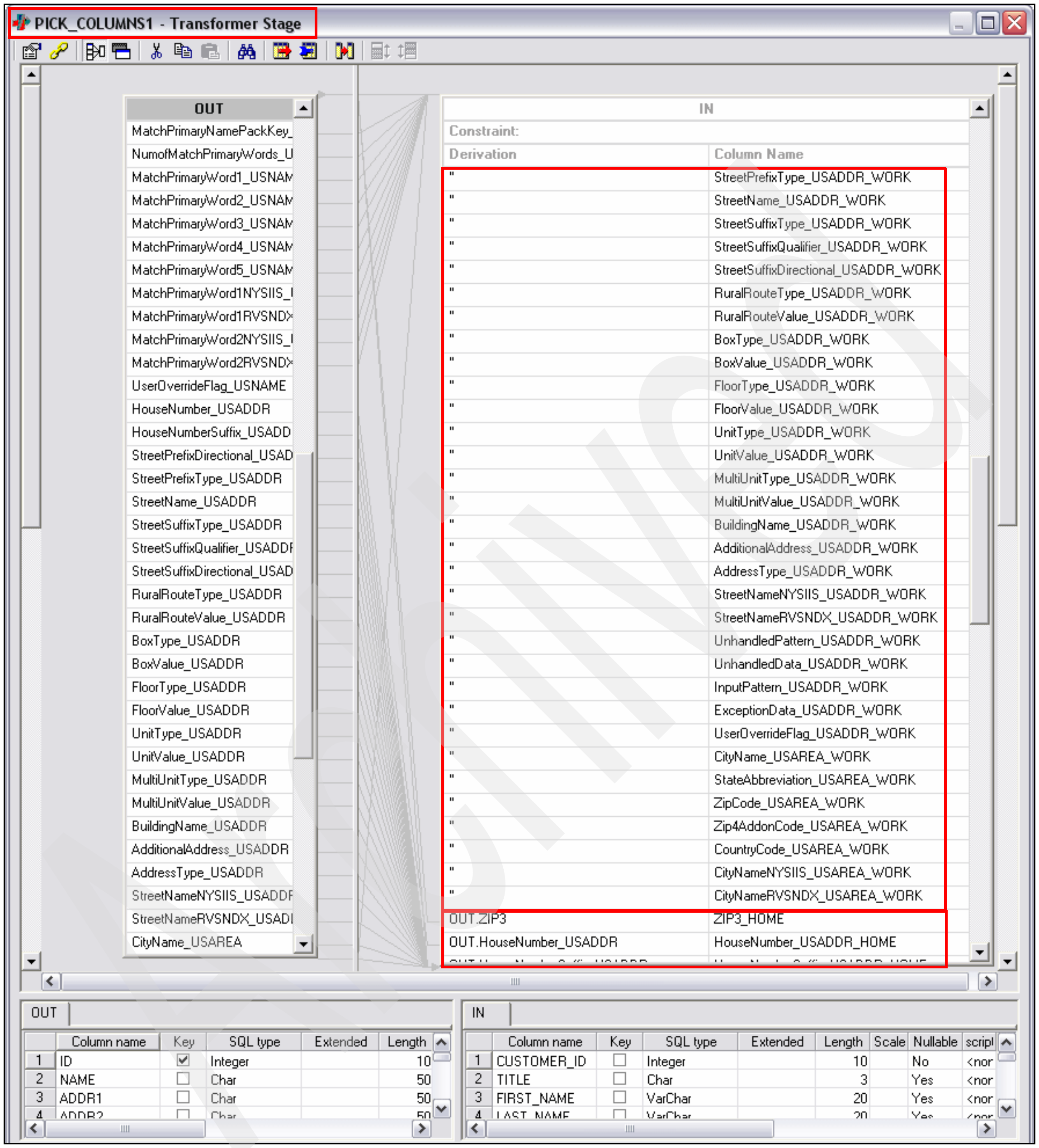
OUT

	Column name	Key	SQL type	Extended	Length
1	ID	✓	Integer		10
2	NAME		Char		50
3	ADDR1		Char		50
4	ADDR2		Char		50

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripl
1	CUSTOMER_ID		Integer		10		No	<nor
2	TITLE		Char		3		Yes	<nor
3	FIRST_NAME		VarChar		20		Yes	<nor
4	LAST_NAME		VarChar		20		Yes	<nor

Figure 2-324 Create j12_PREPARE_NCB_NAB_DATA_FOR_FUNNEL 3/12



PICK_COLUMNS1 - Transformer Stage

OUT
MatchPrimaryNamePackKey_
NumofMatchPrimaryWords_U
MatchPrimaryWord1_USNAM
MatchPrimaryWord2_USNAM
MatchPrimaryWord3_USNAM
MatchPrimaryWord4_USNAM
MatchPrimaryWord5_USNAM
MatchPrimaryWord1NYSIIS_I
MatchPrimaryWord1RVSNDX
MatchPrimaryWord2NYSIIS_I
MatchPrimaryWord2RVSNDX
UserOverrideFlag_USNAME
HouseNumber_USADDR
HouseNumberSuffix_USADD
StreetPrefixDirectional_USAD
StreetPrefixType_USADDR
StreetName_USADDR
StreetSuffixType_USADDR
StreetSuffixQualifier_USADDF
StreetSuffixDirectional_USAD
RuralRouteType_USADDR
RuralRouteValue_USADDR
BoxType_USADDR
BoxValue_USADDR
FloorType_USADDR
FloorValue_USADDR
UnitType_USADDR
UnitValue_USADDR
MultiUnitType_USADDR
MultiUnitValue_USADDR
BuildingName_USADDR
AdditionalAddress_USADDR
AddressType_USADDR
StreetNameNYSIIS_USADDF
StreetNameRVSNDX_USADI
CityName_USAREA

IN

Constraint:

Derivation	Column Name
''	StreetPrefixType_USADDR_WORK
''	StreetName_USADDR_WORK
''	StreetSuffixType_USADDR_WORK
''	StreetSuffixQualifier_USADDR_WORK
''	StreetSuffixDirectional_USADDR_WORK
''	RuralRouteType_USADDR_WORK
''	RuralRouteValue_USADDR_WORK
''	BoxType_USADDR_WORK
''	BoxValue_USADDR_WORK
''	FloorType_USADDR_WORK
''	FloorValue_USADDR_WORK
''	UnitType_USADDR_WORK
''	UnitValue_USADDR_WORK
''	MultiUnitType_USADDR_WORK
''	MultiUnitValue_USADDR_WORK
''	BuildingName_USADDR_WORK
''	AdditionalAddress_USADDR_WORK
''	AddressType_USADDR_WORK
''	StreetNameNYSIIS_USADDR_WORK
''	StreetNameRVSNDX_USADDR_WORK
''	UnhandledPattern_USADDR_WORK
''	UnhandledData_USADDR_WORK
''	InputPattern_USADDR_WORK
''	ExceptionData_USADDR_WORK
''	UserOverrideFlag_USADDR_WORK
''	CityName_USAREA_WORK
''	StateAbbreviation_USAREA_WORK
''	ZipCode_USAREA_WORK
''	Zip4AddonCode_USAREA_WORK
''	CountryCode_USAREA_WORK
''	CityNameNYSIIS_USAREA_WORK
''	CityNameRVSNDX_USAREA_WORK
OUT.ZIP3	ZIP3_HOME
OUT.HouseNumber_USADDR	HouseNumber_USADDR_HOME

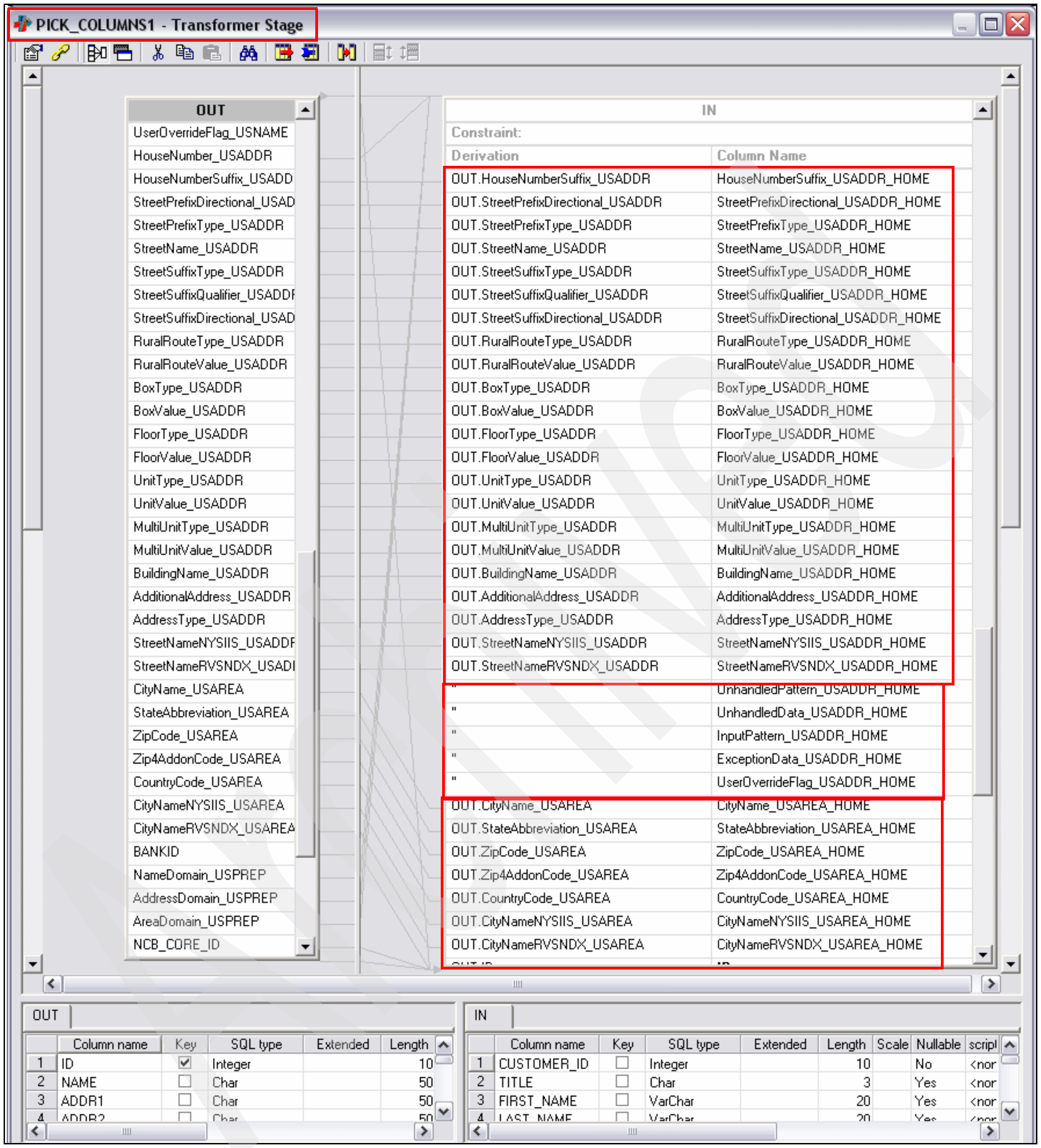
OUT

	Column name	Key	SQL type	Extended	Length
1	ID	✓	Integer		10
2	NAME		Char		50
3	ADDR1		Char		50
4	ADDR2		Char		50

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripl
1	CUSTOMER_ID		Integer		10		No	<nor
2	TITLE		Char		3		Yes	<nor
3	FIRST_NAME		VarChar		20		Yes	<nor
4	LAST_NAME		VarChar		20		Yes	<nor

Figure 2-325 Create j12_PREPARE_NCB_NAB_DATA_FOR_FUNNEL 4/12



PICK_COLUMNS1 - Transformer Stage

OUT
UserOverrideFlag_USNAME
HouseNumber_USADDR
HouseNumberSuffix_USADD
StreetPrefixDirectional_USAD
StreetPrefixType_USADDR
StreetName_USADDR
StreetSuffixType_USADDR
StreetSuffixQualifier_USADDF
StreetSuffixDirectional_USAD
RuralRouteType_USADDR
RuralRouteValue_USADDR
BoxType_USADDR
BoxValue_USADDR
FloorType_USADDR
FloorValue_USADDR
UnitType_USADDR
UnitValue_USADDR
MultiUnitType_USADDR
MultiUnitValue_USADDR
BuildingName_USADDR
AdditionalAddress_USADDR
AddressType_USADDR
StreetNameNYSIIS_USADDF
StreetNameRVSNDX_USADI
CityName_USAREA
StateAbbreviation_USAREA
ZipCode_USAREA
Zip4AddonCode_USAREA
CountryCode_USAREA
CityNameNYSIIS_USAREA
CityNameRVSNDX_USAREA
BANKID
NameDomain_USPREP
AddressDomain_USPREP
AreaDomain_USPREP
NCB_CORE_ID

IN

Constraint:

Derivation	Column Name
OUT.HouseNumberSuffix_USADDR	HouseNumberSuffix_USADDR_HOME
OUT.StreetPrefixDirectional_USADDR	StreetPrefixDirectional_USADDR_HOME
OUT.StreetPrefixType_USADDR	StreetPrefixType_USADDR_HOME
OUT.StreetName_USADDR	StreetName_USADDR_HOME
OUT.StreetSuffixType_USADDR	StreetSuffixType_USADDR_HOME
OUT.StreetSuffixQualifier_USADDR	StreetSuffixQualifier_USADDR_HOME
OUT.StreetSuffixDirectional_USADDR	StreetSuffixDirectional_USADDR_HOME
OUT.RuralRouteType_USADDR	RuralRouteType_USADDR_HOME
OUT.RuralRouteValue_USADDR	RuralRouteValue_USADDR_HOME
OUT.BoxType_USADDR	BoxType_USADDR_HOME
OUT.BoxValue_USADDR	BoxValue_USADDR_HOME
OUT.FloorType_USADDR	FloorType_USADDR_HOME
OUT.FloorValue_USADDR	FloorValue_USADDR_HOME
OUT.UnitType_USADDR	UnitType_USADDR_HOME
OUT.UnitValue_USADDR	UnitValue_USADDR_HOME
OUT.MultiUnitType_USADDR	MultiUnitType_USADDR_HOME
OUT.MultiUnitValue_USADDR	MultiUnitValue_USADDR_HOME
OUT.BuildingName_USADDR	BuildingName_USADDR_HOME
OUT.AdditionalAddress_USADDR	AdditionalAddress_USADDR_HOME
OUT.AddressType_USADDR	AddressType_USADDR_HOME
OUT.StreetNameNYSIIS_USADDR	StreetNameNYSIIS_USADDR_HOME
OUT.StreetNameRVSNDX_USADDR	StreetNameRVSNDX_USADDR_HOME
''	UnhandledPattern_USADDR_HOME
''	UnhandledData_USADDR_HOME
''	InputPattern_USADDR_HOME
''	ExceptionData_USADDR_HOME
''	UserOverrideFlag_USADDR_HOME
OUT.CityName_USAREA	CityName_USAREA_HOME
OUT.StateAbbreviation_USAREA	StateAbbreviation_USAREA_HOME
OUT.ZipCode_USAREA	ZipCode_USAREA_HOME
OUT.Zip4AddonCode_USAREA	Zip4AddonCode_USAREA_HOME
OUT.CountryCode_USAREA	CountryCode_USAREA_HOME
OUT.CityNameNYSIIS_USAREA	CityNameNYSIIS_USAREA_HOME
OUT.CityNameRVSNDX_USAREA	CityNameRVSNDX_USAREA_HOME

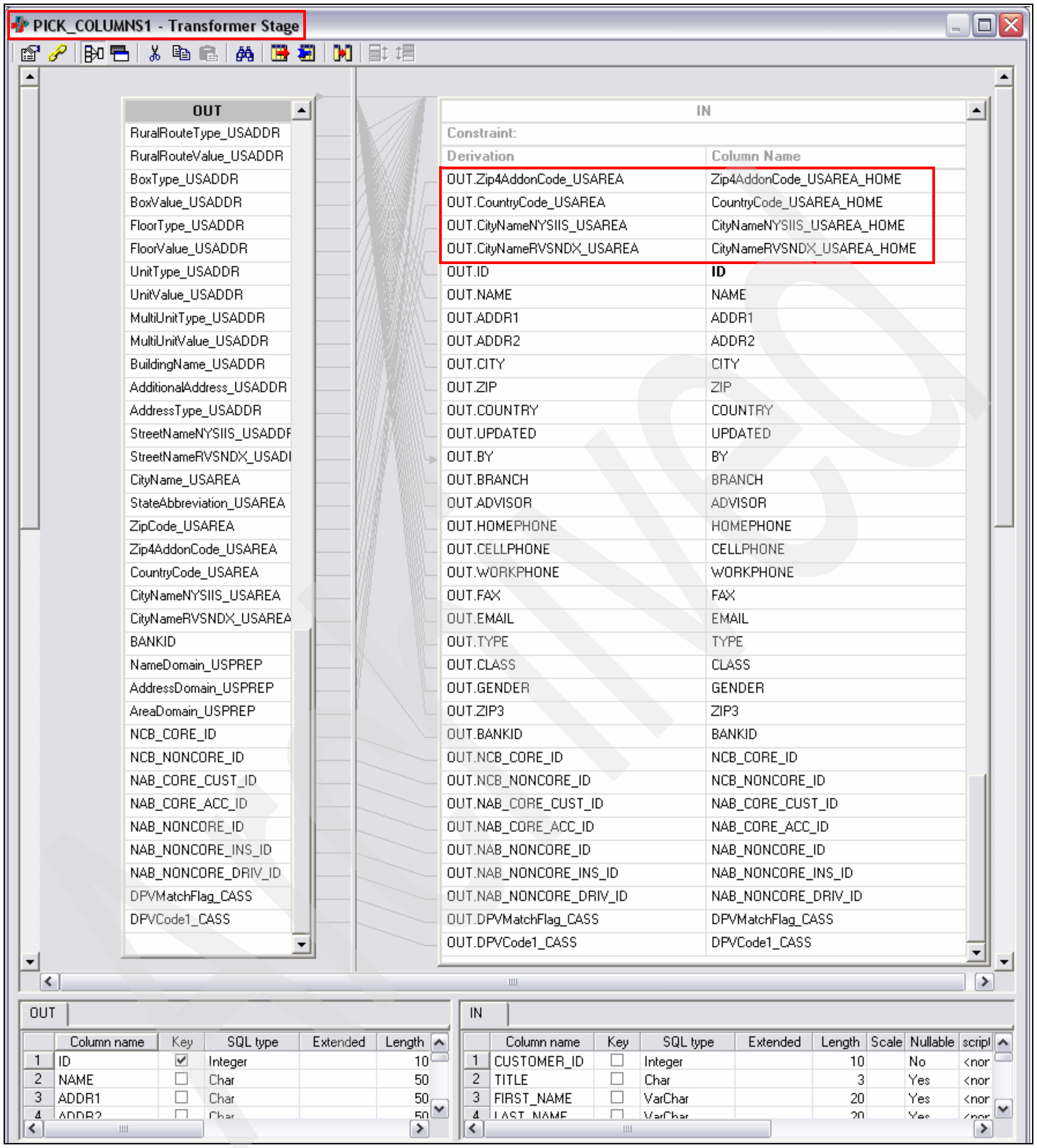
OUT

	Column name	Key	SQL type	Extended	Length
1	ID	✓	Integer		10
2	NAME	☐	Char		50
3	ADDR1	☐	Char		50
4	ADDR2	☐	Char		50

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripi
1	CUSTOMER_ID	☐	Integer		10		No	<nor
2	TITLE	☐	Char		3		Yes	<nor
3	FIRST_NAME	☐	VarChar		20		Yes	<nor
4	LAST_NAME	☐	VarChar		20		Yes	<nor

Figure 2-326 Create j12_PREPARE_NCB_NAB_DATA_FOR_FUNNEL 5/12



PICK_COLUMNS1 - Transformer Stage

OUT
RuralRouteType_USADDR
RuralRouteValue_USADDR
BoxType_USADDR
BoxValue_USADDR
FloorType_USADDR
FloorValue_USADDR
UnitType_USADDR
UnitValue_USADDR
MultiUnitType_USADDR
MultiUnitValue_USADDR
BuildingName_USADDR
AdditionalAddress_USADDR
AddressType_USADDR
StreetNameNYSIIS_USADDF
StreetNameRVSNDX_USADI
CityName_USAREA
StateAbbreviation_USAREA
ZipCode_USAREA
Zip4AddonCode_USAREA
CountryCode_USAREA
CityNameNYSIIS_USAREA
CityNameRVSNDX_USAREA
BANKID
NameDomain_USPREP
AddressDomain_USPREP
AreaDomain_USPREP
NCB_CORE_ID
NCB_NONCORE_ID
NAB_CORE_CUST_ID
NAB_CORE_ACC_ID
NAB_NONCORE_ID
NAB_NONCORE_INS_ID
NAB_NONCORE_DRIV_ID
DPVMatchFlag_CASS
DPVCode1_CASS

IN

Constraint:

Derivation	Column Name
OUT.Zip4AddonCode_USAREA	Zip4AddonCode_USAREA_HOME
OUT.CountryCode_USAREA	CountryCode_USAREA_HOME
OUT.CityNameNYSIIS_USAREA	CityNameNYSIIS_USAREA_HOME
OUT.CityNameRVSNDX_USAREA	CityNameRVSNDX_USAREA_HOME
OUT.ID	**ID**
OUT.NAME	NAME
OUT.ADDR1	ADDR1
OUT.ADDR2	ADDR2
OUT.CITY	CITY
OUT.ZIP	ZIP
OUT.COUNTRY	COUNTRY
OUT.UPDATED	UPDATED
OUT.BY	BY
OUT.BRANCH	BRANCH
OUT.ADVISOR	ADVISOR
OUT.HOMEPHONE	HOMEPHONE
OUT.CELLPHONE	CELLPHONE
OUT.WORKPHONE	WORKPHONE
OUT.FAX	FAX
OUT.EMAIL	EMAIL
OUT.TYPE	TYPE
OUT.CLASS	CLASS
OUT.GENDER	GENDER
OUT.ZIP3	ZIP3
OUT.BANKID	BANKID
OUT.NCB_CORE_ID	NCB_CORE_ID
OUT.NCB_NONCORE_ID	NCB_NONCORE_ID
OUT.NAB_CORE_CUST_ID	NAB_CORE_CUST_ID
OUT.NAB_CORE_ACC_ID	NAB_CORE_ACC_ID
OUT.NAB_NONCORE_ID	NAB_NONCORE_ID
OUT.NAB_NONCORE_INS_ID	NAB_NONCORE_INS_ID
OUT.NAB_NONCORE_DRIV_ID	NAB_NONCORE_DRIV_ID
OUT.DPVMatchFlag_CASS	DPVMatchFlag_CASS
OUT.DPVCode1_CASS	DPVCode1_CASS

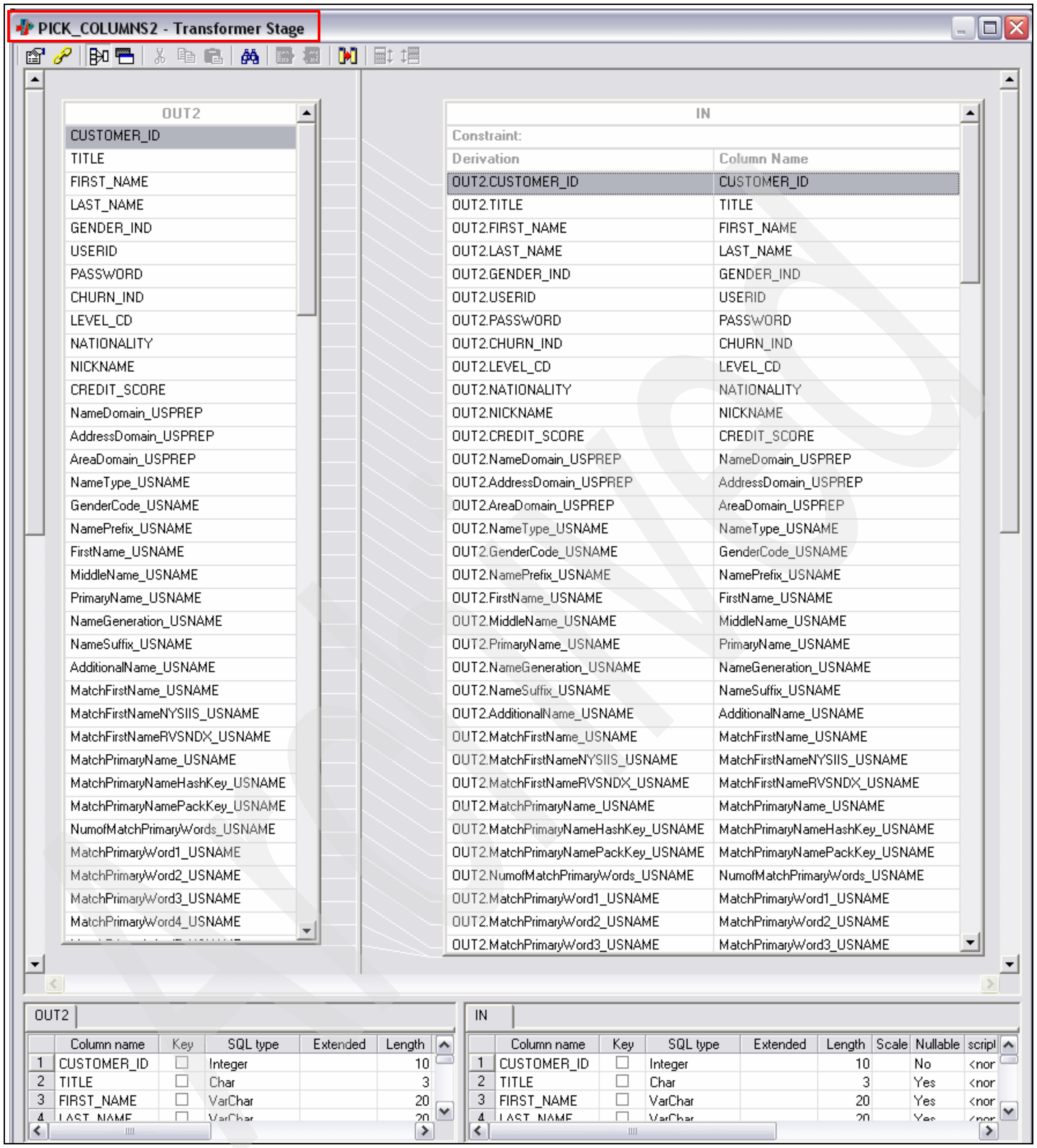
OUT

	Column name	Key	SQL type	Extended	Length
1	ID	✓	Integer		10
2	NAME	☐	Char		50
3	ADDR1	☐	Char		50
4	ADDR2	☐	Char		50

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripl
1	CUSTOMER_ID	☐	Integer		10		No	<nor
2	TITLE	☐	Char		3		Yes	<nor
3	FIRST_NAME	☐	VarChar		20		Yes	<nor
4	LAST_NAME	☐	VarChar		20		Yes	<nor

Figure 2-327 Create j12_PREPARE_NCB_NAB_DATA_FOR_FUNNEL 6/12


PICK_COLUMNS2 - Transformer Stage

OUT2
CUSTOMER_ID
TITLE
FIRST_NAME
LAST_NAME
GENDER_IND
USERID
PASSWORD
CHURN_IND
LEVEL_CD
NATIONALITY
NICKNAME
CREDIT_SCORE
NameDomain_USPREP
AddressDomain_USPREP
AreaDomain_USPREP
NameType_USNAME
GenderCode_USNAME
NamePrefix_USNAME
FirstName_USNAME
MiddleName_USNAME
PrimaryName_USNAME
NameGeneration_USNAME
NameSuffix_USNAME
AdditionalName_USNAME
MatchFirstName_USNAME
MatchFirstNameNYSIIS_USNAME
MatchFirstNameRVSNDX_USNAME
MatchPrimaryName_USNAME
MatchPrimaryNameHashKey_USNAME
MatchPrimaryNamePackKey_USNAME
NumofMatchPrimaryWords_USNAME
MatchPrimaryWord1_USNAME
MatchPrimaryWord2_USNAME
MatchPrimaryWord3_USNAME
MatchPrimaryWord4_USNAME

IN

Constraint:

Derivation	Column Name
OUT2.CUSTOMER_ID	CUSTOMER_ID
OUT2.TITLE	TITLE
OUT2.FIRST_NAME	FIRST_NAME
OUT2.LAST_NAME	LAST_NAME
OUT2.GENDER_IND	GENDER_IND
OUT2.USERID	USERID
OUT2.PASSWORD	PASSWORD
OUT2.CHURN_IND	CHURN_IND
OUT2.LEVEL_CD	LEVEL_CD
OUT2.NATIONALITY	NATIONALITY
OUT2.NICKNAME	NICKNAME
OUT2.CREDIT_SCORE	CREDIT_SCORE
OUT2.NameDomain_USPREP	NameDomain_USPREP
OUT2.AddressDomain_USPREP	AddressDomain_USPREP
OUT2.AreaDomain_USPREP	AreaDomain_USPREP
OUT2.NameType_USNAME	NameType_USNAME
OUT2.GenderCode_USNAME	GenderCode_USNAME
OUT2.NamePrefix_USNAME	NamePrefix_USNAME
OUT2.FirstName_USNAME	FirstName_USNAME
OUT2.MiddleName_USNAME	MiddleName_USNAME
OUT2.PrimaryName_USNAME	PrimaryName_USNAME
OUT2.NameGeneration_USNAME	NameGeneration_USNAME
OUT2.NameSuffix_USNAME	NameSuffix_USNAME
OUT2.AdditionalName_USNAME	AdditionalName_USNAME
OUT2.MatchFirstName_USNAME	MatchFirstName_USNAME
OUT2.MatchFirstNameNYSIIS_USNAME	MatchFirstNameNYSIIS_USNAME
OUT2.MatchFirstNameRVSNDX_USNAME	MatchFirstNameRVSNDX_USNAME
OUT2.MatchPrimaryName_USNAME	MatchPrimaryName_USNAME
OUT2.MatchPrimaryNameHashKey_USNAME	MatchPrimaryNameHashKey_USNAME
OUT2.MatchPrimaryNamePackKey_USNAME	MatchPrimaryNamePackKey_USNAME
OUT2.NumofMatchPrimaryWords_USNAME	NumofMatchPrimaryWords_USNAME
OUT2.MatchPrimaryWord1_USNAME	MatchPrimaryWord1_USNAME
OUT2.MatchPrimaryWord2_USNAME	MatchPrimaryWord2_USNAME
OUT2.MatchPrimaryWord3_USNAME	MatchPrimaryWord3_USNAME

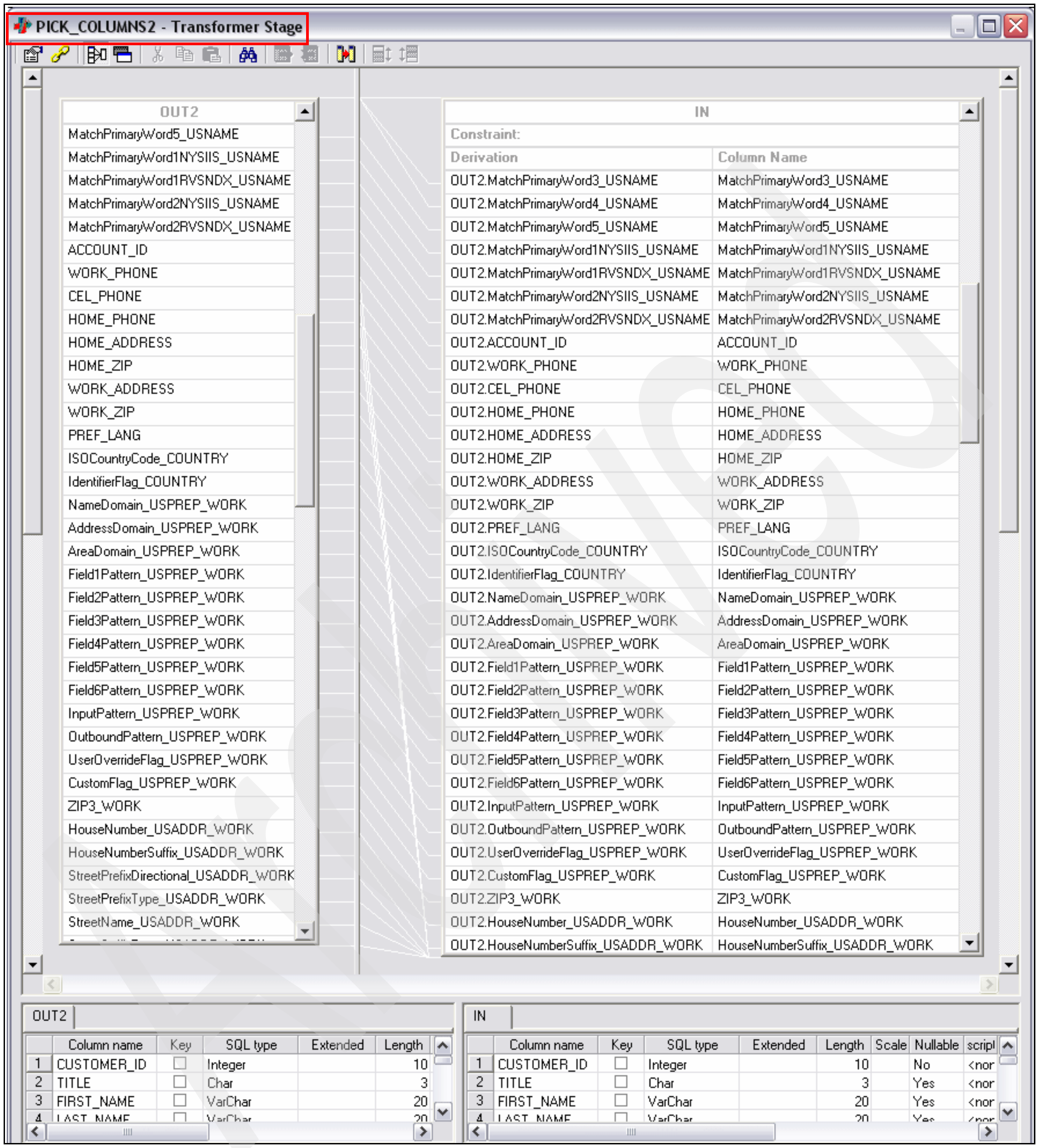
OUT2

	Column name	Key	SQL type	Extended	Length
1	CUSTOMER_ID	☐	Integer		10
2	TITLE	☐	Char		3
3	FIRST_NAME	☐	VarChar		20
4	LAST_NAME	☐	VarChar		20

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripl
1	CUSTOMER_ID	☐	Integer		10		No	<nor
2	TITLE	☐	Char		3		Yes	<nor
3	FIRST_NAME	☐	VarChar		20		Yes	<nor
4	LAST_NAME	☐	VarChar		20		Yes	<nor

Figure 2-328 Create j12_PREPARE_NCB_NAB_DATA_FOR_FUNNEL 7/12



PICK_COLUMNS2 - Transformer Stage

OUT2
MatchPrimaryWord5_USNAME
MatchPrimaryWord1NYSIIS_USNAME
MatchPrimaryWord1RVSNDX_USNAME
MatchPrimaryWord2NYSIIS_USNAME
MatchPrimaryWord2RVSNDX_USNAME
ACCOUNT_ID
WORK_PHONE
CEL_PHONE
HOME_PHONE
HOME_ADDRESS
HOME_ZIP
WORK_ADDRESS
WORK_ZIP
PREF_LANG
ISOCountryCode_COUNTRY
IdentifierFlag_COUNTRY
NameDomain_USPREP_WORK
AddressDomain_USPREP_WORK
AreaDomain_USPREP_WORK
Field1Pattern_USPREP_WORK
Field2Pattern_USPREP_WORK
Field3Pattern_USPREP_WORK
Field4Pattern_USPREP_WORK
Field5Pattern_USPREP_WORK
Field6Pattern_USPREP_WORK
InputPattern_USPREP_WORK
OutboundPattern_USPREP_WORK
UserOverrideFlag_USPREP_WORK
CustomFlag_USPREP_WORK
ZIP3_WORK
HouseNumber_USADDR_WORK
HouseNumberSuffix_USADDR_WORK
StreetPrefixDirectional_USADDR_WORK
StreetPrefixType_USADDR_WORK
StreetName_USADDR_WORK

IN

Constraint:

Derivation	Column Name
OUT2.MatchPrimaryWord3_USNAME	MatchPrimaryWord3_USNAME
OUT2.MatchPrimaryWord4_USNAME	MatchPrimaryWord4_USNAME
OUT2.MatchPrimaryWord5_USNAME	MatchPrimaryWord5_USNAME
OUT2.MatchPrimaryWord1NYSIIS_USNAME	MatchPrimaryWord1NYSIIS_USNAME
OUT2.MatchPrimaryWord1RVSNDX_USNAME	MatchPrimaryWord1RVSNDX_USNAME
OUT2.MatchPrimaryWord2NYSIIS_USNAME	MatchPrimaryWord2NYSIIS_USNAME
OUT2.MatchPrimaryWord2RVSNDX_USNAME	MatchPrimaryWord2RVSNDX_USNAME
OUT2.ACCOUNT_ID	ACCOUNT_ID
OUT2.WORK_PHONE	WORK_PHONE
OUT2.CEL_PHONE	CEL_PHONE
OUT2.HOME_PHONE	HOME_PHONE
OUT2.HOME_ADDRESS	HOME_ADDRESS
OUT2.HOME_ZIP	HOME_ZIP
OUT2.WORK_ADDRESS	WORK_ADDRESS
OUT2.WORK_ZIP	WORK_ZIP
OUT2.PREF_LANG	PREF_LANG
OUT2.ISOCountryCode_COUNTRY	ISOCountryCode_COUNTRY
OUT2.IdentifierFlag_COUNTRY	IdentifierFlag_COUNTRY
OUT2.NameDomain_USPREP_WORK	NameDomain_USPREP_WORK
OUT2.AddressDomain_USPREP_WORK	AddressDomain_USPREP_WORK
OUT2.AreaDomain_USPREP_WORK	AreaDomain_USPREP_WORK
OUT2.Field1Pattern_USPREP_WORK	Field1Pattern_USPREP_WORK
OUT2.Field2Pattern_USPREP_WORK	Field2Pattern_USPREP_WORK
OUT2.Field3Pattern_USPREP_WORK	Field3Pattern_USPREP_WORK
OUT2.Field4Pattern_USPREP_WORK	Field4Pattern_USPREP_WORK
OUT2.Field5Pattern_USPREP_WORK	Field5Pattern_USPREP_WORK
OUT2.Field6Pattern_USPREP_WORK	Field6Pattern_USPREP_WORK
OUT2.InputPattern_USPREP_WORK	InputPattern_USPREP_WORK
OUT2.OutboundPattern_USPREP_WORK	OutboundPattern_USPREP_WORK
OUT2.UserOverrideFlag_USPREP_WORK	UserOverrideFlag_USPREP_WORK
OUT2.CustomFlag_USPREP_WORK	CustomFlag_USPREP_WORK
OUT2.ZIP3_WORK	ZIP3_WORK
OUT2.HouseNumber_USADDR_WORK	HouseNumber_USADDR_WORK
OUT2.HouseNumberSuffix_USADDR_WORK	HouseNumberSuffix_USADDR_WORK

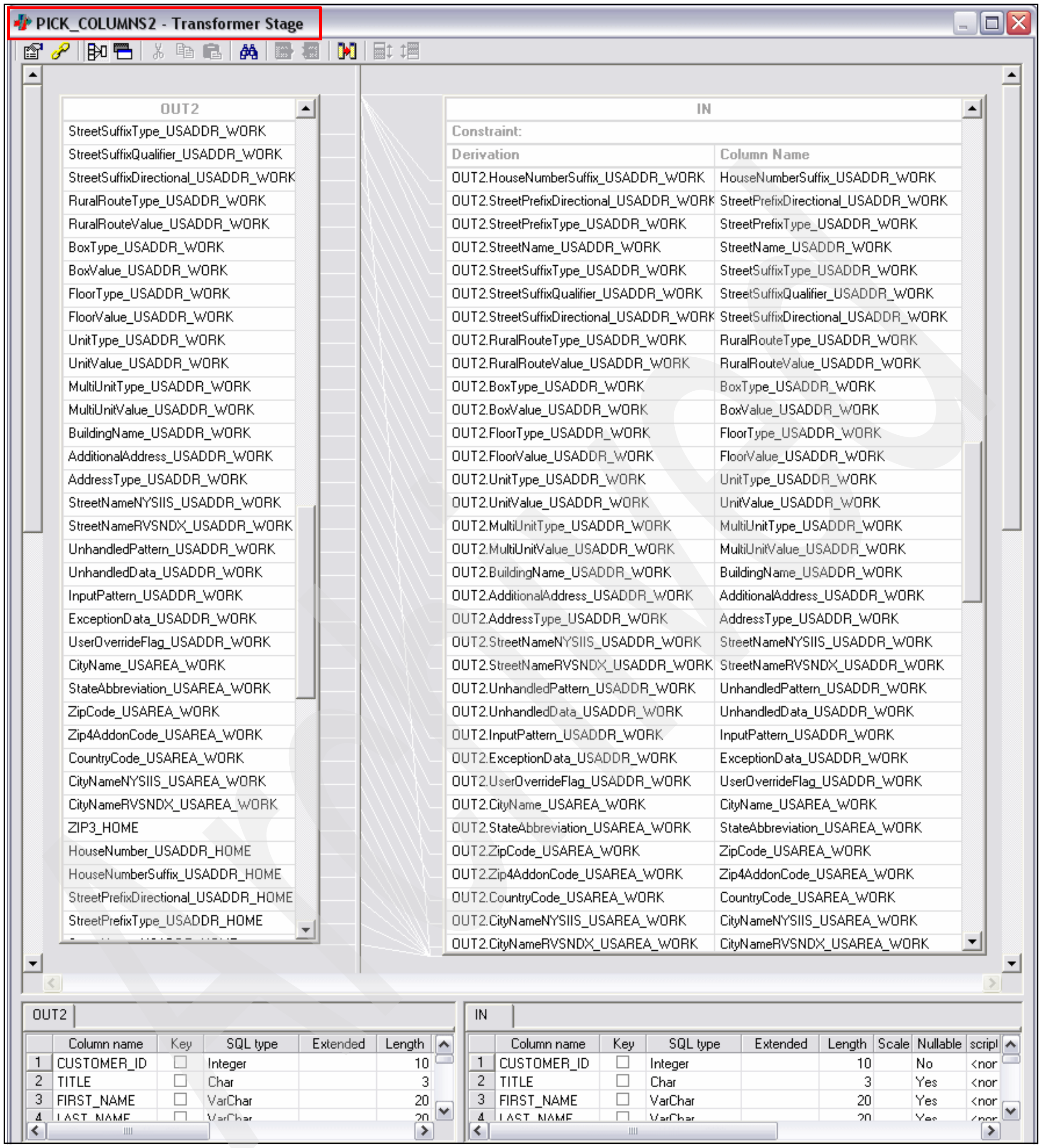
OUT2

	Column name	Key	SQL type	Extended	Length
1	CUSTOMER_ID	☐	Integer		10
2	TITLE	☐	Char		3
3	FIRST_NAME	☐	VarChar		20
4	LAST_NAME	☐	VarChar		20

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripl
1	CUSTOMER_ID	☐	Integer		10		No	<nor
2	TITLE	☐	Char		3		Yes	<nor
3	FIRST_NAME	☐	VarChar		20		Yes	<nor
4	LAST_NAME	☐	VarChar		20		Yes	<nor

Figure 2-329 Create J11_funnel_ncb_data 8/12



PICK_COLUMNS2 - Transformer Stage

OUT2
StreetSuffixType_USADDR_WORK
StreetSuffixQualifier_USADDR_WORK
StreetSuffixDirectional_USADDR_WORK
RuralRouteType_USADDR_WORK
RuralRouteValue_USADDR_WORK
BoxType_USADDR_WORK
BoxValue_USADDR_WORK
FloorType_USADDR_WORK
FloorValue_USADDR_WORK
UnitType_USADDR_WORK
UnitValue_USADDR_WORK
MultiUnitType_USADDR_WORK
MultiUnitValue_USADDR_WORK
BuildingName_USADDR_WORK
AdditionalAddress_USADDR_WORK
AddressType_USADDR_WORK
StreetNameNYSIIS_USADDR_WORK
StreetNameRVSNDX_USADDR_WORK
UnhandledPattern_USADDR_WORK
UnhandledData_USADDR_WORK
InputPattern_USADDR_WORK
ExceptionData_USADDR_WORK
UserOverrideFlag_USADDR_WORK
CityName_USAREA_WORK
StateAbbreviation_USAREA_WORK
ZipCode_USAREA_WORK
Zip4AddonCode_USAREA_WORK
CountryCode_USAREA_WORK
CityNameNYSIIS_USAREA_WORK
CityNameRVSNDX_USAREA_WORK
ZIP3_HOME
HouseNumber_USADDR_HOME
HouseNumberSuffix_USADDR_HOME
StreetPrefixDirectional_USADDR_HOME
StreetPrefixType_USADDR_HOME

IN

Constraint:

Derivation	Column Name
OUT2.HouseNumberSuffix_USADDR_WORK	HouseNumberSuffix_USADDR_WORK
OUT2.StreetPrefixDirectional_USADDR_WORK	StreetPrefixDirectional_USADDR_WORK
OUT2.StreetPrefixType_USADDR_WORK	StreetPrefixType_USADDR_WORK
OUT2.StreetName_USADDR_WORK	StreetName_USADDR_WORK
OUT2.StreetSuffixType_USADDR_WORK	StreetSuffixType_USADDR_WORK
OUT2.StreetSuffixQualifier_USADDR_WORK	StreetSuffixQualifier_USADDR_WORK
OUT2.StreetSuffixDirectional_USADDR_WORK	StreetSuffixDirectional_USADDR_WORK
OUT2.RuralRouteType_USADDR_WORK	RuralRouteType_USADDR_WORK
OUT2.RuralRouteValue_USADDR_WORK	RuralRouteValue_USADDR_WORK
OUT2.BoxType_USADDR_WORK	BoxType_USADDR_WORK
OUT2.BoxValue_USADDR_WORK	BoxValue_USADDR_WORK
OUT2.FloorType_USADDR_WORK	FloorType_USADDR_WORK
OUT2.FloorValue_USADDR_WORK	FloorValue_USADDR_WORK
OUT2.UnitType_USADDR_WORK	UnitType_USADDR_WORK
OUT2.UnitValue_USADDR_WORK	UnitValue_USADDR_WORK
OUT2.MultiUnitType_USADDR_WORK	MultiUnitType_USADDR_WORK
OUT2.MultiUnitValue_USADDR_WORK	MultiUnitValue_USADDR_WORK
OUT2.BuildingName_USADDR_WORK	BuildingName_USADDR_WORK
OUT2.AdditionalAddress_USADDR_WORK	AdditionalAddress_USADDR_WORK
OUT2.AddressType_USADDR_WORK	AddressType_USADDR_WORK
OUT2.StreetNameNYSIIS_USADDR_WORK	StreetNameNYSIIS_USADDR_WORK
OUT2.StreetNameRVSNDX_USADDR_WORK	StreetNameRVSNDX_USADDR_WORK
OUT2.UnhandledPattern_USADDR_WORK	UnhandledPattern_USADDR_WORK
OUT2.UnhandledData_USADDR_WORK	UnhandledData_USADDR_WORK
OUT2.InputPattern_USADDR_WORK	InputPattern_USADDR_WORK
OUT2.ExceptionData_USADDR_WORK	ExceptionData_USADDR_WORK
OUT2.UserOverrideFlag_USADDR_WORK	UserOverrideFlag_USADDR_WORK
OUT2.CityName_USAREA_WORK	CityName_USAREA_WORK
OUT2.StateAbbreviation_USAREA_WORK	StateAbbreviation_USAREA_WORK
OUT2.ZipCode_USAREA_WORK	ZipCode_USAREA_WORK
OUT2.Zip4AddonCode_USAREA_WORK	Zip4AddonCode_USAREA_WORK
OUT2.CountryCode_USAREA_WORK	CountryCode_USAREA_WORK
OUT2.CityNameNYSIIS_USAREA_WORK	CityNameNYSIIS_USAREA_WORK
OUT2.CityNameRVSNDX_USAREA_WORK	CityNameRVSNDX_USAREA_WORK

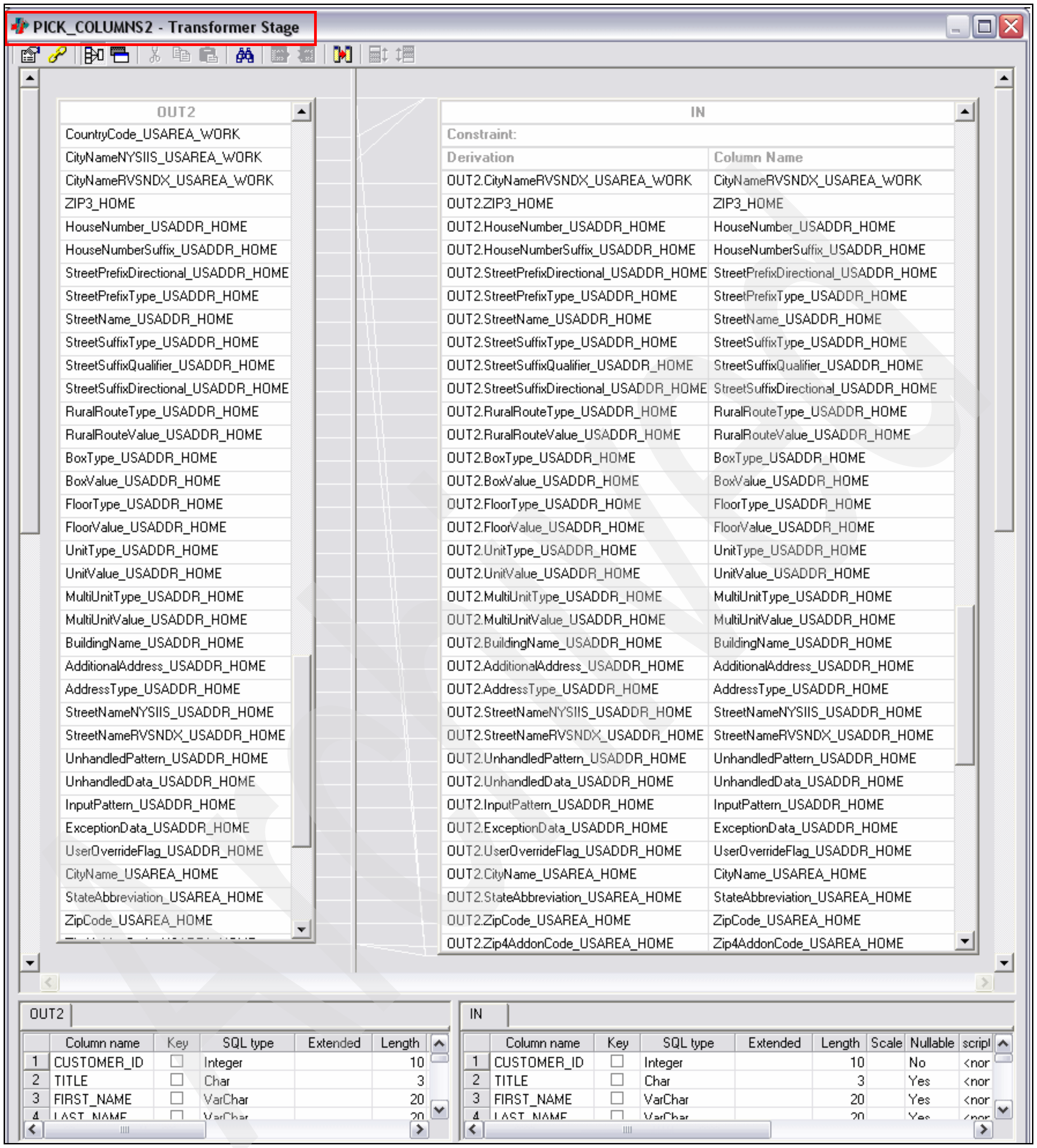
OUT2

	Column name	Key	SQL type	Extended	Length
1	CUSTOMER_ID	☐	Integer		10
2	TITLE	☐	Char		3
3	FIRST_NAME	☐	VarChar		20
4	LAST_NAME	☐	VarChar		20

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripl
1	CUSTOMER_ID	☐	Integer		10		No	<nor
2	TITLE	☐	Char		3		Yes	<nor
3	FIRST_NAME	☐	VarChar		20		Yes	<nor
4	LAST_NAME	☐	VarChar		20		Yes	<nor

Figure 2-330 Create j12_PREPARE_NCB_NAB_DATA_FOR_FUNNEL 9/12


PICK_COLUMNS2 - Transformer Stage

OUT2
CountryCode_USAREA_WORK
CityNameNYSIIS_USAREA_WORK
CityNameRVSNDX_USAREA_WORK
ZIP3_HOME
HouseNumber_USADDR_HOME
HouseNumberSuffix_USADDR_HOME
StreetPrefixDirectional_USADDR_HOME
StreetPrefixType_USADDR_HOME
StreetName_USADDR_HOME
StreetSuffixType_USADDR_HOME
StreetSuffixQualifier_USADDR_HOME
StreetSuffixDirectional_USADDR_HOME
RuralRouteType_USADDR_HOME
RuralRouteValue_USADDR_HOME
BoxType_USADDR_HOME
BoxValue_USADDR_HOME
FloorType_USADDR_HOME
FloorValue_USADDR_HOME
UnitType_USADDR_HOME
UnitValue_USADDR_HOME
MultiUnitType_USADDR_HOME
MultiUnitValue_USADDR_HOME
BuildingName_USADDR_HOME
AdditionalAddress_USADDR_HOME
AddressType_USADDR_HOME
StreetNameNYSIIS_USADDR_HOME
StreetNameRVSNDX_USADDR_HOME
UnhandledPattern_USADDR_HOME
UnhandledData_USADDR_HOME
InputPattern_USADDR_HOME
ExceptionData_USADDR_HOME
UserOverrideFlag_USADDR_HOME
CityName_USAREA_HOME
StateAbbreviation_USAREA_HOME
ZipCode_USAREA_HOME

IN

Constraint:

Derivation	Column Name
OUT2.CityNameRVSNDX_USAREA_WORK	CityNameRVSNDX_USAREA_WORK
OUT2.ZIP3_HOME	ZIP3_HOME
OUT2.HouseNumber_USADDR_HOME	HouseNumber_USADDR_HOME
OUT2.HouseNumberSuffix_USADDR_HOME	HouseNumberSuffix_USADDR_HOME
OUT2.StreetPrefixDirectional_USADDR_HOME	StreetPrefixDirectional_USADDR_HOME
OUT2.StreetPrefixType_USADDR_HOME	StreetPrefixType_USADDR_HOME
OUT2.StreetName_USADDR_HOME	StreetName_USADDR_HOME
OUT2.StreetSuffixType_USADDR_HOME	StreetSuffixType_USADDR_HOME
OUT2.StreetSuffixQualifier_USADDR_HOME	StreetSuffixQualifier_USADDR_HOME
OUT2.StreetSuffixDirectional_USADDR_HOME	StreetSuffixDirectional_USADDR_HOME
OUT2.RuralRouteType_USADDR_HOME	RuralRouteType_USADDR_HOME
OUT2.RuralRouteValue_USADDR_HOME	RuralRouteValue_USADDR_HOME
OUT2.BoxType_USADDR_HOME	BoxType_USADDR_HOME
OUT2.BoxValue_USADDR_HOME	BoxValue_USADDR_HOME
OUT2.FloorType_USADDR_HOME	FloorType_USADDR_HOME
OUT2.FloorValue_USADDR_HOME	FloorValue_USADDR_HOME
OUT2.UnitType_USADDR_HOME	UnitType_USADDR_HOME
OUT2.UnitValue_USADDR_HOME	UnitValue_USADDR_HOME
OUT2.MultiUnitType_USADDR_HOME	MultiUnitType_USADDR_HOME
OUT2.MultiUnitValue_USADDR_HOME	MultiUnitValue_USADDR_HOME
OUT2.BuildingName_USADDR_HOME	BuildingName_USADDR_HOME
OUT2.AdditionalAddress_USADDR_HOME	AdditionalAddress_USADDR_HOME
OUT2.AddressType_USADDR_HOME	AddressType_USADDR_HOME
OUT2.StreetNameNYSIIS_USADDR_HOME	StreetNameNYSIIS_USADDR_HOME
OUT2.StreetNameRVSNDX_USADDR_HOME	StreetNameRVSNDX_USADDR_HOME
OUT2.UnhandledPattern_USADDR_HOME	UnhandledPattern_USADDR_HOME
OUT2.UnhandledData_USADDR_HOME	UnhandledData_USADDR_HOME
OUT2.InputPattern_USADDR_HOME	InputPattern_USADDR_HOME
OUT2.ExceptionData_USADDR_HOME	ExceptionData_USADDR_HOME
OUT2.UserOverrideFlag_USADDR_HOME	UserOverrideFlag_USADDR_HOME
OUT2.CityName_USAREA_HOME	CityName_USAREA_HOME
OUT2.StateAbbreviation_USAREA_HOME	StateAbbreviation_USAREA_HOME
OUT2.ZipCode_USAREA_HOME	ZipCode_USAREA_HOME
OUT2.Zip4AddonCode_USAREA_HOME	Zip4AddonCode_USAREA_HOME

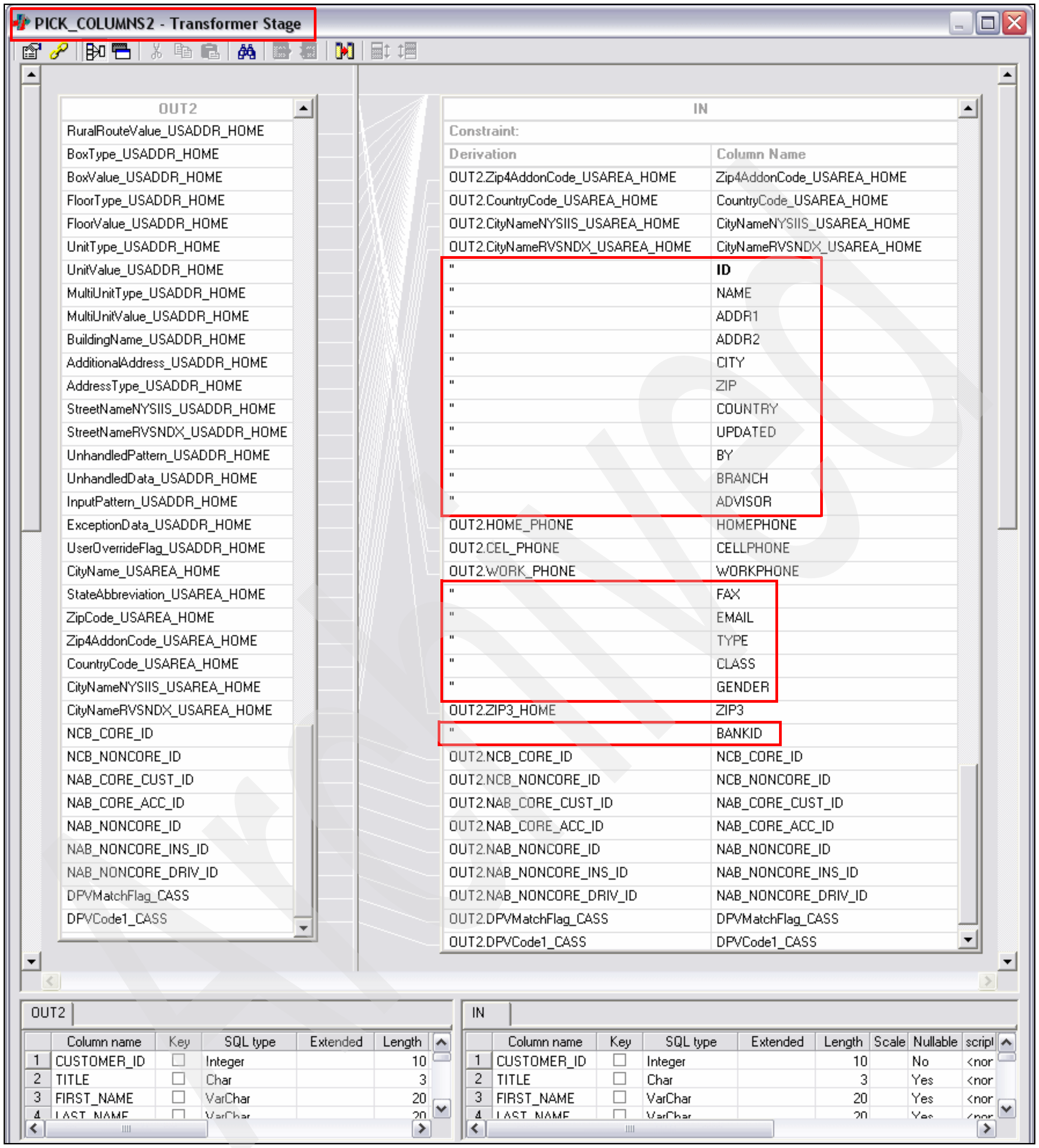
OUT2

	Column name	Key	SQL type	Extended	Length
1	CUSTOMER_ID	☐	Integer		10
2	TITLE	☐	Char		3
3	FIRST_NAME	☐	VarChar		20
4	LAST_NAME	☐	VarChar		20

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scrip
1	CUSTOMER_ID	☐	Integer		10		No	<nor
2	TITLE	☐	Char		3		Yes	<nor
3	FIRST_NAME	☐	VarChar		20		Yes	<nor
4	LAST_NAME	☐	VarChar		20		Yes	<nor

Figure 2-331 Create j12_PREPARE_NCB_NAB_DATA_FOR_FUNNEL 10/12



PICK_COLUMNS2 - Transformer Stage

OUT2
RuralRouteValue_USADDR_HOME
BoxType_USADDR_HOME
BoxValue_USADDR_HOME
FloorType_USADDR_HOME
FloorValue_USADDR_HOME
UnitType_USADDR_HOME
UnitValue_USADDR_HOME
MultiUnitType_USADDR_HOME
MultiUnitValue_USADDR_HOME
BuildingName_USADDR_HOME
AdditionalAddress_USADDR_HOME
AddressType_USADDR_HOME
StreetNameNYSIIS_USADDR_HOME
StreetNameRVSNDX_USADDR_HOME
UnhandledPattern_USADDR_HOME
UnhandledData_USADDR_HOME
InputPattern_USADDR_HOME
ExceptionData_USADDR_HOME
UserOverrideFlag_USADDR_HOME
CityName_USAREA_HOME
StateAbbreviation_USAREA_HOME
ZipCode_USAREA_HOME
Zip4AddonCode_USAREA_HOME
CountryCode_USAREA_HOME
CityNameNYSIIS_USAREA_HOME
CityNameRVSNDX_USAREA_HOME
NCB_CORE_ID
NCB_NONCORE_ID
NAB_CORE_CUST_ID
NAB_CORE_ACC_ID
NAB_NONCORE_ID
NAB_NONCORE_INS_ID
NAB_NONCORE_DRIV_ID
DPVMatchFlag_CASS
DPVCode1_CASS

IN

Constraint:

Derivation	Column Name
OUT2.Zip4AddonCode_USAREA_HOME	Zip4AddonCode_USAREA_HOME
OUT2.CountryCode_USAREA_HOME	CountryCode_USAREA_HOME
OUT2.CityNameNYSIIS_USAREA_HOME	CityNameNYSIIS_USAREA_HOME
OUT2.CityNameRVSNDX_USAREA_HOME	CityNameRVSNDX_USAREA_HOME
"	ID
"	NAME
"	ADDR1
"	ADDR2
"	CITY
"	ZIP
"	COUNTRY
"	UPDATED
"	BY
"	BRANCH
"	ADVISOR
OUT2.HOME_PHONE	HOMEPHONE
OUT2.CEL_PHONE	CELLPHONE
OUT2.WORK_PHONE	WORKPHONE
"	FAX
"	EMAIL
"	TYPE
"	CLASS
"	GENDER
OUT2.ZIP3_HOME	ZIP3
"	BANKID
OUT2.NCB_CORE_ID	NCB_CORE_ID
OUT2.NCB_NONCORE_ID	NCB_NONCORE_ID
OUT2.NAB_CORE_CUST_ID	NAB_CORE_CUST_ID
OUT2.NAB_CORE_ACC_ID	NAB_CORE_ACC_ID
OUT2.NAB_NONCORE_ID	NAB_NONCORE_ID
OUT2.NAB_NONCORE_INS_ID	NAB_NONCORE_INS_ID
OUT2.NAB_NONCORE_DRIV_ID	NAB_NONCORE_DRIV_ID
OUT2.DPVMatchFlag_CASS	DPVMatchFlag_CASS
OUT2.DPVCode1_CASS	DPVCode1_CASS

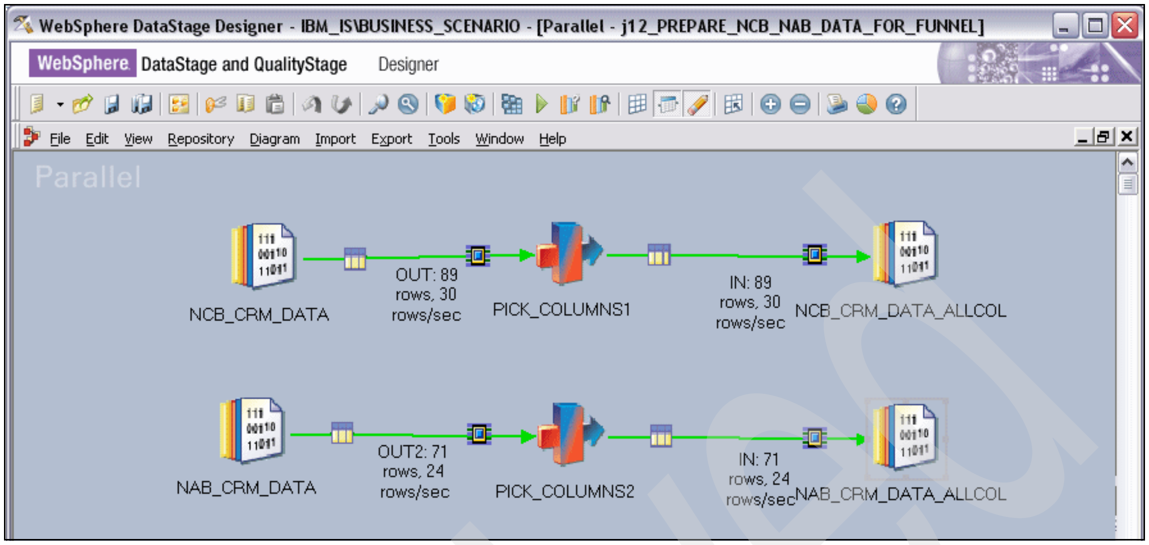
OUT2

	Column name	Key	SQL type	Extended	Length
1	CUSTOMER_ID	☐	Integer		10
2	TITLE	☐	Char		3
3	FIRST_NAME	☐	VarChar		20
4	LAST_NAME	☐	VarChar		20

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripl
1	CUSTOMER_ID	☐	Integer		10		No	<nor
2	TITLE	☐	Char		3		Yes	<nor
3	FIRST_NAME	☐	VarChar		20		Yes	<nor
4	LAST_NAME	☐	VarChar		20		Yes	<nor

Figure 2-332 Create j12_PREPARE_NCB_NAB_DATA_FOR_FUNNEL 11/12



WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j12_PREPARE_NCB_NAB_DATA_FOR_FUNNEL]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NCB_CRM_DATA
OUT: 89 rows, 30 rows/sec
PICK_COLUMNS1
IN: 89 rows, 30 rows/sec
NCB_CRM_DATA_ALLCOL
NAB_CRM_DATA
OUT2: 71 rows, 24 rows/sec
PICK_COLUMNS2
IN: 71 rows, 24 rows/sec
NAB_CRM_DATA_ALLCOL

Figure 2-333 Create j12_PREPARE_NCB_NAB_DATA_FOR_FUNNEL 12/12

j13_FUNNEL_NCB_NAB_CRM_DATA

In this step, we merge the two files that were created in “j12_PREPARE_NCB_NAB_DATA_FOR_FUNNEL” on page 785 using the Funnel stage.

Figure 2-334 on page 798 shows the various stages that are used in this job. It includes the data sets that were created in “j12_PREPARE_NCB_NAB_DATA_FOR_FUNNEL” on page 785, a Funnel stage, a Transformer stage to handle nulls, and an output Data Set stage. We modified the stage names as shown.

Because we have discussed such configurations previously, we do not repeat them here.

Figure 2-334 on page 798 shows the results after compiling and running this job.

Proceed now to “j14_CRM_FREQUENCY” on page 798.

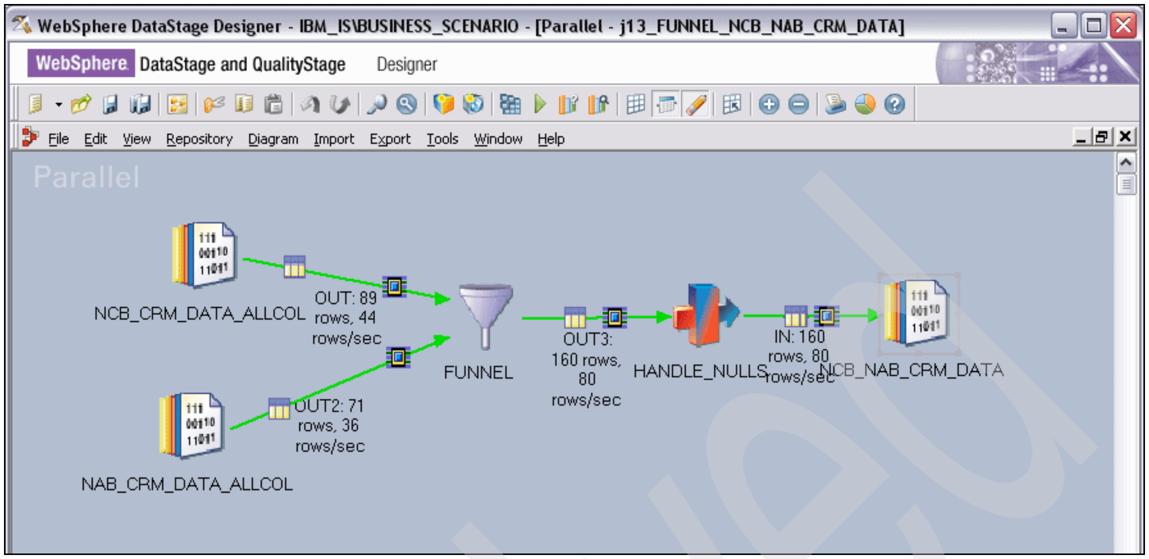

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j13_FUNNEL_NCB_NAB_CRM_DATA]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NCB_CRM_DATA_ALLCOL
OUT: 89 rows, 44 rows/sec
FUNNEL
OUT2: 71 rows, 36 rows/sec
NAB_CRM_DATA_ALLCOL
OUT3: 160 rows, 80 rows/sec
HANDLE_NULLS
IN: 160 rows, 80 rows/sec
NCB_NAB_CRM_DATA

Figure 2-334 Create j13_FUNNEL_NCB_NAB_CRM_DATA

j14_CRM_FREQUENCY

Prior to matching, we generate the frequency distribution on all the columns in the merged file in the previous step "“j13_FUNNEL_NCB_NAB_CRM_DATA” on page 797" using the Match Frequency stage. Here again, the idea is to generate match frequency for all the columns so that it can be used with any match specification.

Figure 2-335 on page 799 shows the various stages that are used in this job, including the data set that was created in "j13_FUNNEL_NCB_NAB_CRM_DATA" on page 797, a Match Frequency stage, and an output Data Set stage. We modified the names of the stages as shown.

Because the configuration of this job is very similar to that described in "J10_MATCHFREQ_STAN_CUSTOMER" on page 266, we do not repeat it here.

After saving, compiling, and running this job (Figure 2-335 on page 799), the contents of the output of this stage are listed in Figure 2-336 on page 800 and Figure 2-337 on page 801. We do not document the interpretation of the format and content of this file.

Proceed to "j14a_MATCHSPEC" on page 801.

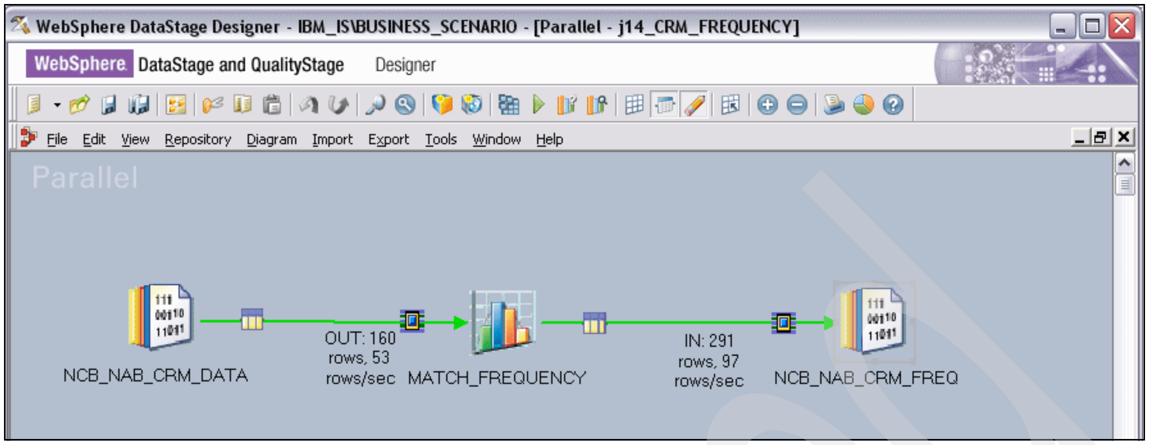
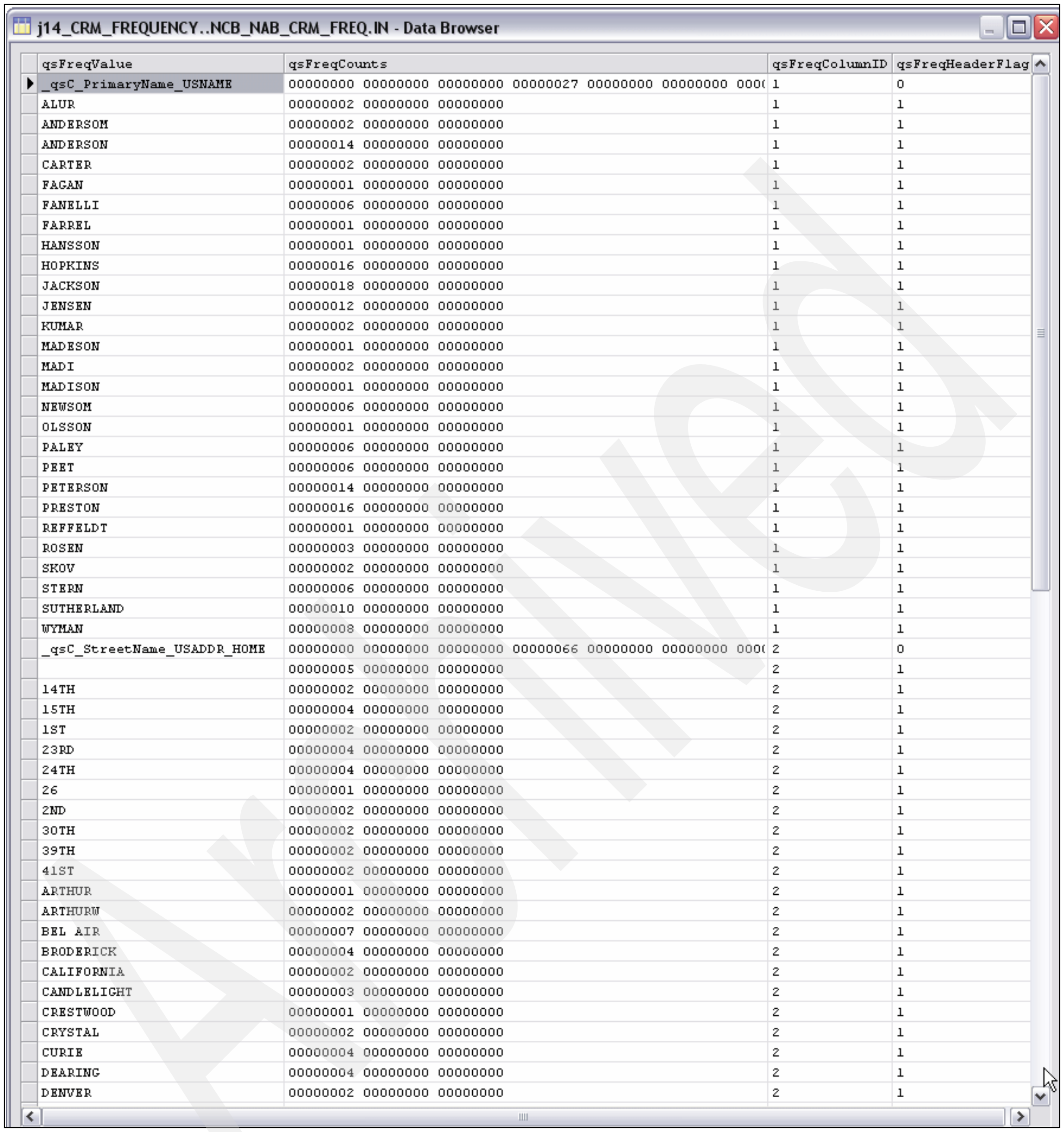

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j14_CRM_FREQUENCY]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NCB_NAB_CRM_DATA
OUT: 160 rows, 53 rows/sec
MATCH_FREQUENCY
IN: 291 rows, 97 rows/sec
NCB_NAB_CRM_FREQ

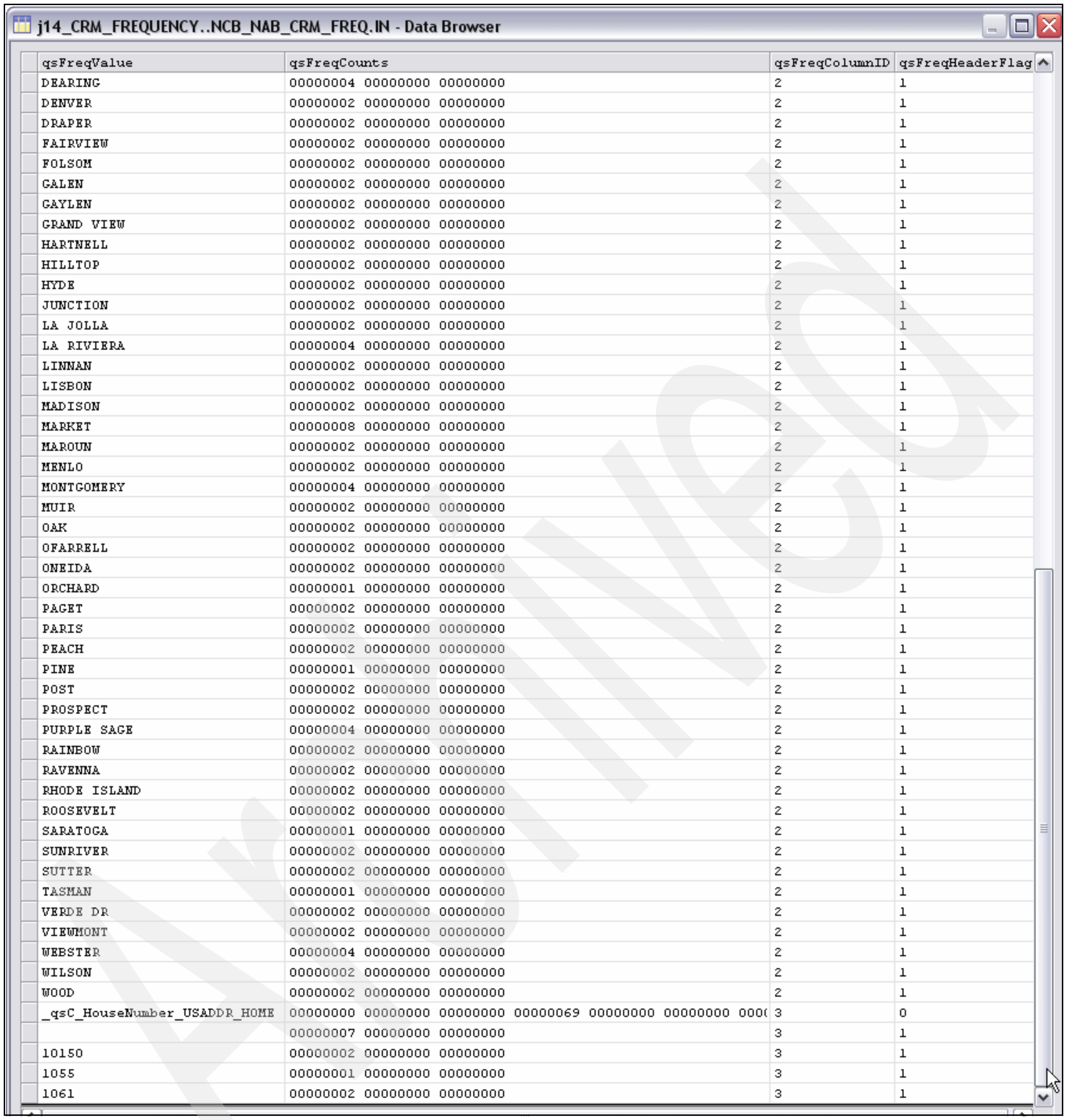
Figure 2-335 Create j14_CRM_FREQUENCY 1/3



j14_CRM_FREQUENCY..NCB_NAB_CRM_FREQ.IN - Data Browser

qsFreqValue	qsFreqCounts	qsFreqColumnID	qsFreqHeaderFlag
_qsC_PrimaryName_USNAME	00000000 00000000 00000000 00000027 00000000 00000000 000(	1	0
ALUR	00000002 00000000 00000000	1	1
ANDERSOM	00000002 00000000 00000000	1	1
ANDERSON	00000014 00000000 00000000	1	1
CARTER	00000002 00000000 00000000	1	1
FAGAN	00000001 00000000 00000000	1	1
FANELLI	00000006 00000000 00000000	1	1
FARREL	00000001 00000000 00000000	1	1
HANSSON	00000001 00000000 00000000	1	1
HOPKINS	00000016 00000000 00000000	1	1
JACKSON	00000018 00000000 00000000	1	1
JENSEN	00000012 00000000 00000000	1	1
KUMAR	00000002 00000000 00000000	1	1
MADESON	00000001 00000000 00000000	1	1
MADI	00000002 00000000 00000000	1	1
MADISON	00000001 00000000 00000000	1	1
NEWSOM	00000006 00000000 00000000	1	1
OLSSON	00000001 00000000 00000000	1	1
PALEY	00000006 00000000 00000000	1	1
PEET	00000006 00000000 00000000	1	1
PETERSON	00000014 00000000 00000000	1	1
PRESTON	00000016 00000000 00000000	1	1
REFFELDT	00000001 00000000 00000000	1	1
ROSEN	00000003 00000000 00000000	1	1
SKOV	00000002 00000000 00000000	1	1
STERN	00000006 00000000 00000000	1	1
SUTHERLAND	00000010 00000000 00000000	1	1
WYMAN	00000008 00000000 00000000	1	1
_qsC_StreetName_USADDR_HOME	00000000 00000000 00000000 00000066 00000000 00000000 000(	2	0
	00000005 00000000 00000000	2	1
14TH	00000002 00000000 00000000	2	1
15TH	00000004 00000000 00000000	2	1
1ST	00000002 00000000 00000000	2	1
23RD	00000004 00000000 00000000	2	1
24TH	00000004 00000000 00000000	2	1
26	00000001 00000000 00000000	2	1
2ND	00000002 00000000 00000000	2	1
30TH	00000002 00000000 00000000	2	1
39TH	00000002 00000000 00000000	2	1
41ST	00000002 00000000 00000000	2	1
ARTHUR	00000001 00000000 00000000	2	1
ARTHURW	00000002 00000000 00000000	2	1
BEL AIR	00000007 00000000 00000000	2	1
BRODERICK	00000004 00000000 00000000	2	1
CALIFORNIA	00000002 00000000 00000000	2	1
CANDLELIGHT	00000003 00000000 00000000	2	1
CRESTWOOD	00000001 00000000 00000000	2	1
CRYSTAL	00000002 00000000 00000000	2	1
CURIE	00000004 00000000 00000000	2	1
DEARING	00000004 00000000 00000000	2	1
DENVER	00000002 00000000 00000000	2	1

Figure 2-336 Create j14_CRM_FREQUENCY 2/3


j14_CRM_FREQUENCY..NCB_NAB_CRM_FREQ.IN - Data Browser

qsFreqValue	qsFreqCounts	qsFreqColumnID	qsFreqHeaderFlag
DEARING	00000004 00000000 00000000	2	1
DENVER	00000002 00000000 00000000	2	1
DRAPER	00000002 00000000 00000000	2	1
FAIRVIEW	00000002 00000000 00000000	2	1
FOLSOM	00000002 00000000 00000000	2	1
GALEN	00000002 00000000 00000000	2	1
GAYLEN	00000002 00000000 00000000	2	1
GRAND VIEW	00000002 00000000 00000000	2	1
HARTNELL	00000002 00000000 00000000	2	1
HILLTOP	00000002 00000000 00000000	2	1
HYDE	00000002 00000000 00000000	2	1
JUNCTION	00000002 00000000 00000000	2	1
LA JOLLA	00000002 00000000 00000000	2	1
LA RIVIERA	00000004 00000000 00000000	2	1
LINNAN	00000002 00000000 00000000	2	1
LISBON	00000002 00000000 00000000	2	1
MADISON	00000002 00000000 00000000	2	1
MARKET	00000008 00000000 00000000	2	1
MAROUN	00000002 00000000 00000000	2	1
MENLO	00000002 00000000 00000000	2	1
MONTGOMERY	00000004 00000000 00000000	2	1
MUIR	00000002 00000000 00000000	2	1
OAK	00000002 00000000 00000000	2	1
OFARRELL	00000002 00000000 00000000	2	1
ONEIDA	00000002 00000000 00000000	2	1
ORCHARD	00000001 00000000 00000000	2	1
PAGET	00000002 00000000 00000000	2	1
PARIS	00000002 00000000 00000000	2	1
PEACH	00000002 00000000 00000000	2	1
PINE	00000001 00000000 00000000	2	1
POST	00000002 00000000 00000000	2	1
PROSPECT	00000002 00000000 00000000	2	1
PURPLE SAGE	00000004 00000000 00000000	2	1
RAINBOW	00000002 00000000 00000000	2	1
RAVENNA	00000002 00000000 00000000	2	1
RHODE ISLAND	00000002 00000000 00000000	2	1
ROOSEVELT	00000002 00000000 00000000	2	1
SARATOGA	00000001 00000000 00000000	2	1
SUNRIVER	00000002 00000000 00000000	2	1
SUTTER	00000002 00000000 00000000	2	1
TASMAN	00000001 00000000 00000000	2	1
VERDE DR	00000002 00000000 00000000	2	1
VIEWMONT	00000002 00000000 00000000	2	1
WEBSTER	00000004 00000000 00000000	2	1
WILSON	00000002 00000000 00000000	2	1
WOOD	00000002 00000000 00000000	2	1
_qsC_HouseNumber_USADDR_HOME	00000000 00000000 00000000 00000069 00000000 00000000 0000(	3	0
	00000007 00000000 00000000	3	1
10150	00000002 00000000 00000000	3	1
1055	00000001 00000000 00000000	3	1
1061	00000002 00000000 00000000	3	1

Figure 2-337 Create j14_CRM_FREQUENCY 3/3

j14a_MATCHSPEC

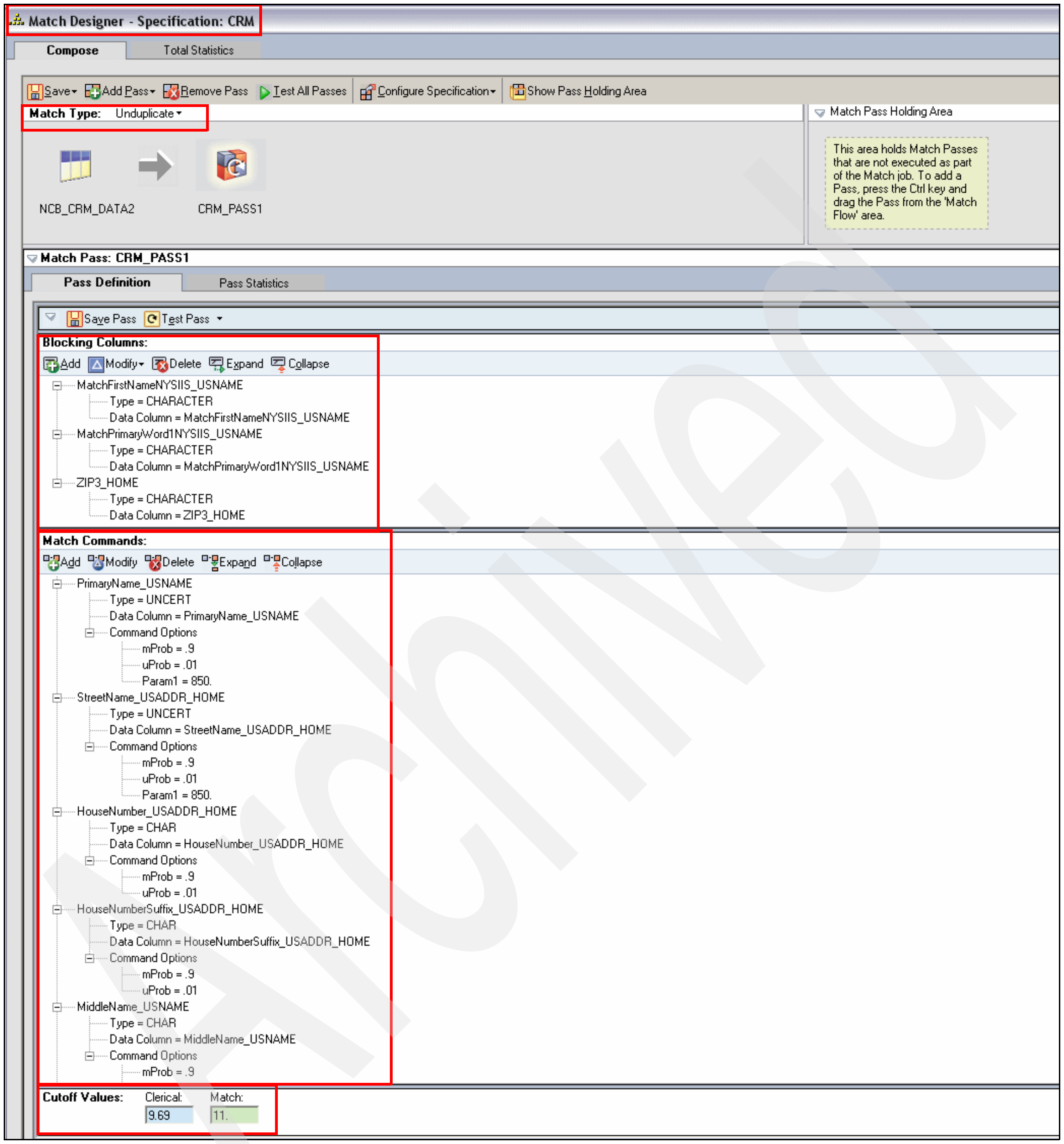
Next, we generate a match specification for an Unduplicate match stage using as input the match frequency data created in the “j14_CRM_FREQUENCY” on page 798 job. The specification included a single pass that blocked on a two phonetic encoding (NYSIIS) columns MatchFirstNameNYSIIS_USNAME, MatchPrimaryWord1NYSIIS_USNAME, and the 3-digit ZIP code column ZIP3_HOME corresponding to the home address.

Note: The work address only appears in the North American Bank's core services and therefore requires no matching and survive. Therefore, the ZIP code of the work address is not relevant here.

Because the creation of a match specification is described in "J15_Undup_MatchSpec_CUSTOMER" on page 318, we do not repeat it here.

Figure 2-338 on page 803 through Figure 2-344 on page 809 show the configuration of the match specification with the Blocking Columns, Match Commands, and Cutoff Values. It also shows the Test Results of the execution of the CRM_PASS1 pass. The Pass Statistics are not shown here.

The test of the match specification with the full volume of data appeared to deliver results that were accurate. This specification was then used in a Unduplicate match stage as described in "j15_UNDUP_CRM" on page 809.



Match Designer - Specification: CRM

Compose | Total Statistics

Save | Add Pass | Remove Pass | Test All Passes | Configure Specification | Show Pass Holding Area

Match Type: Undupliçate

NCB_CRM_DATA2 → CRM_PASS1

Match Pass Holding Area

This area holds Match Passes that are not executed as part of the Match job. To add a Pass, press the Ctrl key and drag the Pass from the 'Match Flow' area.

Match Pass: CRM_PASS1

Pass Definition | Pass Statistics

Save Pass | Test Pass

Blocking Columns:

Add | Modify | Delete | Expand | Collapse

```
MatchFirstNameNYSIIS_USNAME
    Type = CHARACTER
    Data Column = MatchFirstNameNYSIIS_USNAME
MatchPrimaryWord1NYSIIS_USNAME
    Type = CHARACTER
    Data Column = MatchPrimaryWord1NYSIIS_USNAME
ZIP3_HOME
    Type = CHARACTER
    Data Column = ZIP3_HOME
```

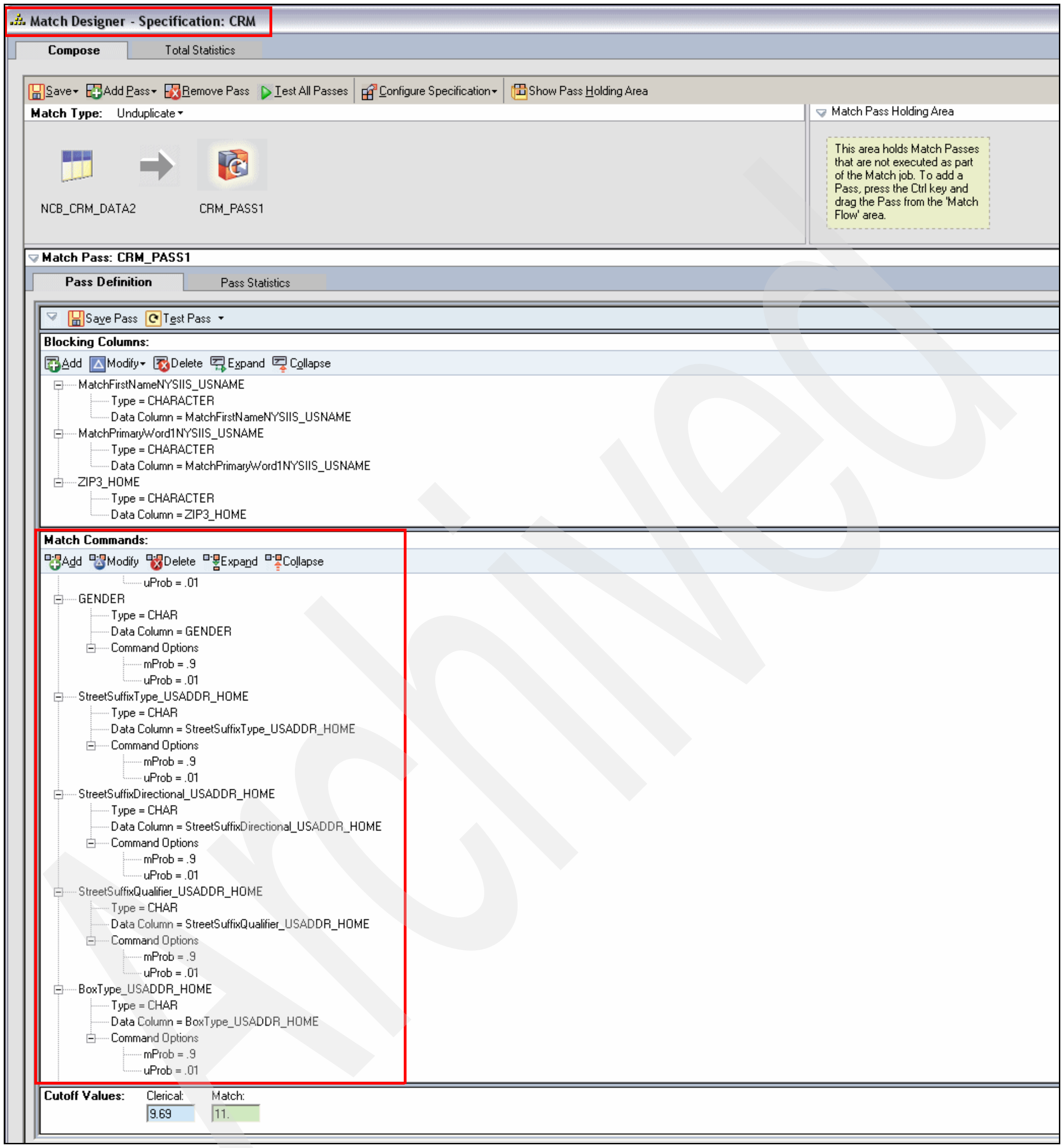
Match Commands:

Add | Modify | Delete | Expand | Collapse

```
PrimaryName_USNAME
    Type = UNCERT
    Data Column = PrimaryName_USNAME
    Command Options
        mProb = .9
        uProb = .01
        Param1 = 850.
StreetName_USADDR_HOME
    Type = UNCERT
    Data Column = StreetName_USADDR_HOME
    Command Options
        mProb = .9
        uProb = .01
        Param1 = 850.
HouseNumber_USADDR_HOME
    Type = CHAR
    Data Column = HouseNumber_USADDR_HOME
    Command Options
        mProb = .9
        uProb = .01
HouseNumberSuffix_USADDR_HOME
    Type = CHAR
    Data Column = HouseNumberSuffix_USADDR_HOME
    Command Options
        mProb = .9
        uProb = .01
MiddleName_USNAME
    Type = CHAR
    Data Column = MiddleName_USNAME
    Command Options
        mProb = .9
```

Cutoff Values: Clerical: 9.69 Match: 11.

Figure 2-338 Create j14a_MATCHSPEC 1/7


Match Designer - Specification: CRM

Compose | Total Statistics

Save | Add Pass | Remove Pass | Test All Passes | Configure Specification | Show Pass Holding Area

Match Type: Undupplicate

NCB_CRM_DATA2 → CRM_PASS1

Match Pass Holding Area

This area holds Match Passes that are not executed as part of the Match job. To add a Pass, press the Ctrl key and drag the Pass from the 'Match Flow' area.

Match Pass: CRM_PASS1

Pass Definition | Pass Statistics

Save Pass | Test Pass

Blocking Columns:

Add | Modify | Delete | Expand | Collapse

```
MatchFirstNameNYSIIS_USNAME
    Type = CHARACTER
    Data Column = MatchFirstNameNYSIIS_USNAME
MatchPrimaryWord1NYSIIS_USNAME
    Type = CHARACTER
    Data Column = MatchPrimaryWord1NYSIIS_USNAME
ZIP3_HOME
    Type = CHARACTER
    Data Column = ZIP3_HOME
```

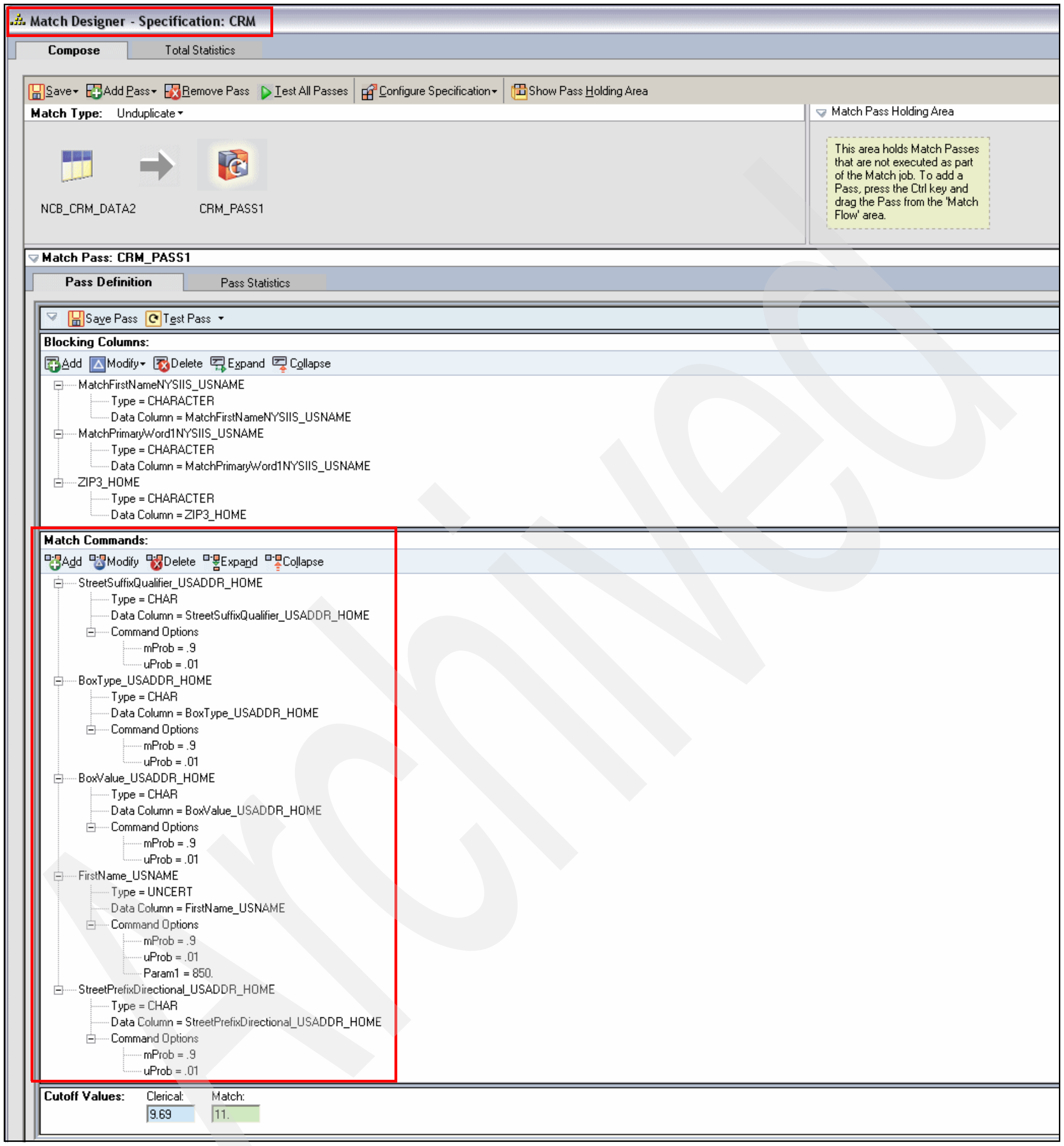
Match Commands:

Add | Modify | Delete | Expand | Collapse

```
        uProb = .01
GENDER
    Type = CHAR
    Data Column = GENDER
    Command Options
        mProb = .9
        uProb = .01
StreetSuffixType_USADDR_HOME
    Type = CHAR
    Data Column = StreetSuffixType_USADDR_HOME
    Command Options
        mProb = .9
        uProb = .01
StreetSuffixDirectional_USADDR_HOME
    Type = CHAR
    Data Column = StreetSuffixDirectional_USADDR_HOME
    Command Options
        mProb = .9
        uProb = .01
StreetSuffixQualifier_USADDR_HOME
    Type = CHAR
    Data Column = StreetSuffixQualifier_USADDR_HOME
    Command Options
        mProb = .9
        uProb = .01
BoxType_USADDR_HOME
    Type = CHAR
    Data Column = BoxType_USADDR_HOME
    Command Options
        mProb = .9
        uProb = .01
```

Cutoff Values: Clerical: 9.69 Match: 11.

Figure 2-339 Create j14a_MATCHSPEC 2/7



Match Type: Undupliate

This area holds Match Passes that are not executed as part of the Match job. To add a Pass, press the Ctrl key and drag the Pass from the 'Match Flow' area.

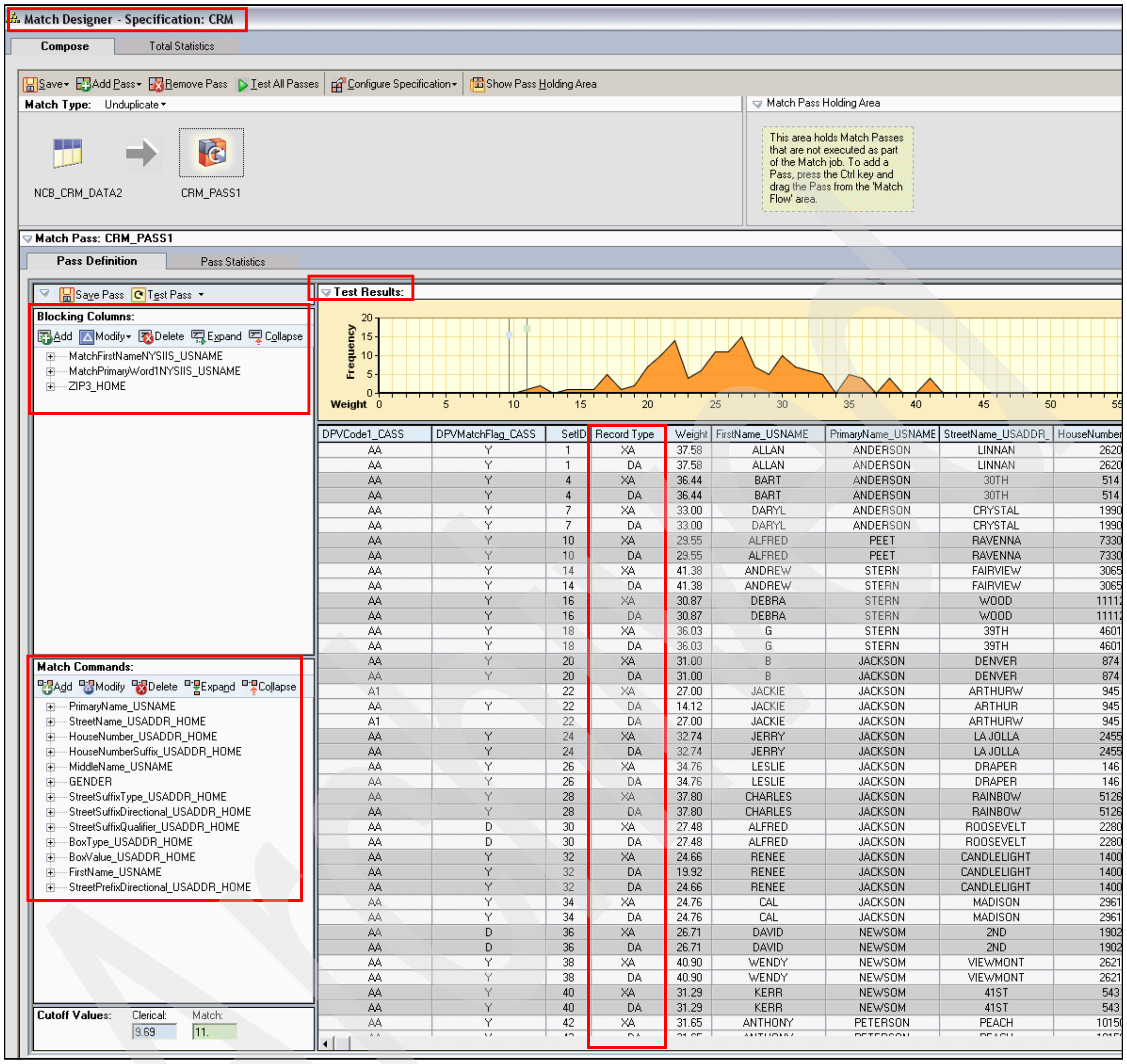
Match Pass: CRM_PASS1

```
Blocking Columns:
MatchFirstNameNYSIIS_USNAME
    Type = CHARACTER
    Data Column = MatchFirstNameNYSIIS_USNAME
MatchPrimaryWord1NYSIIS_USNAME
    Type = CHARACTER
    Data Column = MatchPrimaryWord1NYSIIS_USNAME
ZIP3_HOME
    Type = CHARACTER
    Data Column = ZIP3_HOME

Match Commands:
StreetSuffixQualifier_USADDR_HOME
    Type = CHAR
    Data Column = StreetSuffixQualifier_USADDR_HOME
    Command Options
        mProb = .9
        uProb = .01
BoxType_USADDR_HOME
    Type = CHAR
    Data Column = BoxType_USADDR_HOME
    Command Options
        mProb = .9
        uProb = .01
BoxValue_USADDR_HOME
    Type = CHAR
    Data Column = BoxValue_USADDR_HOME
    Command Options
        mProb = .9
        uProb = .01
FirstName_USNAME
    Type = UNCERT
    Data Column = FirstName_USNAME
    Command Options
        mProb = .9
        uProb = .01
        Param1 = 850.
StreetPrefixDirectional_USADDR_HOME
    Type = CHAR
    Data Column = StreetPrefixDirectional_USADDR_HOME
    Command Options
        mProb = .9
        uProb = .01

Cutoff Values:   Clerical: 9.69   Match: 11.
```

Figure 2-340 Create j14a_MATCHSPEC 3/7



This area holds Match Passes that are not executed as part of the Match job. To add a Pass, press the Ctrl key and drag the Pass from the 'Match Flow' area.

Blocking Columns:

- MatchFirstNameNYSIIS_USNAME
- MatchPrimaryWord1NYSIIS_USNAME
- ZIP3_HOME

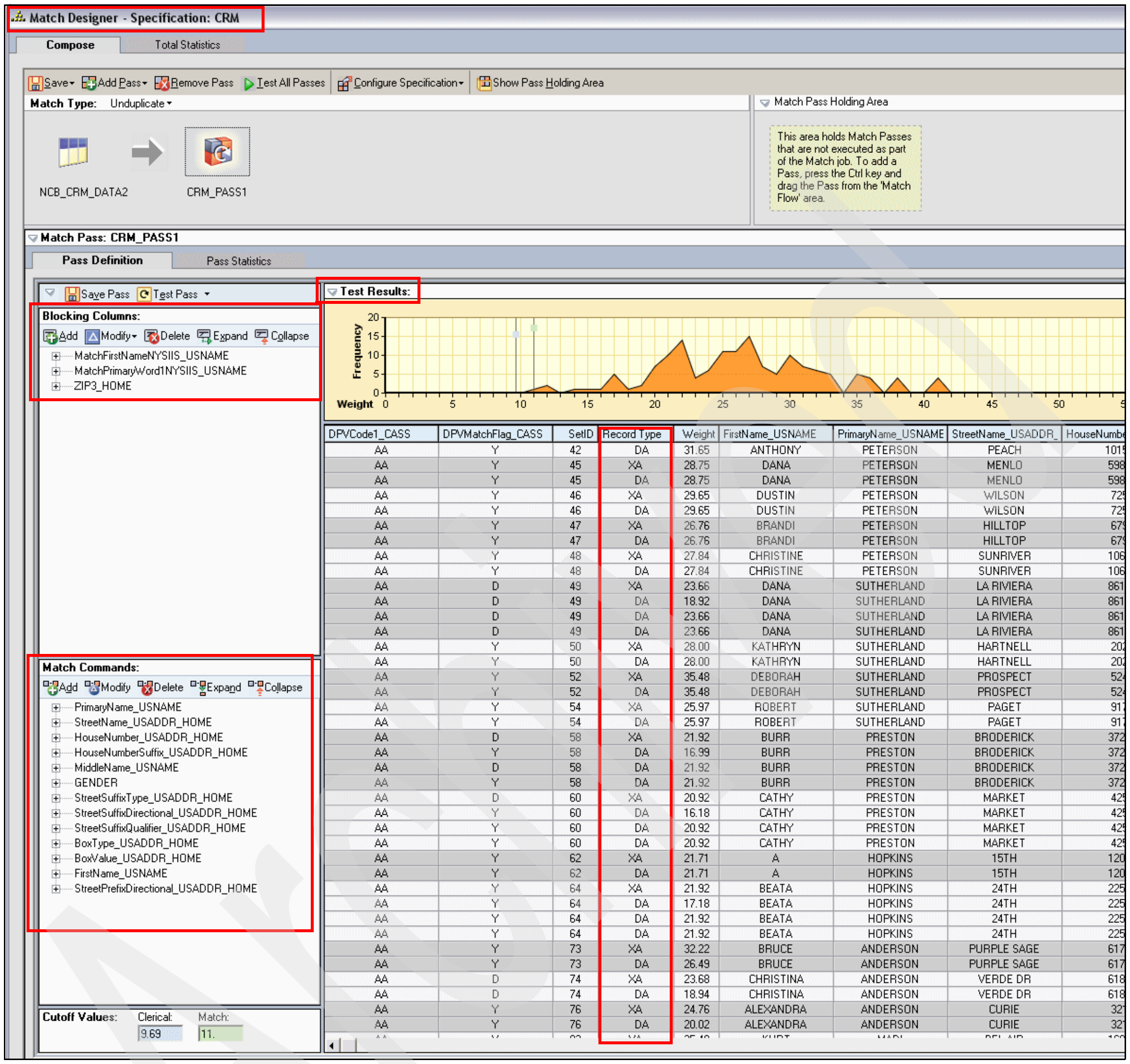
Match Commands:

- PrimaryName_USNAME
- StreetName_USADDR_HOME
- HouseNumber_USADDR_HOME
- HouseNumberSuffix_USADDR_HOME
- MiddleName_USNAME
- GENDER
- StreetSuffixType_USADDR_HOME
- StreetSuffixDirectional_USADDR_HOME
- StreetSuffixQualifier_USADDR_HOME
- BoxType_USADDR_HOME
- BoxValue_USADDR_HOME
- FirstName_USNAME
- StreetPrefixDirectional_USADDR_HOME

Cutoff Values: Clerical: 9.69 Match: 11.

DPVCode1_CASS	DPVMatchFlag_CASS	SetID	Record Type	Weight	FirstName_USNAME	PrimaryName_USNAME	StreetName_USADDR_	HouseNumber
AA	Y	1	XA	37.58	ALLAN	ANDERSON	LINNAN	2620
AA	Y	1	DA	37.58	ALLAN	ANDERSON	LINNAN	2620
AA	Y	4	XA	36.44	BART	ANDERSON	30TH	514
AA	Y	4	DA	36.44	BART	ANDERSON	30TH	514
AA	Y	7	XA	33.00	DARYL	ANDERSON	CRYSTAL	1990
AA	Y	7	DA	33.00	DARYL	ANDERSON	CRYSTAL	1990
AA	Y	10	XA	29.55	ALFRED	PEET	RAVENNA	7330
AA	Y	10	DA	29.55	ALFRED	PEET	RAVENNA	7330
AA	Y	14	XA	41.38	ANDREW	STERN	FAIRVIEW	3065
AA	Y	14	DA	41.38	ANDREW	STERN	FAIRVIEW	3065
AA	Y	16	XA	30.87	DEBRA	STERN	WOOD	1111
AA	Y	16	DA	30.87	DEBRA	STERN	WOOD	1111
AA	Y	18	XA	36.03	G	STERN	39TH	4601
AA	Y	18	DA	36.03	G	STERN	39TH	4601
AA	Y	20	XA	31.00	B	JACKSON	DENVER	874
AA	Y	20	DA	31.00	B	JACKSON	DENVER	874
A1		22	XA	27.00	JACKIE	JACKSON	ARTHURW	945
AA	Y	22	DA	14.12	JACKIE	JACKSON	ARTHUR	945
A1		22	DA	27.00	JACKIE	JACKSON	ARTHURW	945
AA	Y	24	XA	32.74	JERRY	JACKSON	LA JOLLA	2455
AA	Y	24	DA	32.74	JERRY	JACKSON	LA JOLLA	2455
AA	Y	26	XA	34.76	LESLIE	JACKSON	DRAPER	146
AA	Y	26	DA	34.76	LESLIE	JACKSON	DRAPER	146
AA	Y	28	XA	37.80	CHARLES	JACKSON	RAINBOW	5126
AA	Y	28	DA	37.80	CHARLES	JACKSON	RAINBOW	5126
AA	D	30	XA	27.48	ALFRED	JACKSON	ROOSEVELT	2280
AA	D	30	DA	27.48	ALFRED	JACKSON	ROOSEVELT	2280
AA	Y	32	XA	24.66	RENEE	JACKSON	CANDLELIGHT	1400
AA	Y	32	DA	19.92	RENEE	JACKSON	CANDLELIGHT	1400
AA	Y	32	DA	24.66	RENEE	JACKSON	CANDLELIGHT	1400
AA	Y	34	XA	24.76	CAL	JACKSON	MADISON	2961
AA	Y	34	DA	24.76	CAL	JACKSON	MADISON	2961
AA	D	36	XA	26.71	DAVID	NEWSOM	2ND	1902
AA	D	36	DA	26.71	DAVID	NEWSOM	2ND	1902
AA	Y	38	XA	40.90	WENDY	NEWSOM	VIEWMONT	2621
AA	Y	38	DA	40.90	WENDY	NEWSOM	VIEWMONT	2621
AA	Y	40	XA	31.29	KERR	NEWSOM	41ST	543
AA	Y	40	DA	31.29	KERR	NEWSOM	41ST	543
AA	Y	42	XA	31.65	ANTHONY	PETERSON	PEACH	10150

Figure 2-341 Create j14a_MATCHSPEC 4/7



This area holds Match Passes that are not executed as part of the Match job. To add a Pass, press the Ctrl key and drag the Pass from the 'Match Flow' area.

Blocking Columns:

- MatchFirstNameNYSIIS_USNAME
- MatchPrimaryWord1NYSIIS_USNAME
- ZIP3_HOME

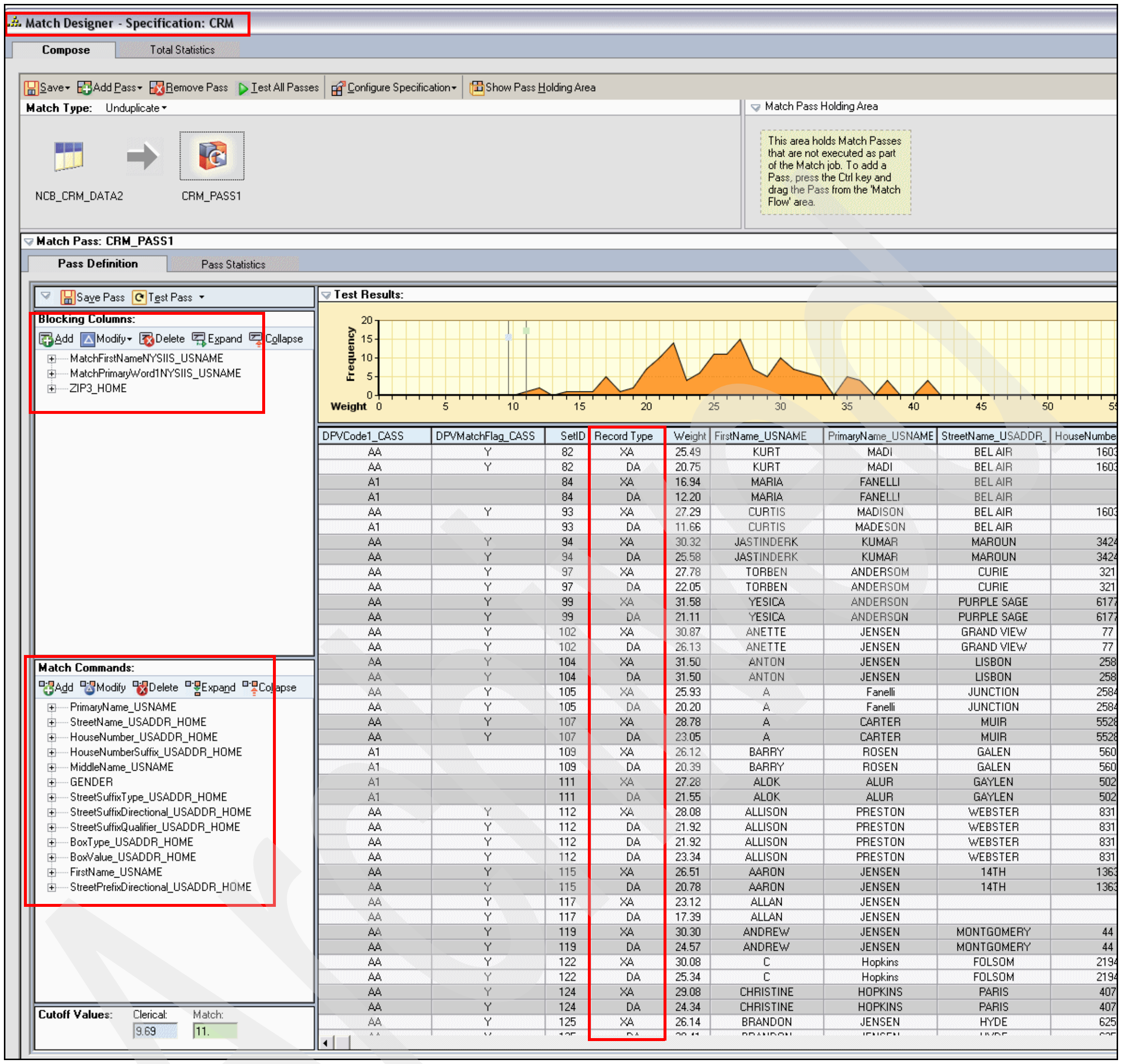
Match Commands:

- PrimaryName_USNAME
- StreetName_USADDR_HOME
- HouseNumber_USADDR_HOME
- HouseNumberSuffix_USADDR_HOME
- MiddleName_USNAME
- GENDER
- StreetSuffixType_USADDR_HOME
- StreetSuffixDirectional_USADDR_HOME
- StreetSuffixQualifier_USADDR_HOME
- BoxType_USADDR_HOME
- BoxValue_USADDR_HOME
- FirstName_USNAME
- StreetPrefixDirectional_USADDR_HOME

Cutoff Values: Clerical: 9.69 Match: 11.

DPVCode1_CASS	DPVMatchFlag_CASS	SetID	Record Type	Weight	FirstName_USNAME	PrimaryName_USNAME	StreetName_USADDR_	HouseNumbe
AA	Y	42	DA	31.65	ANTHONY	PETERSON	PEACH	1015
AA	Y	45	XA	28.75	DANA	PETERSON	MENLO	598
AA	Y	45	DA	28.75	DANA	PETERSON	MENLO	598
AA	Y	46	XA	29.65	DUSTIN	PETERSON	WILSON	725
AA	Y	46	DA	29.65	DUSTIN	PETERSON	WILSON	725
AA	Y	47	XA	26.76	BRANDI	PETERSON	HILLTOP	679
AA	Y	47	DA	26.76	BRANDI	PETERSON	HILLTOP	679
AA	Y	48	XA	27.84	CHRISTINE	PETERSON	SUNRIVER	106
AA	Y	48	DA	27.84	CHRISTINE	PETERSON	SUNRIVER	106
AA	D	49	XA	23.66	DANA	SUTHERLAND	LA RIVIERA	861
AA	D	49	DA	18.92	DANA	SUTHERLAND	LA RIVIERA	861
AA	D	49	DA	23.66	DANA	SUTHERLAND	LA RIVIERA	861
AA	D	49	DA	23.66	DANA	SUTHERLAND	LA RIVIERA	861
AA	Y	50	XA	28.00	KATHRYN	SUTHERLAND	HARTNELL	202
AA	Y	50	DA	28.00	KATHRYN	SUTHERLAND	HARTNELL	202
AA	Y	52	XA	35.48	DEBORAH	SUTHERLAND	PROSPECT	524
AA	Y	52	DA	35.48	DEBORAH	SUTHERLAND	PROSPECT	524
AA	Y	54	XA	25.97	ROBERT	SUTHERLAND	PAGET	917
AA	Y	54	DA	25.97	ROBERT	SUTHERLAND	PAGET	917
AA	D	58	XA	21.92	BURR	PRESTON	BRODERICK	372
AA	Y	58	DA	16.99	BURR	PRESTON	BRODERICK	372
AA	D	58	DA	21.92	BURR	PRESTON	BRODERICK	372
AA	Y	58	DA	21.92	BURR	PRESTON	BRODERICK	372
AA	D	60	XA	20.92	CATHY	PRESTON	MARKET	425
AA	Y	60	DA	16.18	CATHY	PRESTON	MARKET	425
AA	Y	60	DA	20.92	CATHY	PRESTON	MARKET	425
AA	Y	60	DA	20.92	CATHY	PRESTON	MARKET	425
AA	Y	62	XA	21.71	A	HOPKINS	15TH	120
AA	Y	62	DA	21.71	A	HOPKINS	15TH	120
AA	Y	64	XA	21.92	BEATA	HOPKINS	24TH	225
AA	Y	64	DA	17.18	BEATA	HOPKINS	24TH	225
AA	Y	64	DA	21.92	BEATA	HOPKINS	24TH	225
AA	Y	64	DA	21.92	BEATA	HOPKINS	24TH	225
AA	Y	73	XA	32.22	BRUCE	ANDERSON	PURPLE SAGE	617
AA	Y	73	DA	26.49	BRUCE	ANDERSON	PURPLE SAGE	617
AA	D	74	XA	23.68	CHRISTINA	ANDERSON	VERDE DR	618
AA	D	74	DA	18.94	CHRISTINA	ANDERSON	VERDE DR	618
AA	Y	76	XA	24.76	ALEXANDRA	ANDERSON	CURIE	321
AA	Y	76	DA	20.02	ALEXANDRA	ANDERSON	CURIE	321

Figure 2-342 Create j14a_MATCHSPEC 5/7



Match Type: Undunplicate

This area holds Match Passes that are not executed as part of the Match job. To add a Pass, press the Ctrl key and drag the Pass from the 'Match Flow' area.

Match Pass: CRM_PASS1

Blocking Columns:

- MatchFirstNameNYSIIS_USNAME
- MatchPrimaryWord1NYSIIS_USNAME
- ZIP3_HOME

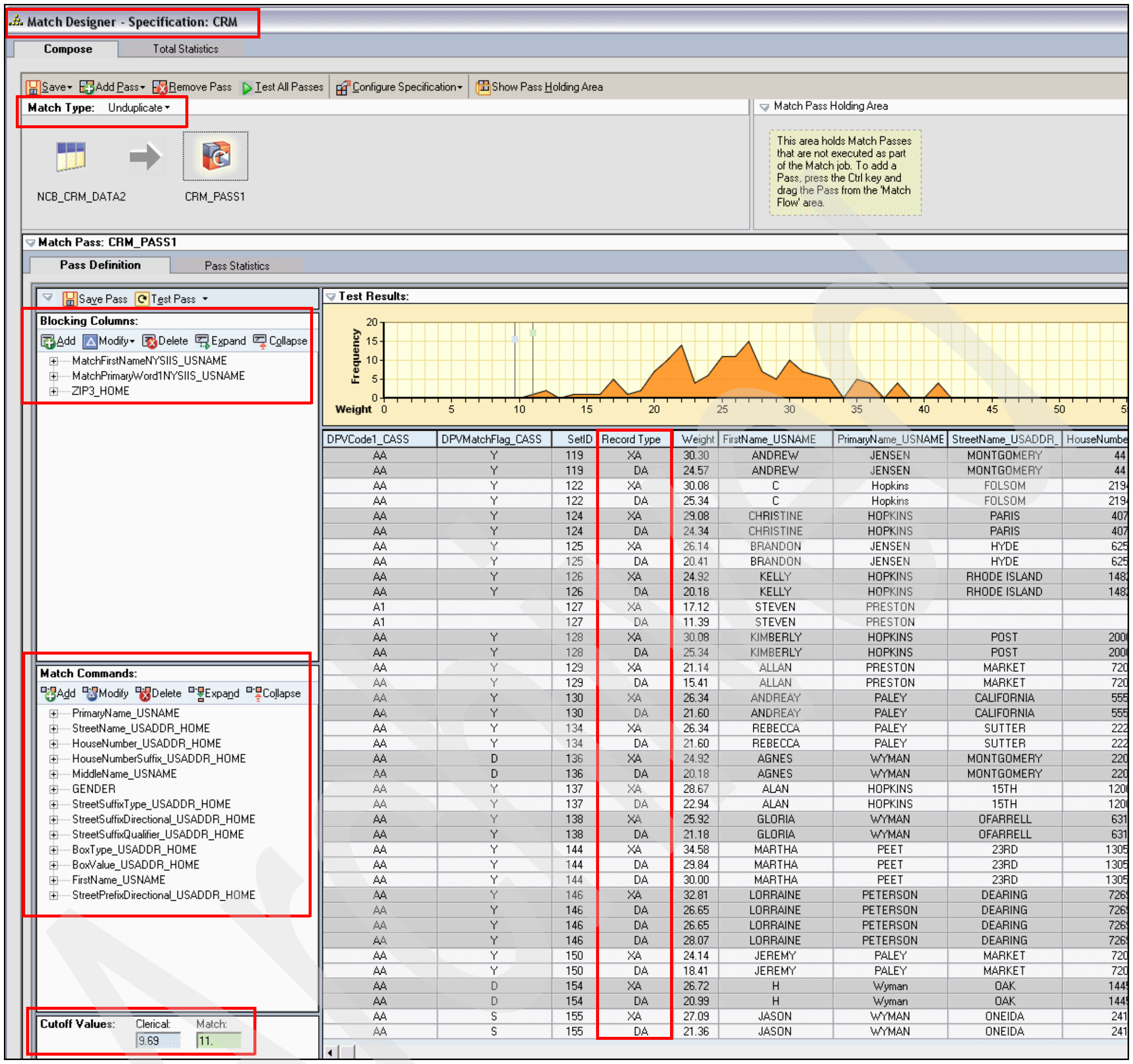
Match Commands:

- PrimaryName_USNAME
- StreetName_USADDR_HOME
- HouseNumber_USADDR_HOME
- HouseNumberSuffix_USADDR_HOME
- MiddleName_USNAME
- GENDER
- StreetSuffixType_USADDR_HOME
- StreetSuffixDirectional_USADDR_HOME
- StreetSuffixQualifier_USADDR_HOME
- BoxType_USADDR_HOME
- BoxValue_USADDR_HOME
- FirstName_USNAME
- StreetPrefixDirectional_USADDR_HOME

Cutoff Values: Clerical: 9.69 Match: 11.

DPVCode1_CASS	DPVMatchFlag_CASS	SetID	Record Type	Weight	FirstName_USNAME	PrimaryName_USNAME	StreetName_USADDR_	HouseNumbe
AA	Y	82	XA	25.49	KURT	MADI	BEL AIR	1603
AA	Y	82	DA	20.75	KURT	MADI	BEL AIR	1603
A1		84	XA	16.94	MARIA	FANELLI	BEL AIR	
A1		84	DA	12.20	MARIA	FANELLI	BEL AIR	
AA	Y	93	XA	27.29	CURTIS	MADISON	BEL AIR	1603
A1		93	DA	11.66	CURTIS	MADESON	BEL AIR	
AA	Y	94	XA	30.32	JASTINDERK	KUMAR	MAROUN	3424
AA	Y	94	DA	25.58	JASTINDERK	KUMAR	MAROUN	3424
AA	Y	97	XA	27.78	TORBEN	ANDERSOM	CURIE	321
AA	Y	97	DA	22.05	TORBEN	ANDERSOM	CURIE	321
AA	Y	99	XA	31.58	YESICA	ANDERSON	PURPLE SAGE	6177
AA	Y	99	DA	21.11	YESICA	ANDERSON	PURPLE SAGE	6177
AA	Y	102	XA	30.87	ANETTE	JENSEN	GRAND VIEW	77
AA	Y	102	DA	26.13	ANETTE	JENSEN	GRAND VIEW	77
AA	Y	104	XA	31.50	ANTON	JENSEN	LISBON	258
AA	Y	104	DA	31.50	ANTON	JENSEN	LISBON	258
AA	Y	105	XA	25.93	A	Fanelli	JUNCTION	2584
AA	Y	105	DA	20.20	A	Fanelli	JUNCTION	2584
AA	Y	107	XA	28.78	A	CARTER	MUIR	5528
AA	Y	107	DA	23.05	A	CARTER	MUIR	5528
A1		109	XA	26.12	BARRY	ROSEN	GALEN	560
A1		109	DA	20.39	BARRY	ROSEN	GALEN	560
A1		111	XA	27.28	ALOK	ALUR	GAYLEN	502
A1		111	DA	21.55	ALOK	ALUR	GAYLEN	502
AA	Y	112	XA	28.08	ALLISON	PRESTON	WEBSTER	831
AA	Y	112	DA	21.92	ALLISON	PRESTON	WEBSTER	831
AA	Y	112	DA	21.92	ALLISON	PRESTON	WEBSTER	831
AA	Y	112	DA	23.34	ALLISON	PRESTON	WEBSTER	831
AA	Y	115	XA	26.51	AARON	JENSEN	14TH	1363
AA	Y	115	DA	20.78	AARON	JENSEN	14TH	1363
AA	Y	117	XA	23.12	ALLAN	JENSEN		
AA	Y	117	DA	17.39	ALLAN	JENSEN		
AA	Y	119	XA	30.30	ANDREW	JENSEN	MONTGOMERY	44
AA	Y	119	DA	24.57	ANDREW	JENSEN	MONTGOMERY	44
AA	Y	122	XA	30.08	C	Hopkins	FOLSOM	2194
AA	Y	122	DA	25.34	C	Hopkins	FOLSOM	2194
AA	Y	124	XA	29.08	CHRISTINE	HOPKINS	PARIS	407
AA	Y	124	DA	24.34	CHRISTINE	HOPKINS	PARIS	407
AA	Y	125	XA	26.14	BRANDON	JENSEN	HYDE	625

Figure 2-343 Create j14a_MATCHSPEC 6/7


Match Designer - Specification: CRM

Compose | Total Statistics

Match Type: Unduplicate

NCB_CRM_DATA2 → CRM_PASS1

Match Pass Holding Area

This area holds Match Passes that are not executed as part of the Match job. To add a Pass, press the Ctrl key and drag the Pass from the 'Match Flow' area.

Match Pass: CRM_PASS1

Pass Definition | Pass Statistics

Blocking Columns:
- MatchFirstNameNYSIIS_USNAME
- MatchPrimaryWord1NYSIIS_USNAME
- ZIP3_HOME

Match Commands:
- PrimaryName_USNAME
- StreetName_USADDR_HOME
- HouseNumber_USADDR_HOME
- HouseNumberSuffix_USADDR_HOME
- MiddleName_USNAME
- GENDER
- StreetSuffixType_USADDR_HOME
- StreetSuffixDirectional_USADDR_HOME
- StreetSuffixQualifier_USADDR_HOME
- BoxType_USADDR_HOME
- BoxValue_USADDR_HOME
- FirstName_USNAME
- StreetPrefixDirectional_USADDR_HOME

Cutoff Values: Clerical: 9.69 Match: 11.

Test Results:

DPVCode1_CASS	DPVMatchFlag_CASS	SetID	Record Type	Weight	FirstName_USNAME	PrimaryName_USNAME	StreetName_USADDR_	HouseNumbe
AA	Y	119	XA	30.30	ANDREW	JENSEN	MONTGOMERY	44
AA	Y	119	DA	24.57	ANDREW	JENSEN	MONTGOMERY	44
AA	Y	122	XA	30.08	C	Hopkins	FOLSOM	219
AA	Y	122	DA	25.34	C	Hopkins	FOLSOM	219
AA	Y	124	XA	29.08	CHRISTINE	HOPKINS	PARIS	407
AA	Y	124	DA	24.34	CHRISTINE	HOPKINS	PARIS	407
AA	Y	125	XA	26.14	BRANDON	JENSEN	HYDE	625
AA	Y	125	DA	20.41	BRANDON	JENSEN	HYDE	625
AA	Y	126	XA	24.92	KELLY	HOPKINS	RHODE ISLAND	148
AA	Y	126	DA	20.18	KELLY	HOPKINS	RHODE ISLAND	148
A1		127	XA	17.12	STEVEN	PRESTON		
A1		127	DA	11.39	STEVEN	PRESTON		
AA	Y	128	XA	30.08	KIMBERLY	HOPKINS	POST	200
AA	Y	128	DA	25.34	KIMBERLY	HOPKINS	POST	200
AA	Y	129	XA	21.14	ALLAN	PRESTON	MARKET	720
AA	Y	129	DA	15.41	ALLAN	PRESTON	MARKET	720
AA	Y	130	XA	26.34	ANDREAY	PALEY	CALIFORNIA	555
AA	Y	130	DA	21.60	ANDREAY	PALEY	CALIFORNIA	555
AA	Y	134	XA	26.34	REBECCA	PALEY	SUTTER	222
AA	Y	134	DA	21.60	REBECCA	PALEY	SUTTER	222
AA	D	136	XA	24.92	AGNES	WYMAN	MONTGOMERY	220
AA	D	136	DA	20.18	AGNES	WYMAN	MONTGOMERY	220
AA	Y	137	XA	28.67	ALAN	HOPKINS	15TH	120
AA	Y	137	DA	22.94	ALAN	HOPKINS	15TH	120
AA	Y	138	XA	25.92	GLORIA	WYMAN	OFARRELL	631
AA	Y	138	DA	21.18	GLORIA	WYMAN	OFARRELL	631
AA	Y	144	XA	34.58	MARTHA	PEET	23RD	1305
AA	Y	144	DA	29.84	MARTHA	PEET	23RD	1305
AA	Y	144	DA	30.00	MARTHA	PEET	23RD	1305
AA	Y	146	XA	32.81	LORRAINE	PETERSON	DEARING	726
AA	Y	146	DA	26.65	LORRAINE	PETERSON	DEARING	726
AA	Y	146	DA	26.65	LORRAINE	PETERSON	DEARING	726
AA	Y	146	DA	28.07	LORRAINE	PETERSON	DEARING	726
AA	Y	150	XA	24.14	JEREMY	PALEY	MARKET	720
AA	Y	150	DA	18.41	JEREMY	PALEY	MARKET	720
AA	D	154	XA	26.72	H	Wyman	OAK	144
AA	D	154	DA	20.99	H	Wyman	OAK	144
AA	S	155	XA	27.09	JASON	WYMAN	ONEIDA	241
AA	S	155	DA	21.36	JASON	WYMAN	ONEIDA	241

Figure 2-344 Create j14a_MATCHSPEC 7/7

j15_UNDUP_CRM

Next, we determine whether there were duplicates in the merged records of the North American Bank and Northern California Bank's core and non-core services customers, using the Unduplicate stage with the match specification and match frequency information that was created in "j14a_MATCHSPEC" on page 801 and "j14_CRM_FREQUENCY" on page 798, respectively. As before, the output is matched records (merge of master and duplicates using a Funnel stage), records for clerical review, and residuals (records that do not match).

Figure 2-345 on page 811 shows the various stages that are used in this job. It includes the data set that was created in “j13_FUNNEL_NCB_NAB_CRM_DATA” on page 797, the match frequency Data Set created in “j14_CRM_FREQUENCY” on page 798, an Unduplicate stage, and a Funnel stage to merge master and duplicate records. Three data sets are created:

- One data set contains the merged master and duplicates by the Funnel stage.
- The other data sets contain the clerical and residual records as the output of the Unduplicate stage.

We modified the names of the stages as shown.

Because the creation of this Unduplicate stage Dependent Match Type is very similar to that described in “J11_UNDUP_DEP_MATCH_CUSTOMER” on page 282, we do not repeat the steps involved here.

After saving, compiling, and running this job (Figure 2-345 on page 811), you view the content of the three data set objects as shown in Figure 2-346 on page 812 through Figure 2-349 on page 814.

Out of a total of 160 records in the input to this process:

- 149 records were masters and duplicates. There are not in sorted order of qsMatchSetId.
- Zero (0) records were for clerical review.
- Remaining eleven 11 records were residuals.

Because there were no records for clerical review, a clean set of master and duplicate records is created, and the “best” information is survived into the master record using the SURVIVE stage as described in “j16_SURVIVE_CRM” on page 814.

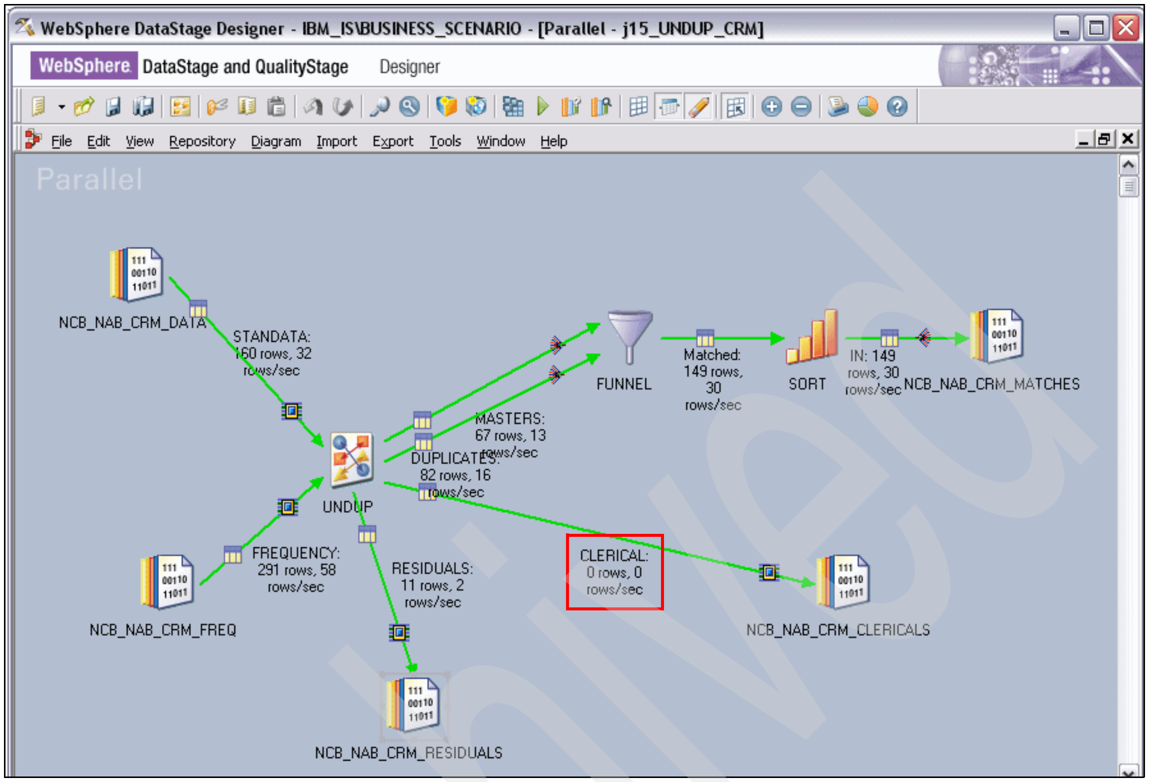
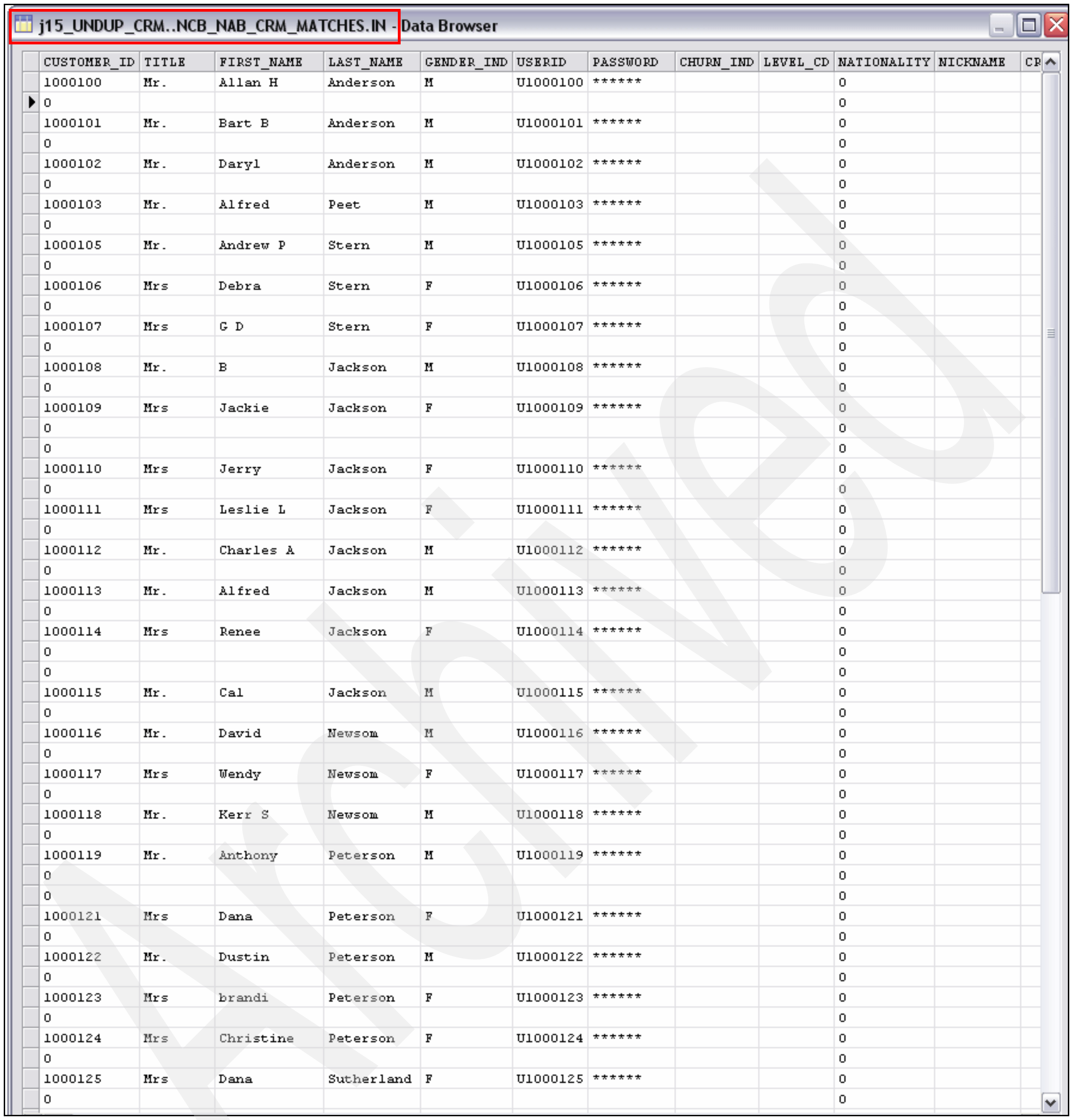
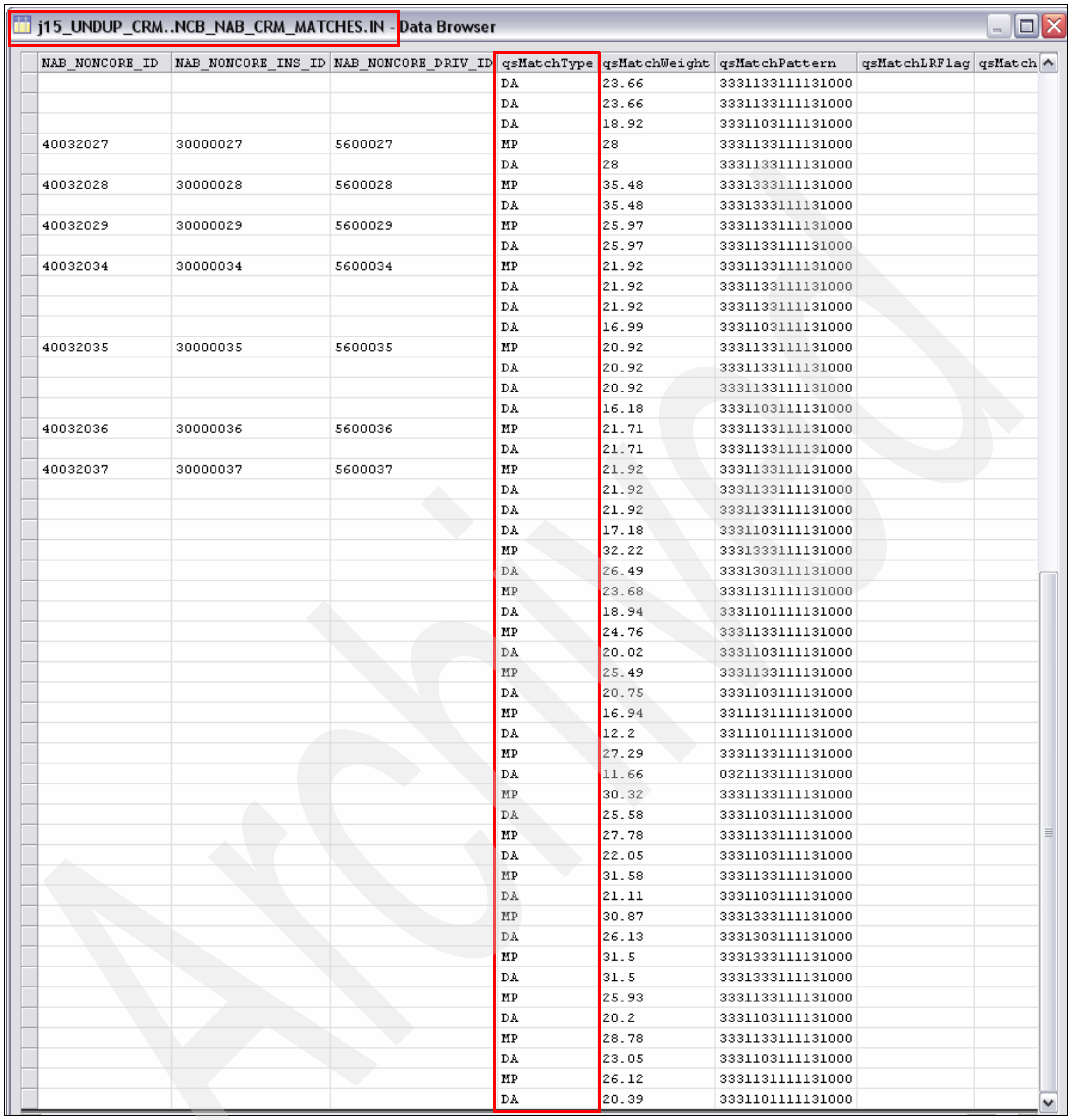

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j15_UNDUP_CRM]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NCB_NAB_CRM_DATA
STANDATA: 160 rows, 32 rows/sec
FUNNEL
Matched: 149 rows, 30 rows/sec
SORT
IN: 149 rows, 30 rows/sec
NCB_NAB_CRM_MATCHES
MASTERS: 67 rows, 13 rows/sec
DUPLICATES: 82 rows, 16 rows/sec
UNDUP
FREQUENCY: 291 rows, 58 rows/sec
RESIDUALS: 11 rows, 2 rows/sec
CLERICAL: 0 rows, 0 rows/sec
NCB_NAB_CRM_FREQ
NCB_NAB_CRM_CLERICALS
NCB_NAB_CRM_RESIDUALS

Figure 2-345 Create j15_UNDUP_CRM 1/5


j15_UNDUP_CRM..NCB_NAB_CRM_MATCHES.IN - Data Browser

CUSTOMER_ID	TITLE	FIRST_NAME	LAST_NAME	GENDER_IND	USERID	PASSWORD	CHURN_IND	LEVEL_CD	NATIONALITY	NICKNAME	CR
1000100	Mr.	Allan H	Anderson	M	U1000100	******			0		
0									0		
1000101	Mr.	Bart B	Anderson	M	U1000101	******			0		
0									0		
1000102	Mr.	Daryl	Anderson	M	U1000102	******			0		
0									0		
1000103	Mr.	Alfred	Peet	M	U1000103	******			0		
0									0		
1000105	Mr.	Andrew P	Stern	M	U1000105	******			0		
0									0		
1000106	Mrs	Debra	Stern	F	U1000106	******			0		
0									0		
1000107	Mrs	G D	Stern	F	U1000107	******			0		
0									0		
1000108	Mr.	B	Jackson	M	U1000108	******			0		
0									0		
1000109	Mrs	Jackie	Jackson	F	U1000109	******			0		
0									0		
0									0		
1000110	Mrs	Jerry	Jackson	F	U1000110	******			0		
0									0		
1000111	Mrs	Leslie L	Jackson	F	U1000111	******			0		
0									0		
1000112	Mr.	Charles A	Jackson	M	U1000112	******			0		
0									0		
1000113	Mr.	Alfred	Jackson	M	U1000113	******			0		
0									0		
1000114	Mrs	Renee	Jackson	F	U1000114	******			0		
0									0		
0									0		
1000115	Mr.	Cal	Jackson	M	U1000115	******			0		
0									0		
1000116	Mr.	David	Newsom	M	U1000116	******			0		
0									0		
1000117	Mrs	Wendy	Newsom	F	U1000117	******			0		
0									0		
1000118	Mr.	Kerr S	Newsom	M	U1000118	******			0		
0									0		
1000119	Mr.	Anthony	Peterson	M	U1000119	******			0		
0									0		
0									0		
1000121	Mrs	Dana	Peterson	F	U1000121	******			0		
0									0		
1000122	Mr.	Dustin	Peterson	M	U1000122	******			0		
0									0		
1000123	Mrs	brandi	Peterson	F	U1000123	******			0		
0									0		
1000124	Mrs	Christine	Peterson	F	U1000124	******			0		
0									0		
1000125	Mrs	Dana	Sutherland	F	U1000125	******			0		
0									0		

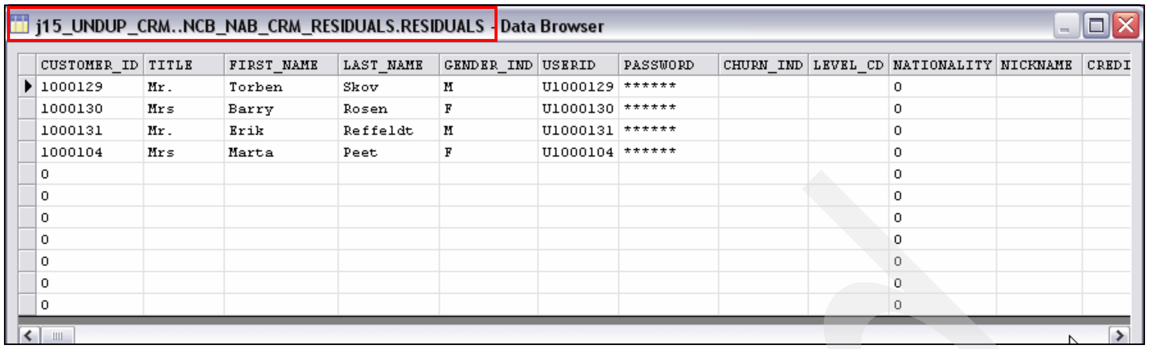
Figure 2-346 Create j15_UNDUP_CRM 2/5



j15_UNDUP_CRM..NCB_NAB_CRM_MATCHES.IN - Data Browser

NAB_NONCORE_ID	NAB_NONCORE_INS_ID	NAB_NONCORE_DRIV_ID	qsMatchType	qsMatchWeight	qsMatchPattern	qsMatchLRFlag	qsMatch
			DA	23.66	3331133111131000		
			DA	23.66	3331133111131000		
			DA	18.92	3331103111131000		
40032027	30000027	5600027	MP	28	3331133111131000		
			DA	28	3331133111131000		
40032028	30000028	5600028	MP	35.48	3331333111131000		
			DA	35.48	3331333111131000		
40032029	30000029	5600029	MP	25.97	3331133111131000		
			DA	25.97	3331133111131000		
40032034	30000034	5600034	MP	21.92	3331133111131000		
			DA	21.92	3331133111131000		
			DA	21.92	3331133111131000		
			DA	16.99	3331103111131000		
40032035	30000035	5600035	MP	20.92	3331133111131000		
			DA	20.92	3331133111131000		
			DA	20.92	3331133111131000		
			DA	16.18	3331103111131000		
40032036	30000036	5600036	MP	21.71	3331133111131000		
			DA	21.71	3331133111131000		
40032037	30000037	5600037	MP	21.92	3331133111131000		
			DA	21.92	3331133111131000		
			DA	21.92	3331133111131000		
			DA	17.18	3331103111131000		
			MP	32.22	3331333111131000		
			DA	26.49	3331303111131000		
			MP	23.68	3331131111131000		
			DA	18.94	3331101111131000		
			MP	24.76	3331133111131000		
			DA	20.02	3331103111131000		
			MP	25.49	3331133111131000		
			DA	20.75	3331103111131000		
			MP	16.94	3311131111131000		
			DA	12.2	3311101111131000		
			MP	27.29	3331133111131000		
			DA	11.66	0321133111131000		
			MP	30.32	3331133111131000		
			DA	25.58	3331103111131000		
			MP	27.78	3331133111131000		
			DA	22.05	3331103111131000		
			MP	31.58	3331133111131000		
			DA	21.11	3331103111131000		
			MP	30.87	3331333111131000		
			DA	26.13	3331303111131000		
			MP	31.5	3331333111131000		
			DA	31.5	3331333111131000		
			MP	25.93	3331133111131000		
			DA	20.2	3331103111131000		
			MP	28.78	3331133111131000		
			DA	23.05	3331103111131000		
			MP	26.12	3331131111131000		
			DA	20.39	3331101111131000		

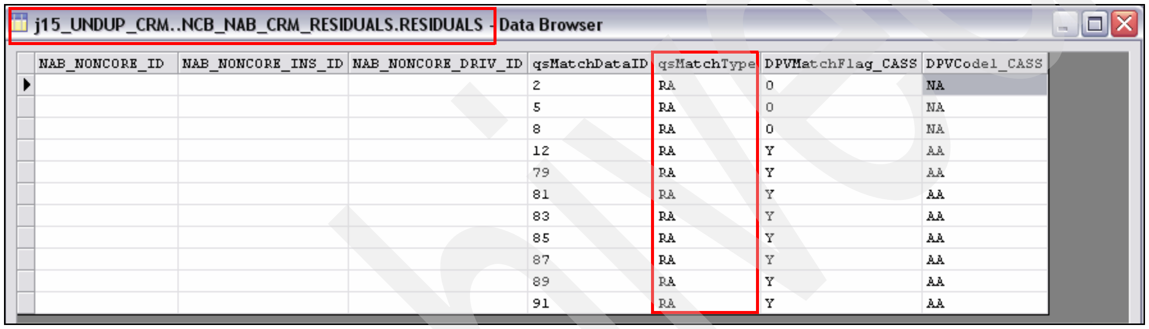
Figure 2-347 Create j15_UNDUP_CRM 3/5



j15_UNDUP_CRM..NCB_NAB_CRM_RESIDUALS.RESIDUALS - Data Browser

CUSTOMER_ID	TITLE	FIRST_NAME	LAST_NAME	GENDER_IND	USERID	PASSWORD	CHURN_IND	LEVEL_CD	NATIONALITY	NICKNAME	CREDI
1000129	Mr.	Torben	Skov	M	U1000129	******			0		
1000130	Mrs	Barry	Rosen	F	U1000130	******			0		
1000131	Mr.	Erik	Reffeldt	M	U1000131	******			0		
1000104	Mrs	Marta	Peet	F	U1000104	******			0		
0									0		
0									0		
0									0		
0									0		
0									0		
0									0		
0									0		

Figure 2-348 Create j15_UNDUP_CRM 4/5



j15_UNDUP_CRM..NCB_NAB_CRM_RESIDUALS.RESIDUALS - Data Browser

NAB_NONCORE_ID	NAB_NONCORE_INS_ID	NAB_NONCORE_DRIV_ID	qsMatchDataID	qsMatchType	DPVMatchFlag_CASS	DPVCode1_CASS
			2	RA	0	NA
			5	RA	0	NA
			8	RA	0	NA
			12	RA	Y	AA
			79	RA	Y	AA
			81	RA	Y	AA
			83	RA	Y	AA
			85	RA	Y	AA
			87	RA	Y	AA
			89	RA	Y	AA
			91	RA	Y	AA

Figure 2-349 Create j15_UNDUP_CRM 5/5

j16_SURVIVE_CRM

The next step is to survive the "best" information from the set of matched records that were created in the "j15_UNDUP_CRM" on page 809 step using the SURVIVE stage. We introduce a Transformer stage to add a column DPVCode_num that contains the recoded value of the contents of the DPVCode1_CASS column—if the column contained `AA`, to put a `1` in the DPVCode_num column, otherwise put a `0`. This was done to circumvent a bug in the Survive Rule Builder that had problems comparing string content, but not numeric content.

Figure 2-350 on page 815 shows the various stages that are used in this job. It includes the input data set that was created in "j15_UNDUP_CRM" on page 809 containing matched and duplicate records, a Transformer stage to recode the DPVCode1_CASS column, a Survive stage, and an output data set containing the survived records. We modified the names of the stages as shown.

Because the creation of this survive job is very similar to that described in "J13_SURVIVE_CUSTOMER" on page 295, we do not repeat the steps involved here. However, some of the configurations of interest are as follows:

- Figure 2-351 on page 816 shows the Transformer stage that recodes the contents of the DPVCode1_CASS column into a new column DPVCode_num for the reasons mentioned earlier.
- Figure 2-352 on page 817 through Figure 2-370 on page 826 show the SURVIVE - Survive Stage windows with the survive rules involving all the fields that exist in multiple records that need to be mapped to the CRM system.
  - The DPVCode rule specifies that if the same customer has an address that is not DPV verified by CASS and an address that is DPV verified, then the DPV verified address should be accepted.

    We could have fixed the address that is not DPV verified, but chose to use this method to demonstrate the use the information generated by CASS in the survive process.
  - The survive rules of the ID columns such as NCB_NONCORE_ID allows the collection of ids from all the rows in each match set. This enables us to determine the tables in which this customer exists, which provides a holistic view of this customer's relationship with the bank.
- After saving, compiling, and running this job (Figure 2-371 on page 826), view the content of the output as shown in Figure 2-372 on page 827 through Figure 2-374 on page 829.

  This report shows the 67 master records with the survived information from the duplicates.

Proceed now to "j17_FUNNEL_UNDUP_RES_DATA" on page 829.

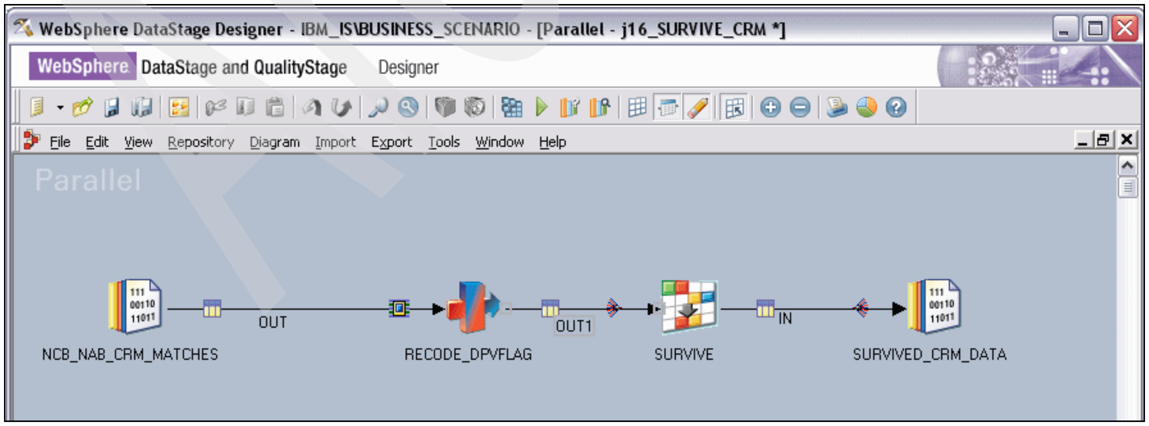
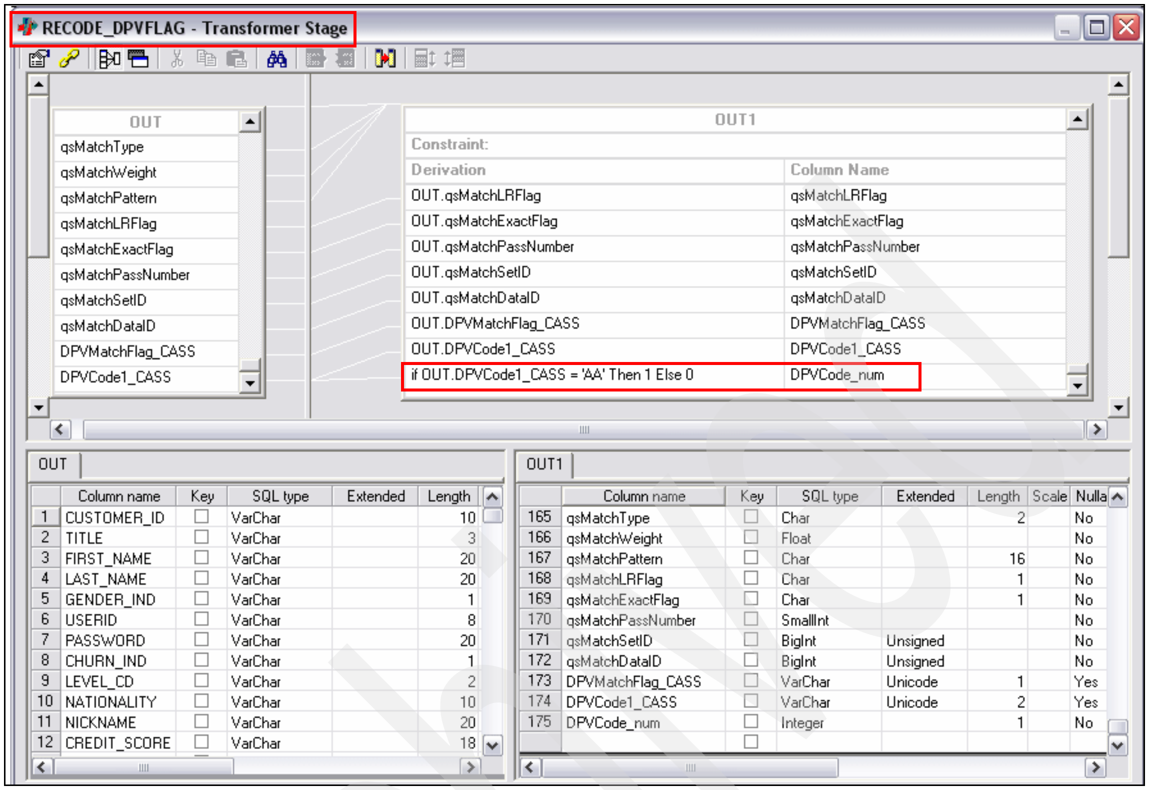

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j16_SURVIVE_CRM *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NCB_NAB_CRM_MATCHES
OUT
RECODE_DPVFLAG
OUT1
SURVIVE
IN
SURVIVED_CRM_DATA

Figure 2-350 Create j16_SURVIVE_CRM 1/25



RECODE_DPVFLAG - Transformer Stage

OUT
qsMatchType
qsMatchWeight
qsMatchPattern
qsMatchLRFlag
qsMatchExactFlag
qsMatchPassNumber
qsMatchSetID
qsMatchDataID
DPVMatchFlag_CASS
DPVCode1_CASS

OUT1

Constraint:

Derivation	Column Name
OUT.qsMatchLRFlag	qsMatchLRFlag
OUT.qsMatchExactFlag	qsMatchExactFlag
OUT.qsMatchPassNumber	qsMatchPassNumber
OUT.qsMatchSetID	qsMatchSetID
OUT.qsMatchDataID	qsMatchDataID
OUT.DPVMatchFlag_CASS	DPVMatchFlag_CASS
OUT.DPVCode1_CASS	DPVCode1_CASS
if OUT.DPVCode1_CASS = 'AA' Then 1 Else 0	DPVCode_num

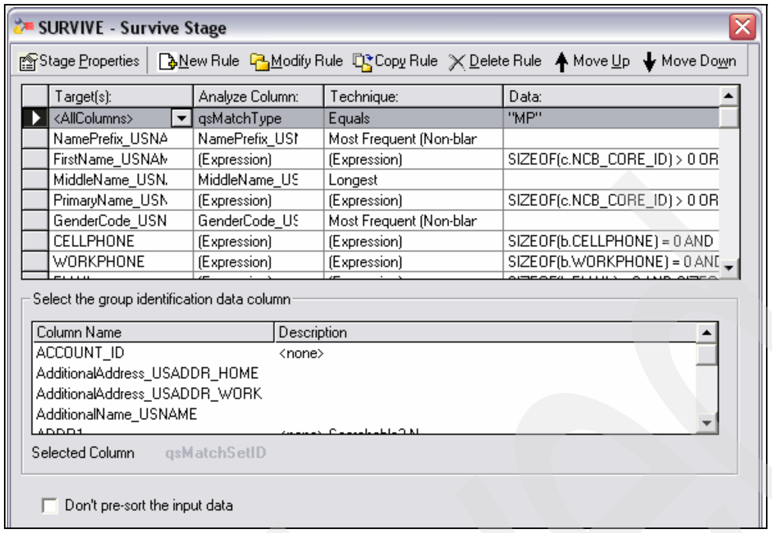
OUT

	Column name	Key	SQL type	Extended	Length
1	CUSTOMER_ID	☐	VarChar		10
2	TITLE	☐	VarChar		3
3	FIRST_NAME	☐	VarChar		20
4	LAST_NAME	☐	VarChar		20
5	GENDER_IND	☐	VarChar		1
6	USERID	☐	VarChar		8
7	PASSWORD	☐	VarChar		20
8	CHURN_IND	☐	VarChar		1
9	LEVEL_CD	☐	VarChar		2
10	NATIONALITY	☐	VarChar		10
11	NICKNAME	☐	VarChar		20
12	CREDIT_SCORE	☐	VarChar		18

OUT1

	Column name	Key	SQL type	Extended	Length	Scale	Nulla
165	qsMatchType	☐	Char		2		No
166	qsMatchWeight	☐	Float				No
167	qsMatchPattern	☐	Char		16		No
168	qsMatchLRFlag	☐	Char		1		No
169	qsMatchExactFlag	☐	Char		1		No
170	qsMatchPassNumber	☐	SmallInt				No
171	qsMatchSetID	☐	BigInt	Unsigned			No
172	qsMatchDataID	☐	BigInt	Unsigned			No
173	DPVMatchFlag_CASS	☐	VarChar	Unicode	1		Yes
174	DPVCode1_CASS	☐	VarChar	Unicode	2		Yes
175	DPVCode_num	☐	Integer		1		No
		☐					

Figure 2-351 Create j16_SURVIVE_CRM 2/25



Target(s):	Analyze Column:	Technique:	Data:
<AllColumns>	qsMatchType	Equals	"MP"
NamePrefix_USNA	NamePrefix_USN	Most Frequent (Non-blar	
FirstName_USNAM	(Expression)	(Expression)	SIZEOF(c.NCB_CORE_ID) > 0 OR
MiddleName_USN.	MiddleName_US	Longest	
PrimaryName_USN	(Expression)	(Expression)	SIZEOF(c.NCB_CORE_ID) > 0 OR
GenderCode_USN	GenderCode_US	Most Frequent (Non-blar	
CELLPHONE	(Expression)	(Expression)	SIZEOF(b.CELLPHONE) = 0 AND
WORKPHONE	(Expression)	(Expression)	SIZEOF(b.WORKPHONE) = 0 AND

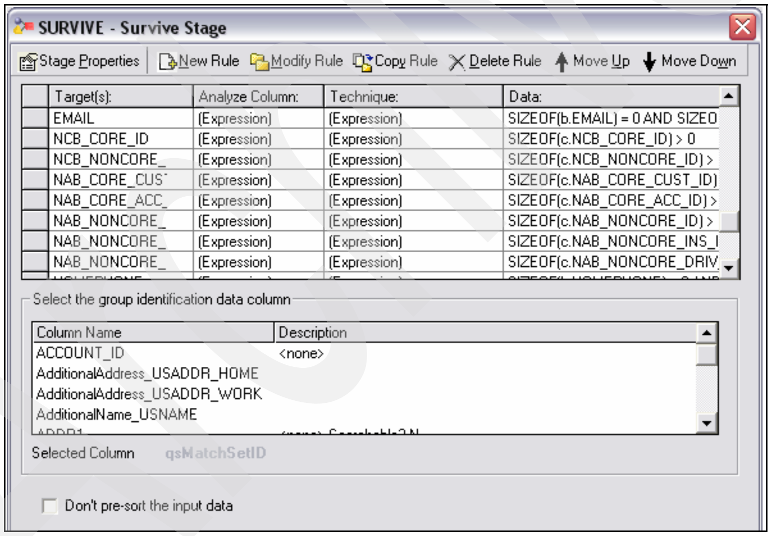
Select the group identification data column

Column Name	Description
ACCOUNT_ID	<none>
AdditionalAddress_USADDR_HOME	
AdditionalAddress_USADDR_WORK	
AdditionalName_USNAME	

Selected Column qsMatchSetID

Don't pre-sort the input data

Figure 2-352 Create j16_SURVIVE_CRM 3/25



Target(s):	Analyze Column:	Technique:	Data:
EMAIL	(Expression)	(Expression)	SIZEOF(b.EMAIL) = 0 AND SIZEO
NCB_CORE_ID	(Expression)	(Expression)	SIZEOF(c.NCB_CORE_ID) > 0
NCB_NONCORE_	(Expression)	(Expression)	SIZEOF(c.NCB_NONCORE_ID) >
NAB_CORE_CUS	(Expression)	(Expression)	SIZEOF(c.NAB_CORE_CUST_ID)
NAB_CORE_ACC_	(Expression)	(Expression)	SIZEOF(c.NAB_CORE_ACC_ID) >
NAB_NONCORE_	(Expression)	(Expression)	SIZEOF(c.NAB_NONCORE_ID) >
NAB_NONCORE_	(Expression)	(Expression)	SIZEOF(c.NAB_NONCORE_INS_I
NAB_NONCORE_	(Expression)	(Expression)	SIZEOF(c.NAB_NONCORE_DRIV

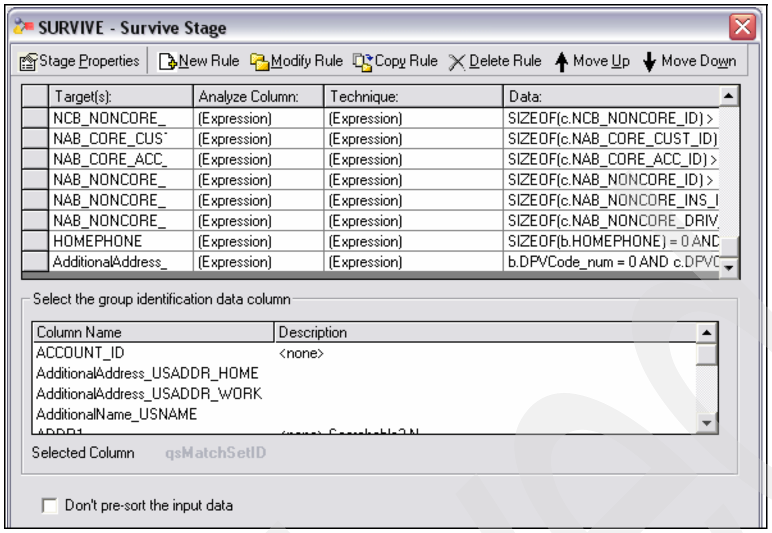
Select the group identification data column

Column Name	Description
ACCOUNT_ID	<none>
AdditionalAddress_USADDR_HOME	
AdditionalAddress_USADDR_WORK	
AdditionalName_USNAME	

Selected Column qsMatchSetID

Don't pre-sort the input data

Figure 2-353 Create j16_SURVIVE_CRM 4/25



SURVIVE - Survive Stage

Target(s):	Analyze Column:	Technique:	Data:
NCB_NONCORE_	(Expression)	(Expression)	SIZEOF(c.NCB_NONCORE_ID) >
NAB_CORE_CUS	(Expression)	(Expression)	SIZEOF(c.NAB_CORE_CUST_ID)
NAB_CORE_ACC_	(Expression)	(Expression)	SIZEOF(c.NAB_CORE_ACC_ID) >
NAB_NONCORE_	(Expression)	(Expression)	SIZEOF(c.NAB_NONCORE_ID) >
NAB_NONCORE_	(Expression)	(Expression)	SIZEOF(c.NAB_NONCORE_INS_I
NAB_NONCORE_	(Expression)	(Expression)	SIZEOF(c.NAB_NONCORE_DRIV
HOMEPHONE	(Expression)	(Expression)	SIZEOF(b.HOMEPHONE) = 0 AND
AdditionalAddress_	(Expression)	(Expression)	b.DPVCode_num = 0 AND c.DPVC

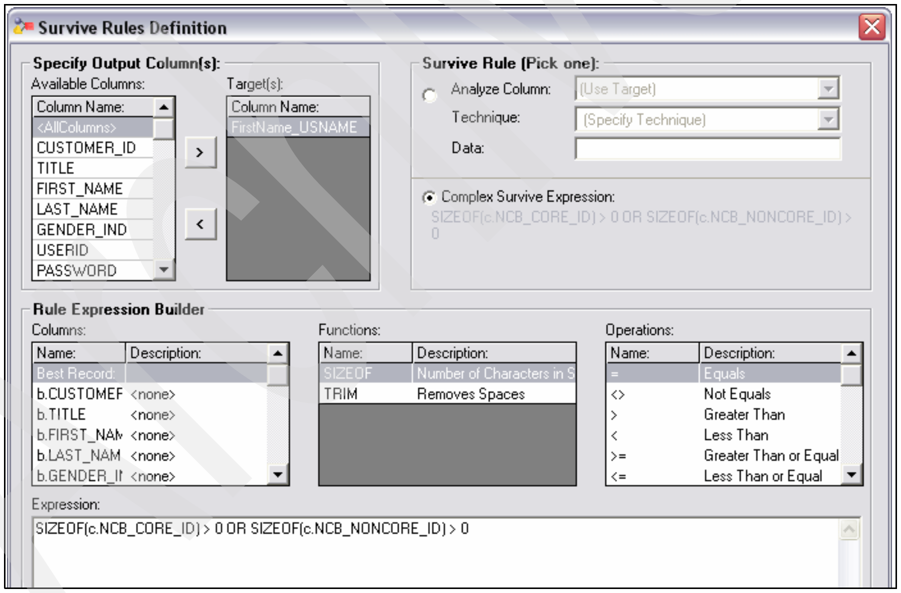
Select the group identification data column

Column Name	Description
ACCOUNT_ID	<none>
AdditionalAddress_USADDR_HOME	
AdditionalAddress_USADDR_WORK	
AdditionalName_USNAME	

Selected Column qsMatchSetID

Don't pre-sort the input data

Figure 2-354 Create j16_SURVIVE_CRM 5/25



Survive Rules Definition

Specify Output Column(s):

Available Columns: <AllColumns>, CUSTOMER_ID, TITLE, FIRST_NAME, LAST_NAME, GENDER_IND, USERID, PASSWORD

Target(s): FirstName_USNAME

Survive Rule (Pick one):

Analyze Column: (Use Target)

Technique: (Specify Technique)

Data:

Complex Survive Expression: SIZEOF(c.NCB_CORE_ID) > 0 OR SIZEOF(c.NCB_NONCORE_ID) > 0

Rule Expression Builder

Columns:

Name:	Description:
Best Record:	
b.CUSTOMEF	<none>
b.TITLE	<none>
b.FIRST_NAN	<none>
b.LAST_NAM	<none>
b.GENDER_II	<none>

Functions:

Name:	Description:
SIZEOF	Number of Characters in S
TRIM	Removes Spaces

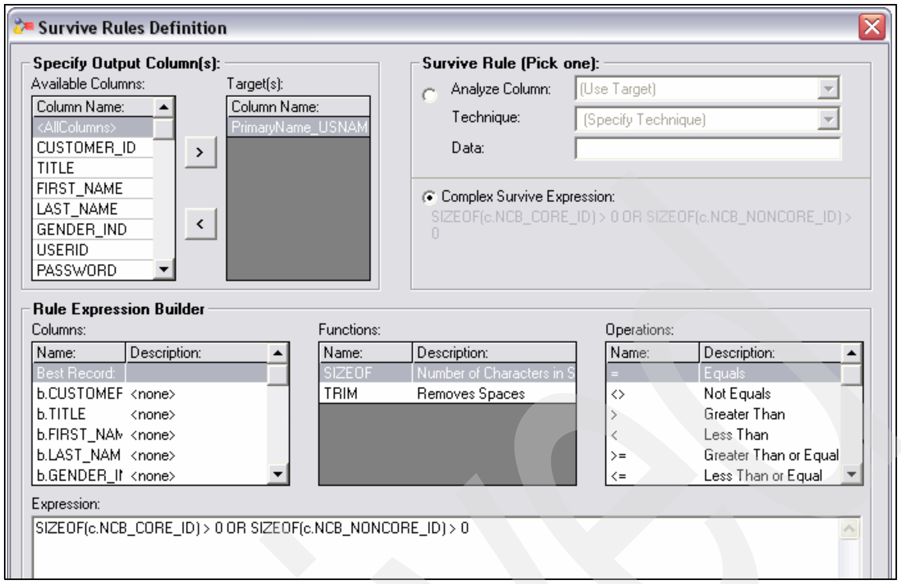
Operations:

Name:	Description:
=	Equals
<>	Not Equals
>	Greater Than
<	Less Than
>=	Greater Than or Equal
<=	Less Than or Equal

Expression:

SIZEOF(c.NCB_CORE_ID) > 0 OR SIZEOF(c.NCB_NONCORE_ID) > 0

Figure 2-355 Create j16_SURVIVE_CRM 6/25



Complex Survive Expression: SIZEOF(c.NCB_CORE_ID) > 0 OR SIZEOF(c.NCB_NONCORE_ID) > 0

Name:	Description:
Best Record:	
b.CUSTOMEF	<none>
b.TITLE	<none>
b.FIRST_NAI	<none>
b.LAST_NAM	<none>
b.GENDER_II	<none>

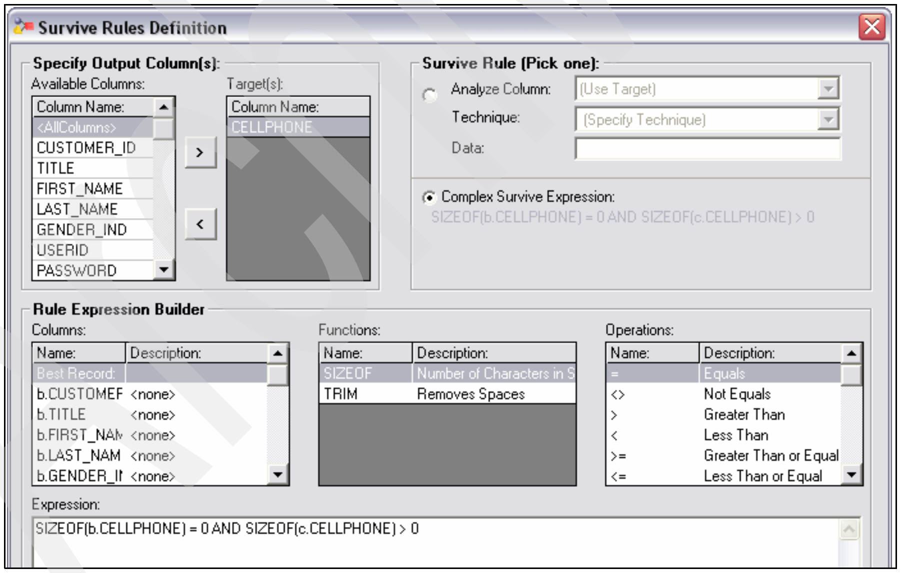
Name:	Description:
SIZEOF	Number of Characters in S
TRIM	Removes Spaces

Name:	Description:
=	Equals
<>	Not Equals
>	Greater Than
<	Less Than
>=	Greater Than or Equal
<=	Less Than or Equal

Expression:

```
SIZEOF(c.NCB_CORE_ID) > 0 OR SIZEOF(c.NCB_NONCORE_ID) > 0
```

Figure 2-356 Create j16_SURVIVE_CRM 7/25



Complex Survive Expression: SIZEOF(b.CELLPHONE) = 0 AND SIZEOF(c.CELLPHONE) > 0

Name:	Description:
Best Record:	
b.CUSTOMEF	<none>
b.TITLE	<none>
b.FIRST_NAI	<none>
b.LAST_NAM	<none>
b.GENDER_II	<none>

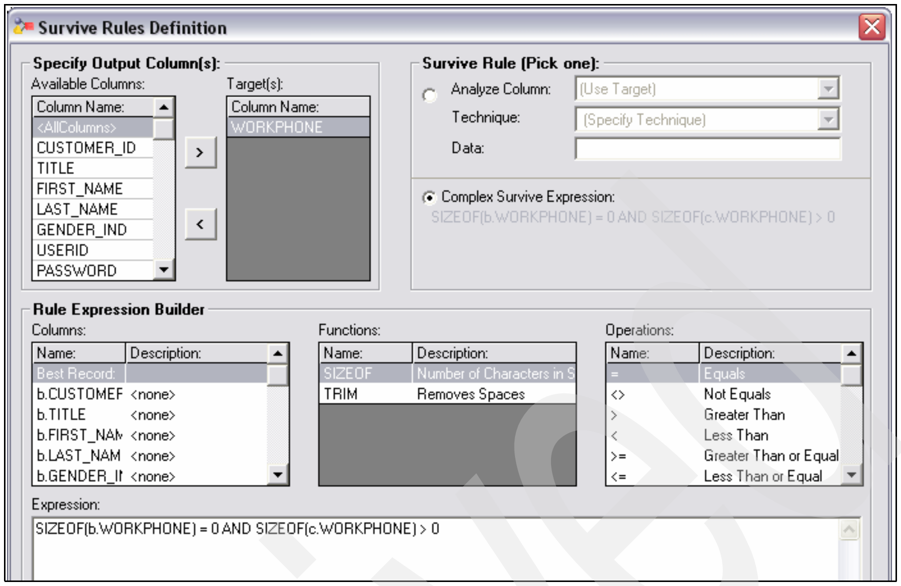
Name:	Description:
SIZEOF	Number of Characters in S
TRIM	Removes Spaces

Name:	Description:
=	Equals
<>	Not Equals
>	Greater Than
<	Less Than
>=	Greater Than or Equal
<=	Less Than or Equal

Expression:

```
SIZEOF(b.CELLPHONE) = 0 AND SIZEOF(c.CELLPHONE) > 0
```

Figure 2-357 Create j16_SURVIVE_CRM 8/25


Survive Rules Definition

Specify Output Column(s):

Available Columns:

Column Name:
<AllColumns>
CUSTOMER_ID
TITLE
FIRST_NAME
LAST_NAME
GENDER_IND
USERID
PASSWORD

Target(s):

Column Name:
WORKPHONE

Survive Rule (Pick one):

( ) Analyze Column: (Use Target)

Technique: (Specify Technique)

Data:

(•) Complex Survive Expression:

SIZEOF(b.WORKPHONE) = 0 AND SIZEOF(c.WORKPHONE) > 0

Rule Expression Builder

Columns:

Name:	Description:
Best Record:	
b.CUSTOMEF	<none>
b.TITLE	<none>
b.FIRST_NAM	<none>
b.LAST_NAM	<none>
b.GENDER_II	<none>

Functions:

Name:	Description:
SIZEOF	Number of Characters in S
TRIM	Removes Spaces

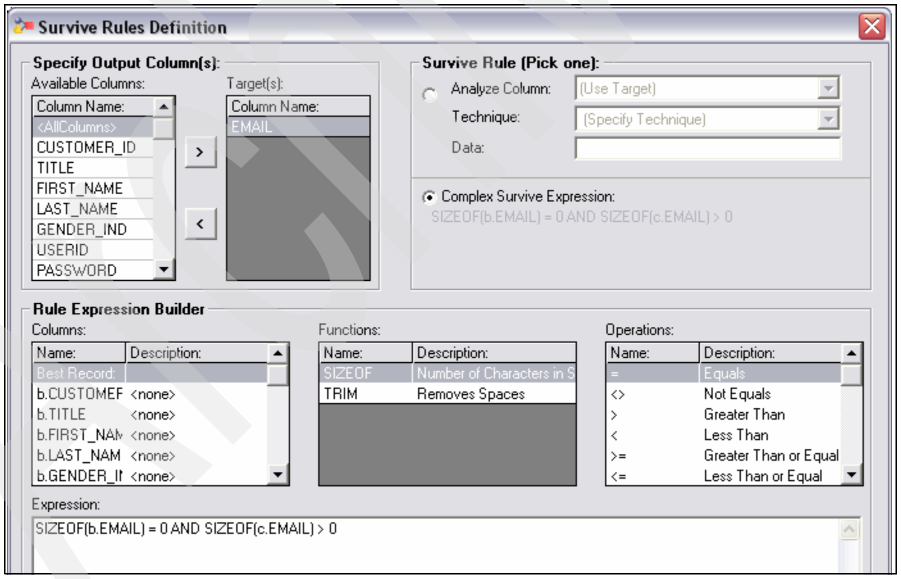
Operations:

Name:	Description:
=	Equals
<>	Not Equals
>	Greater Than
<	Less Than
>=	Greater Than or Equal
<=	Less Than or Equal

Expression:

SIZEOF(b.WORKPHONE) = 0 AND SIZEOF(c.WORKPHONE) > 0

Figure 2-358 Create j16_SURVIVE_CRM 9/25


Survive Rules Definition

Specify Output Column(s):

Available Columns:

Column Name:
<AllColumns>
CUSTOMER_ID
TITLE
FIRST_NAME
LAST_NAME
GENDER_IND
USERID
PASSWORD

Target(s):

Column Name:
EMAIL

Survive Rule (Pick one):

( ) Analyze Column: (Use Target)

Technique: (Specify Technique)

Data:

(•) Complex Survive Expression:

SIZEOF(b.EMAIL) = 0 AND SIZEOF(c.EMAIL) > 0

Rule Expression Builder

Columns:

Name:	Description:
Best Record:	
b.CUSTOMEF	<none>
b.TITLE	<none>
b.FIRST_NAM	<none>
b.LAST_NAM	<none>
b.GENDER_II	<none>

Functions:

Name:	Description:
SIZEOF	Number of Characters in S
TRIM	Removes Spaces

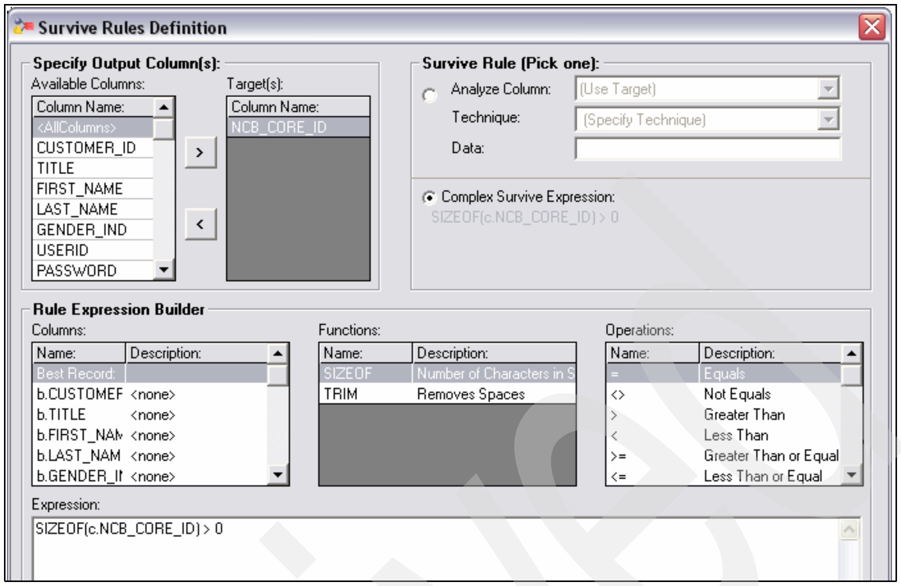
Operations:

Name:	Description:
=	Equals
<>	Not Equals
>	Greater Than
<	Less Than
>=	Greater Than or Equal
<=	Less Than or Equal

Expression:

SIZEOF(b.EMAIL) = 0 AND SIZEOF(c.EMAIL) > 0

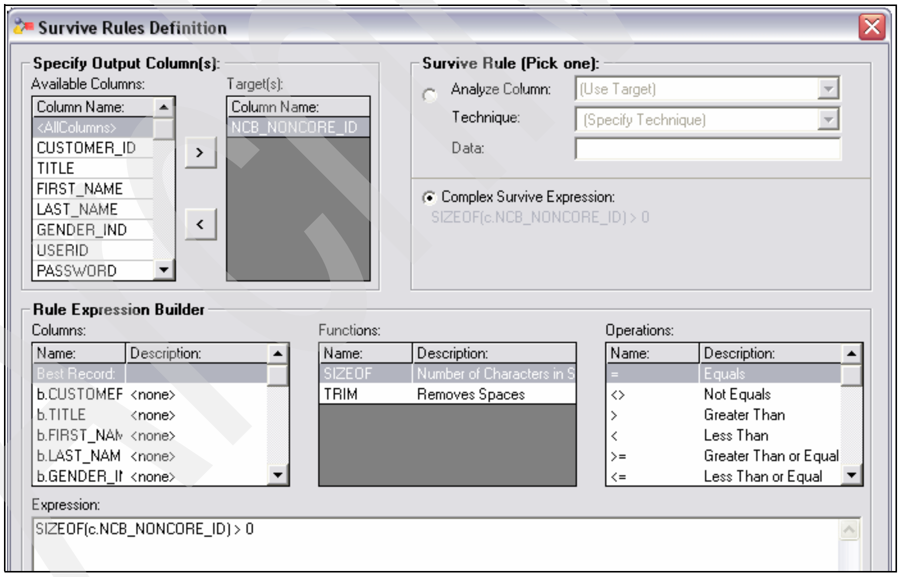
Figure 2-359 Create j16_SURVIVE_CRM 10/25



Survive Rule: [✓ Complex Survive Expression]

```
SIZEOF(c.NCB_CORE_ID) > 0
```

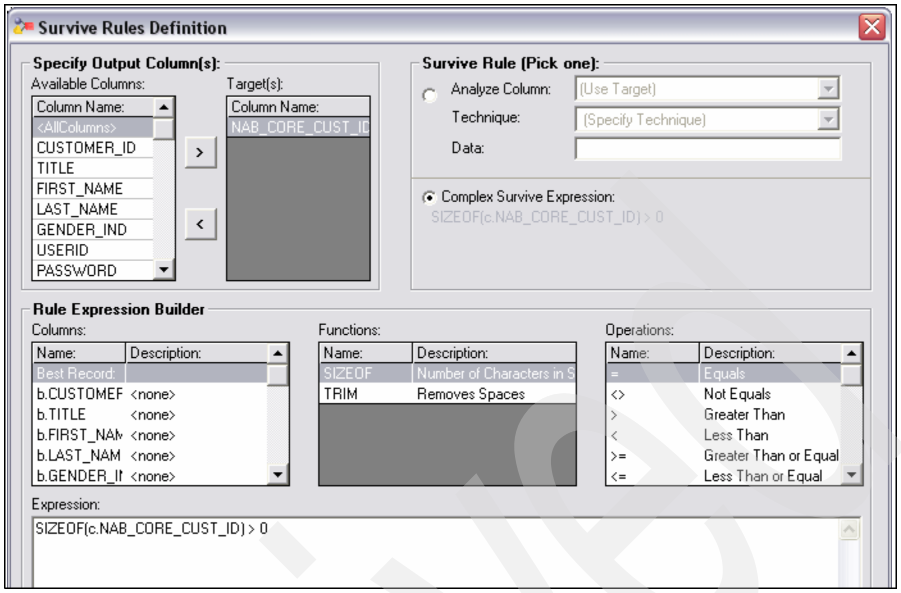
Figure 2-360 Create j16_SURVIVE_CRM 11/25



Survive Rule: [✓ Complex Survive Expression]

```
SIZEOF(c.NCB_NONCORE_ID) > 0
```

Figure 2-361 Create j16_SURVIVE_CRM 12/25



Survive Rules Definition

Specify Output Column(s):

Available Columns:

Column Name:
<AllColumns>
CUSTOMER_ID
TITLE
FIRST_NAME
LAST_NAME
GENDER_IND
USERID
PASSWORD

Target(s):

Column Name:
NAB_CORE_CUST_ID

Survive Rule (Pick one):

Analyze Column: (Use Target)

Technique: (Specify Technique)

Data:

Complex Survive Expression: SIZEOF(c.NAB_CORE_CUST_ID) > 0

Rule Expression Builder

Columns:

Name:	Description:
Best Record:	
b.CUSTOMEF	<none>
b.TITLE	<none>
b.FIRST_NAM	<none>
b.LAST_NAM	<none>
b.GENDER_II	<none>

Functions:

Name:	Description:
SIZEOF	Number of Characters in S
TRIM	Removes Spaces

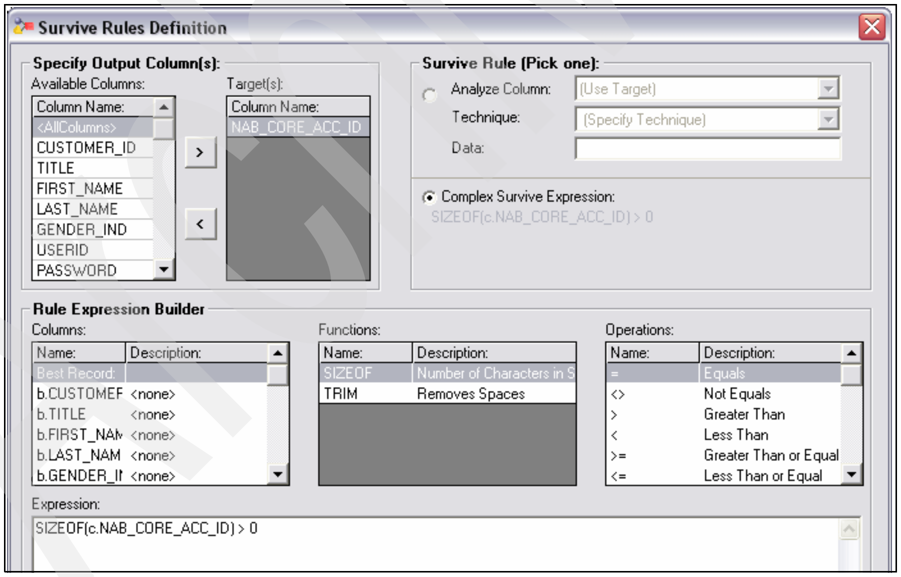
Operations:

Name:	Description:
=	Equals
<>	Not Equals
>	Greater Than
<	Less Than
>=	Greater Than or Equal
<=	Less Than or Equal

Expression:

```
SIZEOF(c.NAB_CORE_CUST_ID) > 0
```

Figure 2-362 Create j16_SURVIVE_CRM 13/25



Survive Rules Definition

Specify Output Column(s):

Available Columns:

Column Name:
<AllColumns>
CUSTOMER_ID
TITLE
FIRST_NAME
LAST_NAME
GENDER_IND
USERID
PASSWORD

Target(s):

Column Name:
NAB_CORE_ACC_ID

Survive Rule (Pick one):

Analyze Column: (Use Target)

Technique: (Specify Technique)

Data:

Complex Survive Expression: SIZEOF(c.NAB_CORE_ACC_ID) > 0

Rule Expression Builder

Columns:

Name:	Description:
Best Record:	
b.CUSTOMEF	<none>
b.TITLE	<none>
b.FIRST_NAM	<none>
b.LAST_NAM	<none>
b.GENDER_II	<none>

Functions:

Name:	Description:
SIZEOF	Number of Characters in S
TRIM	Removes Spaces

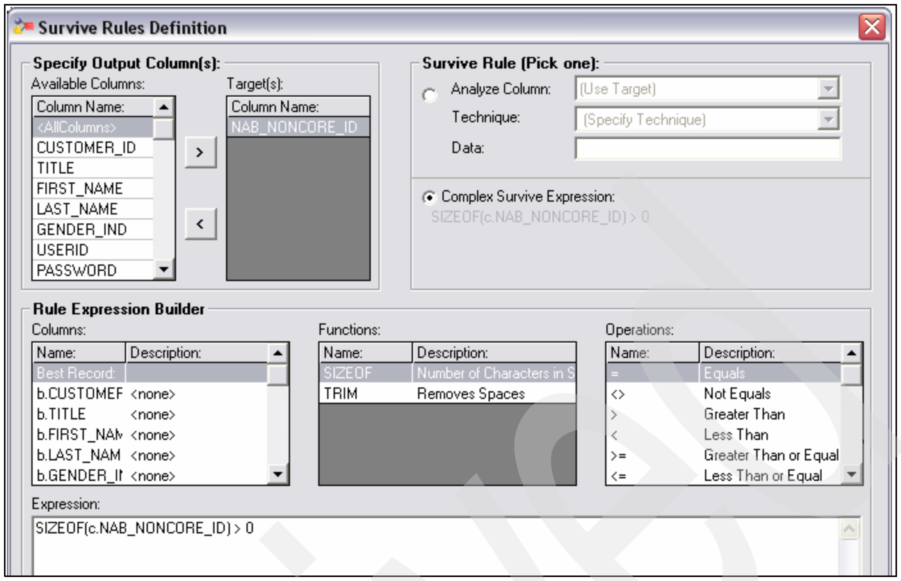
Operations:

Name:	Description:
=	Equals
<>	Not Equals
>	Greater Than
<	Less Than
>=	Greater Than or Equal
<=	Less Than or Equal

Expression:

```
SIZEOF(c.NAB_CORE_ACC_ID) > 0
```

Figure 2-363 Create j16_SURVIVE_CRM 14/25



Complex Survive Expression: SIZEOF(c.NAB_NONCORE_ID) > 0

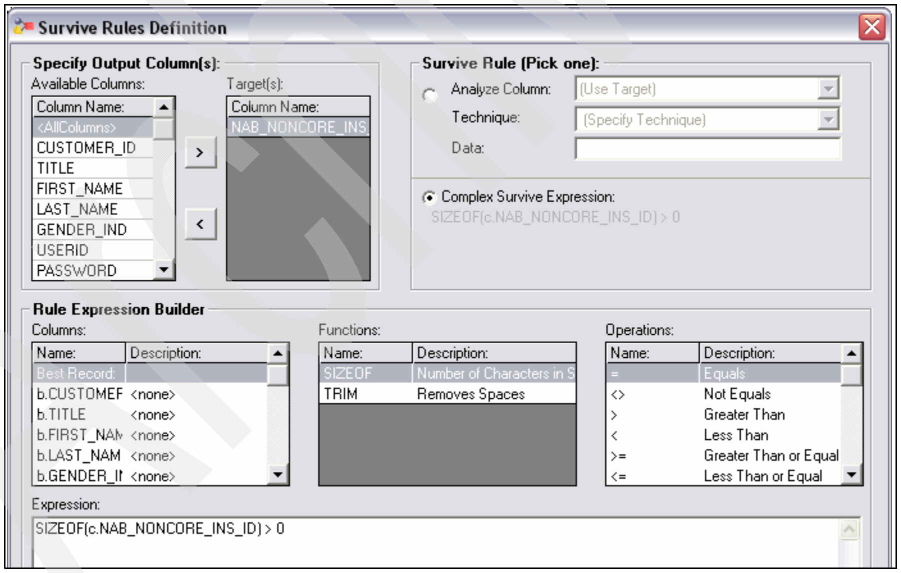
Name:	Description:
Best Record:	
b.CUSTOMEF	<none>
b.TITLE	<none>
b.FIRST_NAM	<none>
b.LAST_NAM	<none>
b.GENDER_II	<none>

Name:	Description:
SIZEOF	Number of Characters in S
TRIM	Removes Spaces

Name:	Description:
=	Equals
<>	Not Equals
>	Greater Than
<	Less Than
>=	Greater Than or Equal
<=	Less Than or Equal

Expression: SIZEOF(c.NAB_NONCORE_ID) > 0

Figure 2-364 Create j16_SURVIVE_CRM 15/25



Complex Survive Expression: SIZEOF(c.NAB_NONCORE_INS_ID) > 0

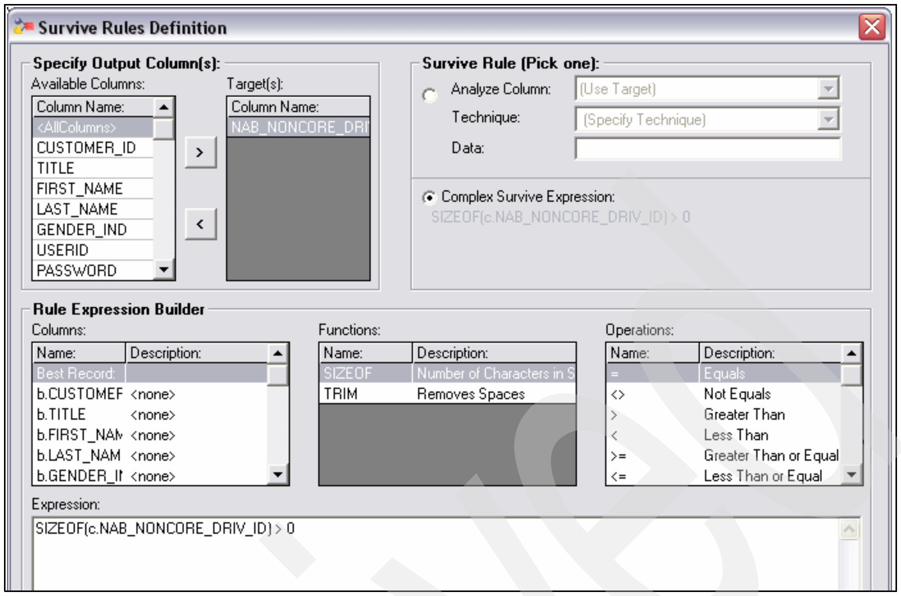
Name:	Description:
Best Record:	
b.CUSTOMEF	<none>
b.TITLE	<none>
b.FIRST_NAM	<none>
b.LAST_NAM	<none>
b.GENDER_II	<none>

Name:	Description:
SIZEOF	Number of Characters in S
TRIM	Removes Spaces

Name:	Description:
=	Equals
<>	Not Equals
>	Greater Than
<	Less Than
>=	Greater Than or Equal
<=	Less Than or Equal

Expression: SIZEOF(c.NAB_NONCORE_INS_ID) > 0

Figure 2-365 Create j16_SURVIVE_CRM 16/25



Survive Rules Definition

Specify Output Column(s):

Available Columns:

Column Name:
<AllColumns>
CUSTOMER_ID
TITLE
FIRST_NAME
LAST_NAME
GENDER_IND
USERID
PASSWORD

Target(s):

Column Name:
NAB_NONCORE_DRI

Survive Rule (Pick one):

Analyze Column: (Use Target)

Technique: (Specify Technique)

Data:

Complex Survive Expression:

SIZEOF(c.NAB_NONCORE_DRIV_ID) > 0

Rule Expression Builder

Columns:

Name:	Description:
Best Record:	
b.CUSTOMEF	<none>
b.TITLE	<none>
b.FIRST_NAI	<none>
b.LAST_NAM	<none>
b.GENDER_II	<none>

Functions:

Name:	Description:
SIZEOF	Number of Characters in S
TRIM	Removes Spaces

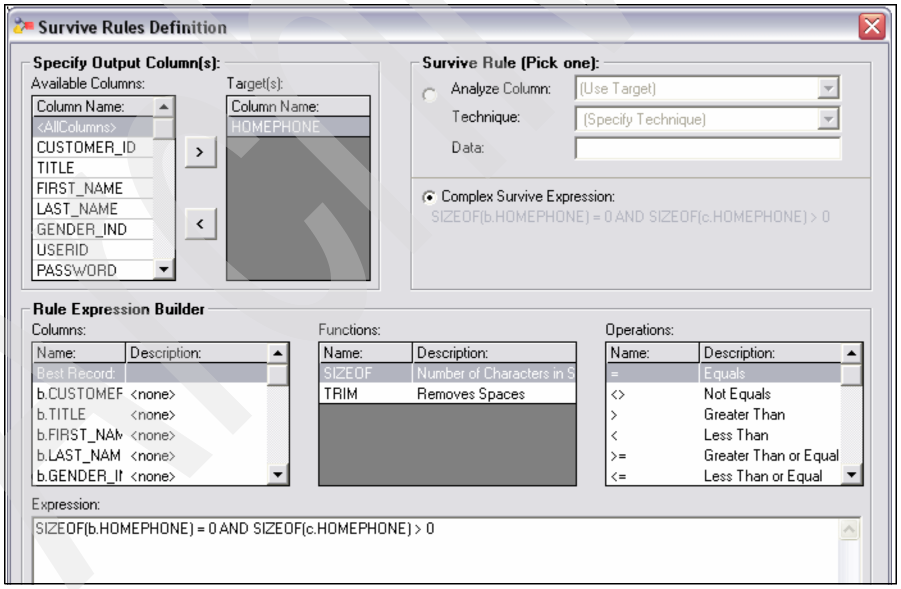
Operations:

Name:	Description:
=	Equals
<>	Not Equals
>	Greater Than
<	Less Than
>=	Greater Than or Equal
<=	Less Than or Equal

Expression:

```
SIZEOF(c.NAB_NONCORE_DRIV_ID) > 0
```

Figure 2-366 Create j16_SURVIVE_CRM 17/25



Survive Rules Definition

Specify Output Column(s):

Available Columns:

Column Name:
<AllColumns>
CUSTOMER_ID
TITLE
FIRST_NAME
LAST_NAME
GENDER_IND
USERID
PASSWORD

Target(s):

Column Name:
HOMEPHONE

Survive Rule (Pick one):

Analyze Column: (Use Target)

Technique: (Specify Technique)

Data:

Complex Survive Expression:

SIZEOF(b.HOMEPHONE) = 0 AND SIZEOF(c.HOMEPHONE) > 0

Rule Expression Builder

Columns:

Name:	Description:
Best Record:	
b.CUSTOMEF	<none>
b.TITLE	<none>
b.FIRST_NAI	<none>
b.LAST_NAM	<none>
b.GENDER_II	<none>

Functions:

Name:	Description:
SIZEOF	Number of Characters in S
TRIM	Removes Spaces

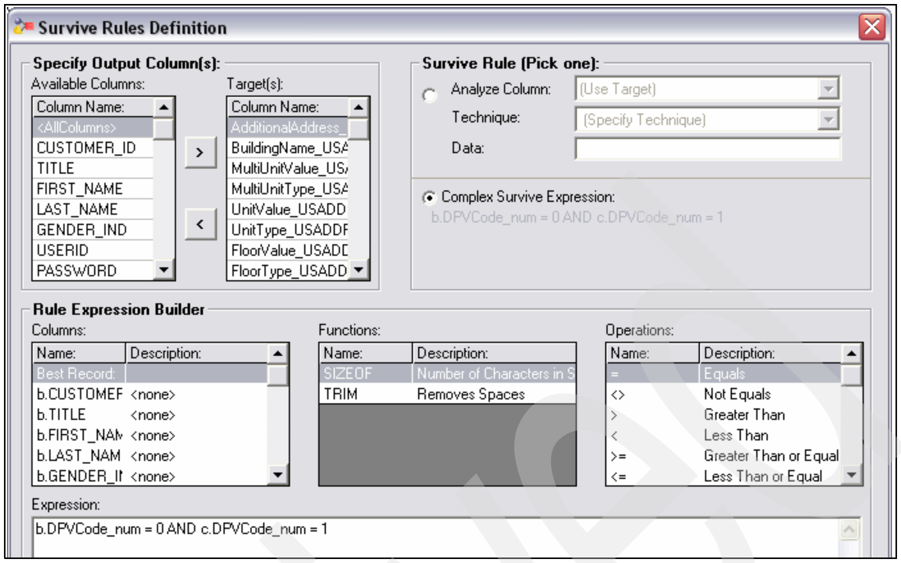
Operations:

Name:	Description:
=	Equals
<>	Not Equals
>	Greater Than
<	Less Than
>=	Greater Than or Equal
<=	Less Than or Equal

Expression:

```
SIZEOF(b.HOMEPHONE) = 0 AND SIZEOF(c.HOMEPHONE) > 0
```

Figure 2-367 Create j16_SURVIVE_CRM 18/25



Available Columns:

Column Name:
<AllColumns>
CUSTOMER_ID
TITLE
FIRST_NAME
LAST_NAME
GENDER_IND
USERID
PASSWORD

Target(s):

Column Name:
AdditionalAddress_
BuildingName_USA
MultiUnitValue_US/
MultiUnitType_USA
UnitValue_USADD
UnitType_USADDF
FloorValue_USADE
FloorType_USADD

Survive Rule (Pick one):
- Analyze Column: (Use Target)
- Technique: (Specify Technique)
- Data:
- [✓ Complex Survive Expression:] b.DPVCode_num = 0 AND c.DPVCode_num = 1

Columns:

Name:	Description:
Best Record:	
b.CUSTOMEF	<none>
b.TITLE	<none>
b.FIRST_NAN	<none>
b.LAST_NAM	<none>
b.GENDER_II	<none>

Functions:

Name:	Description:
SIZEOF	Number of Characters in S
TRIM	Removes Spaces

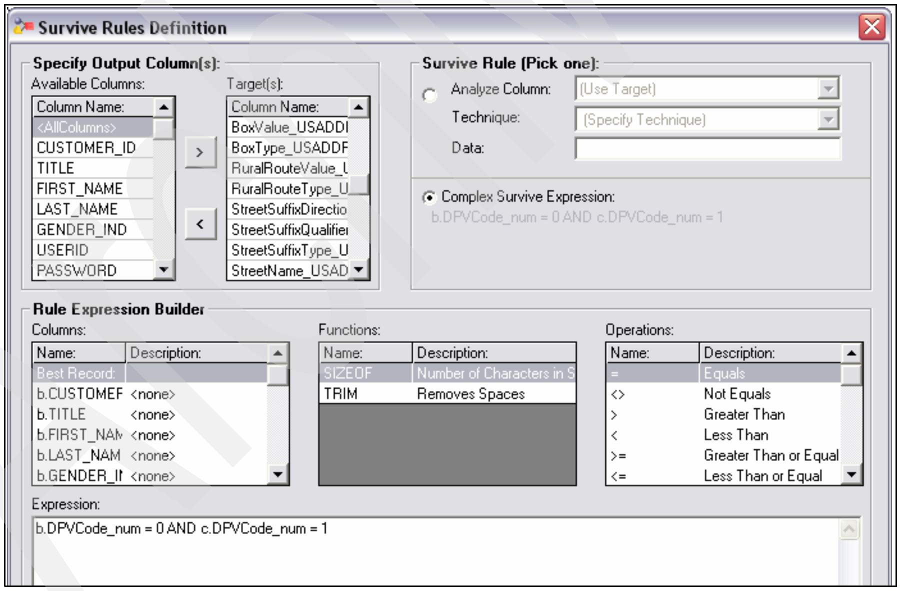
Operations:

Name:	Description:
=	Equals
<>	Not Equals
>	Greater Than
<	Less Than
>=	Greater Than or Equal
<=	Less Than or Equal

Expression:

```
b.DPVCode_num = 0 AND c.DPVCode_num = 1
```

Figure 2-368 Create j16_SURVIVE_CRM 19/25



Available Columns:

Column Name:
<AllColumns>
CUSTOMER_ID
TITLE
FIRST_NAME
LAST_NAME
GENDER_IND
USERID
PASSWORD

Target(s):

Column Name:
BoxValue_USADDI
BoxType_USADDF
RuralRouteValue_l
RuralRouteType_U
StreetSuffixDirectio
StreetSuffixQualifie
StreetSuffixType_U
StreetName_USAD

Survive Rule (Pick one):
- Analyze Column: (Use Target)
- Technique: (Specify Technique)
- Data:
- [✓ Complex Survive Expression:] b.DPVCode_num = 0 AND c.DPVCode_num = 1

Columns:

Name:	Description:
Best Record:	
b.CUSTOMEF	<none>
b.TITLE	<none>
b.FIRST_NAN	<none>
b.LAST_NAM	<none>
b.GENDER_II	<none>

Functions:

Name:	Description:
SIZEOF	Number of Characters in S
TRIM	Removes Spaces

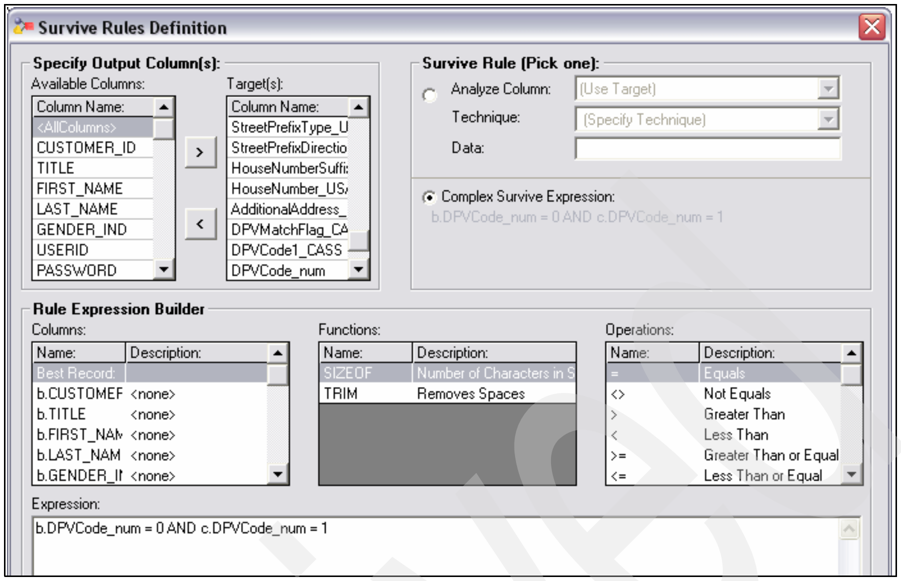
Operations:

Name:	Description:
=	Equals
<>	Not Equals
>	Greater Than
<	Less Than
>=	Greater Than or Equal
<=	Less Than or Equal

Expression:

```
b.DPVCode_num = 0 AND c.DPVCode_num = 1
```

Figure 2-369 Create j16_SURVIVE_CRM 20/25



Survive Rules Definition

Specify Output Column(s):

Available Columns:

Column Name:
<AllColumns>
CUSTOMER_ID
TITLE
FIRST_NAME
LAST_NAME
GENDER_IND
USERID
PASSWORD

Target(s):

Column Name:
StreetPrefixType_U
StreetPrefixDirectio
HouseNumberSuffi:
HouseNumber_US/
AdditionalAddress_
DPVMatchFlag_CA
DPVCode1_CASS
DPVCode_num

Survive Rule (Pick one):

Analyze Column: (Use Target)

Technique: (Specify Technique)

Data:

[✓ Complex Survive Expression:] b.DPVCode_num = 0 AND c.DPVCode_num = 1

Rule Expression Builder

Columns:

Name:	Description:
Best Record:	
b.CUSTOMEF	<none>
b.TITLE	<none>
b.FIRST_NAM	<none>
b.LAST_NAM	<none>
b.GENDER_II	<none>

Functions:

Name:	Description:
SIZEOF	Number of Characters in S
TRIM	Removes Spaces

Operations:

Name:	Description:
=	Equals
<>	Not Equals
>	Greater Than
<	Less Than
>=	Greater Than or Equal
<=	Less Than or Equal

Expression:

```
b.DPVCode_num = 0 AND c.DPVCode_num = 1
```

Figure 2-370 Create j16_SURVIVE_CRM 21/25

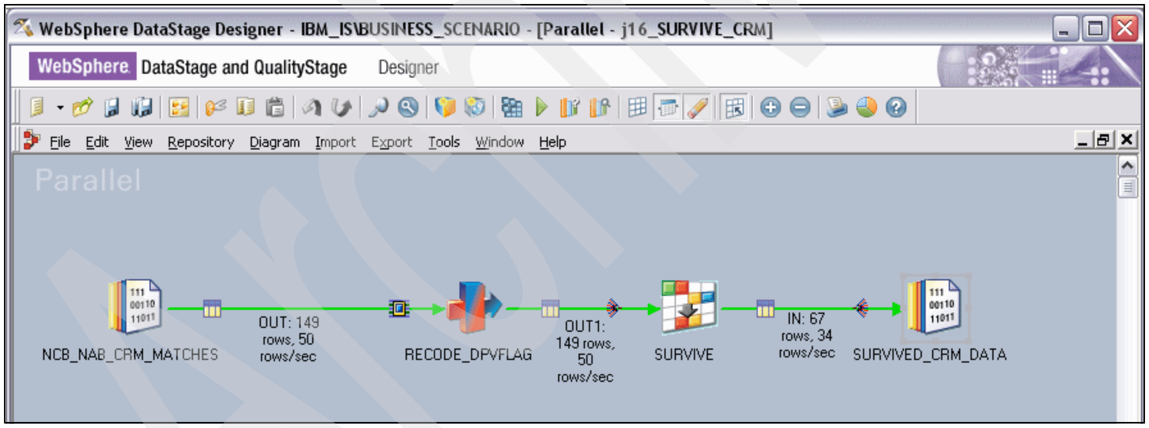
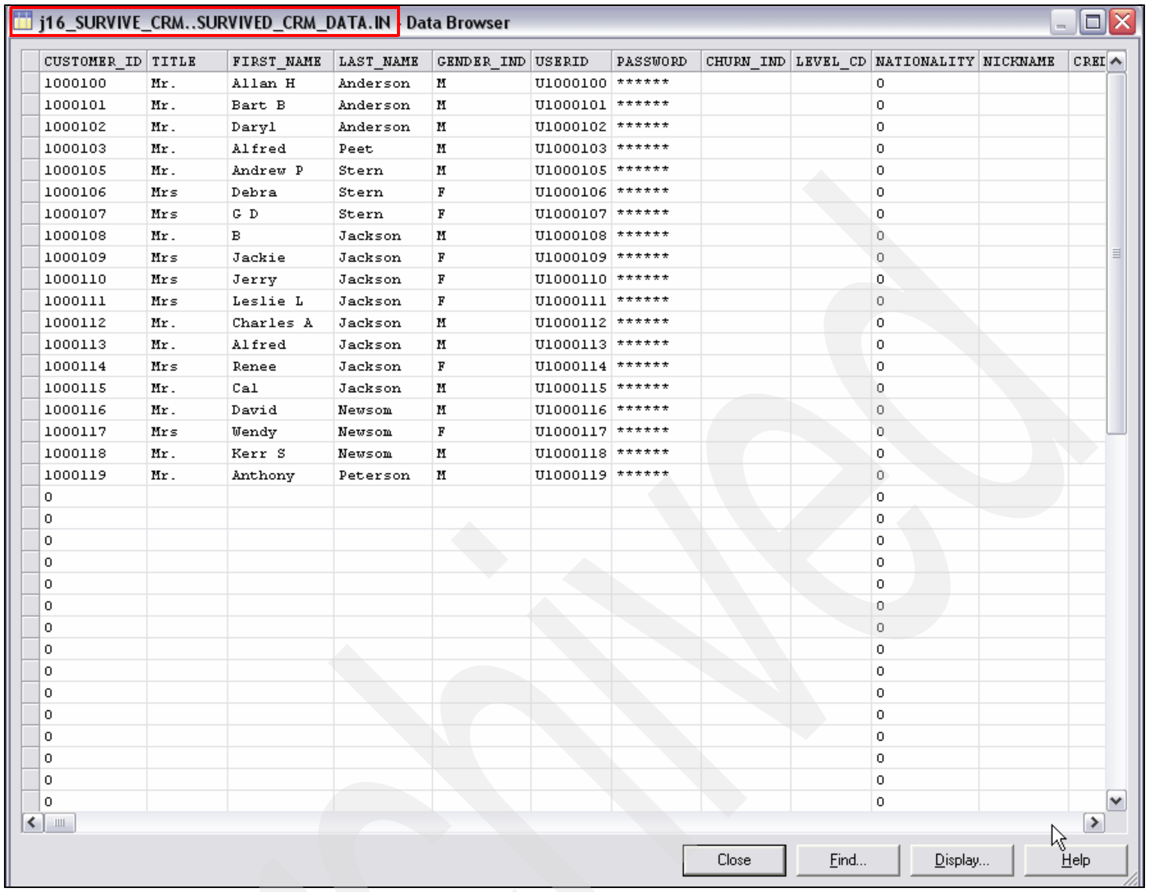
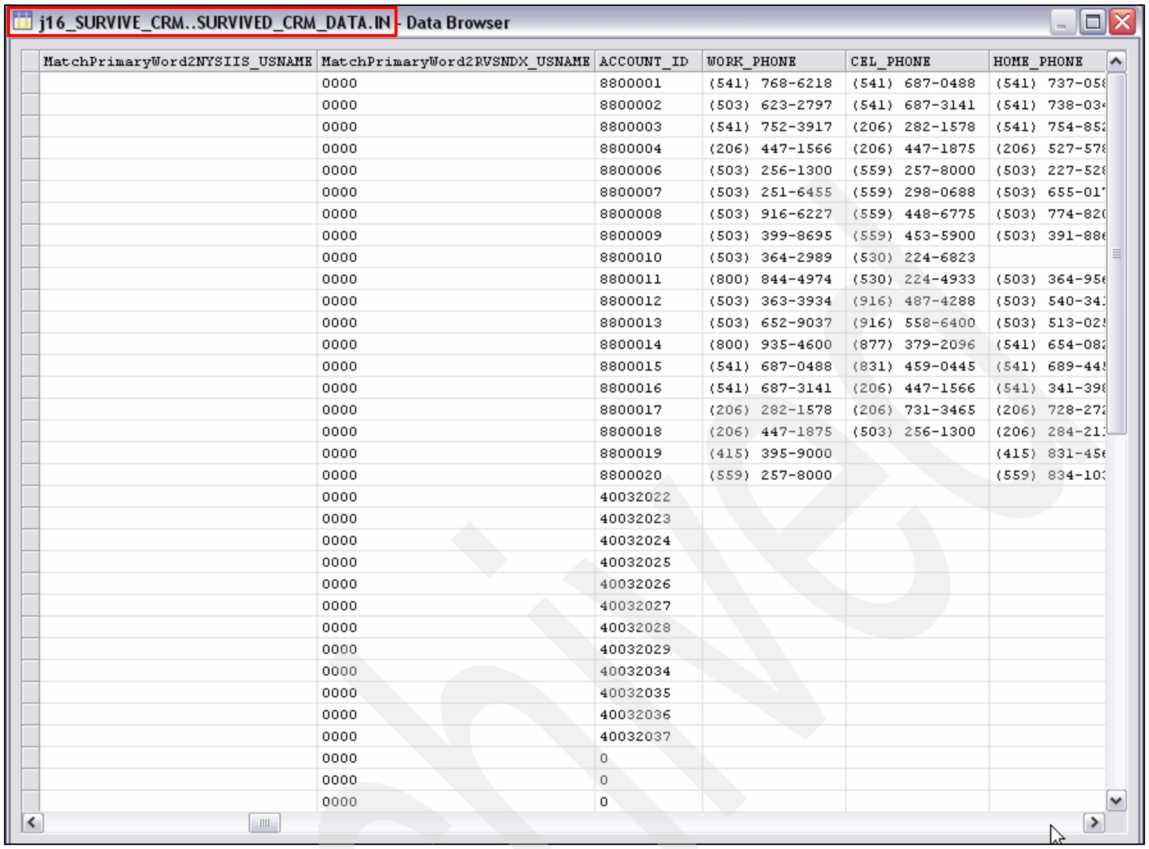

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j16_SURVIVE_CRM]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
NCB_NAB_CRM_MATCHES
OUT: 149 rows, 50 rows/sec
RECODE_DPVFLAG
OUT1: 149 rows, 50 rows/sec
SURVIVE
IN: 67 rows, 34 rows/sec
SURVIVED_CRM_DATA

Figure 2-371 Create j16_SURVIVE_CRM 22/25



CUSTOMER_ID	TITLE	FIRST_NAME	LAST_NAME	GENDER_IND	USERID	PASSWORD	CHURN_IND	LEVEL_CD	NATIONALITY	NICKNAME	CREI
1000100	Mr.	Allan H	Anderson	M	U1000100	******			0		
1000101	Mr.	Bart B	Anderson	M	U1000101	******			0		
1000102	Mr.	Daryl	Anderson	M	U1000102	******			0		
1000103	Mr.	Alfred	Peet	M	U1000103	******			0		
1000105	Mr.	Andrew P	Stern	M	U1000105	******			0		
1000106	Mrs	Debra	Stern	F	U1000106	******			0		
1000107	Mrs	G D	Stern	F	U1000107	******			0		
1000108	Mr.	B	Jackson	M	U1000108	******			0		
1000109	Mrs	Jackie	Jackson	F	U1000109	******			0		
1000110	Mrs	Jerry	Jackson	F	U1000110	******			0		
1000111	Mrs	Leslie L	Jackson	F	U1000111	******			0		
1000112	Mr.	Charles A	Jackson	M	U1000112	******			0		
1000113	Mr.	Alfred	Jackson	M	U1000113	******			0		
1000114	Mrs	Renee	Jackson	F	U1000114	******			0		
1000115	Mr.	Cal	Jackson	M	U1000115	******			0		
1000116	Mr.	David	Newsom	M	U1000116	******			0		
1000117	Mrs	Wendy	Newsom	F	U1000117	******			0		
1000118	Mr.	Kerr S	Newsom	M	U1000118	******			0		
1000119	Mr.	Anthony	Peterson	M	U1000119	******			0		
0									0		
0									0		
0									0		
0									0		
0									0		
0									0		
0									0		
0									0		
0									0		
0									0		
0									0		
0									0		
0									0		
0									0		
0									0		

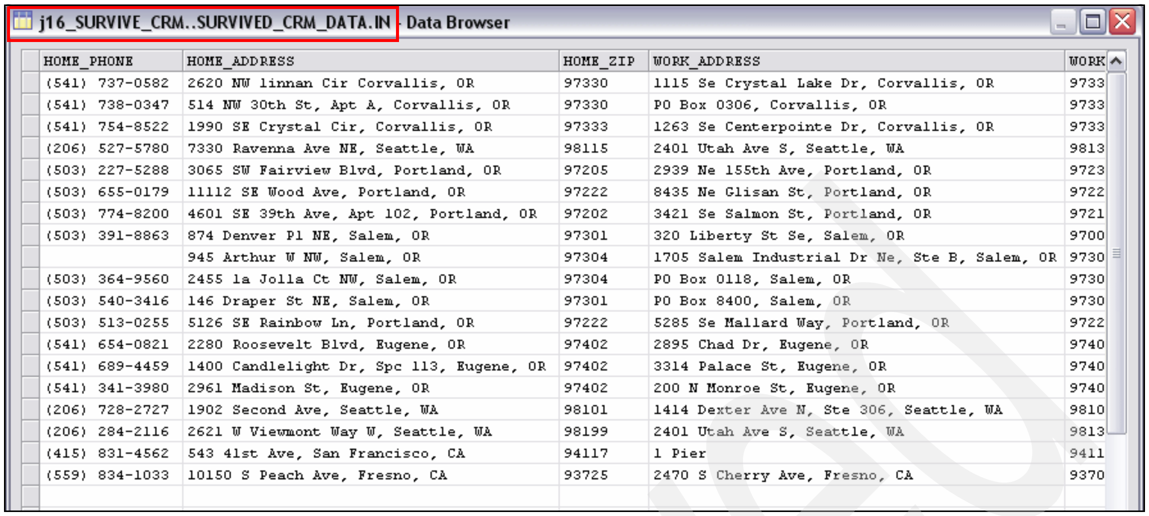
Figure 2-372 Create j16_SURVIVE_CRM 23/25



j16_SURVIVE_CRM..SURVIVED_CRM_DATA.IN - Data Browser

MatchPrimaryWord2NYSIIS_USNAME	MatchPrimaryWord2RVSNDX_USNAME	ACCOUNT_ID	WORK_PHONE	CEL_PHONE	HOME_PHONE
	0000	8800001	(541) 768-6218	(541) 687-0488	(541) 737-05
	0000	8800002	(503) 623-2797	(541) 687-3141	(541) 738-03
	0000	8800003	(541) 752-3917	(206) 282-1578	(541) 754-85
	0000	8800004	(206) 447-1566	(206) 447-1875	(206) 527-57
	0000	8800006	(503) 256-1300	(559) 257-8000	(503) 227-52
	0000	8800007	(503) 251-6455	(559) 298-0688	(503) 655-01
	0000	8800008	(503) 916-6227	(559) 448-6775	(503) 774-82
	0000	8800009	(503) 399-8695	(559) 453-5900	(503) 391-88
	0000	8800010	(503) 364-2989	(530) 224-6823	
	0000	8800011	(800) 844-4974	(530) 224-4933	(503) 364-95
	0000	8800012	(503) 363-3934	(916) 487-4288	(503) 540-34
	0000	8800013	(503) 652-9037	(916) 558-6400	(503) 513-02
	0000	8800014	(800) 935-4600	(877) 379-2096	(541) 654-08
	0000	8800015	(541) 687-0488	(831) 459-0445	(541) 689-44
	0000	8800016	(541) 687-3141	(206) 447-1566	(541) 341-39
	0000	8800017	(206) 282-1578	(206) 731-3465	(206) 728-27
	0000	8800018	(206) 447-1875	(503) 256-1300	(206) 284-21
	0000	8800019	(415) 395-9000		(415) 831-45
	0000	8800020	(559) 257-8000		(559) 834-10
	0000	40032022			
	0000	40032023			
	0000	40032024			
	0000	40032025			
	0000	40032026			
	0000	40032027			
	0000	40032028			
	0000	40032029			
	0000	40032034			
	0000	40032035			
	0000	40032036			
	0000	40032037			
	0000	0			
	0000	0			
	0000	0			

Figure 2-373   Create j16_SURVIVE_CRM 24/25



HOME_PHONE	HOME_ADDRESS	HOME_ZIP	WORK_ADDRESS	WORK
(541) 737-0582	2620 NW linnan Cir Corvallis, OR	97330	1115 Se Crystal Lake Dr, Corvallis, OR	9733
(541) 738-0347	514 NW 30th St, Apt A, Corvallis, OR	97330	PO Box 0306, Corvallis, OR	9733
(541) 754-8522	1990 SE Crystal Cir, Corvallis, OR	97333	1263 Se Centerpointe Dr, Corvallis, OR	9733
(206) 527-5780	7330 Ravenna Ave NE, Seattle, WA	98115	2401 Utah Ave S, Seattle, WA	9813
(503) 227-5288	3065 SW Fairview Blvd, Portland, OR	97205	2939 Ne 155th Ave, Portland, OR	9723
(503) 655-0179	11112 SE Wood Ave, Portland, OR	97222	8435 Ne Glisan St, Portland, OR	9722
(503) 774-8200	4601 SE 39th Ave, Apt 102, Portland, OR	97202	3421 Se Salmon St, Portland, OR	9721
(503) 391-8863	874 Denver Pl NE, Salem, OR	97301	320 Liberty St Se, Salem, OR	9700
	945 Arthur W NW, Salem, OR	97304	1705 Salem Industrial Dr Ne, Ste B, Salem, OR	9730
(503) 364-9560	2455 la Jolla Ct NW, Salem, OR	97304	PO Box 0118, Salem, OR	9730
(503) 540-3416	146 Draper St NE, Salem, OR	97301	PO Box 8400, Salem, OR	9730
(503) 513-0255	5126 SE Rainbow Ln, Portland, OR	97222	5285 Se Mallard Way, Portland, OR	9722
(541) 654-0821	2280 Roosevelt Blvd, Eugene, OR	97402	2895 Chad Dr, Eugene, OR	9740
(541) 689-4459	1400 Candlelight Dr, Spc 113, Eugene, OR	97402	3314 Palace St, Eugene, OR	9740
(541) 341-3980	2961 Madison St, Eugene, OR	97402	200 N Monroe St, Eugene, OR	9740
(206) 728-2727	1902 Second Ave, Seattle, WA	98101	1414 Dexter Ave N, Ste 306, Seattle, WA	9810
(206) 284-2116	2621 W Viewmont Way W, Seattle, WA	98199	2401 Utah Ave S, Seattle, WA	9813
(415) 831-4562	543 41st Ave, San Francisco, CA	94117	1 Pier	9411
(559) 834-1033	10150 S Peach Ave, Fresno, CA	93725	2470 S Cherry Ave, Fresno, CA	9370

Figure 2-374 Create j16_SURVIVE_CRM 25/25

j17_FUNNEL_UNDUP_RES_DATA

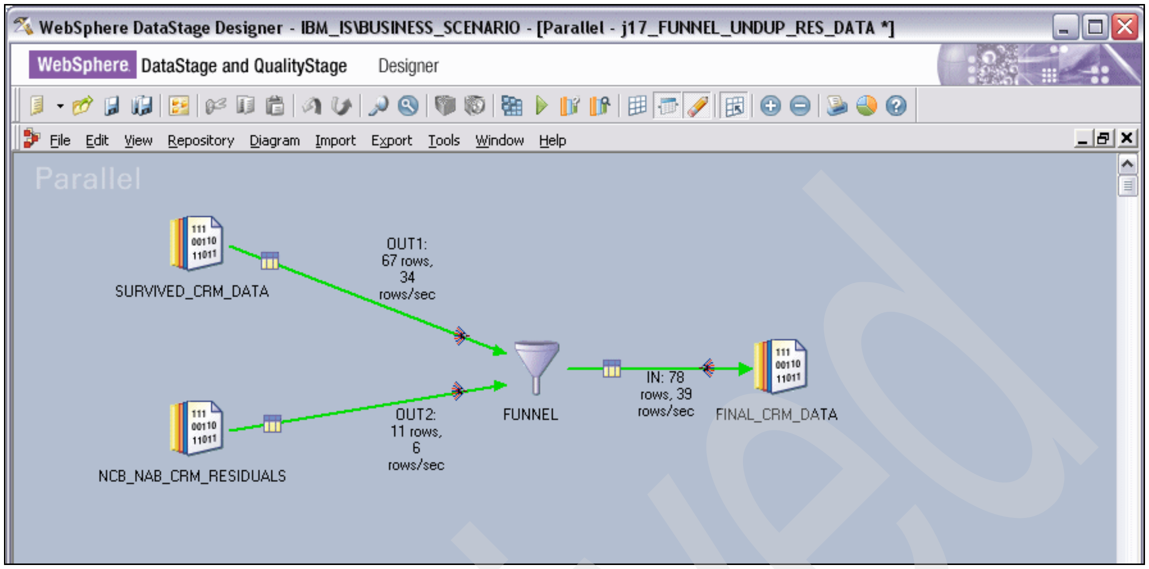
Next, we merged the survived records from “j16_SURVIVE_CRM” on page 814 with the residual records that were created in “j15_UNDUP_CRM” on page 809 using a Funnel stage. The information from these merged files is required to update the CRM system.

Figure 2-375 on page 830 shows the various stages that are used in this job. It includes the data sets that were created in the“j15_UNDUP_CRM” on page 809 and “j16_SURVIVE_CRM” on page 814, a Funnel stage, and an output Data Set stage. We modified the stage names as shown.

Because we have discussed these configurations previously, we do not repeat them here.

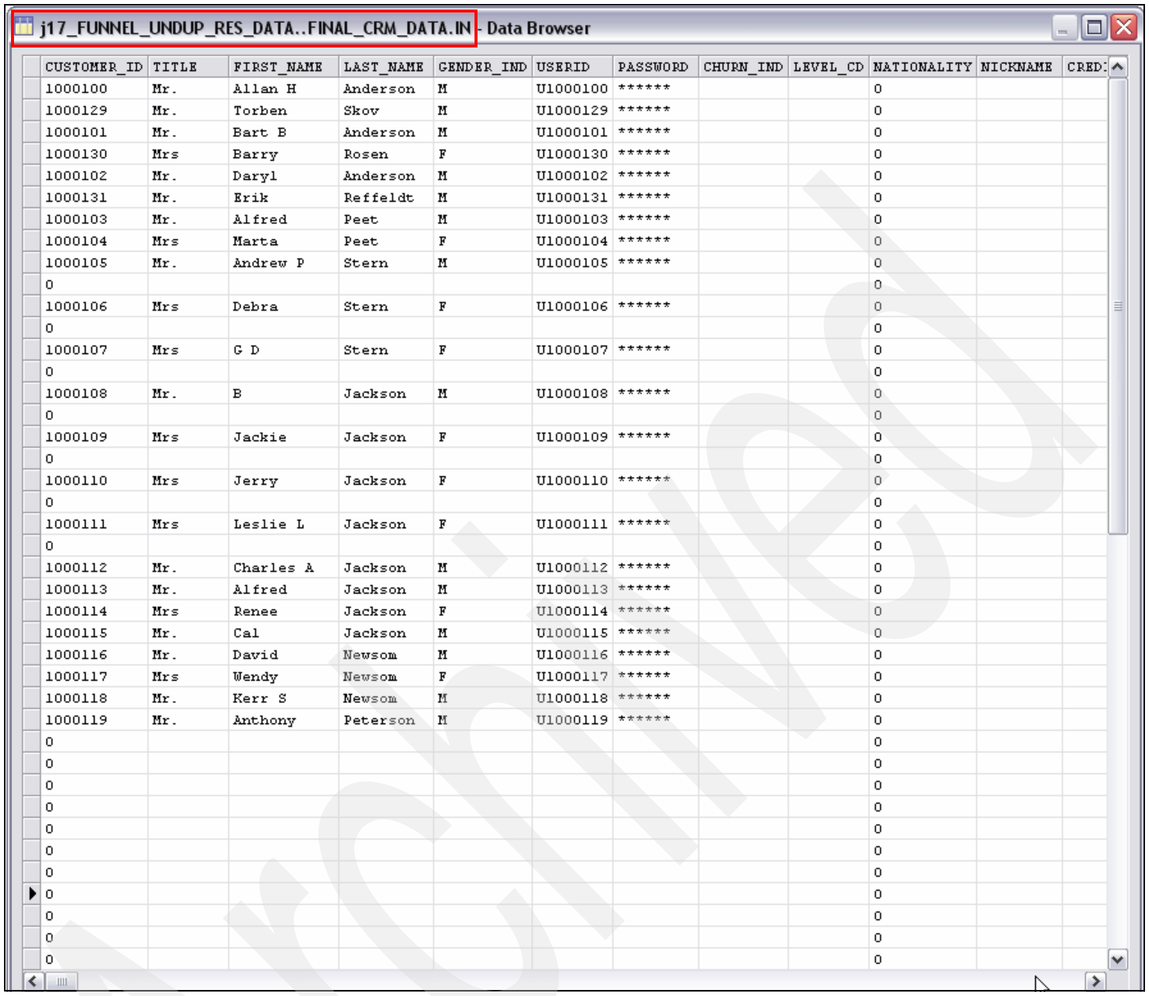
Figure 2-375 on page 830 shows the results after compiling and running this job. The output data set object contents (78 records) are shown in Figure 2-376 on page 831 through Figure 2-377 on page 832.

Proceed now to “j18_CRM_DATA_TRANSFORM” on page 833.



WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j17_FUNNEL_UNDUP_RES_DATA *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
SURVIVED_CRM_DATA
OUT1: 67 rows, 34 rows/sec
NCB_NAB_CRM_RESIDUALS
OUT2: 11 rows, 6 rows/sec
FUNNEL
IN: 78 rows, 39 rows/sec
FINAL_CRM_DATA

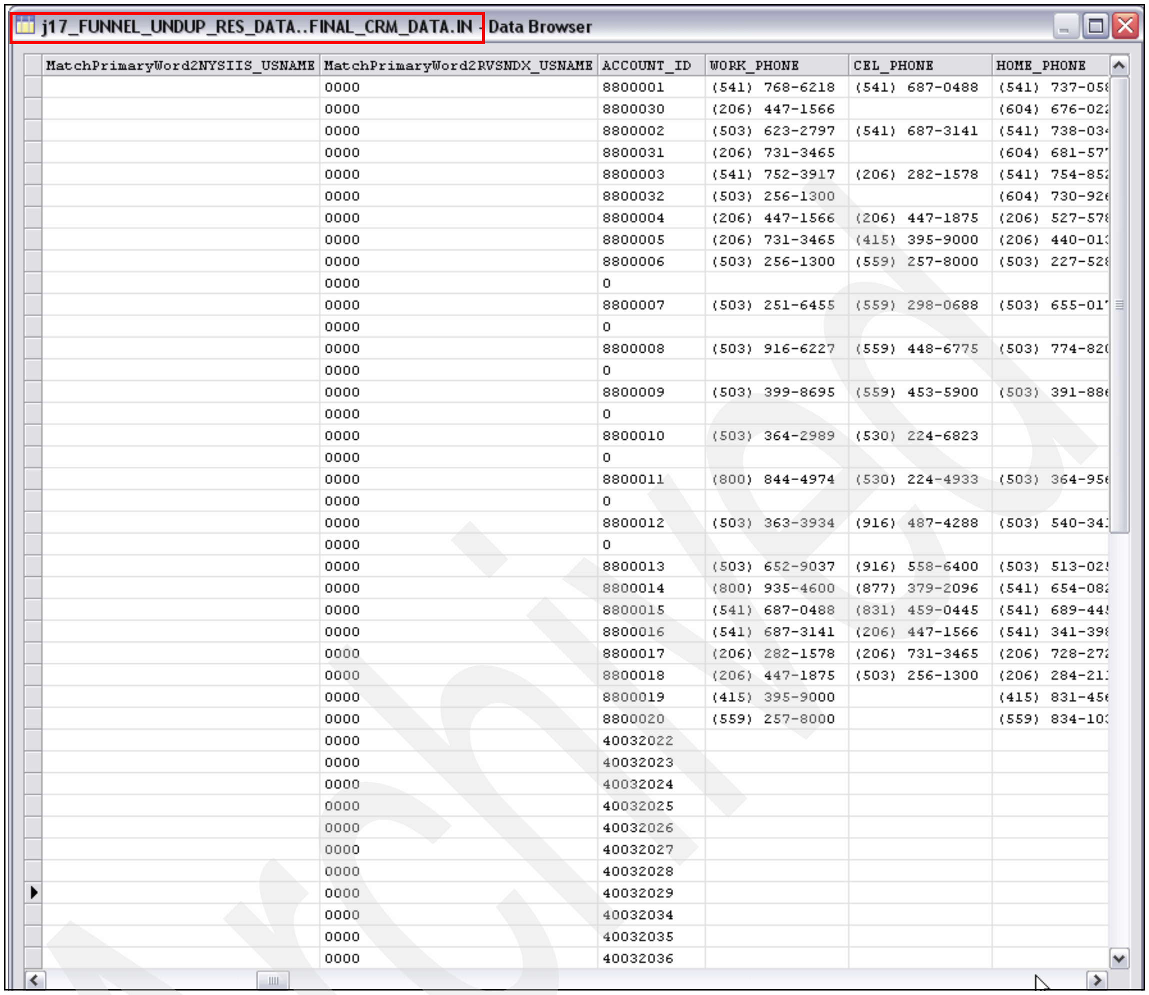
Figure 2-375 Create j17_FUNNEL_UNDUP_RES_DATA 1/4



j17_FUNNEL_UNDUP_RES_DATA..FINAL_CRM_DATA.IN - Data Browser

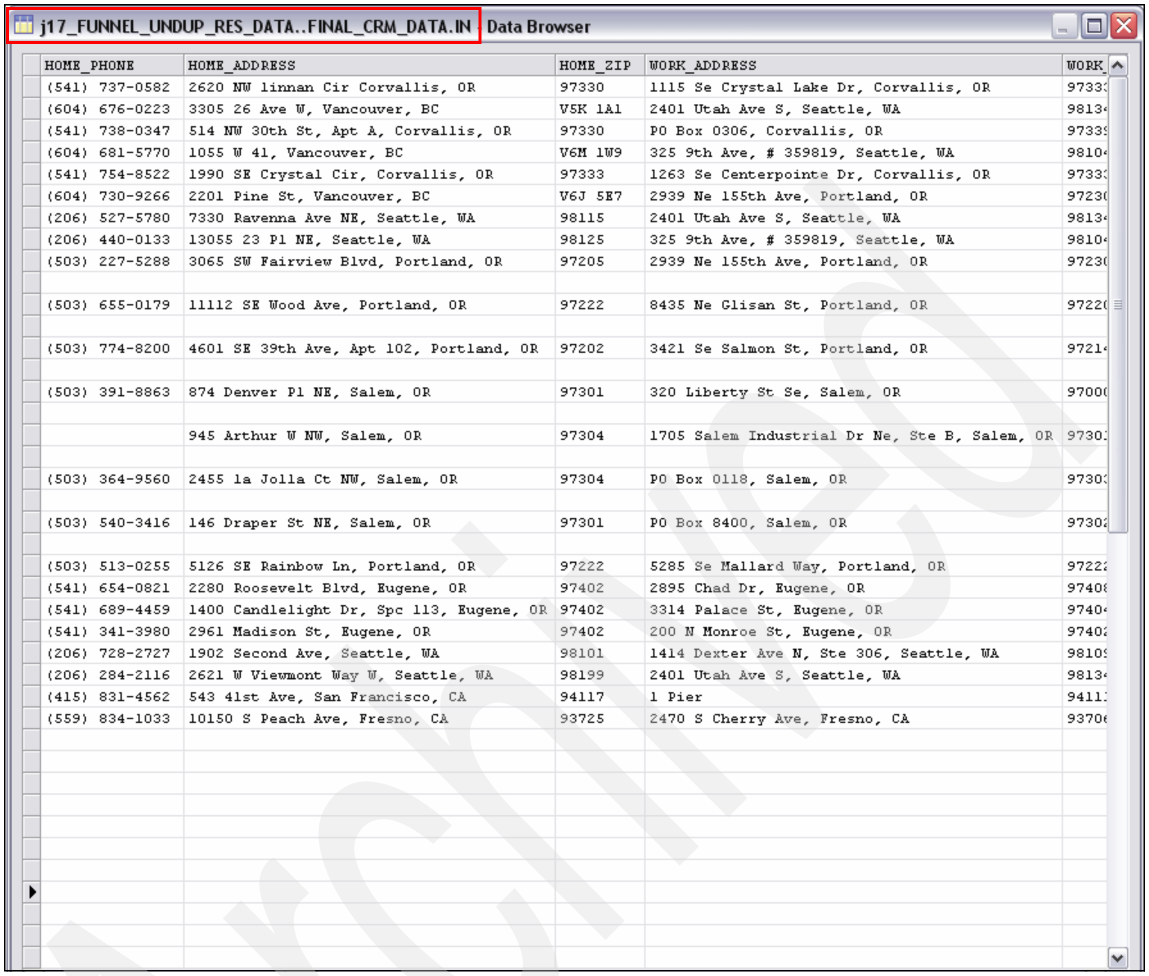
CUSTOMER_ID	TITLE	FIRST_NAME	LAST_NAME	GENDER_IND	USERID	PASSWORD	CHURN_IND	LEVEL_CD	NATIONALITY	NICKNAME	CRED
1000100	Mr.	Allan H	Anderson	M	U1000100	******			0		
1000129	Mr.	Torben	Skov	M	U1000129	******			0		
1000101	Mr.	Bart B	Anderson	M	U1000101	******			0		
1000130	Mrs	Barry	Rosen	F	U1000130	******			0		
1000102	Mr.	Daryl	Anderson	M	U1000102	******			0		
1000131	Mr.	Erik	Reffeldt	M	U1000131	******			0		
1000103	Mr.	Alfred	Peet	M	U1000103	******			0		
1000104	Mrs	Marta	Peet	F	U1000104	******			0		
1000105	Mr.	Andrew P	Stern	M	U1000105	******			0		
0									0		
1000106	Mrs	Debra	Stern	F	U1000106	******			0		
0									0		
1000107	Mrs	G D	Stern	F	U1000107	******			0		
0									0		
1000108	Mr.	B	Jackson	M	U1000108	******			0		
0									0		
1000109	Mrs	Jackie	Jackson	F	U1000109	******			0		
0									0		
1000110	Mrs	Jerry	Jackson	F	U1000110	******			0		
0									0		
1000111	Mrs	Leslie L	Jackson	F	U1000111	******			0		
0									0		
1000112	Mr.	Charles A	Jackson	M	U1000112	******			0		
1000113	Mr.	Alfred	Jackson	M	U1000113	******			0		
1000114	Mrs	Renee	Jackson	F	U1000114	******			0		
1000115	Mr.	Cal	Jackson	M	U1000115	******			0		
1000116	Mr.	David	Newsom	M	U1000116	******			0		
1000117	Mrs	Wendy	Newsom	F	U1000117	******			0		
1000118	Mr.	Kerr S	Newsom	M	U1000118	******			0		
1000119	Mr.	Anthony	Peterson	M	U1000119	******			0		
0									0		
0									0		
0									0		
0									0		
0									0		
0									0		
0									0		
0									0		
0									0		
0									0		
0									0		

Figure 2-376 Create j17_FUNNEL_UNDUP_RES_DATA 2/4


j17_FUNNEL_UNDUP_RES_DATA..FINAL_CRM_DATA.IN - Data Browser

MatchPrimaryWord2NYSIIS_USNAME	MatchPrimaryWord2RVSNDX_USNAME	ACCOUNT_ID	WORK_PHONE	CEL_PHONE	HOME_PHONE
	0000	8800001	(541) 768-6218	(541) 687-0488	(541) 737-05
	0000	8800030	(206) 447-1566		(604) 676-02
	0000	8800002	(503) 623-2797	(541) 687-3141	(541) 738-03
	0000	8800031	(206) 731-3465		(604) 681-57
	0000	8800003	(541) 752-3917	(206) 282-1578	(541) 754-85
	0000	8800032	(503) 256-1300		(604) 730-92
	0000	8800004	(206) 447-1566	(206) 447-1875	(206) 527-57
	0000	8800005	(206) 731-3465	(415) 395-9000	(206) 440-01
	0000	8800006	(503) 256-1300	(559) 257-8000	(503) 227-52
	0000	0			
	0000	8800007	(503) 251-6455	(559) 298-0688	(503) 655-01
	0000	0			
	0000	8800008	(503) 916-6227	(559) 448-6775	(503) 774-82
	0000	0			
	0000	8800009	(503) 399-8695	(559) 453-5900	(503) 391-88
	0000	0			
	0000	8800010	(503) 364-2989	(530) 224-6823	
	0000	0			
	0000	8800011	(800) 844-4974	(530) 224-4933	(503) 364-95
	0000	0			
	0000	8800012	(503) 363-3934	(916) 487-4288	(503) 540-34
	0000	0			
	0000	8800013	(503) 652-9037	(916) 558-6400	(503) 513-02
	0000	8800014	(800) 935-4600	(877) 379-2096	(541) 654-08
	0000	8800015	(541) 687-0488	(831) 459-0445	(541) 689-44
	0000	8800016	(541) 687-3141	(206) 447-1566	(541) 341-39
	0000	8800017	(206) 282-1578	(206) 731-3465	(206) 728-27
	0000	8800018	(206) 447-1875	(503) 256-1300	(206) 284-21
	0000	8800019	(415) 395-9000		(415) 831-45
	0000	8800020	(559) 257-8000		(559) 834-10
	0000	40032022			
	0000	40032023			
	0000	40032024			
	0000	40032025			
	0000	40032026			
	0000	40032027			
	0000	40032028			
	0000	40032029			
	0000	40032034			
	0000	40032035			
	0000	40032036			

Figure 2-377 Create j17_FUNNEL_UNDUP_RES_DATA 3/4


j17_FUNNEL_UNDUP_RES_DATA..FINAL_CRM_DATA.IN - Data Browser

HOME_PHONE	HOME_ADDRESS	HOME_ZIP	WORK_ADDRESS	WORK_
(541) 737-0582	2620 NW linnan Cir Corvallis, OR	97330	1115 Se Crystal Lake Dr, Corvallis, OR	9733
(604) 676-0223	3305 26 Ave W, Vancouver, BC	V5K 1A1	2401 Utah Ave S, Seattle, WA	9813
(541) 738-0347	514 NW 30th St, Apt A, Corvallis, OR	97330	PO Box 0306, Corvallis, OR	9733
(604) 681-5770	1055 W 41, Vancouver, BC	V6M 1W9	325 9th Ave, # 359819, Seattle, WA	9810
(541) 754-8522	1990 SE Crystal Cir, Corvallis, OR	97333	1263 Se Centerpointe Dr, Corvallis, OR	9733
(604) 730-9266	2201 Pine St, Vancouver, BC	V6J 5E7	2939 Ne 155th Ave, Portland, OR	9723
(206) 527-5780	7330 Ravenna Ave NE, Seattle, WA	98115	2401 Utah Ave S, Seattle, WA	9813
(206) 440-0133	13055 23 Pl NE, Seattle, WA	98125	325 9th Ave, # 359819, Seattle, WA	9810
(503) 227-5288	3065 SW Fairview Blvd, Portland, OR	97205	2939 Ne 155th Ave, Portland, OR	9723
(503) 655-0179	11112 SE Wood Ave, Portland, OR	97222	8435 Ne Glisan St, Portland, OR	9722
(503) 774-8200	4601 SE 39th Ave, Apt 102, Portland, OR	97202	3421 Se Salmon St, Portland, OR	9721
(503) 391-8863	874 Denver Pl NE, Salem, OR	97301	320 Liberty St Se, Salem, OR	9700
	945 Arthur W NW, Salem, OR	97304	1705 Salem Industrial Dr Ne, Ste B, Salem, OR	9730
(503) 364-9560	2455 la Jolla Ct NW, Salem, OR	97304	PO Box 0118, Salem, OR	9730
(503) 540-3416	146 Draper St NE, Salem, OR	97301	PO Box 8400, Salem, OR	9730
(503) 513-0255	5126 SE Rainbow Ln, Portland, OR	97222	5285 Se Mallard Way, Portland, OR	9722
(541) 654-0821	2280 Roosevelt Blvd, Eugene, OR	97402	2895 Chad Dr, Eugene, OR	9740
(541) 689-4459	1400 Candlelight Dr, Spc 113, Eugene, OR	97402	3314 Palace St, Eugene, OR	9740
(541) 341-3980	2961 Madison St, Eugene, OR	97402	200 N Monroe St, Eugene, OR	9740
(206) 728-2727	1902 Second Ave, Seattle, WA	98101	1414 Dexter Ave N, Ste 306, Seattle, WA	9810
(206) 284-2116	2621 W Viewmont Way W, Seattle, WA	98199	2401 Utah Ave S, Seattle, WA	9813
(415) 831-4562	543 41st Ave, San Francisco, CA	94117	1 Pier	9411
(559) 834-1033	10150 S Peach Ave, Fresno, CA	93725	2470 S Cherry Ave, Fresno, CA	9370

Figure 2-378 Create j17_FUNNEL_UNDUP_RES_DATA 4/4

j18_CRM_DATA_TRANSFORM

We need to transform the output of the "j17_FUNNEL_UNDUP_RES_DATA" on page 829 step prior to using it for updating the relevant address columns in the CRM system. Two Transformer stages performed these functions. The output of this step contained data to update the CRM system with the "best" information from the North American Bank and Northern California Bank's core and non-core systems.

Figure 2-379 on page 835 shows the various stages that are used in this job. It includes the data set that was created in the "j17_FUNNEL_UNDUP_RES_DATA" on page 829 step, two Transformer stages, a Funnel stage, and an output Data Set stage. We modified the stage names as shown.

The two Transformer stages are described here briefly:

- In the ADD_LEADING_SPACE - Transformer Stage windows in Figure 2-380 on page 836 and Figure 2-381 on page 837 a leading space is added to the components of the standardized address that will eventually need to be concatenated together to update the corresponding column. For example, the following standardized columns need to be concatenated together to update the corresponding HOMESTREET column in the CUSTOMER table in the CRM system:
  - HouseNumberSuffix_USADDR_HOME
  - StreetPrefixDirectional_USADDR_HOME
  - StreetPrefixType_USADDR_HOME
  - StreetName_USADDR_HOME

  For legibility after concatenation, we need to add a leading space to these standardized columns. This is performed in this stage.
- In the PICK_AND_CONCATENATE - Transformer Stage windows in Figure 2-382 on page 838 through Figure 2-387 on page 843 the appropriate columns are mapped to the columns defined in the CRM system. In particular, the HOMESTREET and WORKSTREET columns in the CUSTOMER table of the CRM system require a concatenation of data from multiple columns as shown in Figure 2-386 on page 842 and Figure 2-387 on page 843.

Figure 2-388 on page 844 shows the results after compiling and running this job. The output data set object contents are shown in Figure 2-389 on page 845 through Figure 2-392 on page 848. This information contains the “best” information from the North American Bank and Northern California Bank’s core and non-core systems to update the CRM system.

Note: We do not show the actual update of the CRM system tables using this file. The IBM WebSphere DataStage jobs that are used to perform this update are described in *IBM InfoSphere DataStage Data Flow and Job Design*, SG24-7576.

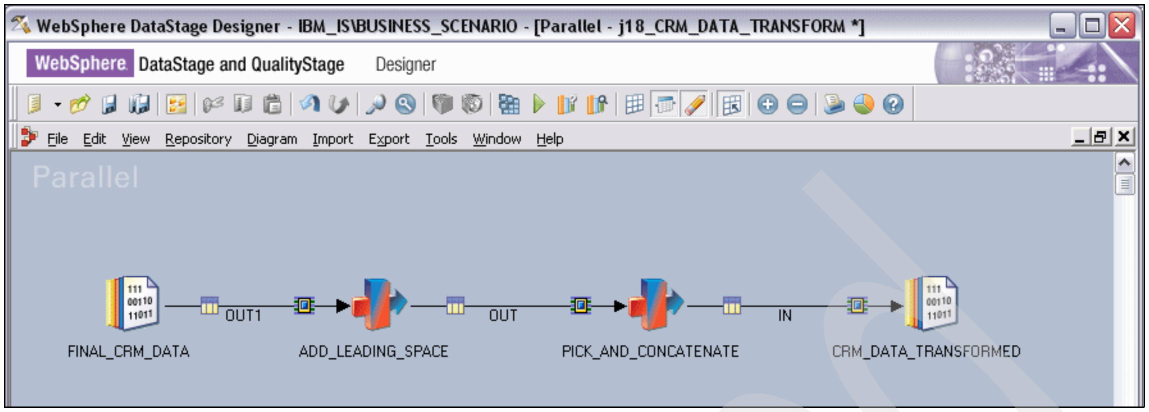
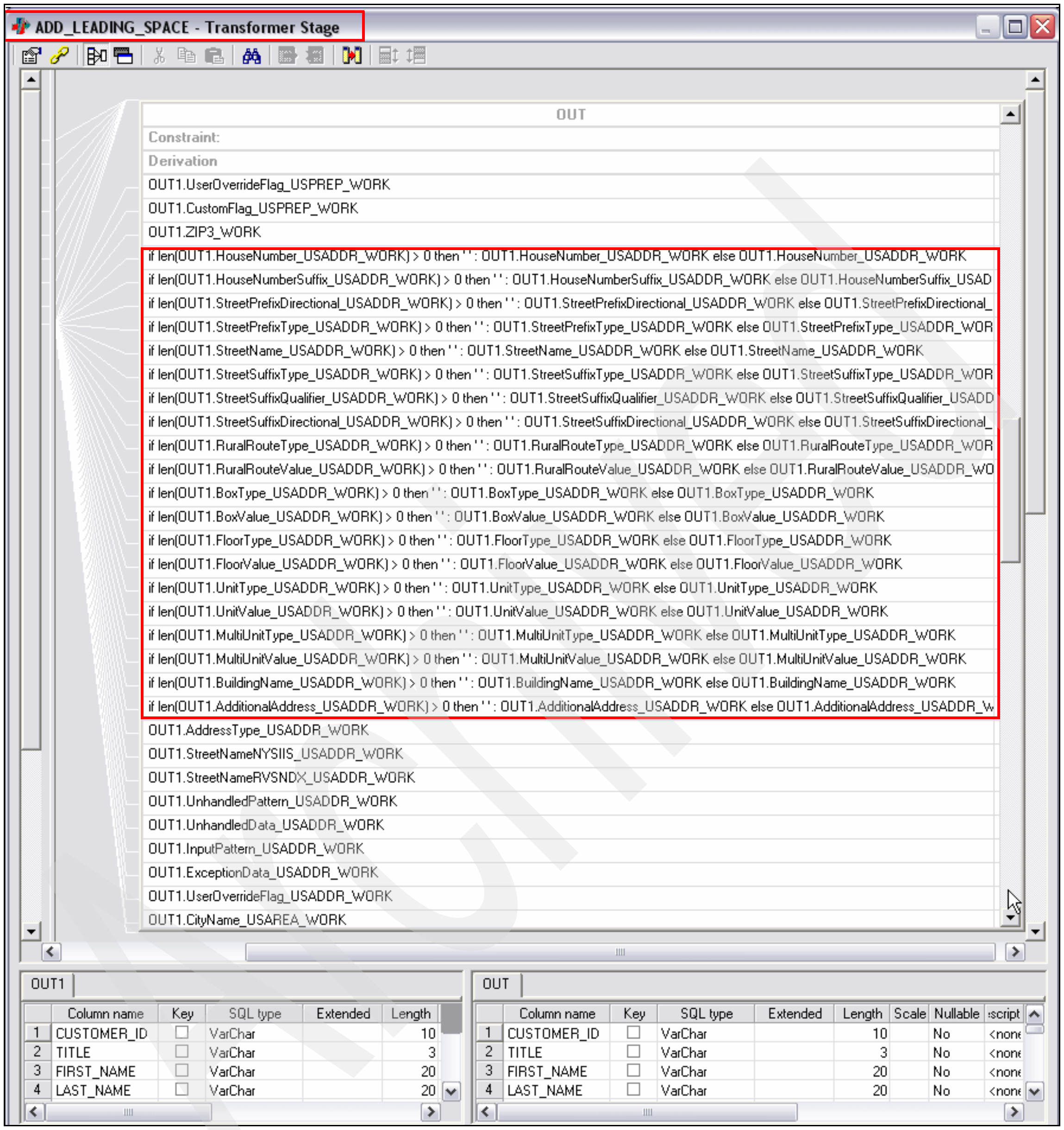

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j18_CRM_DATA_TRANSFORM *]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
FINAL_CRM_DATA
OUT1
ADD_LEADING_SPACE
OUT
PICK_AND_CONCATENATE
IN
CRM_DATA_TRANSFORMED

Figure 2-379 Create j18_CRM_DATA_TRANSFORM 1/14


ADD_LEADING_SPACE - Transformer Stage

OUT

Constraint:

Derivation

```
OUT1.UserOverrideFlag_USPREP_WORK
OUT1.CustomFlag_USPREP_WORK
OUT1.ZIP3_WORK
if len(OUT1.HouseNumber_USADDR_WORK) > 0 then ' ' : OUT1.HouseNumber_USADDR_WORK else OUT1.HouseNumber_USADDR_WORK
if len(OUT1.HouseNumberSuffix_USADDR_WORK) > 0 then ' ' : OUT1.HouseNumberSuffix_USADDR_WORK else OUT1.HouseNumberSuffix_USAD
if len(OUT1.StreetPrefixDirectional_USADDR_WORK) > 0 then ' ' : OUT1.StreetPrefixDirectional_USADDR_WORK else OUT1.StreetPrefixDirectional_
if len(OUT1.StreetPrefixType_USADDR_WORK) > 0 then ' ' : OUT1.StreetPrefixType_USADDR_WORK else OUT1.StreetPrefixType_USADDR_WOR
if len(OUT1.StreetName_USADDR_WORK) > 0 then ' ' : OUT1.StreetName_USADDR_WORK else OUT1.StreetName_USADDR_WORK
if len(OUT1.StreetSuffixType_USADDR_WORK) > 0 then ' ' : OUT1.StreetSuffixType_USADDR_WORK else OUT1.StreetSuffixType_USADDR_WOR
if len(OUT1.StreetSuffixQualifier_USADDR_WORK) > 0 then ' ' : OUT1.StreetSuffixQualifier_USADDR_WORK else OUT1.StreetSuffixQualifier_USADD
if len(OUT1.StreetSuffixDirectional_USADDR_WORK) > 0 then ' ' : OUT1.StreetSuffixDirectional_USADDR_WORK else OUT1.StreetSuffixDirectional_
if len(OUT1.RuralRouteType_USADDR_WORK) > 0 then ' ' : OUT1.RuralRouteType_USADDR_WORK else OUT1.RuralRouteType_USADDR_WOR
if len(OUT1.RuralRouteValue_USADDR_WORK) > 0 then ' ' : OUT1.RuralRouteValue_USADDR_WORK else OUT1.RuralRouteValue_USADDR_WO
if len(OUT1.BoxType_USADDR_WORK) > 0 then ' ' : OUT1.BoxType_USADDR_WORK else OUT1.BoxType_USADDR_WORK
if len(OUT1.BoxValue_USADDR_WORK) > 0 then ' ' : OUT1.BoxValue_USADDR_WORK else OUT1.BoxValue_USADDR_WORK
if len(OUT1.FloorType_USADDR_WORK) > 0 then ' ' : OUT1.FloorType_USADDR_WORK else OUT1.FloorType_USADDR_WORK
if len(OUT1.FloorValue_USADDR_WORK) > 0 then ' ' : OUT1.FloorValue_USADDR_WORK else OUT1.FloorValue_USADDR_WORK
if len(OUT1.UnitType_USADDR_WORK) > 0 then ' ' : OUT1.UnitType_USADDR_WORK else OUT1.UnitType_USADDR_WORK
if len(OUT1.UnitValue_USADDR_WORK) > 0 then ' ' : OUT1.UnitValue_USADDR_WORK else OUT1.UnitValue_USADDR_WORK
if len(OUT1.MultiUnitType_USADDR_WORK) > 0 then ' ' : OUT1.MultiUnitType_USADDR_WORK else OUT1.MultiUnitType_USADDR_WORK
if len(OUT1.MultiUnitValue_USADDR_WORK) > 0 then ' ' : OUT1.MultiUnitValue_USADDR_WORK else OUT1.MultiUnitValue_USADDR_WORK
if len(OUT1.BuildingName_USADDR_WORK) > 0 then ' ' : OUT1.BuildingName_USADDR_WORK else OUT1.BuildingName_USADDR_WORK
if len(OUT1.AdditionalAddress_USADDR_WORK) > 0 then ' ' : OUT1.AdditionalAddress_USADDR_WORK else OUT1.AdditionalAddress_USADDR_W
OUT1.AddressType_USADDR_WORK
OUT1.StreetNameNYSIIS_USADDR_WORK
OUT1.StreetNameRVSNDX_USADDR_WORK
OUT1.UnhandledPattern_USADDR_WORK
OUT1.UnhandledData_USADDR_WORK
OUT1.InputPattern_USADDR_WORK
OUT1.ExceptionData_USADDR_WORK
OUT1.UserOverrideFlag_USADDR_WORK
OUT1.CityName_USAREA_WORK
```

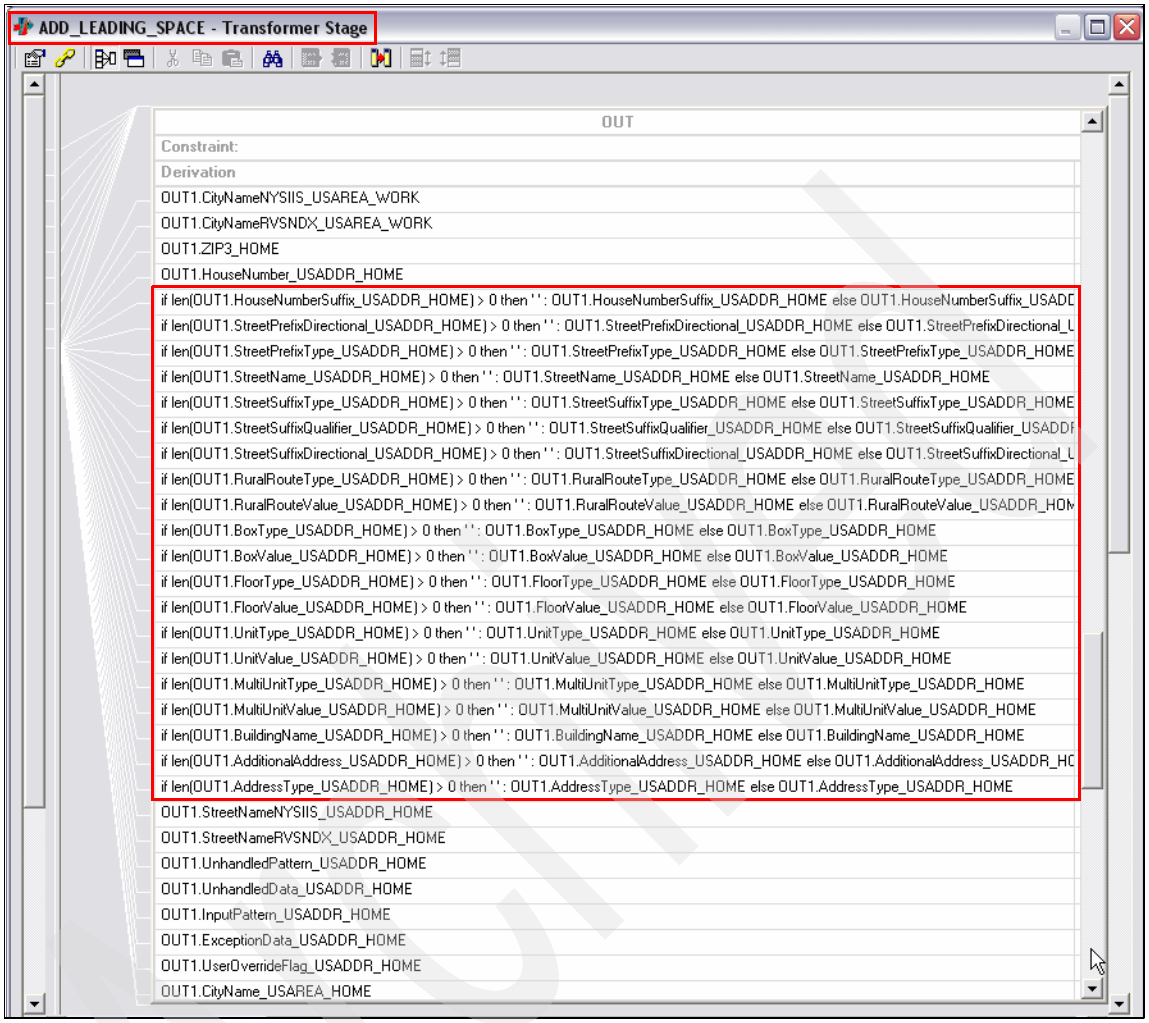
OUT1

	Column name	Key	SQL type	Extended	Length
1	CUSTOMER_ID	☐	VarChar		10
2	TITLE	☐	VarChar		3
3	FIRST_NAME	☐	VarChar		20
4	LAST_NAME	☐	VarChar		20

OUT

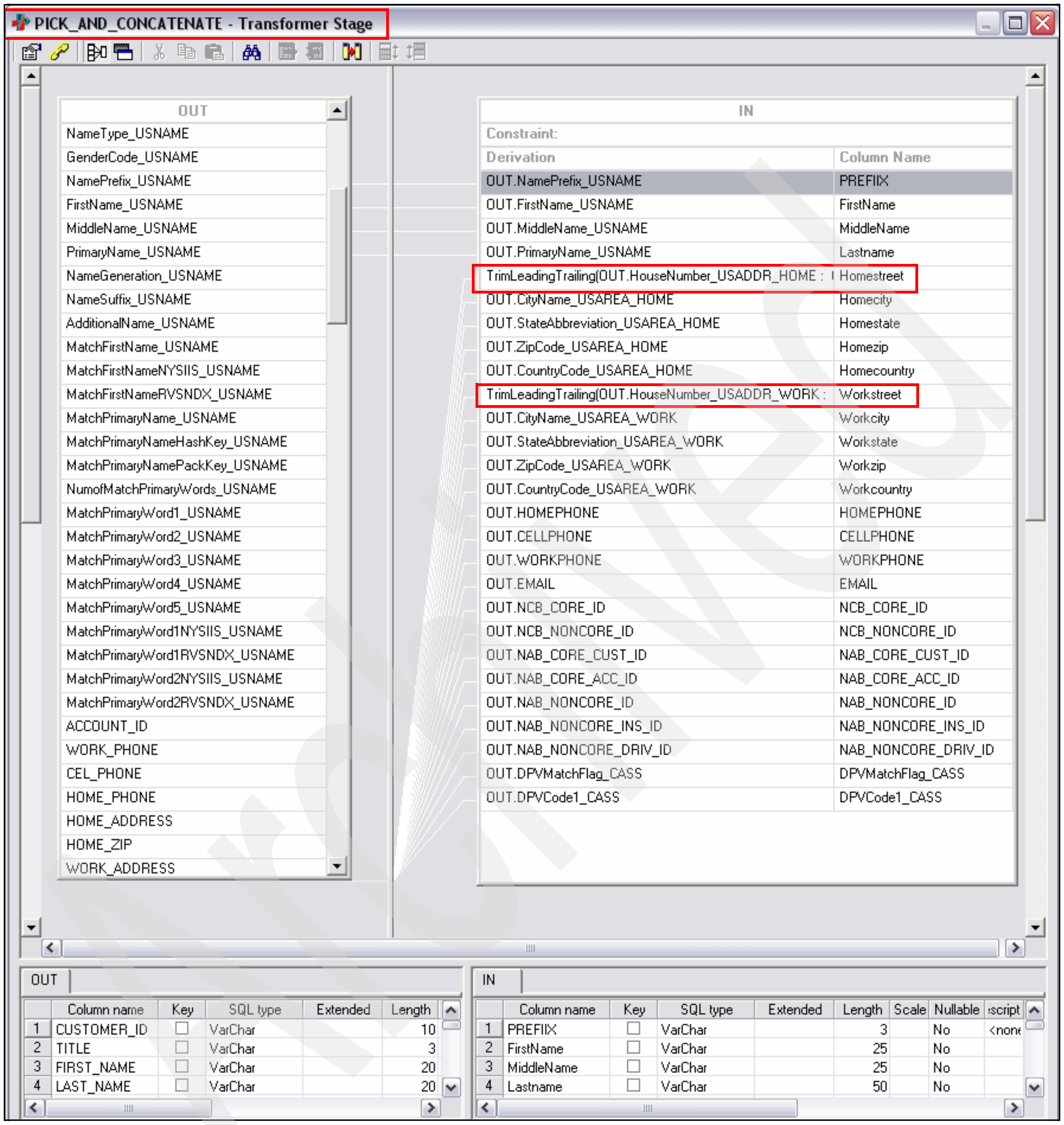
	Column name	Key	SQL type	Extended	Length	Scale	Nullable	script
1	CUSTOMER_ID	☐	VarChar		10		No	<none
2	TITLE	☐	VarChar		3		No	<none
3	FIRST_NAME	☐	VarChar		20		No	<none
4	LAST_NAME	☐	VarChar		20		No	<none

Figure 2-380 Create j18_CRM_DATA_TRANSFORM 2/14


ADD_LEADING_SPACE - Transformer Stage

```
OUT
Constraint:
Derivation
OUT1.CityNameNYSIIS_USAREA_WORK
OUT1.CityNameRVSNDX_USAREA_WORK
OUT1.ZIP3_HOME
OUT1.HouseNumber_USADDR_HOME
if len(OUT1.HouseNumberSuffix_USADDR_HOME) > 0 then ' ' : OUT1.HouseNumberSuffix_USADDR_HOME else OUT1.HouseNumberSuffix_USADE
if len(OUT1.StreetPrefixDirectional_USADDR_HOME) > 0 then ' ' : OUT1.StreetPrefixDirectional_USADDR_HOME else OUT1.StreetPrefixDirectional_L
if len(OUT1.StreetPrefixType_USADDR_HOME) > 0 then ' ' : OUT1.StreetPrefixType_USADDR_HOME else OUT1.StreetPrefixType_USADDR_HOME
if len(OUT1.StreetName_USADDR_HOME) > 0 then ' ' : OUT1.StreetName_USADDR_HOME else OUT1.StreetName_USADDR_HOME
if len(OUT1.StreetSuffixType_USADDR_HOME) > 0 then ' ' : OUT1.StreetSuffixType_USADDR_HOME else OUT1.StreetSuffixType_USADDR_HOME
if len(OUT1.StreetSuffixQualifier_USADDR_HOME) > 0 then ' ' : OUT1.StreetSuffixQualifier_USADDR_HOME else OUT1.StreetSuffixQualifier_USADDI
if len(OUT1.StreetSuffixDirectional_USADDR_HOME) > 0 then ' ' : OUT1.StreetSuffixDirectional_USADDR_HOME else OUT1.StreetSuffixDirectional_L
if len(OUT1.RuralRouteType_USADDR_HOME) > 0 then ' ' : OUT1.RuralRouteType_USADDR_HOME else OUT1.RuralRouteType_USADDR_HOME
if len(OUT1.RuralRouteValue_USADDR_HOME) > 0 then ' ' : OUT1.RuralRouteValue_USADDR_HOME else OUT1.RuralRouteValue_USADDR_HOM
if len(OUT1.BoxType_USADDR_HOME) > 0 then ' ' : OUT1.BoxType_USADDR_HOME else OUT1.BoxType_USADDR_HOME
if len(OUT1.BoxValue_USADDR_HOME) > 0 then ' ' : OUT1.BoxValue_USADDR_HOME else OUT1.BoxValue_USADDR_HOME
if len(OUT1.FloorType_USADDR_HOME) > 0 then ' ' : OUT1.FloorType_USADDR_HOME else OUT1.FloorType_USADDR_HOME
if len(OUT1.FloorValue_USADDR_HOME) > 0 then ' ' : OUT1.FloorValue_USADDR_HOME else OUT1.FloorValue_USADDR_HOME
if len(OUT1.UnitType_USADDR_HOME) > 0 then ' ' : OUT1.UnitType_USADDR_HOME else OUT1.UnitType_USADDR_HOME
if len(OUT1.UnitValue_USADDR_HOME) > 0 then ' ' : OUT1.UnitValue_USADDR_HOME else OUT1.UnitValue_USADDR_HOME
if len(OUT1.MultiUnitType_USADDR_HOME) > 0 then ' ' : OUT1.MultiUnitType_USADDR_HOME else OUT1.MultiUnitType_USADDR_HOME
if len(OUT1.MultiUnitValue_USADDR_HOME) > 0 then ' ' : OUT1.MultiUnitValue_USADDR_HOME else OUT1.MultiUnitValue_USADDR_HOME
if len(OUT1.BuildingName_USADDR_HOME) > 0 then ' ' : OUT1.BuildingName_USADDR_HOME else OUT1.BuildingName_USADDR_HOME
if len(OUT1.AdditionalAddress_USADDR_HOME) > 0 then ' ' : OUT1.AdditionalAddress_USADDR_HOME else OUT1.AdditionalAddress_USADDR_HC
if len(OUT1.AddressType_USADDR_HOME) > 0 then ' ' : OUT1.AddressType_USADDR_HOME else OUT1.AddressType_USADDR_HOME
OUT1.StreetNameNYSIIS_USADDR_HOME
OUT1.StreetNameRVSNDX_USADDR_HOME
OUT1.UnhandledPattern_USADDR_HOME
OUT1.UnhandledData_USADDR_HOME
OUT1.InputPattern_USADDR_HOME
OUT1.ExceptionData_USADDR_HOME
OUT1.UserOverrideFlag_USADDR_HOME
OUT1.CityName_USAREA_HOME
```

Figure 2-381 Create j18_CRM_DATA_TRANSFORM 3/14


PICK_AND_CONCATENATE - Transformer Stage

OUT
NameType_USNAME
GenderCode_USNAME
NamePrefix_USNAME
FirstName_USNAME
MiddleName_USNAME
PrimaryName_USNAME
NameGeneration_USNAME
NameSuffix_USNAME
AdditionalName_USNAME
MatchFirstName_USNAME
MatchFirstNameNYSIIS_USNAME
MatchFirstNameRVSNDX_USNAME
MatchPrimaryName_USNAME
MatchPrimaryNameHashKey_USNAME
MatchPrimaryNamePackKey_USNAME
NumofMatchPrimaryWords_USNAME
MatchPrimaryWord1_USNAME
MatchPrimaryWord2_USNAME
MatchPrimaryWord3_USNAME
MatchPrimaryWord4_USNAME
MatchPrimaryWord5_USNAME
MatchPrimaryWord1NYSIIS_USNAME
MatchPrimaryWord1RVSNDX_USNAME
MatchPrimaryWord2NYSIIS_USNAME
MatchPrimaryWord2RVSNDX_USNAME
ACCOUNT_ID
WORK_PHONE
CEL_PHONE
HOME_PHONE
HOME_ADDRESS
HOME_ZIP
WORK_ADDRESS

IN

Constraint:

Derivation	Column Name
OUT.NamePrefix_USNAME	PREFIIX
OUT.FirstName_USNAME	FirstName
OUT.MiddleName_USNAME	MiddleName
OUT.PrimaryName_USNAME	Lastname
TrimLeadingTrailing(OUT.HouseNumber_USADDR_HOME :	Homestreet
OUT.CityName_USAREA_HOME	Homecity
OUT.StateAbbreviation_USAREA_HOME	Homestate
OUT.ZipCode_USAREA_HOME	Homezip
OUT.CountryCode_USAREA_HOME	Homecountry
TrimLeadingTrailing(OUT.HouseNumber_USADDR_WORK :	Workstreet
OUT.CityName_USAREA_WORK	Workcity
OUT.StateAbbreviation_USAREA_WORK	Workstate
OUT.ZipCode_USAREA_WORK	Workzip
OUT.CountryCode_USAREA_WORK	Workcountry
OUT.HOMEPHONE	HOMEPHONE
OUT.CELLPHONE	CELLPHONE
OUT.WORKPHONE	WORKPHONE
OUT.EMAIL	EMAIL
OUT.NCB_CORE_ID	NCB_CORE_ID
OUT.NCB_NONCORE_ID	NCB_NONCORE_ID
OUT.NAB_CORE_CUST_ID	NAB_CORE_CUST_ID
OUT.NAB_CORE_ACC_ID	NAB_CORE_ACC_ID
OUT.NAB_NONCORE_ID	NAB_NONCORE_ID
OUT.NAB_NONCORE_INS_ID	NAB_NONCORE_INS_ID
OUT.NAB_NONCORE_DRIV_ID	NAB_NONCORE_DRIV_ID
OUT.DPVMatchFlag_CASS	DPVMatchFlag_CASS
OUT.DPVCode1_CASS	DPVCode1_CASS

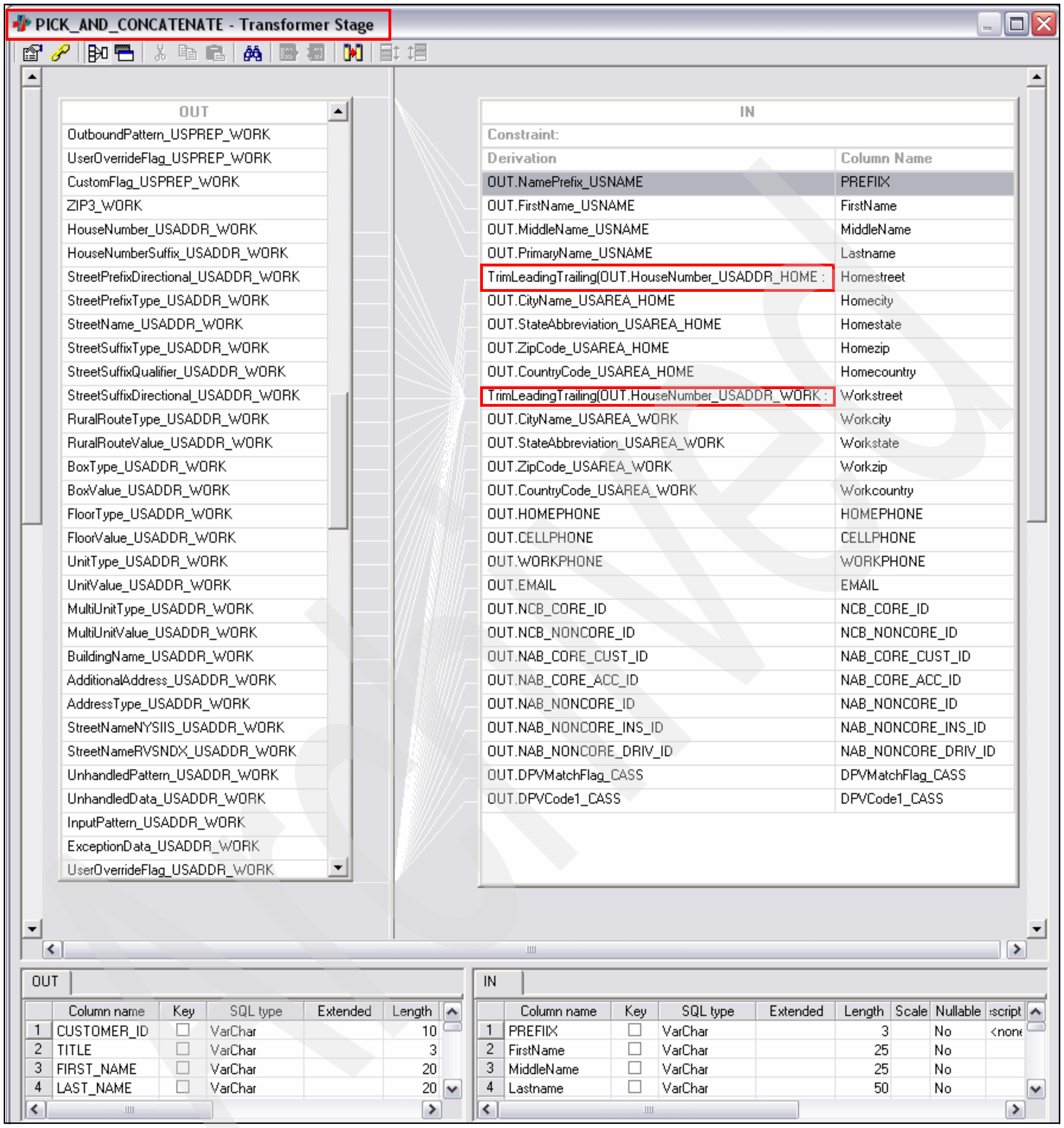
OUT

	Column name	Key	SQL type	Extended	Length
1	CUSTOMER_ID	☐	VarChar		10
2	TITLE	☐	VarChar		3
3	FIRST_NAME	☐	VarChar		20
4	LAST_NAME	☐	VarChar		20

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	:script
1	PREFIIX	☐	VarChar		3		No	<none
2	FirstName	☐	VarChar		25		No	
3	MiddleName	☐	VarChar		25		No	
4	Lastname	☐	VarChar		50		No	

Figure 2-382 Create j18_CRM_DATA_TRANSFORM 4/14



PICK_AND_CONCATENATE - Transformer Stage

OUT
OutboundPattern_USPREP_WORK
UserOverrideFlag_USPREP_WORK
CustomFlag_USPREP_WORK
ZIP3_WORK
HouseNumber_USADDR_WORK
HouseNumberSuffix_USADDR_WORK
StreetPrefixDirectional_USADDR_WORK
StreetPrefixType_USADDR_WORK
StreetName_USADDR_WORK
StreetSuffixType_USADDR_WORK
StreetSuffixQualifier_USADDR_WORK
StreetSuffixDirectional_USADDR_WORK
RuralRouteType_USADDR_WORK
RuralRouteValue_USADDR_WORK
BoxType_USADDR_WORK
BoxValue_USADDR_WORK
FloorType_USADDR_WORK
FloorValue_USADDR_WORK
UnitType_USADDR_WORK
UnitValue_USADDR_WORK
MultiUnitType_USADDR_WORK
MultiUnitValue_USADDR_WORK
BuildingName_USADDR_WORK
AdditionalAddress_USADDR_WORK
AddressType_USADDR_WORK
StreetNameNYSIIS_USADDR_WORK
StreetNameRVSNDX_USADDR_WORK
UnhandledPattern_USADDR_WORK
UnhandledData_USADDR_WORK
InputPattern_USADDR_WORK
ExceptionData_USADDR_WORK
UserOverrideFlag_USADDR_WORK

IN

Constraint:

Derivation	Column Name
OUT.NamePrefix_USNAME	PREFIX
OUT.FirstName_USNAME	FirstName
OUT.MiddleName_USNAME	MiddleName
OUT.PrimaryName_USNAME	Lastname
TrimLeadingTrailing(OUT.HouseNumber_USADDR_HOME :	Homestreet
OUT.CityName_USAREA_HOME	Homecity
OUT.StateAbbreviation_USAREA_HOME	Homestate
OUT.ZipCode_USAREA_HOME	Homezip
OUT.CountryCode_USAREA_HOME	Homecountry
TrimLeadingTrailing(OUT.HouseNumber_USADDR_WORK :	Workstreet
OUT.CityName_USAREA_WORK	Workcity
OUT.StateAbbreviation_USAREA_WORK	Workstate
OUT.ZipCode_USAREA_WORK	Workzip
OUT.CountryCode_USAREA_WORK	Workcountry
OUT.HOMEPHONE	HOMEPHONE
OUT.CELLPHONE	CELLPHONE
OUT.WORKPHONE	WORKPHONE
OUT.EMAIL	EMAIL
OUT.NCB_CORE_ID	NCB_CORE_ID
OUT.NCB_NONCORE_ID	NCB_NONCORE_ID
OUT.NAB_CORE_CUST_ID	NAB_CORE_CUST_ID
OUT.NAB_CORE_ACC_ID	NAB_CORE_ACC_ID
OUT.NAB_NONCORE_ID	NAB_NONCORE_ID
OUT.NAB_NONCORE_INS_ID	NAB_NONCORE_INS_ID
OUT.NAB_NONCORE_DRIV_ID	NAB_NONCORE_DRIV_ID
OUT.DPVMatchFlag_CASS	DPVMatchFlag_CASS
OUT.DPVCode1_CASS	DPVCode1_CASS

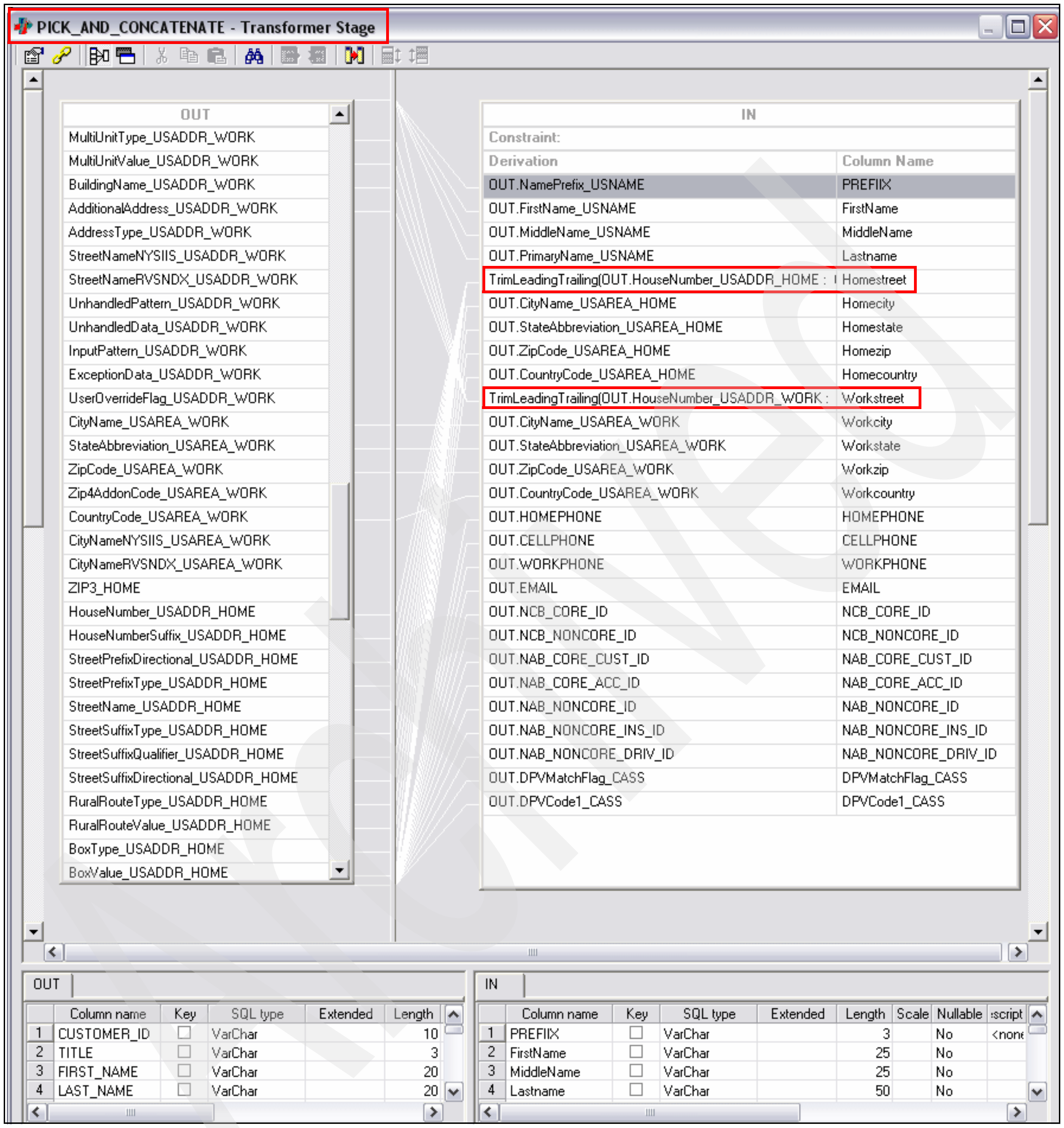
OUT

	Column name	Key	SQL type	Extended	Length
1	CUSTOMER_ID	☐	VarChar		10
2	TITLE	☐	VarChar		3
3	FIRST_NAME	☐	VarChar		20
4	LAST_NAME	☐	VarChar		20

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	:script
1	PREFIX	☐	VarChar		3		No	<none
2	FirstName	☐	VarChar		25		No	
3	MiddleName	☐	VarChar		25		No	
4	Lastname	☐	VarChar		50		No	

Figure 2-383 Create j18_CRM_DATA_TRANSFORM 5/14



PICK_AND_CONCATENATE - Transformer Stage

OUT
MultiUnitType_USADDR_WORK
MultiUnitValue_USADDR_WORK
BuildingName_USADDR_WORK
AdditionalAddress_USADDR_WORK
AddressType_USADDR_WORK
StreetNameNYSIIS_USADDR_WORK
StreetNameRVSNDX_USADDR_WORK
UnhandledPattern_USADDR_WORK
UnhandledData_USADDR_WORK
InputPattern_USADDR_WORK
ExceptionData_USADDR_WORK
UserOverrideFlag_USADDR_WORK
CityName_USAREA_WORK
StateAbbreviation_USAREA_WORK
ZipCode_USAREA_WORK
Zip4AddonCode_USAREA_WORK
CountryCode_USAREA_WORK
CityNameNYSIIS_USAREA_WORK
CityNameRVSNDX_USAREA_WORK
ZIP3_HOME
HouseNumber_USADDR_HOME
HouseNumberSuffix_USADDR_HOME
StreetPrefixDirectional_USADDR_HOME
StreetPrefixType_USADDR_HOME
StreetName_USADDR_HOME
StreetSuffixType_USADDR_HOME
StreetSuffixQualifier_USADDR_HOME
StreetSuffixDirectional_USADDR_HOME
RuralRouteType_USADDR_HOME
RuralRouteValue_USADDR_HOME
BoxType_USADDR_HOME
BoxValue_USADDR_HOME

IN

Constraint:

Derivation	Column Name
OUT.NamePrefix_USNAME	PREFIX
OUT.FirstName_USNAME	FirstName
OUT.MiddleName_USNAME	MiddleName
OUT.PrimaryName_USNAME	Lastname
TrimLeadingTrailing(OUT.HouseNumber_USADDR_HOME : (	Homestreet
OUT.CityName_USAREA_HOME	Homecity
OUT.StateAbbreviation_USAREA_HOME	Homestate
OUT.ZipCode_USAREA_HOME	Homezip
OUT.CountryCode_USAREA_HOME	Homecountry
TrimLeadingTrailing(OUT.HouseNumber_USADDR_WORK :	Workstreet
OUT.CityName_USAREA_WORK	Workcity
OUT.StateAbbreviation_USAREA_WORK	Workstate
OUT.ZipCode_USAREA_WORK	Workzip
OUT.CountryCode_USAREA_WORK	Workcountry
OUT.HOMEPHONE	HOMEPHONE
OUT.CELLPHONE	CELLPHONE
OUT.WORKPHONE	WORKPHONE
OUT.EMAIL	EMAIL
OUT.NCB_CORE_ID	NCB_CORE_ID
OUT.NCB_NONCORE_ID	NCB_NONCORE_ID
OUT.NAB_CORE_CUST_ID	NAB_CORE_CUST_ID
OUT.NAB_CORE_ACC_ID	NAB_CORE_ACC_ID
OUT.NAB_NONCORE_ID	NAB_NONCORE_ID
OUT.NAB_NONCORE_INS_ID	NAB_NONCORE_INS_ID
OUT.NAB_NONCORE_DRIV_ID	NAB_NONCORE_DRIV_ID
OUT.DPVMatchFlag_CASS	DPVMatchFlag_CASS
OUT.DPVCode1_CASS	DPVCode1_CASS

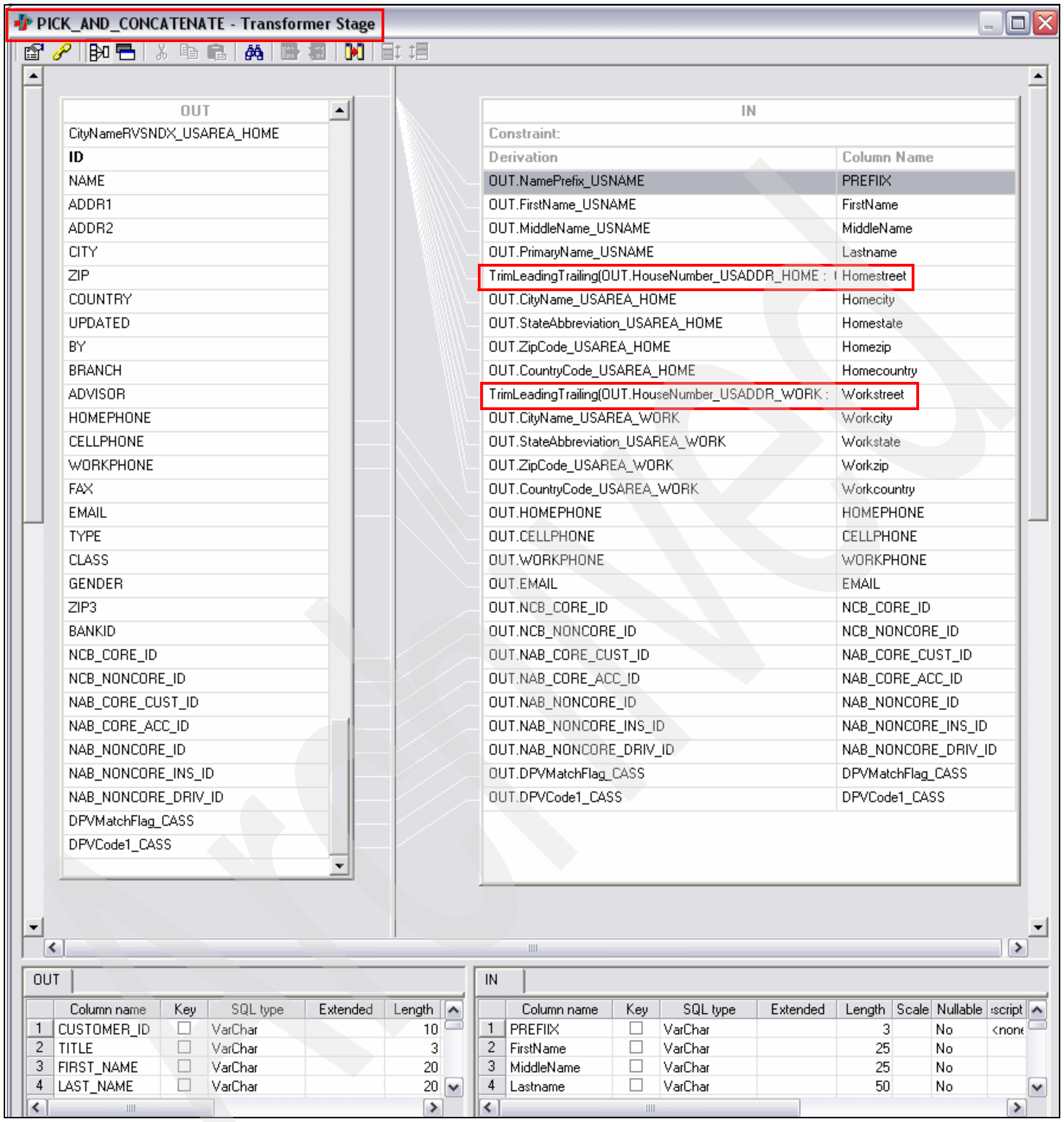
OUT

	Column name	Key	SQL type	Extended	Length
1	CUSTOMER_ID	☐	VarChar		10
2	TITLE	☐	VarChar		3
3	FIRST_NAME	☐	VarChar		20
4	LAST_NAME	☐	VarChar		20

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	:script
1	PREFIX	☐	VarChar		3		No	<none
2	FirstName	☐	VarChar		25		No	
3	MiddleName	☐	VarChar		25		No	
4	Lastname	☐	VarChar		50		No	

Figure 2-384 Create j18_CRM_DATA_TRANSFORM 6/14


PICK_AND_CONCATENATE - Transformer Stage

OUT
CityNameRVSNDX_USAREA_HOME
ID
NAME
ADDR1
ADDR2
CITY
ZIP
COUNTRY
UPDATED
BY
BRANCH
ADVISOR
HOMEPHONE
CELLPHONE
WORKPHONE
FAX
EMAIL
TYPE
CLASS
GENDER
ZIP3
BANKID
NCB_CORE_ID
NCB_NONCORE_ID
NAB_CORE_CUST_ID
NAB_CORE_ACC_ID
NAB_NONCORE_ID
NAB_NONCORE_INS_ID
NAB_NONCORE_DRIV_ID
DPVMatchFlag_CASS
DPVCode1_CASS

IN

Constraint:

Derivation	Column Name
OUT.NamePrefix_USNAME	PREFIX
OUT.FirstName_USNAME	FirstName
OUT.MiddleName_USNAME	MiddleName
OUT.PrimaryName_USNAME	Lastname
TrimLeadingTrailing(OUT.HouseNumber_USADDR_HOME :	Homestreet
OUT.CityName_USAREA_HOME	Homecity
OUT.StateAbbreviation_USAREA_HOME	Homestate
OUT.ZipCode_USAREA_HOME	Homezip
OUT.CountryCode_USAREA_HOME	Homecountry
TrimLeadingTrailing(OUT.HouseNumber_USADDR_WORK :	Workstreet
OUT.CityName_USAREA_WORK	Workcity
OUT.StateAbbreviation_USAREA_WORK	Workstate
OUT.ZipCode_USAREA_WORK	Workzip
OUT.CountryCode_USAREA_WORK	Workcountry
OUT.HOMEPHONE	HOMEPHONE
OUT.CELLPHONE	CELLPHONE
OUT.WORKPHONE	WORKPHONE
OUT.EMAIL	EMAIL
OUT.NCB_CORE_ID	NCB_CORE_ID
OUT.NCB_NONCORE_ID	NCB_NONCORE_ID
OUT.NAB_CORE_CUST_ID	NAB_CORE_CUST_ID
OUT.NAB_CORE_ACC_ID	NAB_CORE_ACC_ID
OUT.NAB_NONCORE_ID	NAB_NONCORE_ID
OUT.NAB_NONCORE_INS_ID	NAB_NONCORE_INS_ID
OUT.NAB_NONCORE_DRIV_ID	NAB_NONCORE_DRIV_ID
OUT.DPVMatchFlag_CASS	DPVMatchFlag_CASS
OUT.DPVCode1_CASS	DPVCode1_CASS

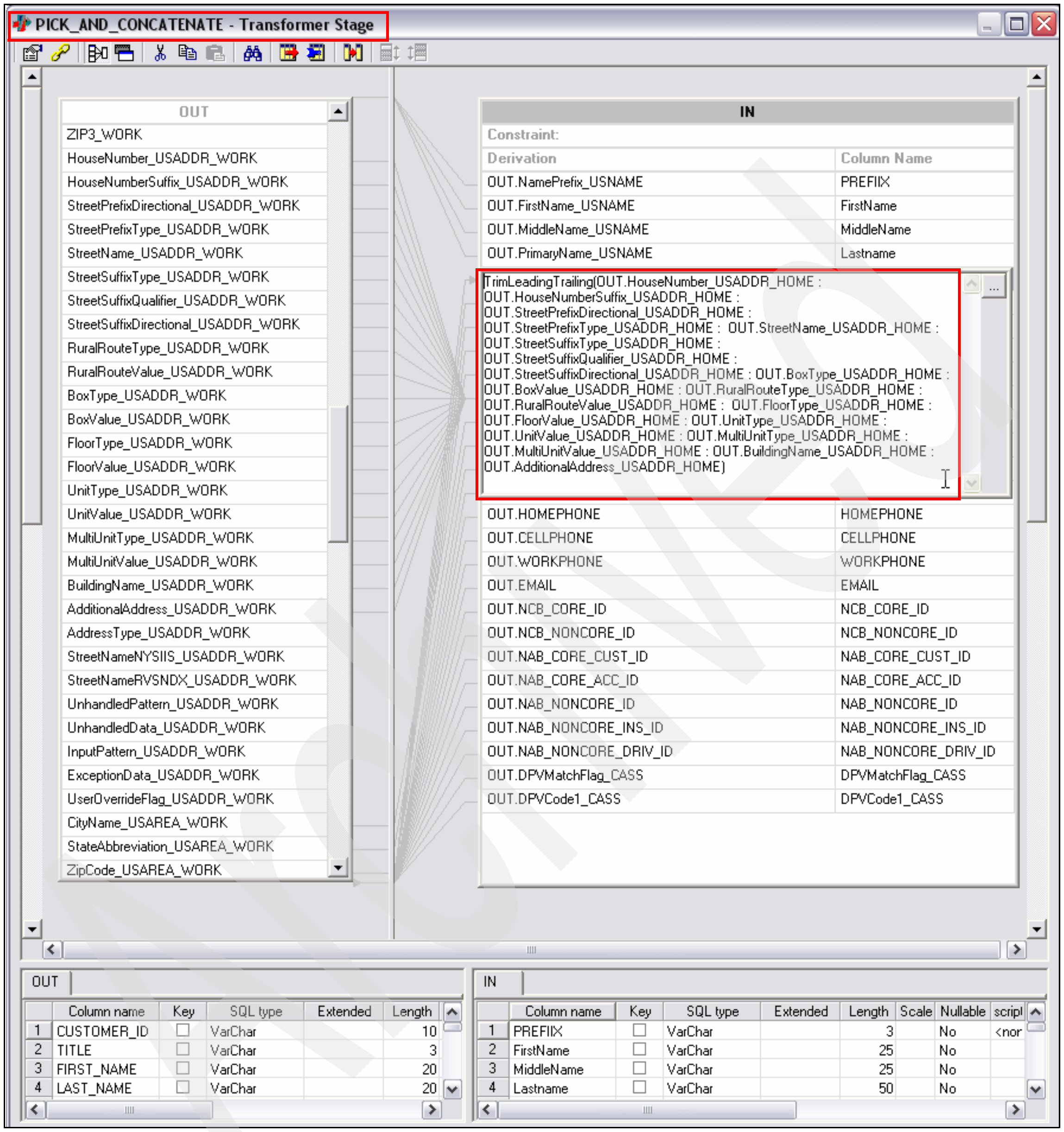
OUT

	Column name	Key	SQL type	Extended	Length
1	CUSTOMER_ID	☐	VarChar		10
2	TITLE	☐	VarChar		3
3	FIRST_NAME	☐	VarChar		20
4	LAST_NAME	☐	VarChar		20

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	script
1	PREFIX	☐	VarChar		3		No	<none
2	FirstName	☐	VarChar		25		No	
3	MiddleName	☐	VarChar		25		No	
4	Lastname	☐	VarChar		50		No	

Figure 2-385 Create j18_CRM_DATA_TRANSFORM 7/14



PICK_AND_CONCATENATE - Transformer Stage

OUT
ZIP3_WORK
HouseNumber_USADDR_WORK
HouseNumberSuffix_USADDR_WORK
StreetPrefixDirectional_USADDR_WORK
StreetPrefixType_USADDR_WORK
StreetName_USADDR_WORK
StreetSuffixType_USADDR_WORK
StreetSuffixQualifier_USADDR_WORK
StreetSuffixDirectional_USADDR_WORK
RuralRouteType_USADDR_WORK
RuralRouteValue_USADDR_WORK
BoxType_USADDR_WORK
BoxValue_USADDR_WORK
FloorType_USADDR_WORK
FloorValue_USADDR_WORK
UnitType_USADDR_WORK
UnitValue_USADDR_WORK
MultiUnitType_USADDR_WORK
MultiUnitValue_USADDR_WORK
BuildingName_USADDR_WORK
AdditionalAddress_USADDR_WORK
AddressType_USADDR_WORK
StreetNameNYSIIS_USADDR_WORK
StreetNameRVSNDX_USADDR_WORK
UnhandledPattern_USADDR_WORK
UnhandledData_USADDR_WORK
InputPattern_USADDR_WORK
ExceptionData_USADDR_WORK
UserOverrideFlag_USADDR_WORK
CityName_USAREA_WORK
StateAbbreviation_USAREA_WORK
ZipCode_USAREA_WORK

IN

Constraint:

Derivation	Column Name
OUT.NamePrefix_USNAME	PREFIX
OUT.FirstName_USNAME	FirstName
OUT.MiddleName_USNAME	MiddleName
OUT.PrimaryName_USNAME	Lastname

```
TrimLeadingTrailing(OUT.HouseNumber_USADDR_HOME :
OUT.HouseNumberSuffix_USADDR_HOME :
OUT.StreetPrefixDirectional_USADDR_HOME :
OUT.StreetPrefixType_USADDR_HOME :  OUT.StreetName_USADDR_HOME :
OUT.StreetSuffixType_USADDR_HOME :
OUT.StreetSuffixQualifier_USADDR_HOME :
OUT.StreetSuffixDirectional_USADDR_HOME : OUT.BoxType_USADDR_HOME :
OUT.BoxValue_USADDR_HOME : OUT.RuralRouteType_USADDR_HOME :
OUT.RuralRouteValue_USADDR_HOME :  OUT.FloorType_USADDR_HOME :
OUT.FloorValue_USADDR_HOME : OUT.UnitType_USADDR_HOME :
OUT.UnitValue_USADDR_HOME : OUT.MultiUnitType_USADDR_HOME :
OUT.MultiUnitValue_USADDR_HOME : OUT.BuildingName_USADDR_HOME :
OUT.AdditionalAddress_USADDR_HOME)
```

Derivation	Column Name
OUT.HOMEPHONE	HOMEPHONE
OUT.CELLPHONE	CELLPHONE
OUT.WORKPHONE	WORKPHONE
OUT.EMAIL	EMAIL
OUT.NCB_CORE_ID	NCB_CORE_ID
OUT.NCB_NONCORE_ID	NCB_NONCORE_ID
OUT.NAB_CORE_CUST_ID	NAB_CORE_CUST_ID
OUT.NAB_CORE_ACC_ID	NAB_CORE_ACC_ID
OUT.NAB_NONCORE_ID	NAB_NONCORE_ID
OUT.NAB_NONCORE_INS_ID	NAB_NONCORE_INS_ID
OUT.NAB_NONCORE_DRIV_ID	NAB_NONCORE_DRIV_ID
OUT.DPVMatchFlag_CASS	DPVMatchFlag_CASS
OUT.DPVCode1_CASS	DPVCode1_CASS

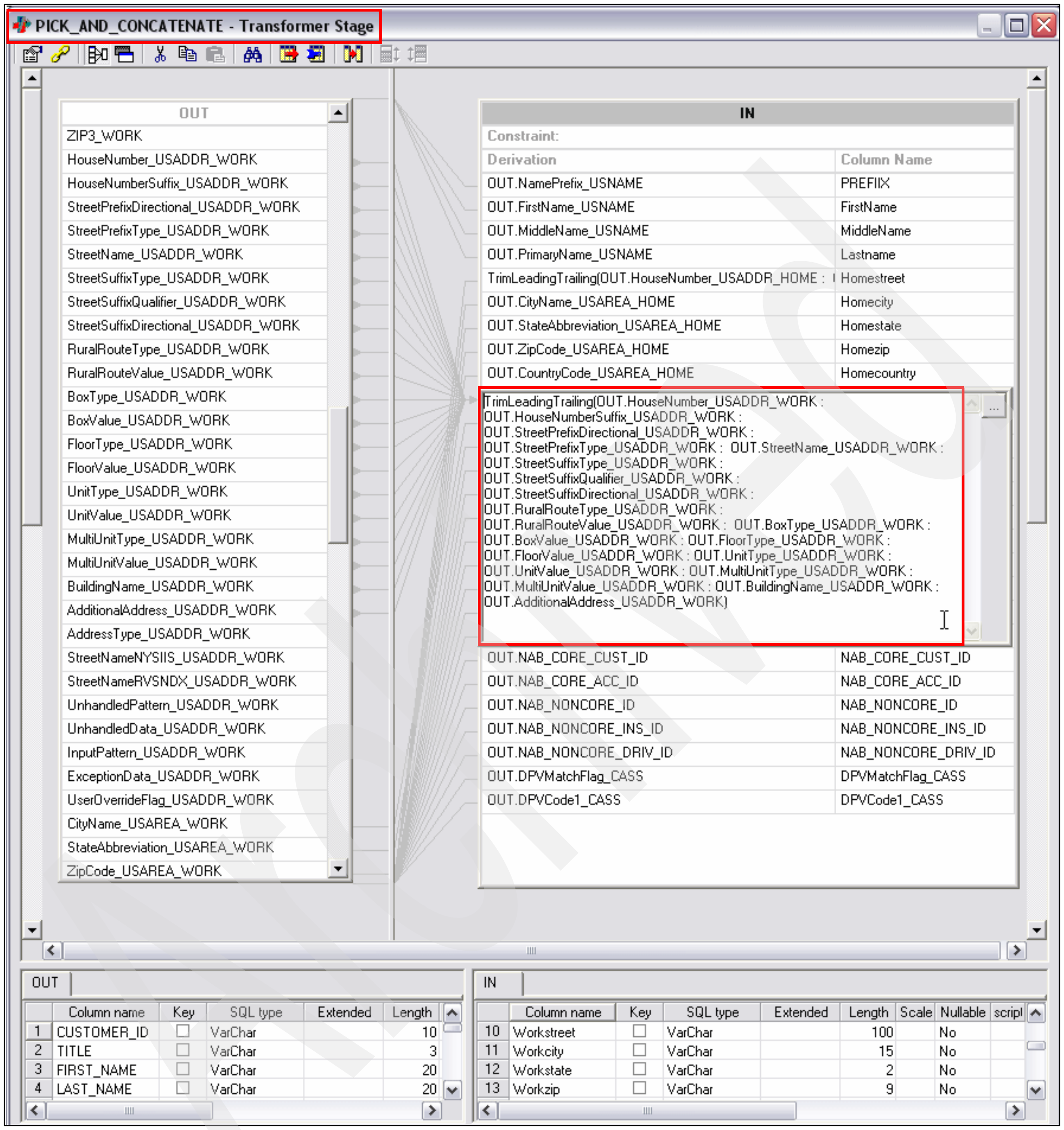
OUT

	Column name	Key	SQL type	Extended	Length
1	CUSTOMER_ID	☐	VarChar		10
2	TITLE	☐	VarChar		3
3	FIRST_NAME	☐	VarChar		20
4	LAST_NAME	☐	VarChar		20

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	scripl
1	PREFIX	☐	VarChar		3		No	<nor
2	FirstName	☐	VarChar		25		No	
3	MiddleName	☐	VarChar		25		No	
4	Lastname	☐	VarChar		50		No	

Figure 2-386 Create j18_CRM_DATA_TRANSFORM 8/14


PICK_AND_CONCATENATE - Transformer Stage

OUT
ZIP3_WORK
HouseNumber_USADDR_WORK
HouseNumberSuffix_USADDR_WORK
StreetPrefixDirectional_USADDR_WORK
StreetPrefixType_USADDR_WORK
StreetName_USADDR_WORK
StreetSuffixType_USADDR_WORK
StreetSuffixQualifier_USADDR_WORK
StreetSuffixDirectional_USADDR_WORK
RuralRouteType_USADDR_WORK
RuralRouteValue_USADDR_WORK
BoxType_USADDR_WORK
BoxValue_USADDR_WORK
FloorType_USADDR_WORK
FloorValue_USADDR_WORK
UnitType_USADDR_WORK
UnitValue_USADDR_WORK
MultiUnitType_USADDR_WORK
MultiUnitValue_USADDR_WORK
BuildingName_USADDR_WORK
AdditionalAddress_USADDR_WORK
AddressType_USADDR_WORK
StreetNameNYSIIS_USADDR_WORK
StreetNameRVSNDX_USADDR_WORK
UnhandledPattern_USADDR_WORK
UnhandledData_USADDR_WORK
InputPattern_USADDR_WORK
ExceptionData_USADDR_WORK
UserOverrideFlag_USADDR_WORK
CityName_USAREA_WORK
StateAbbreviation_USAREA_WORK
ZipCode_USAREA_WORK

IN

Constraint:

Derivation	Column Name
OUT.NamePrefix_USNAME	PREFIX
OUT.FirstName_USNAME	FirstName
OUT.MiddleName_USNAME	MiddleName
OUT.PrimaryName_USNAME	Lastname
TrimLeadingTrailing(OUT.HouseNumber_USADDR_HOME :	Homestreet
OUT.CityName_USAREA_HOME	Homecity
OUT.StateAbbreviation_USAREA_HOME	Homestate
OUT.ZipCode_USAREA_HOME	Homezip
OUT.CountryCode_USAREA_HOME	Homecountry
TrimLeadingTrailing(OUT.HouseNumber_USADDR_WORK : OUT.HouseNumberSuffix_USADDR_WORK : OUT.StreetPrefixDirectional_USADDR_WORK : OUT.StreetPrefixType_USADDR_WORK : OUT.StreetName_USADDR_WORK : OUT.StreetSuffixType_USADDR_WORK : OUT.StreetSuffixQualifier_USADDR_WORK : OUT.StreetSuffixDirectional_USADDR_WORK : OUT.RuralRouteType_USADDR_WORK : OUT.RuralRouteValue_USADDR_WORK : OUT.BoxType_USADDR_WORK : OUT.BoxValue_USADDR_WORK : OUT.FloorType_USADDR_WORK : OUT.FloorValue_USADDR_WORK : OUT.UnitType_USADDR_WORK : OUT.UnitValue_USADDR_WORK : OUT.MultiUnitType_USADDR_WORK : OUT.MultiUnitValue_USADDR_WORK : OUT.BuildingName_USADDR_WORK : OUT.AdditionalAddress_USADDR_WORK)	
OUT.NAB_CORE_CUST_ID	NAB_CORE_CUST_ID
OUT.NAB_CORE_ACC_ID	NAB_CORE_ACC_ID
OUT.NAB_NONCORE_ID	NAB_NONCORE_ID
OUT.NAB_NONCORE_INS_ID	NAB_NONCORE_INS_ID
OUT.NAB_NONCORE_DRIV_ID	NAB_NONCORE_DRIV_ID
OUT.DPVMatchFlag_CASS	DPVMatchFlag_CASS
OUT.DPVCode1_CASS	DPVCode1_CASS

OUT

	Column name	Key	SQL type	Extended	Length
1	CUSTOMER_ID	☐	VarChar		10
2	TITLE	☐	VarChar		3
3	FIRST_NAME	☐	VarChar		20
4	LAST_NAME	☐	VarChar		20

IN

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	script
10	Workstreet	☐	VarChar		100		No	
11	Workcity	☐	VarChar		15		No	
12	Workstate	☐	VarChar		2		No	
13	Workzip	☐	VarChar		9		No	

Figure 2-387   Create j18_CRM_DATA_TRANSFORM 9/14

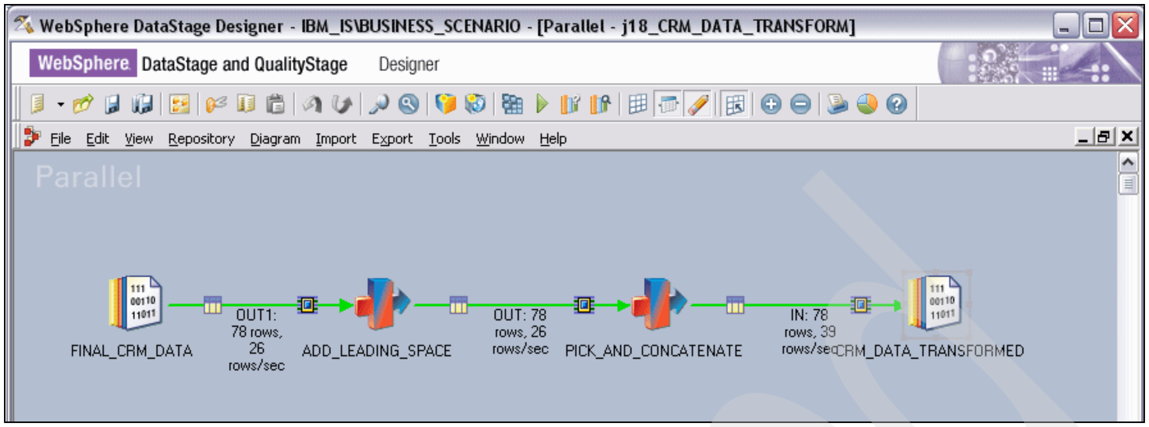
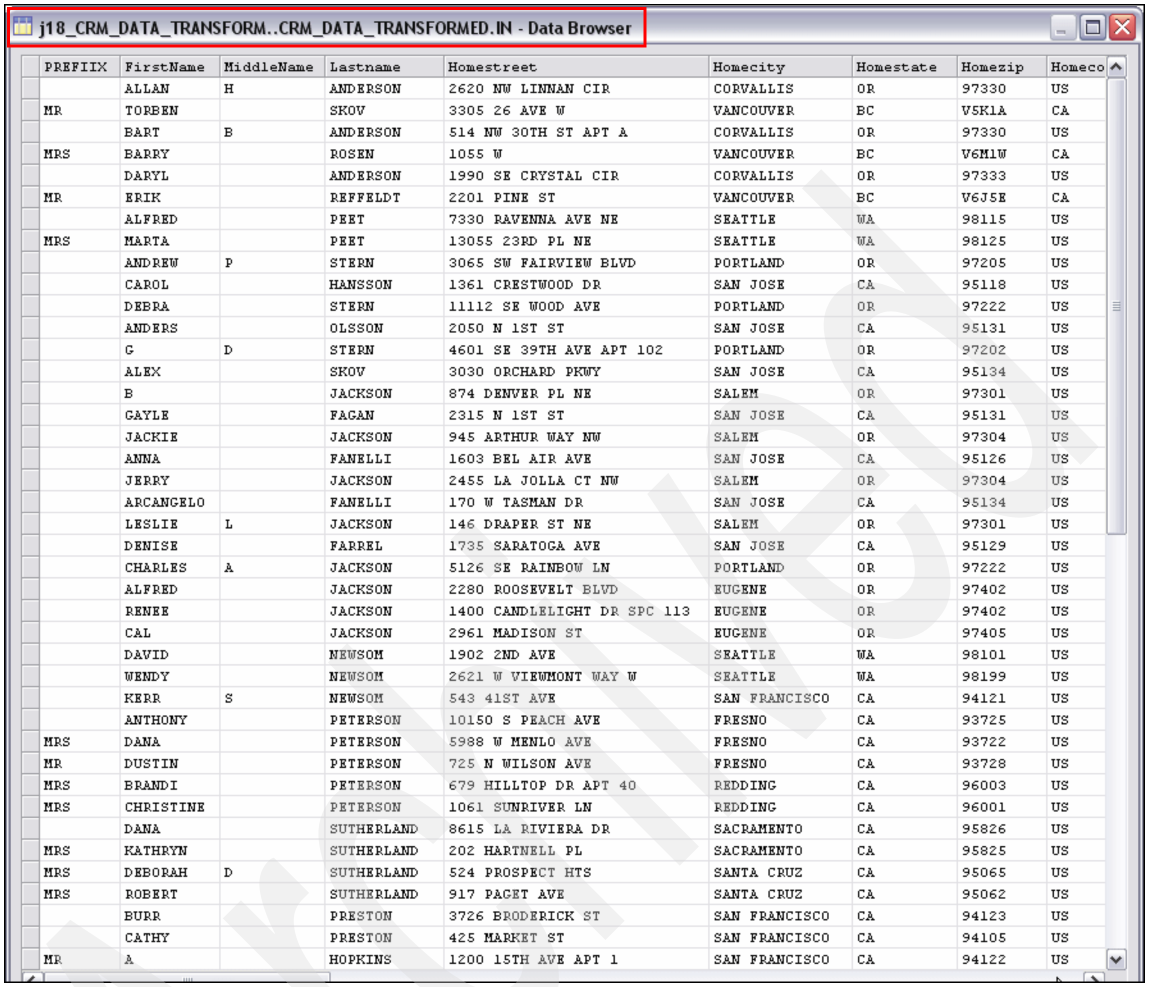
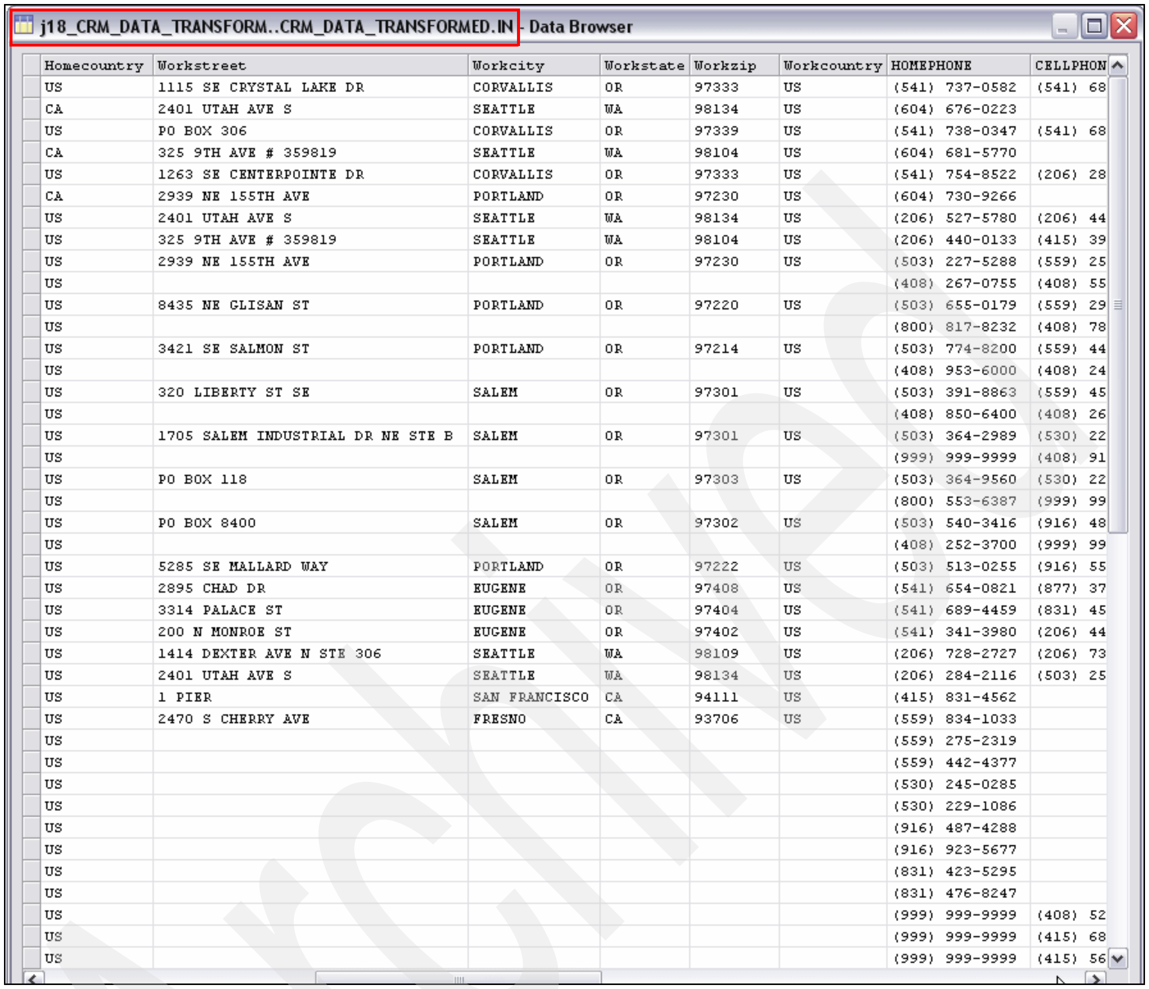

WebSphere DataStage Designer - IBM_IS\BUSINESS_SCENARIO - [Parallel - j18_CRM_DATA_TRANSFORM]
WebSphere DataStage and QualityStage Designer
File Edit View Repository Diagram Import Export Tools Window Help
Parallel
FINAL_CRM_DATA
OUT1: 78 rows, 26 rows/sec
ADD_LEADING_SPACE
OUT: 78 rows, 26 rows/sec
PICK_AND_CONCATENATE
IN: 78 rows, 39 rows/sec
CRM_DATA_TRANSFORMED

Figure 2-388 Create j18_CRM_DATA_TRANSFORM 10/14


j18_CRM_DATA_TRANSFORM..CRM_DATA_TRANSFORMED.IN - Data Browser

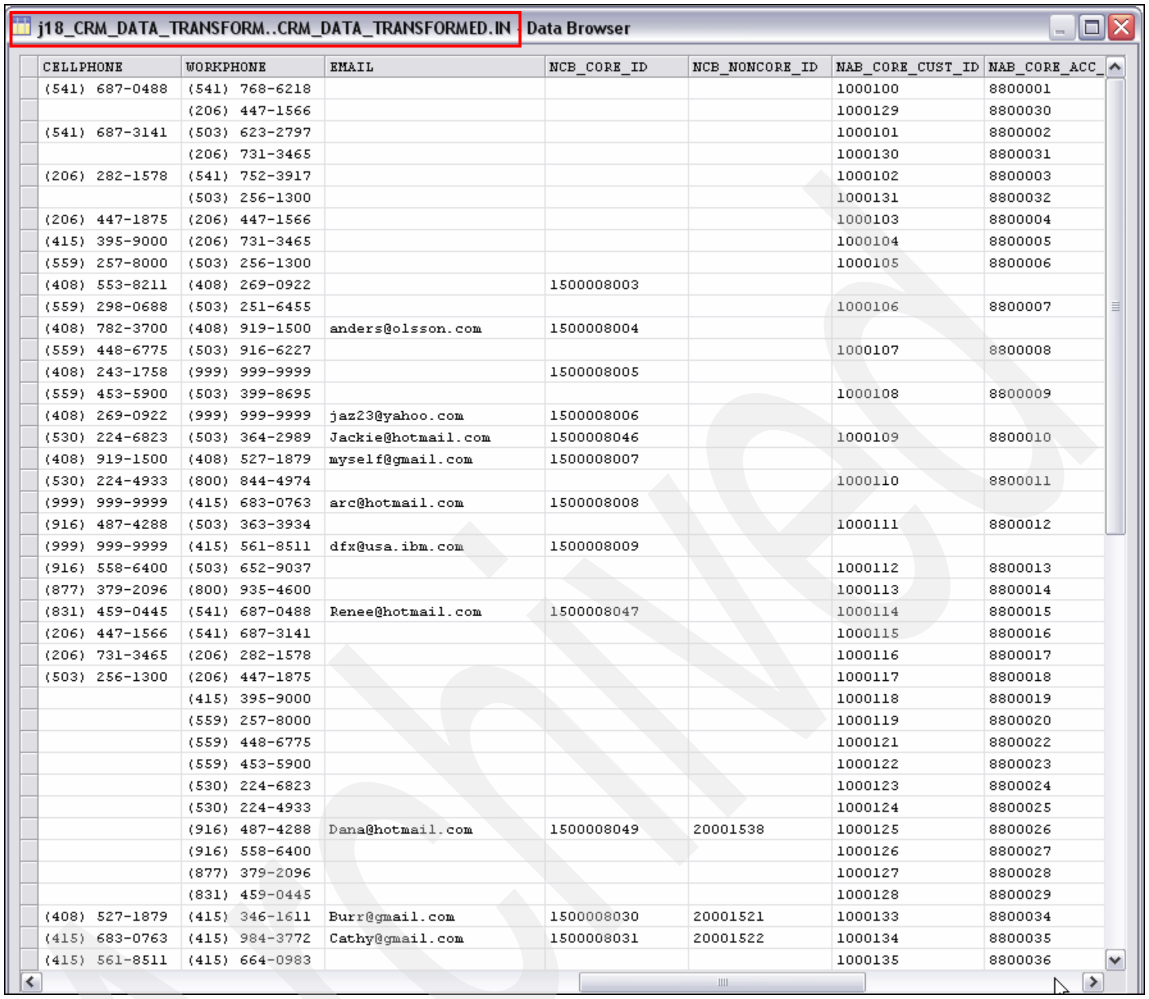
PREFIX	FirstName	MiddleName	Lastname	Homestreet	Homecity	Homestate	Homezip	Homeco
	ALLAN	H	ANDERSON	2620 NW LINNAN CIR	CORVALLIS	OR	97330	US
MR	TORBEN		SKOV	3305 26 AVE W	VANCOUVER	BC	V5K1A	CA
	BART	B	ANDERSON	514 NW 30TH ST APT A	CORVALLIS	OR	97330	US
MRS	BARRY		ROSEN	1055 W	VANCOUVER	BC	V6M1W	CA
	DARYL		ANDERSON	1990 SE CRYSTAL CIR	CORVALLIS	OR	97333	US
MR	ERIK		REFFELDT	2201 PINE ST	VANCOUVER	BC	V6J5E	CA
	ALFRED		PEET	7330 RAVENNA AVE NE	SEATTLE	WA	98115	US
MRS	MARTA		PEET	13055 23RD PL NE	SEATTLE	WA	98125	US
	ANDREW	P	STERN	3065 SW FAIRVIEW BLVD	PORTLAND	OR	97205	US
	CAROL		HANSSON	1361 CRESTWOOD DR	SAN JOSE	CA	95118	US
	DEBRA		STERN	11112 SE WOOD AVE	PORTLAND	OR	97222	US
	ANDERS		OLSSON	2050 N 1ST ST	SAN JOSE	CA	95131	US
	G	D	STERN	4601 SE 39TH AVE APT 102	PORTLAND	OR	97202	US
	ALEX		SKOV	3030 ORCHARD PKWY	SAN JOSE	CA	95134	US
	B		JACKSON	874 DENVER PL NE	SALEM	OR	97301	US
	GAYLE		FAGAN	2315 N 1ST ST	SAN JOSE	CA	95131	US
	JACKIE		JACKSON	945 ARTHUR WAY NW	SALEM	OR	97304	US
	ANNA		FANELLI	1603 BEL AIR AVE	SAN JOSE	CA	95126	US
	JERRY		JACKSON	2455 LA JOLLA CT NW	SALEM	OR	97304	US
	ARCANGELO		FANELLI	170 W TASMAN DR	SAN JOSE	CA	95134	US
	LESLIE	L	JACKSON	146 DRAPER ST NE	SALEM	OR	97301	US
	DENISE		FARREL	1735 SARATOGA AVE	SAN JOSE	CA	95129	US
	CHARLES	A	JACKSON	5126 SE RAINBOW LN	PORTLAND	OR	97222	US
	ALFRED		JACKSON	2280 ROOSEVELT BLVD	EUGENE	OR	97402	US
	RENEE		JACKSON	1400 CANDLELIGHT DR SPC 113	EUGENE	OR	97402	US
	CAL		JACKSON	2961 MADISON ST	EUGENE	OR	97405	US
	DAVID		NEWSOM	1902 2ND AVE	SEATTLE	WA	98101	US
	WENDY		NEWSOM	2621 W VIEWMONT WAY W	SEATTLE	WA	98199	US
	KERR	S	NEWSOM	543 41ST AVE	SAN FRANCISCO	CA	94121	US
	ANTHONY		PETERSON	10150 S PEACH AVE	FRESNO	CA	93725	US
MRS	DANA		PETERSON	5988 W MENLO AVE	FRESNO	CA	93722	US
MR	DUSTIN		PETERSON	725 N WILSON AVE	FRESNO	CA	93728	US
MRS	BRANDI		PETERSON	679 HILLTOP DR APT 40	REDDING	CA	96003	US
MRS	CHRISTINE		PETERSON	1061 SUNRIVER LN	REDDING	CA	96001	US
	DANA		SUTHERLAND	8615 LA RIVIERA DR	SACRAMENTO	CA	95826	US
MRS	KATHRYN		SUTHERLAND	202 HARTNELL PL	SACRAMENTO	CA	95825	US
MRS	DEBORAH	D	SUTHERLAND	524 PROSPECT HTS	SANTA CRUZ	CA	95065	US
MRS	ROBERT		SUTHERLAND	917 PAGET AVE	SANTA CRUZ	CA	95062	US
	BURR		PRESTON	3726 BRODERICK ST	SAN FRANCISCO	CA	94123	US
	CATHY		PRESTON	425 MARKET ST	SAN FRANCISCO	CA	94105	US
MR	A		HOPKINS	1200 15TH AVE APT 1	SAN FRANCISCO	CA	94122	US

Figure 2-389 Create j18_CRM_DATA_TRANSFORM 11/14


j18_CRM_DATA_TRANSFORM..CRM_DATA_TRANSFORMED.IN - Data Browser

Homecountry	Workstreet	Workcity	Workstate	Workzip	Workcountry	HOMEPHONE	CELLPHON
US	1115 SE CRYSTAL LAKE DR	CORVALLIS	OR	97333	US	(541) 737-0582	(541) 68
CA	2401 UTAH AVE S	SEATTLE	WA	98134	US	(604) 676-0223	
US	PO BOX 306	CORVALLIS	OR	97339	US	(541) 738-0347	(541) 68
CA	325 9TH AVE # 359819	SEATTLE	WA	98104	US	(604) 681-5770	
US	1263 SE CENTERPOINTE DR	CORVALLIS	OR	97333	US	(541) 754-8522	(206) 28
CA	2939 NE 155TH AVE	PORTLAND	OR	97230	US	(604) 730-9266	
US	2401 UTAH AVE S	SEATTLE	WA	98134	US	(206) 527-5780	(206) 44
US	325 9TH AVE # 359819	SEATTLE	WA	98104	US	(206) 440-0133	(415) 39
US	2939 NE 155TH AVE	PORTLAND	OR	97230	US	(503) 227-5288	(559) 25
US						(408) 267-0755	(408) 55
US	8435 NE GLISAN ST	PORTLAND	OR	97220	US	(503) 655-0179	(559) 29
US						(800) 817-8232	(408) 78
US	3421 SE SALMON ST	PORTLAND	OR	97214	US	(503) 774-8200	(559) 44
US						(408) 953-6000	(408) 24
US	320 LIBERTY ST SE	SALEM	OR	97301	US	(503) 391-8863	(559) 45
US						(408) 850-6400	(408) 26
US	1705 SALEM INDUSTRIAL DR NE STE B	SALEM	OR	97301	US	(503) 364-2989	(530) 22
US						(999) 999-9999	(408) 91
US	PO BOX 118	SALEM	OR	97303	US	(503) 364-9560	(530) 22
US						(800) 553-6387	(999) 99
US	PO BOX 8400	SALEM	OR	97302	US	(503) 540-3416	(916) 48
US						(408) 252-3700	(999) 99
US	5285 SE MALLARD WAY	PORTLAND	OR	97222	US	(503) 513-0255	(916) 55
US	2895 CHAD DR	EUGENE	OR	97408	US	(541) 654-0821	(877) 37
US	3314 PALACE ST	EUGENE	OR	97404	US	(541) 689-4459	(831) 45
US	200 N MONROE ST	EUGENE	OR	97402	US	(541) 341-3980	(206) 44
US	1414 DEXTER AVE N STE 306	SEATTLE	WA	98109	US	(206) 728-2727	(206) 73
US	2401 UTAH AVE S	SEATTLE	WA	98134	US	(206) 284-2116	(503) 25
US	1 PIER	SAN FRANCISCO	CA	94111	US	(415) 831-4562	
US	2470 S CHERRY AVE	FRESNO	CA	93706	US	(559) 834-1033	
US						(559) 275-2319	
US						(559) 442-4377	
US						(530) 245-0285	
US						(530) 229-1086	
US						(916) 487-4288	
US						(916) 923-5677	
US						(831) 423-5295	
US						(831) 476-8247	
US						(999) 999-9999	(408) 52
US						(999) 999-9999	(415) 68
US						(999) 999-9999	(415) 56

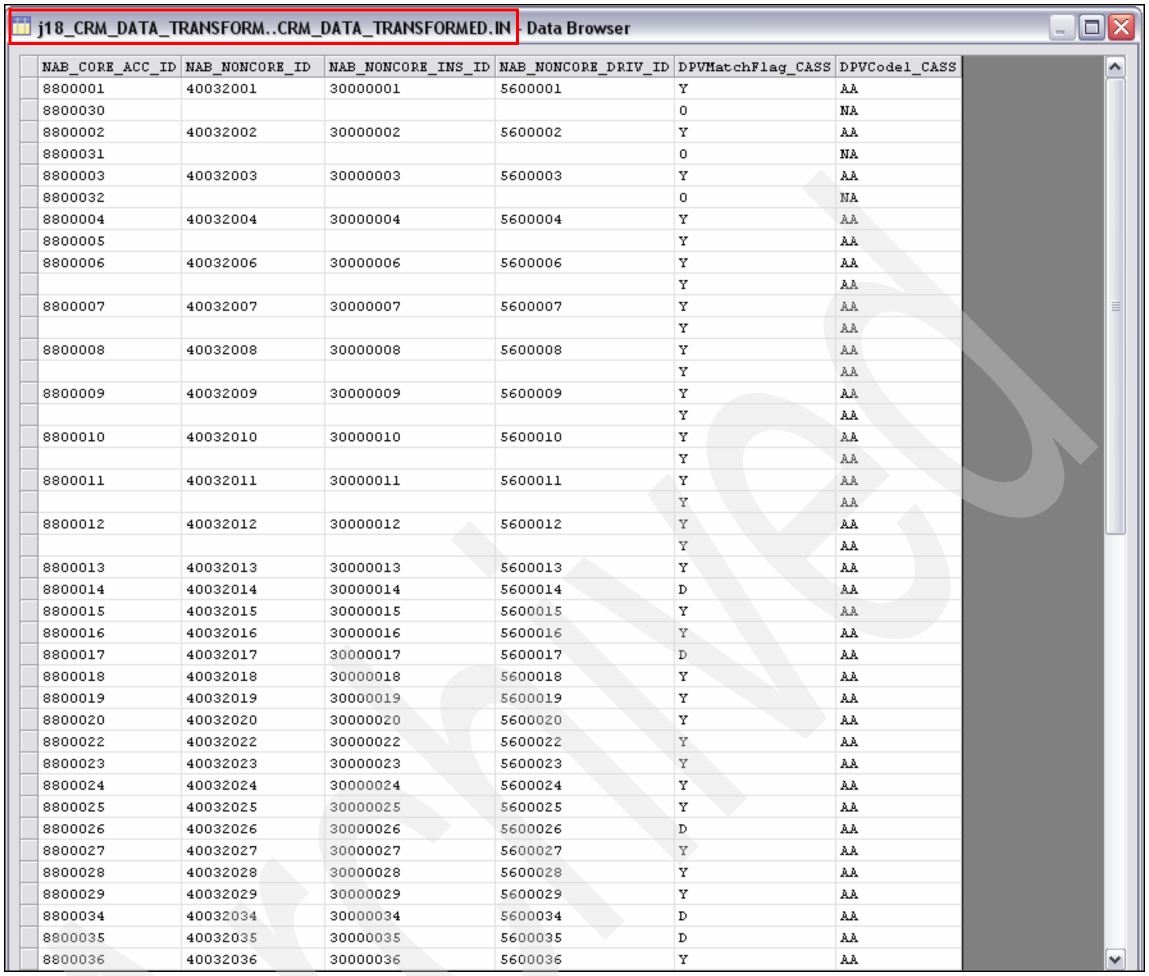
Figure 2-390 Create j18_CRM_DATA_TRANSFORM 12/14



j18_CRM_DATA_TRANSFORM..CRM_DATA_TRANSFORMED.IN - Data Browser

CELLPHONE	WORKPHONE	EMAIL	NCB_CORE_ID	NCB_NONCORE_ID	NAB_CORE_CUST_ID	NAB_CORE_ACC_
(541) 687-0488	(541) 768-6218				1000100	8800001
	(206) 447-1566				1000129	8800030
(541) 687-3141	(503) 623-2797				1000101	8800002
	(206) 731-3465				1000130	8800031
(206) 282-1578	(541) 752-3917				1000102	8800003
	(503) 256-1300				1000131	8800032
(206) 447-1875	(206) 447-1566				1000103	8800004
(415) 395-9000	(206) 731-3465				1000104	8800005
(559) 257-8000	(503) 256-1300				1000105	8800006
(408) 553-8211	(408) 269-0922		1500008003			
(559) 298-0688	(503) 251-6455				1000106	8800007
(408) 782-3700	(408) 919-1500	anders@olsson.com	1500008004			
(559) 448-6775	(503) 916-6227				1000107	8800008
(408) 243-1758	(999) 999-9999		1500008005			
(559) 453-5900	(503) 399-8695				1000108	8800009
(408) 269-0922	(999) 999-9999	jaz23@yahoo.com	1500008006			
(530) 224-6823	(503) 364-2989	Jackie@hotmail.com	1500008046		1000109	8800010
(408) 919-1500	(408) 527-1879	myself@gmail.com	1500008007			
(530) 224-4933	(800) 844-4974				1000110	8800011
(999) 999-9999	(415) 683-0763	arc@hotmail.com	1500008008			
(916) 487-4288	(503) 363-3934				1000111	8800012
(999) 999-9999	(415) 561-8511	dfx@usa.ibm.com	1500008009			
(916) 558-6400	(503) 652-9037				1000112	8800013
(877) 379-2096	(800) 935-4600				1000113	8800014
(831) 459-0445	(541) 687-0488	Renee@hotmail.com	1500008047		1000114	8800015
(206) 447-1566	(541) 687-3141				1000115	8800016
(206) 731-3465	(206) 282-1578				1000116	8800017
(503) 256-1300	(206) 447-1875				1000117	8800018
	(415) 395-9000				1000118	8800019
	(559) 257-8000				1000119	8800020
	(559) 448-6775				1000121	8800022
	(559) 453-5900				1000122	8800023
	(530) 224-6823				1000123	8800024
	(530) 224-4933				1000124	8800025
	(916) 487-4288	Dana@hotmail.com	1500008049	20001538	1000125	8800026
	(916) 558-6400				1000126	8800027
	(877) 379-2096				1000127	8800028
	(831) 459-0445				1000128	8800029
(408) 527-1879	(415) 346-1611	Burr@gmail.com	1500008030	20001521	1000133	8800034
(415) 683-0763	(415) 984-3772	Cathy@gmail.com	1500008031	20001522	1000134	8800035
(415) 561-8511	(415) 664-0983				1000135	8800036

Figure 2-391 Create j18_CRM_DATA_TRANSFORM 13/14



j18_CRM_DATA_TRANSFORM..CRM_DATA_TRANSFORMED.IN - Data Browser

NAB_CORE_ACC_ID	NAB_NONCORE_ID	NAB_NONCORE_INS_ID	NAB_NONCORE_DRIV_ID	DPVMatchFlag_CASS	DPVCode1_CASS
8800001	40032001	30000001	5600001	Y	AA
8800030				0	NA
8800002	40032002	30000002	5600002	Y	AA
8800031				0	NA
8800003	40032003	30000003	5600003	Y	AA
8800032				0	NA
8800004	40032004	30000004	5600004	Y	AA
8800005				Y	AA
8800006	40032006	30000006	5600006	Y	AA
				Y	AA
8800007	40032007	30000007	5600007	Y	AA
				Y	AA
8800008	40032008	30000008	5600008	Y	AA
				Y	AA
8800009	40032009	30000009	5600009	Y	AA
				Y	AA
8800010	40032010	30000010	5600010	Y	AA
				Y	AA
8800011	40032011	30000011	5600011	Y	AA
				Y	AA
8800012	40032012	30000012	5600012	Y	AA
				Y	AA
8800013	40032013	30000013	5600013	Y	AA
8800014	40032014	30000014	5600014	D	AA
8800015	40032015	30000015	5600015	Y	AA
8800016	40032016	30000016	5600016	Y	AA
8800017	40032017	30000017	5600017	D	AA
8800018	40032018	30000018	5600018	Y	AA
8800019	40032019	30000019	5600019	Y	AA
8800020	40032020	30000020	5600020	Y	AA
8800022	40032022	30000022	5600022	Y	AA
8800023	40032023	30000023	5600023	Y	AA
8800024	40032024	30000024	5600024	Y	AA
8800025	40032025	30000025	5600025	Y	AA
8800026	40032026	30000026	5600026	D	AA
8800027	40032027	30000027	5600027	Y	AA
8800028	40032028	30000028	5600028	Y	AA
8800029	40032029	30000029	5600029	Y	AA
8800034	40032034	30000034	5600034	D	AA
8800035	40032035	30000035	5600035	D	AA
8800036	40032036	30000036	5600036	Y	AA

Figure 2-392 Create j18_CRM_DATA_TRANSFORM 14/14

2.7 Post migration from North American Bank systems' to Northern California Bank

As described in the 2.2, "Business requirement" on page 480, migration of the North American Bank's core services systems to that of the Northern California Bank occurs in parallel with data integration that builds the CRM system. To expedite the process, we assume that the migration occurs before attempting to cleanse the data in the migrated Northern California Bank's core services systems.

Table 2-5 and Table 2-6 on page 852 is a summary of differences between the source North American Bank's core services systems and the target Northern California Bank's core services systems. These differences were identified in *IBM WebSphere Information Analyzer and Data Quality Assessment*, SG24-7508. The actual IBM WebSphere DataStage jobs that are used to migrate the data are discussed in *IBM InfoSphere DataStage Data Flow and Job Design*, SG24-7576.

Given that we do not have a migrated system, we describe the steps that are involved in cleansing the data in the migrated system using the jobs created in the data integration scenario in 2.6, "Data integration of North American Bank and Northern California Bank systems" on page 494. This is described in 2.7.1, "Cleansing names and addresses, matching, and surviving data in the migrated system" on page 854.

Table 2-5 Summary of differences between source and target core services and the action to be taken

North American Bank (source)				Northern California Bank (target)			Action to be taken
Column in the table	Metadata		Data Content example	Column in the table	Metadata Defined	Data Content example	
	Defined	Inferred					
CUSTOMER_ID in CUSTOMER	INT32	INT16	4079	ID in CUSTOMER	INT32	10003828	Generate new keys for the records in the source in the target. If required, create a cross reference table to map the source keys to the target keys until the transition of account numbers is fully achieved
ACCOUNT_ID in ACCOUNT	INT32	INT16	216	ID in ACCOUNT	INT32	11001500	Generate new keys for the records in the source in the target. Create a cross reference table to map the source keys to the target keys until the transition of account numbers is fully achieved

North American Bank (source)				Northern California Bank (target)			Action to be taken
ACCOUNT_ID,LOAN_I D in LOAN	(INT32, INT32)	(INT16, INT16)	(3295,2197)	LOAN_ID in LOAN	INT32	11001500	Generate new keys for the records in the source in the target. Create a cross reference table to map the source keys to the target keys until the transition of account numbers is fully achieved
ACCOUNT_ID,LOAN_I D,TRANSACTION_ID in LOAN_TRANSACTION	(INT32,INT32, INT32)	(INT16, INT16, INT32)	(152,102,13767)	ACCOUNT, UPDATED in TRANSACTION	(INTEGER, TIMESTAMP)	(11001583, 2005-11-19 11:28:29.745877)	Generate new keys for the records in the source in the target. Create a cross reference table to map the source keys to the target keys until the transition of account numbers is fully achieved
ACCOUNT_ID, TRANSACTION_ID in TRANSACTION	(INT32,INT32)	(INT16,INT32)	(100,27397)	ACCOUNT, UPDATED in TRANSACTION	(INT32, TIMESTAMP)	(11001583, 2005-11-19 11:28:29.745877)	Generate new keys for the records in the source in the target. Create a cross reference table to map the source keys to the target keys until the transition of account numbers is fully achieved
BRANCH_ID in BRANCH	INT32	INT8	51	ID in BRANCH	INT32	12001536	Generate new keys for the records in the source in the target. If necessary, create a cross reference table to map the source keys to the target keys until the transition of account numbers is fully achieved
GENDER_IND in CUSTOMER	CHAR(1)	CHAR(1)	F	GENDER in CUSTOMER	CHAR(1)	1	Transform the values from the source to the target.
LEVEL_CD in CUSTOMER	CHAR(2)	CHAR(2)	SL	CLASS in CUSTOMER	INT32	7	Map with data type transformation

North American Bank (source)				Northern California Bank (target)			Action to be taken
TITLE in CUSTOMER	CHAR(3)	CHAR(2)	MS	NAME in CUSTOMER	CHAR(50)	Jazmine Fisher	Map the TITLE, FIRST_NAME and LAST_NAME columns in the source into the NAME column in the target
FIRST_NAME in CUSTOMER	VARCHAR(20)	CHAR(12)	SHAYLA	NAME in CUSTOMER	CHAR(50)	Jazmine Fisher	
LAST_NAME in CUSTOMER	VARCHAR(20)	CHAR(16)	VAN DER ZIJDEN	NAME in CUSTOMER	CHAR(50)	Jazmine Fisher	
HOME_ADDRESS in CONTACT_INFO	VARCHAR(50)	VARCHAR(36)	1301 Evans Avenue, San Francisco, CA	ADDR1 in CUSTOMER	VARCHAR(50)	397 CHISANA STREET	Direct mapping
HOME_ZIP in CONTACT_INFO	CHAR(9)	INT32 length 5	94124	ZIP in CUSTOMER	CHAR(10)	90028	Transform by including a hyphen
HOME_PHONE in CONTACT_INFO	CHAR(15)	INT64 length 10	8005553717	HOMEPHONE in CUSTOMER	CHAR(15)	517-555-1385	Map and include hyphen
WORK_PHONE in CONTACT_INFO	CHAR(15)	CHAR(10)	8145553731,NA	WORKPHONE in CUSTOMER	CHAR(15)	000-000-0000	Map and include hyphen
CELL_PHONE in CONTACT_INFO	CHAR(15)	CHAR(10)	8175553734	CELLPHONE in CUSTOMER	CHAR(15)	517-555-4641	Map and include hyphen
TYPE_IND in ACCOUNT	CHAR(1)	CHAR(1)	C	TYPE in ACCOUNT	CHAR(2)	SS	Transform the values from the source to the target.
RATES in LOAN	DECIMAL(8,5)	DECIMAL (7,5)	2.15	INTEREST_RATE in LOAN	CHAR(20)	19.75	Map with data type transformation
INITIAL_VALUE in LOAN	DECIMAL(9,2)	DECIMAL(8,2)	1000	INITIAL_LOAN_VALUE in LOAN	CHAR(20)	100000	Map with data type transformation
LATE_FEE in LOAN	DECIMAL(9,2)	DECIMAL(6,2)	100	LATE_FEE in LOAN	CHAR(20)	50	Map with data type transformation
LATE_RATE in LOAN	DECIMAL(8,5)	DECIMAL(7,5)	5.123	LATE_INTEREST_RATE in LOAN	CHAR(20)	10	Map with data type transformation
BALANCE in LOAN	DECIMAL(9,2)	DECIMAL(8,2)	500	BALANCE in LOAN	CHAR(20)	75000	Map with data type transformation
TRANS_TYPE_CD in LOAN_TRANSACTION	CHAR(2)	CHAR(2)	D	CODE in TRANSACTION	CHAR(1)	C	Transform the values from the source to the target.
AMOUNT in LOAN_TRANSACTION	DECIMAL(9,2)	DECIMAL(8,2)	400.00	BALANCE in TRANSACTION	CHAR(20)	300	Map with data type transformation
TRANS_TYPE_CD in TRANSACTION	CHAR(2)	CHAR(2)	D	CODE in TRANSACTION	CHAR(1)	C	Transform the values from the source to the target.
AMOUNT in TRANSACTION	DECIMAL(9,2)	DECIMAL(7,2)	400.00	BALANCE in TRANSACTION	CHAR(20)	300	Map with data type transformation
TRANS_TYPE_CD in TRANSACTION_TYPE_REF	CHAR(2)	CHAR(2)	D	CODE in TRANSACTION	CHAR(1)	C	Transform the values from the source to the target.

North American Bank (source)				Northern California Bank (target)			Action to be taken
LEVEL_CD in LEVEL_REF	CHAR(2)	CHAR(2)	PL				Ignore this column because it is part of a reference table

Table 2-6 Missing information in source or target bank relating to core services and action to be taken

Data element	North American Bank (source)	Northern California Bank (target)	Action to be taken
Customer's preferred language for interaction	Not defined	ISO code (PREF_LANG column in the CUSTOMER table defined as NOT NULL WITH DEFAULT 'ENG')	No action because default will be applied
Currency of deposits	Not defined	ISO code (CURRENCY column in the ACCOUNT and PORTFOLIO tables defined as nullable; CURRENCY column in the ACCTYPE table defined as NOT NULL WITH DEFAULT ' ')	No action because null or default will be applied
Country	Not defined	ISO code (COUNTRY column in the BRANCH table defined as nullable; CTRY2, CTRY3, and CTRYN columns in the COUNTRY lookup table defined as NOT NULL)	No action because null or default will be applied for the COUNTRY column in the BRANCH table. CTRY2, CTRY3, and CTRYN columns are in the COUNTRY lookup table and need no action
Nationality	Free text field (NATIONALITY column in the CUSTOMER table)	Not defined	Ignore this field during migration

Data element	North American Bank (source)	Northern California Bank (target)	Action to be taken
Nickname	Free text field (NICKNAME column in the CUSTOMER table)	Not defined	Ignore this field during migration
Credit score	Free text field (CREDIT_SCORE column in the CUSTOMER table)	Not defined	Ignore this field during migration
Churn indicator	CHAR(1) (CHURN_IND column in the CUSTOMER table)	Not defined	Ignore this field during migration
Automatic debit indicator	CHAR(1) (AUTOMAT_DEBIT_IND column in the LOAN table)	Not defined	Ignore this field during migration
Work address	Free text field (WORK_ADDRESS column in the CONTACT_INFO table)	Not defined	Ignore this field during migration
Work ZIP code	Free text field (WORK_ZIP column in the CONTACT_INFO table)	Not defined	Ignore this field during migration
Account fee frequency	Not defined	*Y* for yearly, *M* for monthly, and *C* per check (FEEFRQ column in the ACCTYPE table defined as nullable)	No action
City	Not defined	Separate field (CITY column in the CUSTOMER table defined as NOT NULL)	CITY column in the CUSTOMER table will have to be populated by extracting this information from the HOME_ADDRESS column in the CONTACT_INFO table in the North American Bank core services

Data element	North American Bank (source)	Northern California Bank (target)	Action to be taken
Customer type	Not defined	"P" for person, "O" for organization (TYPE column in the CUSTOMER table defined as NOT NULL WITH DEFAULT '-')	No action because default will be applied
Employee information	Defined	Defined	Not going to migrate the HR system.
Various amounts	▸ DECIMAL(9,2) (BALANCE in ACCOUNT) ▸ DECIMAL(9,2) (MIN_AMOUNT in ACCOUNT) ▸ DECIMAL(9,2) (OVERDRAFT in ACCOUNT) ▸ DECIMAL(9,2) (OVERDRAFT_LIMIT in ACCOUNT) ▸ DECIMAL(8,5) (OVERDRAFT_RATE in ACCOUNT) ▸ DECIMAL(9,2) (OVERDRAFT_FEE in ACCOUNT)		Modify target system with the correct data type, precision and scale. Significant impact.

2.7.1 Cleansing names and addresses, matching, and surviving data in the migrated system

We need to process the migrated data in the DB2 tables using many of the same steps described in 2.6.2, "Cleansing Northern California Bank's core and non-core services" on page 706 and 2.6.3, "Matching and surviving Northern California Bank and Northern California Bank information" on page 783. Specifically, the migrated data can be cleansed, matched (identification of duplicates), and survived as follows:

1. The migrated data should be processed by jobs "j00_SRC_NCB" on page 709 through "j11_FUNNEL_NCB_DATA" on page 782. This results in a merged file of the CUSTOMER and BCUSTOMER files that contains standardized name and address information.

2. This merged file should then be processed by jobs “j14_CRM_FREQUENCY” on page 798 through “j17_FUNNEL_UNDUP_RES_DATA” on page 829 to get a cleansed file of migrated data.

   A set of IBM WebSphere DataStage jobs then needs to be developed to update Northern California Bank’s systems from the file that was created in the “j17_FUNNEL_UNDUP_RES_DATA” on page 829 job. This task is described in *IBM InfoSphere DataStage Data Flow and Job Design*, SG24-7576.

A

IBM Information Server overview

In this chapter we provide an overview of IBM Information Server architecture and processing flow.

We cover the following topics:

- Introduction
- IBM Information Server architecture
- Configuration flow
- Runtime flow

A.1 Introduction

Over the years, most organizations have made significant investments in enterprise resource planning, customer relationship management, and supply chain management packages in addition to their home grown applications. This has resulted in larger amounts of data being captured about their businesses. To turn all this data into consistent, timely, and accurate information for decision-making requires an effective means of integrating information. Statutory compliance requirements such as Basel II and Sarbanes-Oxley place additional demands for consistent, complete, and trustworthy information.

IBM Information Server addresses these critical information integration requirements of consistent, complete, and trustworthy information with a comprehensive, unified foundation for enterprise information architectures, capable of scaling to meet any information volume requirement so that companies can deliver business results faster and with higher quality results for all the following initiatives:

- Business intelligence

  IBM Information Server makes it easier to develop a unified view of the business for better decisions. It helps you understand existing data sources, cleanse, correct, and standardize information, and load analytical views that can be reused throughout the enterprise.

- Master data management

  IBM Information Server simplifies the development of authoritative master data by showing where and how information is stored across source systems. It also consolidates disparate data into a single, reliable record, cleanses and standardizes information, removes duplicates, and links records together across systems. This master record can be loaded into operational data stores, data warehouses, or master data applications such as WebSphere Customer Center and WebSphere Product Center. The record can also be assembled, completely or partially, on demand.

- Infrastructure rationalization

  IBM Information Server aids in reducing operating costs by showing relationships between systems and by defining migration rules to consolidate instances or move data from obsolete systems. Data cleansing and matching ensure high-quality data in the new system.

- Business transformation

  IBM Information Server can speed development and increase business agility by providing reusable information services that can be plugged into applications, business processes, and portals. These standards-based information services are maintained centrally by information specialists but are widely accessible throughout the enterprise.

- Risk and compliance

  IBM Information Server helps improve visibility and data governance by enabling complete, authoritative views of information with proof of lineage and quality. These views can be made widely available and reusable as shared services, while the rules inherent in them are maintained centrally.

IBM Information Server combines the technologies of key information integration functions within the IBM Information Integration Solutions portfolio into a single unified platform that enables companies to understand, cleanse, transform, move, and deliver trustworthy and context-rich information as shown in Figure A-1 on page 861.

IBM Information Server includes the following product modules:

- IBM WebSphere DataStage

  It enables organizations to design data flows that extract information from multiple source systems, transform it in ways that make it more valuable, and then deliver it to one or more target databases or applications.

- IBM WebSphere QualityStage

  Designed to help organizations understand and improve the overall quality of their data assets, WebSphere QualityStage provides advanced features to help investigate, repair, consolidate, and validate heterogeneous data within an integration workflow.

- IBM WebSphere Federation Server

  It enables applications to access and integrate diverse data and content sources as though they were a single resource—regardless of where the information resides—while retaining the autonomy and integrity of the heterogeneous data and content sources. This enabling technology is transparent, heterogeneous, and extensible, and provides high function and high performance.

- IBM WebSphere Information Services Director

  IBM Information Server provides a unified mechanism for publishing and managing shared service-oriented architecture (SOA) services across data quality, data transformation, and federation functions, allowing information specialists to easily deploy services for any information integration task and consistently manage them. This enables developers to take data integration logic built using IBM Information Server and publish it as an *always on* service—in minutes. The common services also include the metadata services, which provide standard service-oriented access and analysis of metadata across the platform.

- IBM WebSphere Information Analyzer

  IBM WebSphere Information Analyzer profiles and analyzes data so that you can deliver trusted information to your users. It can automatically scan samples of your data to determine their quality and structure. This analysis aids you in understanding the inputs to your integration process, ranging from individual fields to high-level data entities. Information analysis also enables you to correct problems with structure or validity before they affect your project. While analysis of source data is a critical first step in any integration project, you must continually monitor the quality of the data. IBM WebSphere Information Analyzer enables you to treat profiling and analysis as an ongoing process and create business metrics that you can run and track over time.

- IBM WebSphere Business Glossary

  IBM Information Server provides a Web-based tool that enables business analysts and subject-matter experts to create, manage, and share a common enterprise vocabulary and classification system. WebSphere Business Glossary functionality is powered by and actively connected to WebSphere Metadata Server. This enables users to link business terms to more technical artifacts managed by WebSphere Metadata Server. The Metadata Server also enables sharing of the business terms by IBM Rational® Data Architect and WebSphere Information Analyzer, creating a common set of semantic tags for reuse by data modelers, data analysts, business analysts, and end users.

- IBM WebSphere Metadata Server

  IBM Information Server provides the next-generation metadata repository that is fully integrated and common across all product modules, including WebSphere Information Analyzer, WebSphere QualityStage, WebSphere DataStage, and WebSphere Business Glossary. The metadata services infrastructure of IBM Information Server is designed to allow metadata to be more easily managed, accessed by those who need it, and shared across heterogeneous technologies through an SOA.

- IBM WebSphere DataStage MVS™ Edition

  IBM Information Server brings data transformation capabilities to the mainframe with its WebSphere DataStage MVS Edition product module. WebSphere DataStage MVS Edition consolidates, collects, and centralizes information from various systems and mainframes using native execution, from a single design environment.

Note: For complete details about these product modules, refer to the documentation in the Web site:

`http://www.ibm.com/software/data/integration/info_server/`

A number of companion products support IBM Information Server, such as Rational Data Architect and WebSphere Replication Server Event Publisher.

A.2 IBM Information Server architecture

IBM Information Server provides a unified architecture that works with all types of information integration as shown in Figure A-1. A unified user interface, common services, key integration functions (understand, cleanse, transform and move, and deliver), unified parallel processing, and unified metadata are at the core of the architecture.

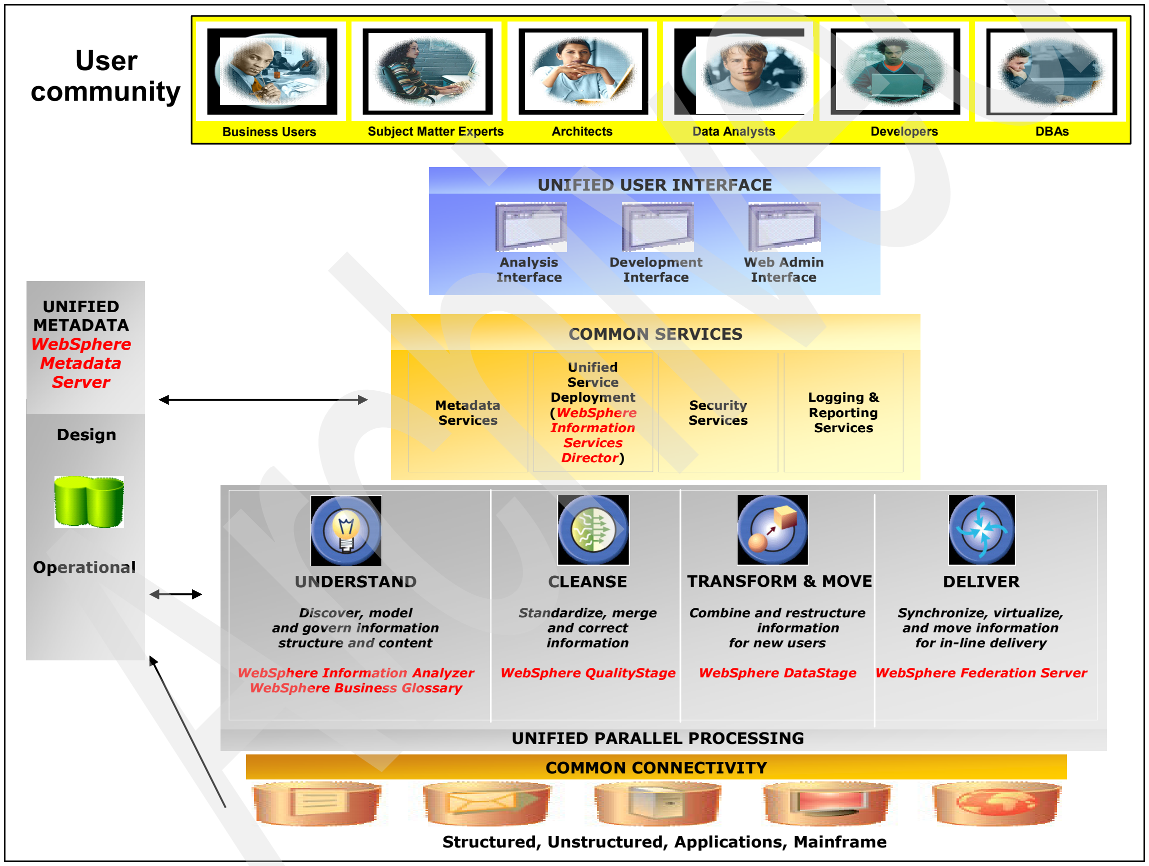

User community
Business Users
Subject Matter Experts
Architects
Data Analysts
Developers
DBAs
UNIFIED USER INTERFACE
Analysis Interface
Development Interface
Web Admin Interface
UNIFIED METADATA
WebSphere Metadata Server
Design
Operational
COMMON SERVICES
Metadata Services
Unified Service Deployment (WebSphere Information Services Director)
Security Services
Logging & Reporting Services
UNDERSTAND
Discover, model and govern information structure and content
WebSphere Information Analyzer
WebSphere Business Glossary
CLEANSE
Standardize, merge and correct information
WebSphere QualityStage
TRANSFORM & MOVE
Combine and restructure information for new users
WebSphere DataStage
DELIVER
Synchronize, virtualize, and move information for in-line delivery
WebSphere Federation Server
UNIFIED PARALLEL PROCESSING
COMMON CONNECTIVITY
Structured, Unstructured, Applications, Mainframe

Figure A-1 IBM Information Server architecture

The architecture is service oriented, enabling IBM Information Server to work within an organization's evolving enterprise SOAs. An SOA also connects the individual product modules of IBM Information Server. By eliminating duplication of functions, the architecture efficiently uses resources and reduces the amount of development and administrative effort that are required to deploy an integration solution.

In this section, we describe each of the following components of the architecture briefly:

- Unified user interface
- Common services
- Key integration functions
- Unified parallel processing
- Unified metadata
- Common connectivity
- Client application access to services

A.2.1 Unified user interface

The unified user interface enables an organization's entire user community of business users, subject matter experts, architects, data analysts, developers, and database administrators (DBAs) to collaborate, administer, and query information within the enterprise. A security infrastructure ensures that users are permitted to access information and perform tasks for which they are authorized.

The face of IBM Information Server is a common graphical interface and tool framework. Shared interfaces such as the IBM Information Server console and Web console provide a common look and feel, visual controls, and user experience across products. Common functions such as catalog browsing, metadata import, query, and data browsing all expose underlying common services in a uniform way. IBM Information Center provides rich client interfaces for highly detailed development work and thin clients that run in Web browsers for administration. Application programming interfaces (APIs) support a variety of interface styles that include standard request-reply, service-oriented, event-driven, and scheduled task invocation.

The three broad user interface categories are the analysis interface, development interface, and Web Admin interface as shown in Figure A-1 on page 861.

A.2.2 Common services

IBM Information Server is built entirely on a set of shared services that centralize core tasks across the platform. These include administrative tasks such as unified service deployment, security, user administration, logging, and reporting.

The common services provides flexible, configurable interconnections among the many parts of the architecture

Shared services allow these tasks to be managed and controlled in one place, regardless of which product module is being used. The common services also include the metadata services, which provide standard service-oriented access and analysis of metadata across the platform. In addition, the common services layer manages how services are deployed from any of the product functions, allowing cleansing and transformation rules or federated queries to be published as shared services within an SOA, using a consistent and easy-to-use mechanism.

The common services layer is deployed on J2EE™-compliant application servers such as IBM WebSphere Application Server.

Attention: Today, common services are consumed exclusively by the various components of IBM Information Server. These common services are currently *not* exposed as public SOA services, and therefore cannot be invoked by applications or tools.

IBM Information Server products can access four general categories of service such as design, execution, metadata, and unified service deployment as follows:

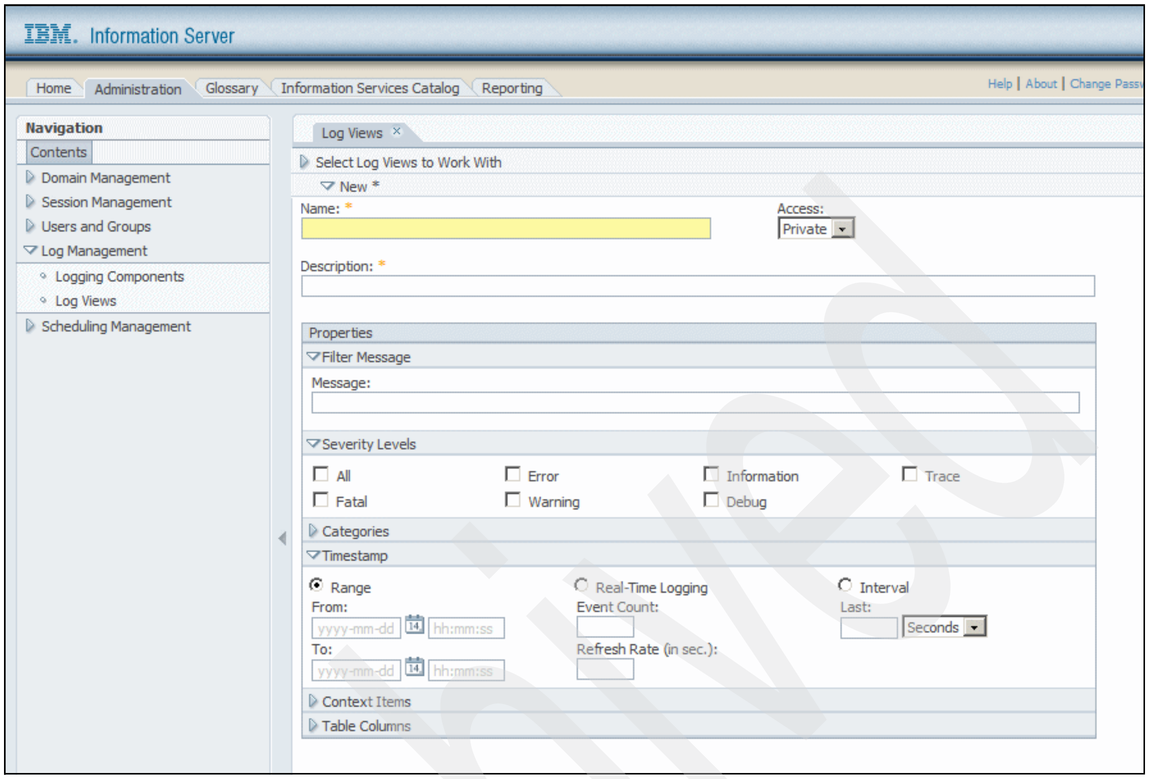
A.2.2.1 Design services

Design services help developers create function-specific services that can also be shared. For example, WebSphere Information Analyzer calls a column analyzer service that was created for enterprise data analysis but can be integrated with other parts of IBM Information Server because it exhibits common SOA characteristics.

A.2.2.2 Execution services

Execution services include logging, scheduling, monitoring, reporting, security, and Web framework.

- Log services help you manage logs across all of the IBM Information Server suite components. The Web console shown in Figure A-2 on page 864 provides a central place to view logs and resolve problems. Logs are stored in the common repository, and each IBM Information Server suite component defines relevant logging categories. You can configure which categories of logging messages are saved in the repository. Log views are saved queries that an administrator can create to help with common tasks. For example, you might want to display all of the errors in DataStage jobs that ran in the past 24 hours. Logging is organized by server components. The Web console displays default and active configurations for each component.



Access: Private

[✓ Range] Real-Time Logging Interval

Last: Seconds

Figure A-2 Web console for setting up logs

- Scheduling services help plan and track activities such as logging and reporting, and suite component tasks such data monitoring and trending. Schedules are maintained using the IBM Information Server console shown in Figure A-3 on page 865, which helps you define schedules, view their status, history, and forecast, and purge them from the system.

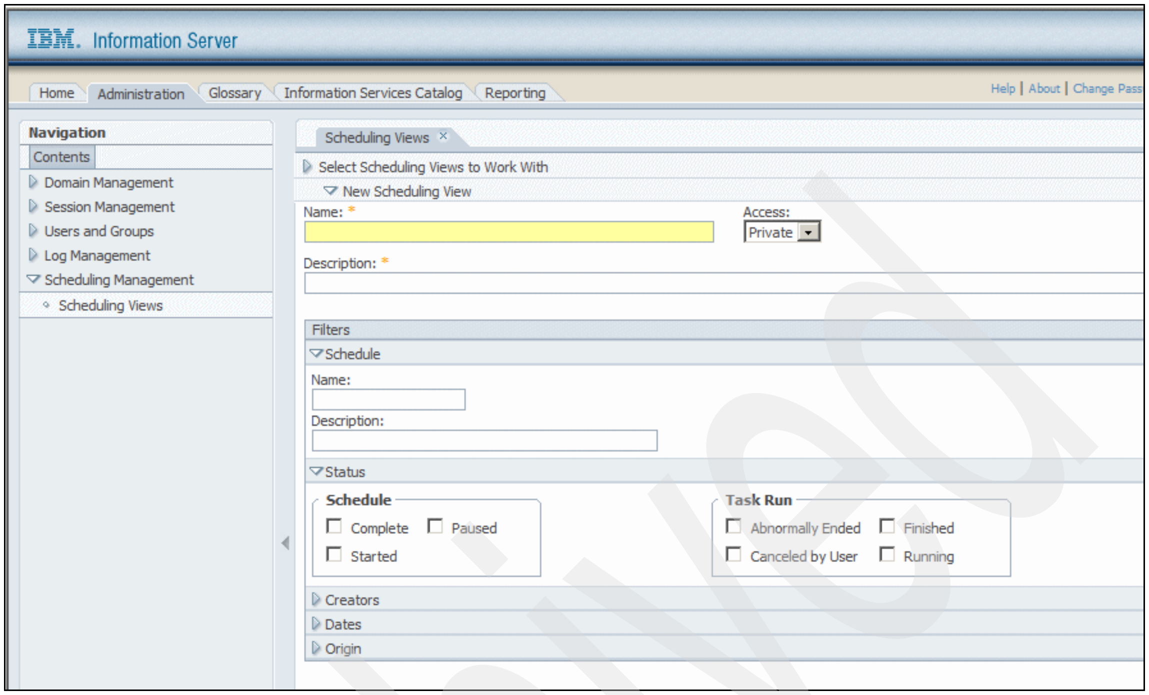
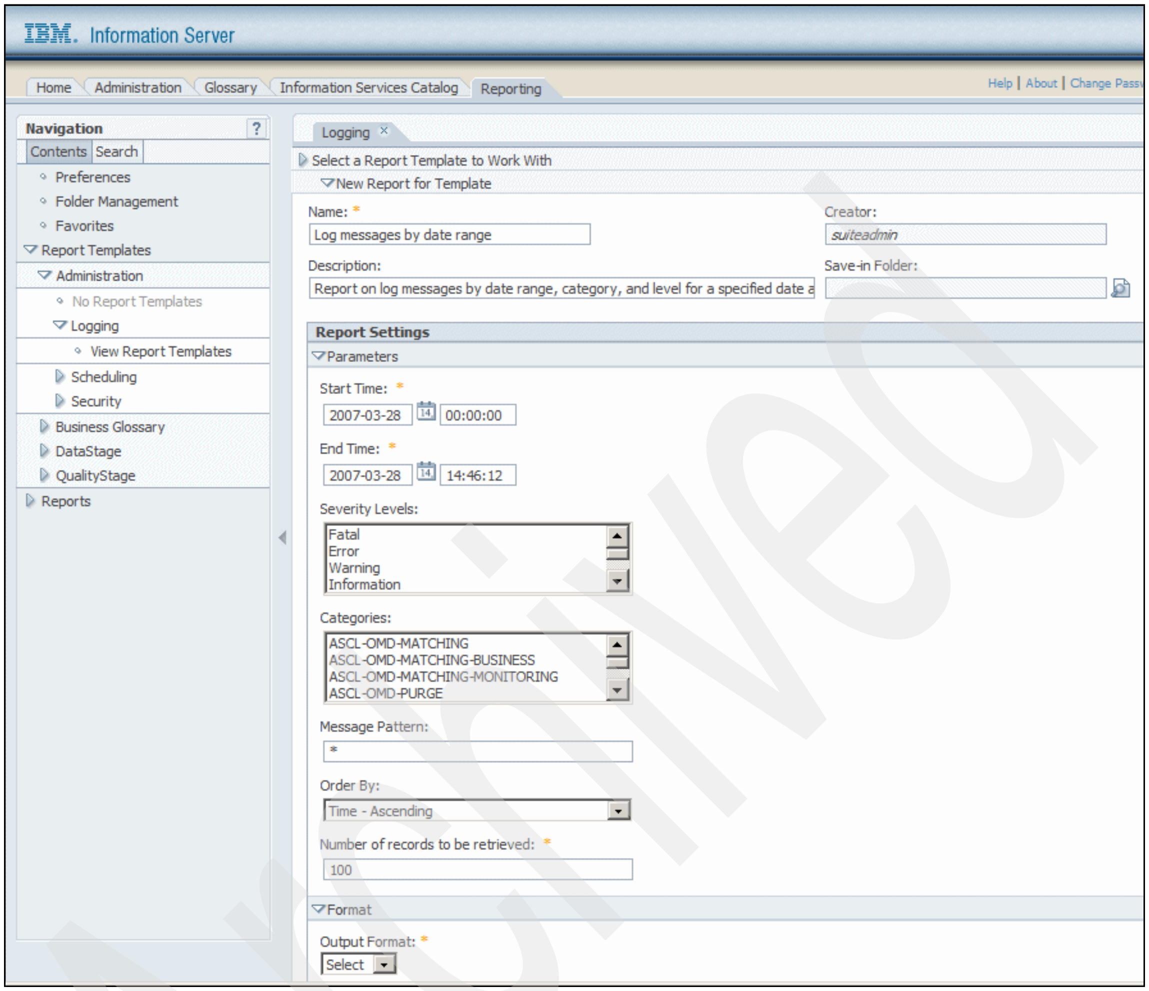


Figure A-3 Web console scheduling view creation

- Reporting services manage run time and administrative aspects of reporting for IBM Information Server. You can create product-specific reports for WebSphere DataStage, WebSphere QualityStage, and WebSphere Information Analyzer, and cross-product reports for logging, monitoring, scheduling, and security services. All reporting tasks are set up and run from a single interface, the IBM Information Server Web console. You can retrieve and view reports and schedule reports to run at a specific time and frequency. You define reports by choosing from a set of predefined parameters and templates as shown in Figure A-4 on page 866. You can specify a history policy that determines how the report will be archived and when it expires. Reports can be formatted as HTML, PDF, or Microsoft Word documents.



Name: * Log messages by date range

Creator: suiteadmin

Description: Report on log messages by date range, category, and level for a specified date a

Save-in Folder:

Start Time: * 2007-03-28 00:00:00

End Time: * 2007-03-28 14:46:12

Severity Levels: Fatal, Error, Warning, Information

Categories: ASCL-OMD-MATCHING, ASCL-OMD-MATCHING-BUSINESS, ASCL-OMD-MATCHING-MONITORING, ASCL-OMD-PURGE

Message Pattern: *

Order By: Time - Ascending

Number of records to be retrieved: * 100

Output Format: * Select

Figure A-4 Web console logging report creation

- Security services support role-based authentication of users, access-control services, and encryption that complies with many privacy and security regulations. The Web console shown in Figure A-5 on page 867 helps administrators add users, groups, and roles and lets administrators browse, create, delete, and update operations within Information Server. Directory services act as a central authority that can authenticate resources and manage identities and relationships among identities. You can base directories on IBM Information Server's own internal directory or on external directories that are based on LDAP, Microsoft's Active Directory®, or UNIX®. Users only use one credential to access all the components of Information

Server. A set of credentials is stored for each user to provide single sign-on to the products registered with the domain.

Note: A white paper is currently being developed to provide guidelines about implementing authentication, access control, and encryption security within an IBM Information Server environment. A reference to this white paper will be provided here when it becomes publicly available.

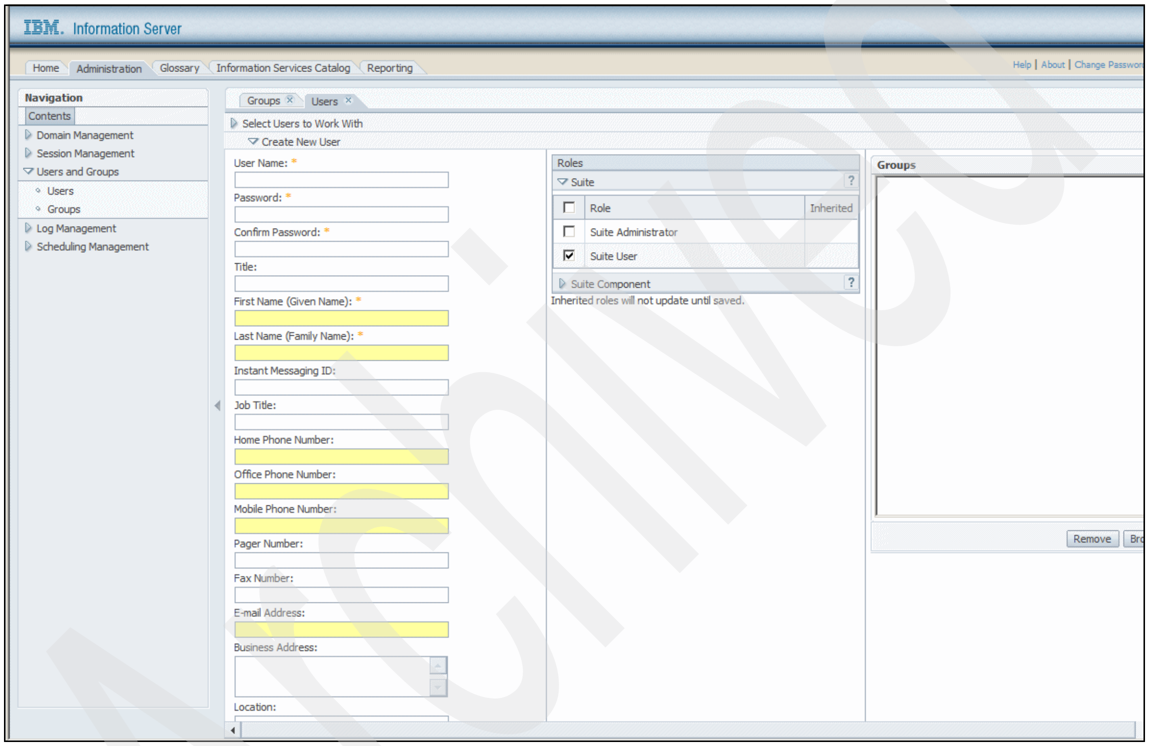


Figure A-5 Web console to administer users and groups

A.2.2.3 Metadata services

Metadata services enable metadata to be shared “live” across tools so that changes made in one IBM Information Server product are instantly visible across all of the product modules. Metadata services are tightly integrated with the common repository and are packaged in WebSphere Metadata Server. You can also exchange metadata with external tools by using metadata services.

The major metadata services components of IBM Information Server are WebSphere Business Glossary, WebSphere Metadata Server, and WebSphere MetaBrokers and bridges.

- WebSphere Business Glossary is a Web-based application that provides a business-oriented view into the data integration environment. By using WebSphere Business Glossary, you can view and update business descriptions and access technical metadata. Metadata is best managed by business analysts who understand the meaning and importance of the information assets to the business. Designed for collaborative authoring, WebSphere Business Glossary gives users the ability to share insights and experiences about data. It provides users with the following information about data resources:
  - Business meaning and descriptions of data
  - Stewardship of data and processes
  - Standard business hierarchies
  - Approved terms

  WebSphere Business Glossary is organized and searchable according to the semantics that are defined by a controlled vocabulary, which you can create using the Web console.

- WebSphere Metadata Server provides a variety of services to other components of IBM Information Server:
  - Metadata access
  - Metadata integration
  - Metadata import and export
  - Impact analysis
  - Search and query

  WebSphere Metadata Server provides a common repository with facilities that are capable of sourcing, sharing, storing, and reconciling a comprehensive spectrum of metadata including business metadata and technical metadata as follows:

  - Business metadata provides business context for information technology assets and adds business meaning to the artifacts that are created and managed by other IT applications. Business metadata includes controlled vocabularies, taxonomies, stewardship, examples, and business definitions.
  - Technical metadata provides details about source and target systems, their table and field structures, attributes, derivations, and dependencies. Technical metadata also includes details about profiling, quality, and ETL processes, projects, and users.

- WebSphere MetaBrokers and bridges provide semantic model mapping technology that allows metadata to be shared among applications for all products that are used in the data integration life cycle:
  - Data modeling or case tools
  - Business intelligence applications
  - Data marts and data warehouses
  - Enterprise applications
  - Data integration tools

These components can be used to establish common data definitions across business and IT functions:

- Drive consistency throughout the data integration life cycle
- Deliver business-oriented and IT-oriented reporting
- Provide enterprise visibility for change management
- Easily extend to new, existing, and homegrown metadata sources

A.2.2.4 Unified service deployment

IBM Information Server provides an SOA infrastructure that exposes data transformation processes,[1] federated queries,[2] and database stored procedures as a set of shared services and operations. This is performed by using a consistent and intuitive graphical interface, and managed after publication using the same user interface.

IBM Information Server provides standard service-oriented interfaces for enterprise data integration. The built-in integration logic of IBM Information Server can easily be encapsulated as service objects that are embedded in user applications. These service objects have the following characteristics:

- Always on

  By definition, the services are always running and waiting for requests. This ability removes the overhead of batch startup and shutdown and enables services to respond instantaneously to requests.

- Scalable

  The services distribute request processing and stop and start jobs across multiple WebSphere DataStage servers, enabling high performance with large, unpredictable volumes of requests.

- Standards-based

  The services are based on open standards and can easily be invoked by standards-based technologies including Web Services Description Language

[1] Created from new or existing WebSphere DataStage or WebSphere QualityStage jobs
[2] Created by WebSphere Federation Server

(WSDL), enterprise application integration (EAI), and enterprise service bus (ESB) platforms, applications, and portals.

- Manageable

  Monitoring services coordinate timely reporting of system performance data.

- Flexible

  You can invoke the services by using multiple mechanisms (bindings) and choose from many options for using the services.

- Reliable and highly available

  If any WebSphere DataStage server becomes unavailable, it routes service requests to a different server in the pool.

- Reusable

  The services publish their own metadata, enabling them to be found and called across any network.

- High performance

  Load balancing and the underlying parallel processing capabilities of IBM Information Server create high performance for any type of data payload.

A data integration service is created by designing the data integration process logic in IBM Information Server and publishing it as a service. These services can then be accessed by external projects and technologies.

WebSphere Information Services Director provides a foundation for information services by allowing you to leverage the other components of IBM Information Server for understanding, cleansing, and transforming information and deploying those integration tasks as consistent and reusable information services.

WebSphere Information Services Director provides an integrated environment for designing services that enables you to rapidly deploy integration logic as services without assuming extensive development skills. With a simple, wizard-driven interface, in a few minutes you can attach a specific binding and deploy a reusable integration service. WebSphere Information Services Director also provides these features:

- Administrator services for cataloging and registering services.
- Shared reporting and security services.
- A metadata services layer that promotes reuse of the information services by actually defining what the service does and what information it delivers.

A.2.3 Key integration functions

We describe the four key integration functions shown in Figure A-1 on page 861 briefly here:

- Understand your data

  IBM Information Server helps you to automatically discover, define, and model information content and structure and understand and analyze the meaning, relationships, and lineage of information. By automating data profiling and data-quality auditing within systems, organizations can achieve the following goals:

  – Understand data sources and relationships
  – Eliminate the risk of using or proliferating bad data
  – Improve productivity through automation
  – Take advantage of existing IT investments

  IBM Information Server makes it easier for businesses to collaborate across roles. Data analysts can use analysis and reporting functionality, generating integration specifications and business rules that they can monitor over time. Subject matter experts can use Web-based tools to define, annotate, and report on fields of business data. A common metadata foundation makes it easier for different types of users to create and manage metadata by using tools that are optimized for their roles.

  The upcoming WebSphere Information Analyzer product module will provide this functionality.

- Cleanse your information

  IBM Information Server supports information quality and consistency by standardizing, validating, matching, and merging data. It can certify and enrich common data elements, use trusted data such as postal records for name and address information, and match records across or within data sources. IBM Information Server allows a single record to survive from the best information across sources for each unique entity, helping you to create a single, comprehensive, and accurate view of information across source systems.

  The WebSphere QualityStage product module currently provides this functionality.

- Transform your data into information and move

  IBM Information Server transforms and enriches information to ensure that it is in the proper context for new uses. Hundreds of prebuilt transformation functions combine, restructure, and aggregate information.

  Transformation functionality is broad and flexible, to meet the requirements of varied integration scenarios. For example, IBM Information Server provides inline validation and transformation of complex data types such as U.S. Health Insurance Portability and Accountability Act (HIPAA), along with high-speed joins and sorts of heterogeneous data. IBM Information Server also provides high-volume, complex data transformation and movement functionality that can be used for standalone extract/transform/load (ETL) scenarios, or as a real-time data processing engine for applications or processes.

The WebSphere DataStage product modules currently provide this functionality.

- Deliver your information

  IBM Information Server provides the ability to virtualize, synchronize, or move information to the people, processes, or applications that need it. Information can be delivered through federation or time-based or event-based processing, moved in large bulk volumes from location to location, or accessed in place when it cannot be consolidated. IBM Information Server provides direct, native access to a wide variety of information sources, both mainframe and distributed. It provides access to databases, files, services and packaged applications, and to content repositories and collaboration systems. Companion products allow high-speed replication, synchronization and distribution across databases, change data capture, and event-based publishing of information.

  The WebSphere Federation Server product module currently provides this functionality.

A.2.4 Unified parallel processing

Much of the work that IBM Information Server does takes place within the parallel processing engine. The engine handles data processing needs as diverse as performing analysis of large databases for WebSphere Information Analyzer, data cleansing for WebSphere QualityStage, and complex transformations for WebSphere DataStage. This parallel processing engine is designed to deliver:

- Parallelism and pipelining to complete increasing volumes of work in decreasing time windows.
  - Data partitioning is an approach to parallelism that involves breaking the record set into partitions, or subsets of records. Data partitioning generally provides linear increases in application performance.

    IBM Information Server automatically partitions data based on the type of partition that the stage requires. In a well-designed, scalable architecture, the developer does not need to be concerned about the number of partitions that will run, the ability to increase the number of partitions, or re-partitioning data.

  – Data pipelining is the process of pulling records from the source system and moving them through the sequence of processing functions that are defined in the data-flow (the job). Because records are flowing through the pipeline, they can be processed without writing the records to disk,

- Scalability by adding hardware (for example, processors or nodes in a grid) with no changes to the data integration design.
- Optimized database, file, and queue processing to handle large files that cannot fit in memory all at once or with large numbers of small files.

Note: The dynamic parallelization of all potential service implementations is an objective of this architecture.

A.2.5 Unified metadata

IBM Information Server is built on a unified metadata infrastructure that enables shared understanding between business and technical domains. This infrastructure reduces development time and provides a persistent record that can improve confidence in information.

All functions of IBM Information Server share the same metamodel, making it easier for different roles and functions to collaborate. A common metadata repository provides persistent storage for all IBM Information Server product modules. All of the products depend on the repository to navigate, query, and update metadata.

The repository contains two kinds of metadata:

- Dynamic metadata that includes design-time information.
- Operational metadata that includes performance monitoring, audit and log data, and data profiling sample data.

Because the repository is shared by all product modules, profiling information that is created by WebSphere Information Analyzer is instantly available to users of WebSphere DataStage and QualityStage, for example.The repository is a J2EE application that uses a standard relational database such as IBM DB2, Oracle®, or SQL Server® for persistence (DB2 is provided with IBM Information Server). These databases provide backup, administration, scalability, parallel access, transactions, and concurrent access.

A.2.6 Common connectivity

IBM Information Server connects to information sources whether they are structured, unstructured, on the mainframe, or applications.

Metadata-driven connectivity is shared across the product modules, and connection objects are reusable across functions. Connectors provide design-time importing of metadata, data browsing and sampling, run-time dynamic metadata access, error handling, as well as high functionality and high performance run-time data access.

Prebuilt interfaces for packaged applications called *Packs* provide adapters to SAP, Siebel®, Oracle, and others, enabling integration with enterprise applications and associated reporting and analytical systems.

A.2.7 Client application access to services

After an information service is enabled by IBM Information Server, any enterprise application, .Net or Java™ developer, Microsoft Office, or integration software can invoke the service by using a binding protocol such as SOAP over HTTP or EJB™.

Figure A-6 on page 875 shows how IBM Information Server information services participate in the SOA Reference Architecture. Briefly:

- An information service (blue dots) can access content systems and data systems, while other (non IBM Information Server) services (pink dots) can access applications and registry services. Applications will most likely access data or content using “proprietary” APIs.
- Information services can be invoked by other (non IBM Information Server) services.
- Business processes can invoke information services and other (non IBM Information Server) services.
- Service consumers can invoke information services, business processes, or other (non IBM Information Server) services directly or indirectly.

The Enterprise Service Bus (ESB) layer enables the integration of services through the introduction of a reliable set of capabilities such as intelligent routing, protocol mediation, and other transformation mechanisms. An ESB provides a location independent mechanism for integration.

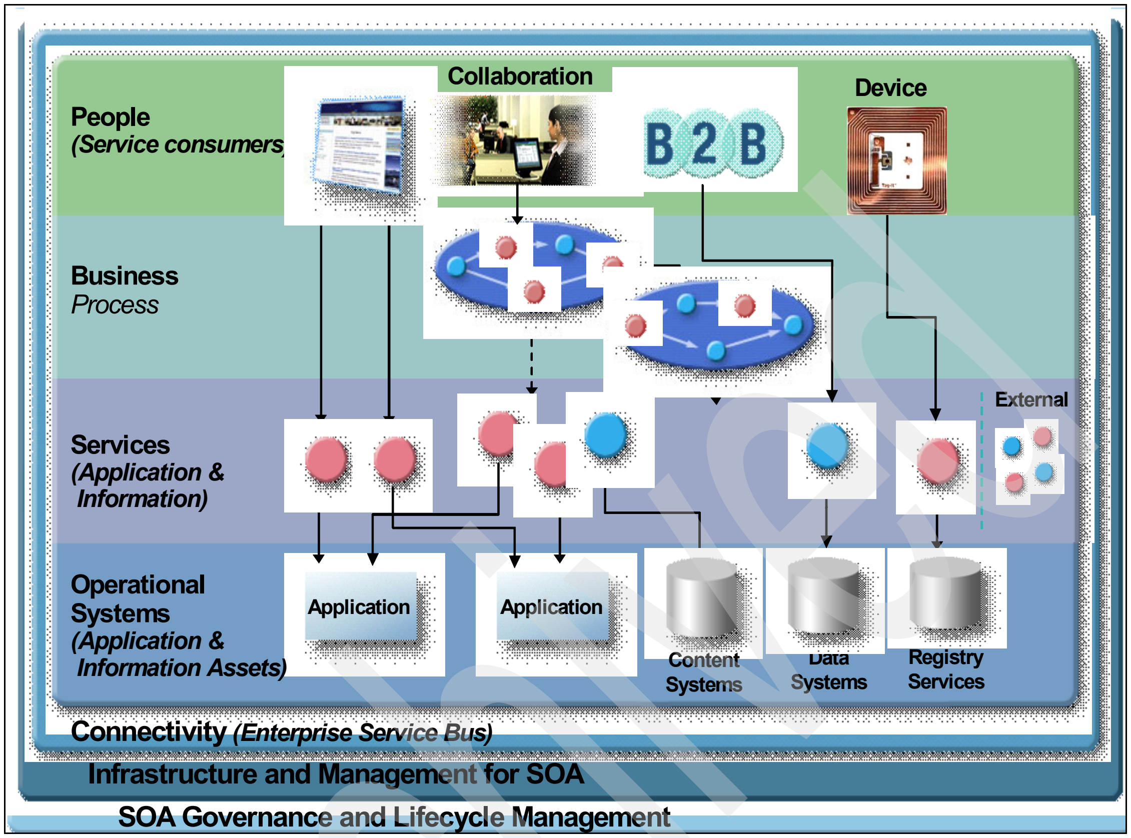


Figure A-6 Information Services in the SOA Reference Architecture

A service-ready data integration job accepts requests from client applications, mapping request data to:

- Input rows and passing them to the underlying jobs in the case of DataStage and QualityStage jobs.
- Input parameters that are executed against a federated database in the case of a federated query or stored procedure.

A job instance can include database access (federated queries), transformations (DataStage jobs), data standardization and matching (QualityStage jobs), and other data integration tasks (database stored procedures) that are supplied by IBM Information Server.

The design of a real-time job determines whether it is always running or runs once to completion. All jobs that are exposed as services process requests on a 24-hour basis.

The SOA infrastructure supports three job topologies for different load and work style requirements, which relates specifically to DataStage and QualityStage jobs, and not to federated queries or stored procedures:

- Batch jobs

  This topology uses new or existing batch jobs that are exposed as services. A batch job starts on demand. Each service request starts one instance of the job that runs to completion. This job typically initiates a batch process from a real-time process that does not need direct feedback on the results. This topology is tailored for processing bulk data sets and is capable of accepting job parameters as input arguments.

- Batch jobs with a Service Output stage

  This topology uses an existing batch job and adds an output stage. The Service Output stage is the exit point from the job, returning one or more rows to the client application as a service response. These jobs typically initiate a batch process from a real-time process that requires feedback or data from the results. This topology is designed to process large data sets and can accept job parameters as input arguments.

- Jobs with a Service Input stage and Service Output stage

  This topology uses both a Service Input stage and a Service Output stage. The Service Input stage is the entry point to a job, accepting one or more rows during a service request. These jobs are always running. This topology is typically used to process high volumes of smaller transactions where response time is important. It is tailored to process many small requests rather than a few large requests.

Current restriction: Client applications can only access IBM Information Server services in synchronous mode, where the model requires feedback to be received for any request made before the client application can proceed to its next course of action. IBM Information Server services that are long running tasks (batch jobs and batch jobs with service output stage topologies) or tasks where no feedback is returned to the requestor (batch jobs topology) need special handling if the client application is to avoid "waiting." Such topology jobs must be redesigned to return feedback to the requestor as soon as the request has been processed, and the client application will have to be designed to check on the status of the job at some later point in time. An upcoming release will provide asynchronous support with JMS binding.

A.3 Configuration flow

This section describes the processing flow involved when configuring a service.

Multiple steps are involved in configuring a service before it can become the target of a client application invocation.

There is a hierarchy of containers when defining an information service—**Project** → **Application** → **Service** → **Operation**. This is reflected in the main steps in creating an SOA service using IBM Information Server shown in Figure A-7.

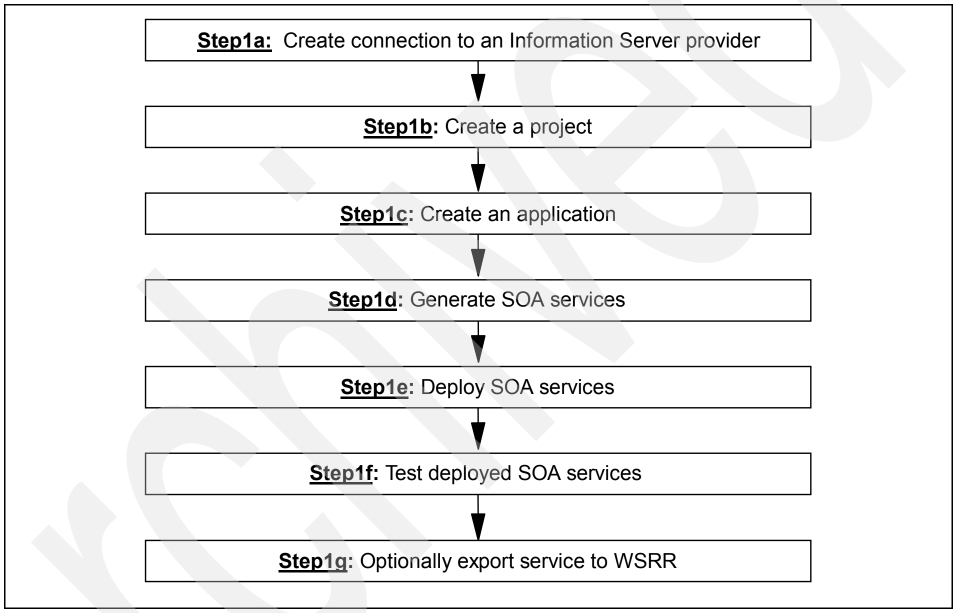

Step1a: Create connection to an Information Server provider
Step1b: Create a project
Step1c: Create an application
Step1d: Generate SOA services
Step1e: Deploy SOA services
Step1f: Test deployed SOA services
Step1g: Optionally export service to WSRR

Figure A-7 Steps in creating SOA services

A.3.1 Step 1a: Create connection to an Information Server provider

An information provider is both the server that contains functions that you can expose as services and the functions themselves, such as WebSphere DataStage and WebSphere QualityStage jobs, database stored procedures, or federated SQL queries.

Before an SOA service can be generated for a function, the information provider must be enabled using WebSphere Information Services Director.

There are two types of Information Server providers—a “DataStage and QualityStage” type for DataStage and QualityStage jobs, and a “DB2 or Federation Server” type for database stored procedures and federated queries.

A.3.2 Step 1b: Create a project

A project is a collaborative environment that you use to design applications, services, and operations. All project information that you create is saved in the common metadata repository so that it can easily be shared among other IBM Information Server components. You can export a project to back up your work or share work with other IBM Information Server users. The export file includes applications, services, operations, and binding information.

Therefore, you must create a project first.

A.3.3 Step 1c: Create an application

An application is a container for a set of services and operations. An application contains one or more services that you want to deploy together as an Enterprise Archive (EAR) file on an application server.

All design-time activity occurs in the context of applications:

- Creating services and operations
- Describing how message payloads and transport protocols are used to expose a service
- Attaching a reference provider, such as a WebSphere DataStage job or an SQL query, to an operation

You can also export services from an application before it is deployed and import the services into another application.

Therefore, you must create an application in the project created.

A.3.4 Step 1d: Generate SOA services

An information service exposes results from processing by information providers such as DataStage servers and federated servers. A deployed service runs on an application server and processes requests from service client applications.

An information service is a collection of operations that are selected from jobs, federated queries, or other information providers. You can group operations in the same information service or design them in separate services. You create an information service for a set of operations that you want to deploy together. You also specify the bindings (SOAP over HTTP or EJB) for the service.

As mentioned earlier, an information service is associated with a particular application.

A.3.5 Step 1e: Deploy SOA services

After the information service is generated, it must be deployed. You deploy an application on WebSphere Application Server to enable the information services that are contained in the application to receive service requests. You can exclude one or more services, bindings, and operations from the deployment, change runtime properties such as minimum number of job instances, or, for WebSphere DataStage jobs, set constant values for job parameters. WebSphere Information Services Director deploys the Enterprise Archive (EAR) file on the application server.

A.3.6 Step 1f: Test deployed SOA services

After deployment, we strongly recommend that you test the deployed information service before making it available to client applications.

A.3.7 Step 1g: Optionally export service to IBM WebSphere Service Registry Repository

Note: WebSphere Service Registry and Repository is not a pre-requisite for IBM Information Server, nor is it mandatory for SOA. If your organization has implemented a WebSphere Server Registry and Repository, you can choose to export the IBM Information Server service you generated to it.

The WebSphere Server Registry and Repository is a separate entity from IBM Information Server that can serve as the master metadata repository (not related in any way to the IBM Information Server metadata repository mentioned earlier) for service descriptions. As the integration point for service metadata, WebSphere Server Registry and Repository establishes a central point for finding and managing service metadata acquired from a number of sources, including service application deployments and other service metadata and endpoint registries and repositories, such as UDDI. It is where service metadata that is scattered across an enterprise is brought together to provide a single, comprehensive description of a service. After that happens, visibility is controlled, versions are managed, proposed changes are analyzed and communicated, usage is monitored and other parts of the SOA foundation can access service metadata with the confidence that they have found the copy of record.

In this context, WebSphere Server Registry and Repository handles the metadata management aspects of operational services and provides the system of record of these metadata artifacts—the place where anybody looking for a catalog of all services deployed in or used by the enterprise would go first. The WebSphere Server Registry and Repository provides registry functions supporting publication of metadata about services, their capabilities, requirements, and semantics of services that enable service consumers to find services or to analyze their relationships.

A.4 Runtime flow

This section provides an overview of the runtime flow associated with processing an invocation of an IBM Information Server service by a client application.

A brief overview of the service artifacts is covered before describing the flow of a request through the system.

A.4.1 Service artifacts

Every IBM Information Server service generated by WebSphere Information Services Director is generated as an EJB session bean (Service Session Bean in Figure A-8 on page 881) regardless of whether the function is a DataStage job, QualityStage job, database stored procedure, or federated query. After the service session bean has been generated, additional artifacts are created depending upon whether SOAP over HTTP or EJB binding is requested for the service:

- With SOAP over HTTP binding, a router servlet and facade session bean is generated. The servlet invokes the facade session bean that in turn invokes the service session bean.
- With EJB binding, a facade session bean is generated that invokes the service session bean.

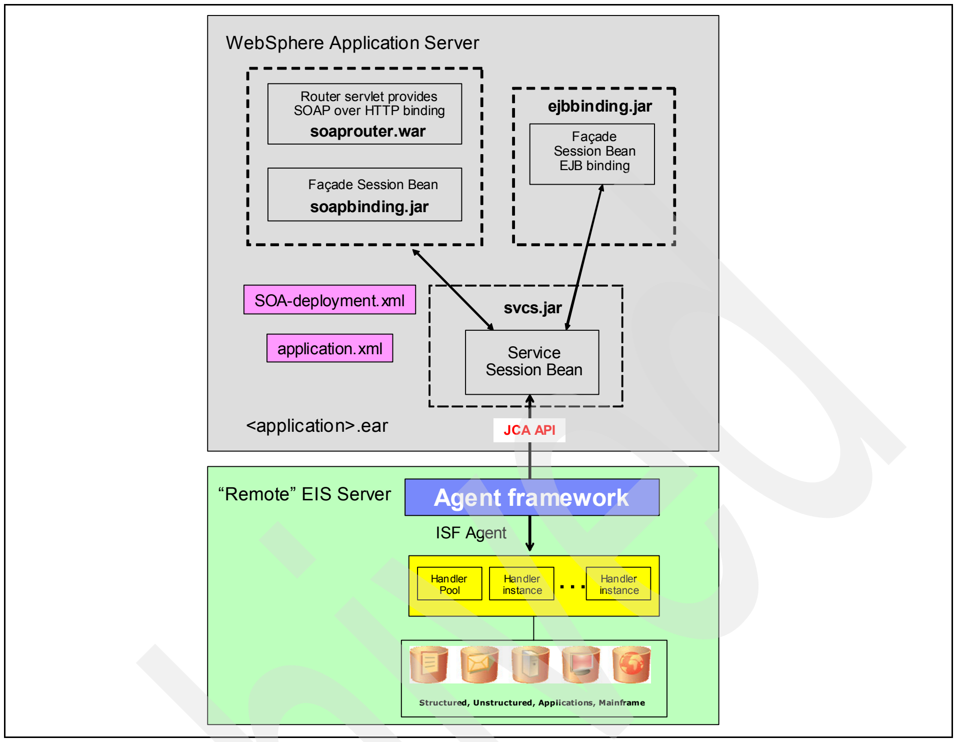


Figure A-8 Partial contents of IBM Information Server application EAR file

Note: The .jar and .war files shown in Example A-1 through Example A-4 relate to the creation of an information service named AccountService with SOAP over HTTP and EJB bindings in the BrokerageApp application within the A2ZProject.

The service session bean is packaged into an svcs.jar file along with other files as shown in Example A-1.

Example: A-1 The svcs.jar file contents

```
AccountService.class
AccountServiceBean.class
AccountServiceForWSDL.class
AccountServiceHome.class
AccountServiceRemote.class
ejb-jar.xml
ibm-ejb-jar-bnd.xmi
Manifest.mf
Response.class
RTIServiceEJBBase.class
```

The router servlet is packaged into a soaprouter.war file along with other files as shown in Example A-2.

Example: A-2 The soaprouter.war file contents

```
ibm-web-bnd.xml
ibm-web-ext.xml
Manifest.mf
web.xml
```

The facade session bean is packaged into a soapbinding.jar file along with other files as shown in Example A-3.

Example: A-3 The soapbinding.jar file contents

```
AccountService.wsdl
AccountService_mapping.xml
AccountServiceSOAPBindingBean.class
ejb-jar.xml
ibm-ejb-jar-bnd.xmi
ibm-webservices-bnd.xmi
ibm-webservices-ext.xmi
Manifest.mf
webservices.xml
```

The facade session bean is packaged into an ejbbinding.jar file along with other files as shown in Example A-4.

Example: A-4 The ejbbinding.jar file contents

```
AccountServiceBean.class
ejb-jar.xml
ibm-ejb-jar-bnd.xmi
Manifest.mf
```

All these .jar files along with a soa-deployment.xml descriptor (shown in Example A-5 on page 883 as including both SOAP over HTTP and EJB bindings), and application.xml (Example A-7 on page 884) descriptor are packaged into A2ZBrokerageApp.ear file (shown in Example A-6 on page 884) that eventually gets deployed on the WebSphere Application Server (associated with IBM Information Server) by WebSphere Information Services Director.

Important: We strongly recommend that you do *not* modify the various descriptors in the IBM Information Server <application>.ear file that is generated by WebSphere Information Services Director and attempt to deploy it in WebSphere Application Server using other tools. The results can be unpredictable.

Example: A-5 The soa-deployment.xml descriptor

```
<?xml version="1.0" ?>
- <soa-descriptor name="A2ZBrokerageApp">
- <service-descriptor name="AccountService" type="value">
- <description>
- <![CDATA[ Service for opening accounts
  ]]>
  </description>
- <entry-point>
  <home>com.ibm.isd.A2ZBrokerageApp.AccountService.server.AccountServiceHome</home>
  <remote>com.ibm.isd.A2ZBrokerageApp.AccountService.server.AccountServiceRemote</remote>
  <ejb-class>com.ibm.isd.A2ZBrokerageApp.AccountService.server.impl.AccountServiceBean</ejb-class>
  <business>com.ibm.isd.A2ZBrokerageApp.AccountService.AccountService</business>
  </entry-point>
- <j2ee-descriptor reference="true">
  <jndi-name>ascential/rti/A2ZBrokerageApp/AccountService</jndi-name>
  <ejb-name>com.ibm.isd.A2ZBrokerageApp.AccountService.AccountService</ejb-name>
  <bean-type value="stateless" />
  </j2ee-descriptor>
  <category>/RTI</category>
  <initialization jndiName="" priority="1" />
  <allowable-binding>EJB</allowable-binding>
- <binding name="EJB">
  <property name="BeanDescription" value="" />
  <property name="JNDIName" value="ejb/A2ZBrokerageApp/AccountService" />
  <property name="Package" value="com.ibm.isd.A2ZBrokerageApp.AccountService.ejb" />
  </binding>
  <allowable-binding>SOAPHttp</allowable-binding>
- <binding name="SOAPHttp">
  <property name="Package" value="com.ibm.isd.A2ZBrokerageApp.AccountService" />
  <property name="TargetNameSpace"
value="http://AccountService.A2ZBrokerageApp.isd.ibm.com/soapoverhttp/" />
  <property name="UriRoot" value="wisd" />
  <property name="WSDLClass"
value="com.ibm.isd.A2ZBrokerageApp.AccountService.AccountServiceForWSDL" />
  <property name="SOAPStyle" value="DOCLIT" />
  <property name="SOAPAction" value="NONE" />
  </binding>
  </service-descriptor>
  </soa-descriptor>
```

Example: A-6   BrokerageApp.ear file

```
A2ZBrokerageApp_client.jar
application.xml
ejbbinding.jar
Manifest.mf
soa-deployment.xml
soapbinding.jar
soaprouter.war
svcs.jar
```

Example: A-7   Information Server application.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
- <application xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" id="Application_ID"
version="1.4" xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/application_1_4.xsd">
  <display-name>A2ZBrokerageApp</display-name>
- <module>
  <ejb>svcs.jar</ejb>
  </module>
- <module>
- <web>
  <web-uri>soaprouter.war</web-uri>
  <context-root>/wisd/A2ZBrokerageApp</context-root>
  </web>
  </module>
- <module>
  <ejb>soapbinding.jar</ejb>
  </module>
- <module>
  <ejb>ejbbinding.jar</ejb>
  </module>
  </application>
```

A.4.2 Flow of a request

A request for an IBM Information Server service can be invoked using SOAP over HTTP binding or EJB binding.

With SOAP over HTTP binding, an incoming request from a remote client is de-serialized by the WebSphere Application Server SOAP stack (associated with IBM Information Server) and passed to the IBM Information Server and Information Services Framework as follows:

1. Invokes the router servlet with the interface parameters.
2. The router servlet then invokes the facade session bean
3. Facade session bean invokes the service session bean using EJB binding.
4. The service session bean connects to the Information Services Framework (ISF) Agent[3] using the Agent framework using a J2EE Connector Architecture[4] (JCA) API to send a request to the back-end system and obtain a response.

> **Note:** As shown in Figure A-8 on page 881, the Agent Framework, ISF Agent and back-end data sources reside on a logically "remote" EIS server, which means that these components can be located physically on a separate server or co-located on the same server as the IBM Information Server.

The ISF Agent provides a framework for sending requests from the IBM Information Server to remote (ISF) clients (where the ISF Agent is located) without the need for a full J2EE Application Server at each client location. The ISF Agent utilizes a plug-in architecture to allow different types of requests to

3 There is one ISF Agent associated with a DataStage server or Federation Server. If both DataStage server and Federation Server are installed on a server, only a single ISF Agent is installed on that server. An ISF Agent can be configured to access only one IBM Information Server. This architecture allows DataStage and Federation servers to be installed on servers distinct from where IBM Information Server is installed. A discussion of configuring IBM Information Server with multiple ISF Agents is beyond the scope of this publication.

4 The J2EE Connector Architecture specifies a standard architecture for accessing resources in diverse Enterprise Information Systems (EIS) such as ERP systems, mainframe transaction processing systems, existing applications and non-relational database systems. The Connector Architecture defines a common interface (using the JCA API) between application servers and EIS systems, implemented in EIS specific resource adapters. A resource adapter is a system library specific to an EIS system such as SAP, and provides connectivity to that EIS via the JCA API. It is somewhat similar to a JDBC™ driver. The interface between the resource adapter and the EIS is typically specific to the underlying EIS. A Connector Architecture compliant resource adapter works with any J2EE server. A single resource adapter is provided with IBM Information Server that handles both the DataStage/QualityStage data source, and database stored procedure/federated query.

be passed from the IBM Information Server. The ISF Agent framework also provides load balancing and pooling of resources.

The code that processes a request in the ISF Agent is called a Handler. Each ISF Agent can be configured to support multiple different Handlers at the same time. Currently, there are two handlers:

– A DataStage/QualityStage handler
– A database stored procedure and federated query handler.

The ISF Agent framework takes care of routing requests and returning any responses. The clients (service session bean in our case) of the ISF Agent use the Java Connection Architecture (JCA) to send data. This consists of obtaining a Connection in much the same way that a JDBC Connection is obtained.

5. As each request arrives, the ISF Agent framework selects an instance of the requested Handler to process it. It does that by requesting a Handler instance from the handler pool. The handler pool in turn can decide to create a new instance or reuse an existing instance.
6. The Handler instance then processes the request and returns a response.

With EJB binding, an application such as a servlet or JSP™ invoke the facade session bean directly which invokes the service session bean. Thereafter, the processing is be identical to that of SOAP over HTTP.

B

Code and scripts used in the financial services business scenario

In this appendix we document some of the code and scripts that are used in the migration and data integration business scenarios.

B.1 Introduction

This appendix documents some of the code and scripts used in the migration and data integration business scenarios. as follows:

- Example B-1 on page 888 shows the DDL for creating the tables in the Northern California Bank data model.
- Example B-2 on page 897 and Example B-3 on page 903 show the DDL for creating the tables in the North American Bank data model and VSAM file definition respectively.
- Example B-4 on page 904 shows the DDL for creating the tables in the CRM data model.

Example: B-1 Fields in the tables in the Northern California Bank data model

```
-- This CLP file was created using DB2LOOK Version 9.1
-- Timestamp: Tue Sep 11 11:32:35 CDT 2007
-- Database Name: REDBANK
-- Database Manager Version: DB2/AIX64 Version 9.1.3
-- Database Codepage: 1252
-- Database Collating Sequence is: UNIQUE
CONNECT TO REDBANK;
------------------------------------------------
-- DDL Statements for table "DB2INST1"."ACCTYPE"
------------------------------------------------
CREATE TABLE "DB2INST1"."ACCTYPE"  (
        "TYPE" CHAR(2) NOT NULL ,
        "DESCRIP" CHAR(50) NOT NULL ,
        "INTR" INTEGER ,
        "FEE" CHAR(20) ,
        "FEEFRQ" CHAR(1) ,
        "UPDATED" TIMESTAMP NOT NULL WITH DEFAULT  ,
        "BY" CHAR(8) NOT NULL ,
        "CURRENCY" CHAR(3) NOT NULL WITH DEFAULT '' )
       IN "USERSPACE1" ;
-- DDL Statements for primary key on Table "DB2INST1"."ACCTYPE"
ALTER TABLE "DB2INST1"."ACCTYPE"
   ADD CONSTRAINT "ACCTYPE_PK" PRIMARY KEY
      ("TYPE");
------------------------------------------------
-- DDL Statements for table "DB2INST1"."CURRENCY"
------------------------------------------------

CREATE TABLE "DB2INST1"."CURRENCY"  (
        "CURRENCY" CHAR(3) NOT NULL ,
        "CTRY" VARCHAR(30) NOT NULL ,
        "NAME" VARCHAR(30) NOT NULL ,
        "UPDATED" TIMESTAMP NOT NULL WITH DEFAULT  ,
        "BY" CHAR(8) NOT NULL WITH DEFAULT 'USER5555' )
       IN "USERSPACE1" ;
------------------------------------------------
-- DDL Statements for table "DB2INST1"."COUNTRY"
------------------------------------------------
CREATE TABLE "DB2INST1"."COUNTRY"  (
        "COUNTRY" VARCHAR(30) NOT NULL ,
        "CTRY2" CHAR(2) NOT NULL ,
        "CTRY3" CHAR(3) NOT NULL ,
        "CTRYN" CHAR(3) NOT NULL ,
        "UPDATED" TIMESTAMP NOT NULL WITH DEFAULT  ,
        "BY" CHAR(8) NOT NULL WITH DEFAULT 'USER5555' )
       IN "USERSPACE1" ;
------------------------------------------------
-- DDL Statements for table "DB2INST1"."LANGUAGES"
------------------------------------------------
CREATE TABLE "DB2INST1"."LANGUAGES"  (
        "LAN3" CHAR(3) NOT NULL ,
        "LANGUAGE" VARCHAR(30) NOT NULL ,
        "UPDATED" TIMESTAMP NOT NULL WITH DEFAULT  ,
        "BY" CHAR(8) NOT NULL WITH DEFAULT 'USER5555' )
       IN "USERSPACE1" ;
------------------------------------------------
-- DDL Statements for table "DB2INST1"."BACCOUNT"
------------------------------------------------
CREATE TABLE "DB2INST1"."BACCOUNT"  (
        "ID" INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY (
          START WITH +21001500
          INCREMENT BY +1
          MINVALUE +21001500
          MAXVALUE +2147483647
          NO CYCLE
          CACHE 20
           ) ,
        "TYPE" CHAR(2) NOT NULL ,
        "UPDATED" TIMESTAMP NOT NULL WITH DEFAULT  ,
        "BY" CHAR(8) NOT NULL )
       IN "USERSPACE1" ;
ALTER TABLE "DB2INST1"."BACCOUNT" ALTER COLUMN "ID" RESTART WITH 
21001539;
------------------------------------------------
-- DDL Statements for table "DB2INST1"."BROKERAGE"
```

```
------------------------------------------------
CREATE TABLE "DB2INST1"."BROKERAGE"  (
        "OWNER" INTEGER NOT NULL ,
        "ACCOUNT" INTEGER NOT NULL ,
        "PORTFOLIO" INTEGER NOT NULL ,
        "UPDATED" TIMESTAMP NOT NULL WITH DEFAULT  ,
        "BY" CHAR(8) NOT NULL )
       IN "USERSPACE1" ;
-- DDL Statements for indexes on Table "DB2INST1"."BROKERAGE"

CREATE INDEX "DB2INST1"."XBROKERAGE" ON "DB2INST1"."BROKERAGE"
      ("OWNER" ASC,
       "ACCOUNT" ASC,
       "PORTFOLIO" ASC)
      PCTFREE 10
ALLOW REVERSE SCANS;
------------------------------------------------
-- DDL Statements for table "DB2INST1"."BRANCH"
------------------------------------------------
CREATE TABLE "DB2INST1"."BRANCH"  (
        "ID" INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY (
          START WITH +12001500
          INCREMENT BY +1
          MINVALUE +12001500
          MAXVALUE +2147483647
          NO CYCLE
          CACHE 20
           ) ,
        "NAME" CHAR(50) NOT NULL ,
        "ADDR1" CHAR(50) NOT NULL ,
        "ADDR2" CHAR(50) ,
        "CITY" CHAR(30) NOT NULL ,
        "ZIP" CHAR(10) NOT NULL ,
        "COUNTRY" CHAR(3) ,
        "UPDATED" TIMESTAMP NOT NULL WITH DEFAULT  ,
        "BY" CHAR(8) NOT NULL )
       IN "USERSPACE1" ;
-- DDL Statements for primary key on Table "DB2INST1"."BRANCH"
ALTER TABLE "DB2INST1"."BRANCH"
   ADD CONSTRAINT "BRANCH_PK" PRIMARY KEY
      ("ID");
ALTER TABLE "DB2INST1"."BRANCH" ALTER COLUMN "ID" RESTART WITH
12001559;
------------------------------------------------
-- DDL Statements for table "DB2INST1"."CUSTOMER"
```

```
------------------------------------------------
CREATE TABLE "DB2INST1"."CUSTOMER"  (
        "ID" INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY (
          START WITH +10001500
          INCREMENT BY +1
          MINVALUE +10001500
          MAXVALUE +2147483647
          NO CYCLE
          CACHE 20
           ) ,
        "NAME" CHAR(50) NOT NULL ,
        "ADDR1" CHAR(50) NOT NULL ,
        "ADDR2" CHAR(50) ,
        "CITY" CHAR(30) NOT NULL ,
        "ZIP" CHAR(10) NOT NULL ,
        "COUNTRY" CHAR(30) ,
        "UPDATED" TIMESTAMP NOT NULL WITH DEFAULT  ,
        "BY" CHAR(8) NOT NULL ,
        "BRANCH" INTEGER ,
        "ADVISOR" INTEGER ,
        "HOMEPHONE" CHAR(15) ,
        "CELLPHONE" CHAR(15) ,
        "WORKPHONE" CHAR(15) ,
        "FAX" CHAR(15) ,
        "EMAIL" VARCHAR(50) ,
        "TYPE" CHAR(1) NOT NULL WITH DEFAULT '-' ,
        "CLASS" INTEGER NOT NULL WITH DEFAULT 0 ,
        "GENDER" CHAR(1) NOT NULL WITH DEFAULT '-' ,
        "PREF_LANG" CHAR(3) NOT NULL WITH DEFAULT 'ENG' )
       IN "USERSPACE1" ;
-- DDL Statements for primary key on Table "DB2INST1"."CUSTOMER"
ALTER TABLE "DB2INST1"."CUSTOMER"
   ADD CONSTRAINT "CUSTOMER_PK" PRIMARY KEY
      ("ID");
------------------------------------------------
-- DDL Statements for table "DB2INST1"."EMPLOYEE"
------------------------------------------------
CREATE TABLE "DB2INST1"."EMPLOYEE"  (
        "ID" INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY (
          START WITH +13001500
          INCREMENT BY +1
          MINVALUE +13001500
          MAXVALUE +2147483647
          NO CYCLE
          CACHE 20
```

```
                ) ,
             "NAME" CHAR(50) NOT NULL ,
             "USERID" CHAR(8) ,
             "BRANCH" INTEGER ,
             "UPDATED" TIMESTAMP NOT NULL WITH DEFAULT  ,
             "BY" CHAR(8) NOT NULL )
            IN "USERSPACE1" ;
-- DDL Statements for indexes on Table "DB2INST1"."EMPLOYEE"
CREATE INDEX "DB2INST1"."IEMPLOYEE1" ON "DB2INST1"."EMPLOYEE"
      ("ID" ASC)
      PCTFREE 10
ALLOW REVERSE SCANS;
-- DDL Statements for primary key on Table "DB2INST1"."EMPLOYEE"
ALTER TABLE "DB2INST1"."EMPLOYEE"
   ADD CONSTRAINT "EMPLOYEE_PK" PRIMARY KEY
      ("ID");
ALTER TABLE "DB2INST1"."EMPLOYEE" ALTER COLUMN "ID" RESTART WITH
13001979;
------------------------------------------------
-- DDL Statements for table "DB2INST1"."ACCOUNT"
------------------------------------------------
CREATE TABLE "DB2INST1"."ACCOUNT"  (
             "ID" INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY (
               START WITH +11001500
               INCREMENT BY +1
               MINVALUE +11001500
               MAXVALUE +2147483647
               NO CYCLE
               CACHE 20
                ) ,
             "OWNER" INTEGER NOT NULL ,
             "TYPE" CHAR(2) NOT NULL ,
             "SEC_OWNER" INTEGER ,
             "UPDATED" TIMESTAMP NOT NULL WITH DEFAULT  ,
             "BY" CHAR(8) NOT NULL ,
             "CURRENCY" CHAR(3) )
            IN "USERSPACE1" ;
-- DDL Statements for primary key on Table "DB2INST1"."ACCOUNT"
ALTER TABLE "DB2INST1"."ACCOUNT"
   ADD CONSTRAINT "ACCOUNT_PK" PRIMARY KEY
      ("ID");
-- DDL Statements for indexes on Table "DB2INST1"."ACCOUNT"
CREATE INDEX "DB2INST1"."IACCOUNT2" ON "DB2INST1"."ACCOUNT"
      ("ID" ASC,
       "OWNER" ASC,
       "TYPE" ASC)
      PCTFREE 10
ALLOW REVERSE SCANS;
ALTER TABLE "DB2INST1"."ACCOUNT" ALTER COLUMN "ID" RESTART WITH
11027739;
------------------------------------------------
-- DDL Statements for table "DB2INST1"."COLLATERAL"
------------------------------------------------
CREATE TABLE "DB2INST1"."COLLATERAL"  (
        "ACCOUNT" INTEGER NOT NULL ,
        "TYPE" CHAR(2) NOT NULL ,
        "STATUS" CHAR(1) NOT NULL ,
        "EST_VAL" CHAR(20) NOT NULL ,
        "DESC" VARCHAR(200) NOT NULL ,
        "UPDATED" TIMESTAMP NOT NULL WITH DEFAULT  ,
        "BY" CHAR(8) NOT NULL )
       IN "USERSPACE1" ;
-- DDL Statements for primary key on Table "DB2INST1"."COLLATERAL"
ALTER TABLE "DB2INST1"."COLLATERAL"
   ADD CONSTRAINT "COLLATERAL_PK" PRIMARY KEY
      ("ACCOUNT",
       "UPDATED");
------------------------------------------------
-- DDL Statements for table "DB2INST1"."TRANSACTION"
------------------------------------------------
CREATE TABLE "DB2INST1"."TRANSACTION"  (
        "ACCOUNT" INTEGER NOT NULL ,
        "DESCR" CHAR(50) NOT NULL ,
        "CODE" CHAR(1) NOT NULL ,
        "CHANGE" CHAR(20) NOT NULL ,
        "BALANCE" CHAR(20) NOT NULL ,
        "UPDATED" TIMESTAMP NOT NULL WITH DEFAULT  ,
        "BY" CHAR(8) NOT NULL )
       IN "USERSPACE1" ;
-- DDL Statements for primary key on Table "DB2INST1"."TRANSACTION"
ALTER TABLE "DB2INST1"."TRANSACTION"
   ADD CONSTRAINT "TRANSACTION_PK" PRIMARY KEY
      ("ACCOUNT",
       "UPDATED");
------------------------------------------------
-- DDL Statements for table "DB2INST1"."BCUSTOMER"
------------------------------------------------
CREATE TABLE "DB2INST1"."BCUSTOMER"  (
        "ID" INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY (
          START WITH +20001500
          INCREMENT BY +1
          MINVALUE +20001500
          MAXVALUE +2147483647
          NO CYCLE
          CACHE 20
           ) ,
        "UPDATED" TIMESTAMP NOT NULL WITH DEFAULT CURRENT TIMESTAMP ,
        "BY" CHAR(8) NOT NULL ,
        "BRANCH" INTEGER ,
        "ADVISOR" INTEGER ,
        "NAME" VARCHAR(40) NOT NULL ,
        "ADDR1" VARCHAR(40) NOT NULL ,
        "ADDR2" VARCHAR(40) ,
        "CITY" VARCHAR(30) NOT NULL ,
        "ZIP" CHAR(10) NOT NULL ,
        "COUNTRY" VARCHAR(30) ,
        "EMAIL" VARCHAR(50) ,
        "BANKID" INTEGER )
       IN "USERSPACE1" ;
-- DDL Statements for primary key on Table "DB2INST1"."BCUSTOMER"
ALTER TABLE "DB2INST1"."BCUSTOMER"
   ADD CONSTRAINT "BCUSTOMER_PK" PRIMARY KEY
      ("ID");
-- DDL Statements for indexes on Table "DB2INST1"."BCUSTOMER"
CREATE INDEX "DB2INST1"."BCUSTOMER_PK" ON "DB2INST1"."BCUSTOMER"
      ("ID" ASC,
       "UPDATED" ASC)
      ALLOW REVERSE SCANS;
------------------------------------------------
-- DDL Statements for table "DB2INST1"."PORTFOLIO"
------------------------------------------------
CREATE TABLE "DB2INST1"."PORTFOLIO"  (
        "ID" INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY (
          START WITH +22001500
          INCREMENT BY +1
          MINVALUE +22001500
          MAXVALUE +2147483647
          NO CYCLE
          CACHE 20
           ) ,
        "NAME" VARCHAR(40) NOT NULL ,
        "SYMBOL" CHAR(8) NOT NULL ,
        "ORDERED" DATE NOT NULL ,
        "PURCHASED" DATE ,
        "SELL_BY_DATE" DATE ,
        "SELL_BY_PRICE" CHAR(10) ,
        "SIZE" CHAR(20) NOT NULL ,
        "QUANTITY" CHAR(20) NOT NULL ,
        "PRICE" CHAR(20) ,
        "UPDATED" TIMESTAMP NOT NULL WITH DEFAULT  ,
        "BY" CHAR(8) ,
        "CURRENCY" CHAR(3) )
       IN "USERSPACE1" ;
-- DDL Statements for indexes on Table "DB2INST1"."PORTFOLIO"
CREATE INDEX "DB2INST1"."XPORTFOLIO" ON "DB2INST1"."PORTFOLIO"
      ("ID" ASC,
       "UPDATED" ASC)
      PCTFREE 10
ALLOW REVERSE SCANS;
ALTER TABLE "DB2INST1"."PORTFOLIO" ALTER COLUMN "ID" RESTART WITH
22028339;

------------------------------------------------
-- DDL Statements for table "DB2INST1"."LOAN"
------------------------------------------------
CREATE TABLE "DB2INST1"."LOAN"  (
        "LOAN_ID" INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY (
          START WITH +1
          INCREMENT BY +1
          MINVALUE +1
          MAXVALUE +2147483647
          NO CYCLE
          CACHE 20
           ) ,
        "ACCOUNT_ID" INTEGER NOT NULL ,
        "DESCRIPTION" CHAR(50) ,
        "INTEREST_RATE" CHAR(20) ,
        "INITIAL_LOAN_VALUE" CHAR(20) ,
        "OPENING_FEE" CHAR(20) ,
        "LATE_FEE" CHAR(20) ,
        "LATE_INTEREST_RATE" CHAR(20) ,
        "BALANCE" CHAR(20) )
       IN "USERSPACE1" ;
-- DDL Statements for primary key on Table "DB2INST1"."LOAN"
ALTER TABLE "DB2INST1"."LOAN"
   ADD PRIMARY KEY
      ("LOAN_ID");
-- DDL Statements for foreign keys on Table "DB2INST1"."CUSTOMER"
ALTER TABLE "DB2INST1"."CUSTOMER"
   ADD CONSTRAINT "ADVISOR_FK" FOREIGN KEY
```

```
      ("ADVISOR")
   REFERENCES "DB2INST1"."EMPLOYEE"
      ("ID")
   ON DELETE NO ACTION
   ON UPDATE NO ACTION;
ALTER TABLE "DB2INST1"."CUSTOMER"
   ADD CONSTRAINT "BRANCH_FK" FOREIGN KEY
      ("BRANCH")
   REFERENCES "DB2INST1"."BRANCH"
      ("ID")
   ON DELETE NO ACTION
   ON UPDATE NO ACTION;
-- DDL Statements for foreign keys on Table "DB2INST1"."EMPLOYEE"
ALTER TABLE "DB2INST1"."EMPLOYEE"
   ADD CONSTRAINT "CC1185467343863" FOREIGN KEY
      ("BRANCH")
   REFERENCES "DB2INST1"."BRANCH"
      ("ID")
   ON DELETE NO ACTION
   ON UPDATE NO ACTION;
-- DDL Statements for foreign keys on Table "DB2INST1"."ACCOUNT"
ALTER TABLE "DB2INST1"."ACCOUNT"
   ADD CONSTRAINT "ACCOUNTOWNER_FK" FOREIGN KEY
      ("OWNER")
   REFERENCES "DB2INST1"."CUSTOMER"
      ("ID")
   ON DELETE NO ACTION
   ON UPDATE NO ACTION;
ALTER TABLE "DB2INST1"."ACCOUNT"
   ADD CONSTRAINT "ACCOUNTTYPE_FK" FOREIGN KEY
      ("TYPE")
   REFERENCES "DB2INST1"."ACCTYPE"
      ("TYPE")
   ON DELETE NO ACTION
   ON UPDATE NO ACTION;
-- DDL Statements for foreign keys on Table "DB2INST1"."COLLATERAL"
ALTER TABLE "DB2INST1"."COLLATERAL"
   ADD CONSTRAINT "COLLATERAL_FK" FOREIGN KEY
      ("ACCOUNT")
   REFERENCES "DB2INST1"."ACCOUNT"
      ("ID")
   ON DELETE NO ACTION
   ON UPDATE NO ACTION;
-- DDL Statements for foreign keys on Table "DB2INST1"."TRANSACTION"
ALTER TABLE "DB2INST1"."TRANSACTION"
```

```
   ADD CONSTRAINT "ACCOUNT_FK" FOREIGN KEY
      ("ACCOUNT")
   REFERENCES "DB2INST1"."ACCOUNT"
      ("ID")
   ON DELETE NO ACTION
   ON UPDATE NO ACTION;
-- DDL Statements for foreign keys on Table "DB2INST1"."LOAN"
ALTER TABLE "DB2INST1"."LOAN"
   ADD CONSTRAINT "SQL070911112510600" FOREIGN KEY
      ("ACCOUNT_ID")
   REFERENCES "DB2INST1"."ACCOUNT"
      ("ID")
   ON DELETE NO ACTION
   ON UPDATE NO ACTION;
```

Example: B-2 Fields in the tables in the North American Bank data model

```
-- This CLP file was created using DB2LOOK Version 9.1
-- Timestamp: 13-09-2007 14:27:14
-- Database Name: DB8A
-- Database Manager Version: DB2 Version 8.1.5
-- Database Codepage: 1208
------------------------------------------------
-- DDL Statements for table "SG247508"."ACCOUNT"
------------------------------------------------
CREATE TABLE "SG247508"."ACCOUNT"
      (
       "ACCOUNT_ID"  INTEGER NOT NULL ,
       "BRANCH_ID"  INTEGER NOT NULL ,
       "ACTIVE_IND"  CHAR(1) ,
       "BALANCE"  DECIMAL(9,2) ,
       "MIN_AMOUNT"  DECIMAL(9,2) ,
       "OVERDRAF"  DECIMAL(9,2) ,
       "OVERDRAF_LIMIT"  DECIMAL(9,2) ,
       "OVERDRAF_RATE"  DECIMAL(8,5) ,
       "OVERDRAF_FEE"  DECIMAL(9,2) ,
       "TYPE_IND"  CHAR(1)
      );
CREATE UNIQUE INDEX "NALUR1"."PKACCOUNT" ON "SG247508"."ACCOUNT"
      ( "ACCOUNT_ID" ASC);
------------------------------------------------
-- DDL Statements for table "SG247508"."BRANCH"
------------------------------------------------
```

```
CREATE TABLE "SG247508"."BRANCH"
      (
       "BRANCH_ID"  INTEGER NOT NULL ,
       "BRANCH_DESCRIPTION"  CHAR(18) ,
       "WORK_ADDRESS"  CHAR(18) ,
       "WORK_ZIP"  CHAR(18)
      );
CREATE UNIQUE INDEX "NALUR1"."PKBRANCH" ON "SG247508"."BRANCH"
      ( "BRANCH_ID" ASC);
------------------------------------------------
-- DDL Statements for table "SG247508"."CARD"
------------------------------------------------
CREATE TABLE "SG247508"."CARD"
      (
       "CARD_ID"  CHAR(16)  NOT NULL ,
       "PIN"  CHAR(4) ,
       "EXPIRE_DT"  TIMESTAMP ,
       "CARD_TYPE_CD"  CHAR(2)  NOT NULL ,
       "LEVEL_CD"  CHAR(2)  NOT NULL ,
       "CUSTOMER_ID"  INTEGER NOT NULL ,
       "CARD_CUST_NAME"  CHAR(18) ,
       "ACCOUNT_ID"  INTEGER NOT NULL ,
       "LIMIT"  DECIMAL(9,2) ,
       "WITHDRAW_LIMIT"  DECIMAL(9,2) ,
       "SECURITY_NUM"  CHAR(4) ,
       "LIMIT_BALANCE"  DECIMAL(9,2) ,
       "LIMIT_W_BALANCE"  DECIMAL(9,2) ,
       "FLAG_IND"  CHAR(1) ,
       "INTL_IND"  CHAR(1) ,
       "AUTOMAT_DEBIT_IND"  CHAR(1) ,
       "REWARDS_IND"  CHAR(1) ,
       "REWARDS_NUM"  VARCHAR(20) ,
       "REWARDS_CD"  CHAR(3)
      );
CREATE UNIQUE INDEX "NALUR1"."PKCARD" ON "SG247508"."CARD"
      ( "CARD_ID" ASC,    "CARD_TYPE_CD" ASC,    "CUSTOMER_ID" ASC,
"ACCOUNT_ID" ASC);
------------------------------------------------
-- DDL Statements for table "SG247508"."CARD_TRANSACTION"
------------------------------------------------
CREATE TABLE "SG247508"."CARD_TRANSACTION"
      (
       "CARD_ID"  CHAR(16)  NOT NULL ,
       "CARD_TYPE_CD"  CHAR(2)  NOT NULL ,
       "CUSTOMER_ID"  INTEGER NOT NULL ,
       "ACCOUNT_ID"  INTEGER NOT NULL ,
       "TRANSACTION_ID"  INTEGER NOT NULL ,
       "DESCRIPTION"  VARCHAR(20) ,
       "TRANSACTION_DT"  TIMESTAMP ,
       "VENDOR_NAME"  VARCHAR(50) ,
       "VENDOR_ID"  INTEGER,
       "INTL_IND"  CHAR(1) ,
       "AMOUNT"  DECIMAL(9,2) ,
       "TRANS_TYPE_CD"  CHAR(2)  NOT NULL ,
       "CUST_REFUSAL_IND"  CHAR(1) ,
       "LOCAL_CURRENCY_AMOUNT"  DECIMAL(9,2) ,
       "EXCHANGE_CURR_USED"  DECIMAL(9,2)
      );
CREATE UNIQUE INDEX "NALUR1"."PKCARDTRANS" ON
"SG247508"."CARD_TRANSACTION"
      ( "CARD_ID" ASC,    "CARD_TYPE_CD" ASC,    "CUSTOMER_ID" ASC,
"ACCOUNT_ID" ASC,    "TRANSACTION_ID" ASC);
------------------------------------------------
-- DDL Statements for table "SG247508"."CARD_TYPE_REF"
------------------------------------------------
CREATE TABLE "SG247508"."CARD_TYPE_REF"
      (
       "CARD_TYPE_CD"  CHAR(2)  NOT NULL ,
       "DESCIPTION"  VARCHAR(20)
      );
CREATE UNIQUE INDEX "NALUR1"."PKCARDREF" ON "SG247508"."CARD_TYPE_REF"
      ( "CARD_TYPE_CD" ASC);
------------------------------------------------
-- DDL Statements for table "SG247508"."CAR_INSURANCE"
------------------------------------------------
CREATE TABLE "SG247508"."CAR_INSURANCE"
      (
       "ACCOUNT_ID"  INTEGER NOT NULL ,
       "INSURANCE_ID"  INTEGER NOT NULL ,
       "CAR_PLATE"  CHAR(10) ,
       "START_DT"  DATE,
       "END_DT"  DATE,
       "CAR_VALUE"  DECIMAL(9,2) ,
       "CLAIM_VALUE"  DECIMAL(9,2) ,
       "FULL_COVERAGE_IND"  CHAR(1) ,
       "THIRD_COVERAGE_LIMIT"  DECIMAL(9,2) ,
       "INSURANCE_COVERAGE"  DECIMAL(9,2) ,
       "INSURANCE_VALUE"  DECIMAL(9,2) ,
       "AUTOMAT_DEBIT_IND"  CHAR(1)
      );
```

```
CREATE UNIQUE INDEX "NALUR1"."PKCARINS" ON "SG247508"."CAR_INSURANCE"
      ( "ACCOUNT_ID" ASC,    "INSURANCE_ID" ASC);
------------------------------------------------
-- DDL Statements for table "SG247508"."CONTACT_INFO"
------------------------------------------------
CREATE TABLE "SG247508"."CONTACT_INFO"
      (
       "CUSTOMER_ID"  INTEGER NOT NULL ,
       "ACCOUNT_ID"  INTEGER NOT NULL ,
       "WORK_PHONE"  CHAR(15) ,
       "CELL_PHONE"  CHAR(15) ,
       "HOME_PHONE"  CHAR(15) ,
       "HOME_ADDRESS"  VARCHAR(50) ,
       "HOME_ZIP"  CHAR(9) ,
       "WORK_ADDRESS"  VARCHAR(50) ,
       "WORK_ZIP"  CHAR(9) ,
       "PREF_LANG"  CHAR(3)  NOT NULL  WITH DEFAULT 'ENG'
      );
CREATE UNIQUE INDEX "NALUR1"."PKCUSTINFO" ON "SG247508"."CONTACT_INFO"
      ( "CUSTOMER_ID" ASC,    "ACCOUNT_ID" ASC);
------------------------------------------------
-- DDL Statements for table "SG247508"."CUSTOMER"
------------------------------------------------
CREATE TABLE "SG247508"."CUSTOMER"
      (
       "CUSTOMER_ID"  INTEGER NOT NULL ,
       "TITLE"  CHAR(3) ,
       "FIRST_NAME"  VARCHAR(20) ,
       "LAST_NAME"  VARCHAR(20) ,
       "GENDER_IND"  CHAR(1) ,
       "USERID"  VARCHAR(8) ,
       "PASSWORD"  VARCHAR(20) ,
       "CHURN_IND"  CHAR(1)  NOT NULL  WITH DEFAULT,
       "LEVEL_CD"  CHAR(2)  NOT NULL  WITH DEFAULT,
       "NICKNAME"  VARCHAR(20) ,
       "CREDIT_SCORE"  CHAR(18) ,
       "NATIONALITY"  VARCHAR(20)
      );

CREATE UNIQUE INDEX "NALUR1"."PKCUSTOMER" ON "SG247508"."CUSTOMER"
      ( "CUSTOMER_ID" ASC);

------------------------------------------------
-- DDL Statements for table "SG247508"."CUST_ACC"
------------------------------------------------
```

```
CREATE TABLE "SG247508"."CUST_ACC"
      (
       "CUSTOMER_ID"  INTEGER NOT NULL ,
       "ACCOUNT_ID"  INTEGER NOT NULL
      );
CREATE INDEX "SG247508"."IXCUSTAC2" ON "SG247508"."CUST_ACC"
      ( "ACCOUNT_ID" ASC);
CREATE INDEX "SG247508"."IXCUSTACC" ON "SG247508"."CUST_ACC"
      ( "CUSTOMER_ID" ASC);
CREATE UNIQUE INDEX "NALUR1"."PKCUST_ACC" ON "SG247508"."CUST_ACC"
      ( "CUSTOMER_ID" ASC,    "ACCOUNT_ID" ASC);
------------------------------------------------
-- DDL Statements for table "SG247508"."DRIVER"
------------------------------------------------
CREATE TABLE "SG247508"."DRIVER"
      (
       "ACCOUNT_ID"  INTEGER NOT NULL ,
       "INSURANCE_ID"  INTEGER NOT NULL ,
       "DRIVER_ID"  INTEGER NOT NULL ,
       "NAME"  VARCHAR(50) ,
       "SSN"  CHAR(11) ,
       "BIRTH_DT"  DATE,
       "GENDER"  CHAR(1) ,
       "START_DRIVING"  DATE,
       "ADDRESS"  VARCHAR(50) ,
       "CITY"  VARCHAR(40) ,
       "STATE"  CHAR(2) ,
       "ZIP"  CHAR(9) ,
       "CORRECTIVE_LENSES_IND"  CHAR(1) ,
       "HAIR_COLOR"  VARCHAR(10) ,
       "HEIGHT"  VARCHAR(10) ,
       "WEIGHT"  VARCHAR(10)
      );
CREATE UNIQUE INDEX "NALUR1"."PKDRIVER" ON "SG247508"."DRIVER"
      ( "ACCOUNT_ID" ASC,    "INSURANCE_ID" ASC,    "DRIVER_ID" ASC);
------------------------------------------------
-- DDL Statements for table "SG247508"."LEVEL_REF"
------------------------------------------------
CREATE TABLE "SG247508"."LEVEL_REF"
      (
       "LEVEL_CD"  CHAR(2)  NOT NULL ,
       "DESCRIPTON"  VARCHAR(20)
      );
------------------------------------------------
-- DDL Statements for table "SG247508"."LOAN"
```

```
------------------------------------------------
CREATE TABLE "SG247508"."LOAN"
      (
       "ACCOUNT_ID"  INTEGER NOT NULL ,
       "LOAN_ID"  INTEGER NOT NULL ,
       "DESCRIPTION"  VARCHAR(20) ,
       "RATES"  DECIMAL(8,5) ,
       "INITIAL_VALUE"  DECIMAL(9,2) ,
       "LATE_FEE"  DECIMAL(9,2) ,
       "LATE_RATE"  DECIMAL(8,5) ,
       "BALANCE"  DECIMAL(9,2) ,
       "AUTOMAT_DEBIT_IND"  CHAR(1)
      );
CREATE UNIQUE INDEX "NALUR1"."PKLOAN" ON "SG247508"."LOAN"
      ( "ACCOUNT_ID" ASC,    "LOAN_ID" ASC);
------------------------------------------------
-- DDL Statements for table "SG247508"."LOAN_TRANSACTION"
------------------------------------------------
CREATE TABLE "SG247508"."LOAN_TRANSACTION"
      (
       "ACCOUNT_ID"  INTEGER NOT NULL ,
       "LOAN_ID"  INTEGER NOT NULL ,
       "TRANSACTION_ID"  INTEGER NOT NULL ,
       "DESCRIPTION"  VARCHAR(20) ,
       "TRANSACTION_DT"  TIMESTAMP ,
       "AMOUNT"  DECIMAL(9,2) ,
       "TRANS_TYPE_CD"  CHAR(2)  NOT NULL
      );

CREATE UNIQUE INDEX "NALUR1"."PKLOAN_TRANS" ON
"SG247508"."LOAN_TRANSACTION"
      ( "ACCOUNT_ID" ASC,    "LOAN_ID" ASC,    "TRANSACTION_ID" ASC);
------------------------------------------------
-- DDL Statements for table "SG247508"."REWARD_REF"
------------------------------------------------
CREATE TABLE "SG247508"."REWARD_REF"
      (
       "REWARDS_CD"  CHAR(3)  NOT NULL ,
       "DESCRIPTION"  VARCHAR(50)
      );
CREATE UNIQUE INDEX "NALUR1"."PKREWARD" ON "SG247508"."REWARD_REF"
      ( "REWARDS_CD" ASC);
------------------------------------------------
-- DDL Statements for table "SG247508"."TRANSACTION"
------------------------------------------------
```

```
CREATE TABLE "SG247508"."TRANSACTION"
      (
       "ACCOUNT_ID"  INTEGER NOT NULL ,
       "TRANSACTION_ID"  INTEGER NOT NULL ,
       "TRANSACTION_DT"  TIMESTAMP ,
       "TRANS_TYPE_CD"  CHAR(2)  NOT NULL ,
       "DESCRIPTION"  VARCHAR(20) ,
       "AMOUNT"  DECIMAL(9,2) ,
       "PAID_TO"  CHAR(18)
      );
CREATE UNIQUE INDEX "NALUR1"."PKTRANSA" ON "SG247508"."TRANSACTION"
      ( "ACCOUNT_ID" ASC,    "TRANSACTION_ID" ASC);
------------------------------------------------
-- DDL Statements for table "SG247508"."TRANSACTION_TYPE_REF"
------------------------------------------------
CREATE TABLE "SG247508"."TRANSACTION_TYPE_REF"
      (
       "TRANS_TYPE_CD"  CHAR(2)  NOT NULL ,
       "DESCRIPTION"  VARCHAR(20)
      );
```

Example: B-3   VSAM file containing EMPLOYEE records

```
CREATE TABLE "CAC"."EMPLOYEE" DBTYPE VSAM
   DS "CAC.VSAM.EMPLOYEE"
   (
   "EMPNAME" SOURCE DEFINITION
      DATAMAP OFFSET 0 LENGTH 21
      DATATYPE C
      USE AS CHAR(21),
   "DEPTNAME" SOURCE DEFINITION
      DATAMAP OFFSET 47 LENGTH 18
      DATATYPE C
      USE AS CHAR(18),
   "EMPNO" SOURCE DEFINITION
      DATAMAP OFFSET 72 LENGTH 8
      DATATYPE C
      USE AS CHAR(8));
```

Example: B-4 Fields in the tables in the CRM data model

```
--ISO code for languages
create table nabncb.iso_language (
       id            char(3) not null,
       name          varchar(30) not null,
       primary key(id)
       );
-- Home address, work address, home phone, call phone, e-mail
create table nabncb.contacttype (
       id            integer not null,
       description   varchar(50) not null,
       primary key(id)
       );
-- Commercial banking, incurance, brokerage
create table nabncb.lineofbusiness (
       id            integer not null,
       description   varchar(50) not null,
       primary key(id)
       );
-- Cross reference between CRM and other core and non-core systems
create table nabncb.custkeyxref (
       crmid            integer not null,
       nabcoreid        integer,
       nabnoncoreid     integer,
       ncbcoreid        integer,
       ncbnoncoreid     integer,
       primary key(crmid)
       );
-- Person, organization
create table nabncb.customertype (
       id            integer not null,
       description   varchar(50) not null,
       primary key(id)
       );
-- Account owner, beneficiary, stakeholder, insurer
create table nabncb.role (
       id            integer not null,
       description   varchar(50) not null,
       primary key(id)
       );
-- Currency, information on collateral for loans etc.
create table nabncb.item (
       id            integer not null,
       description   varchar(50) not null,
       primary key(id)
       );
-- Types of relationships between customers; member of same household,
-- owner of business
create table nabncb.relationtype (
       id            integer not null,
       description   varchar(50) not null,
       primary key(id)
       );
-- Branch information
create table nabncb.branch (
      id        integer not null,
      name      char(50) not null,
      primary key (id)
      );
-- Employee information
create table nabncb.employee (
      id        integer not null,
      name      char(50) not null,
      userid    char(8) not null,
      branch    integer not null,
      business  integer not null,
      primary key (id),
      foreign key (branch) references nabncb.branch (id),
      foreign key (business) references nabncb.lineofbusiness (id)
      );
-- Customer information. Rating from one to five stars, the more the
-- better
create table nabncb.customer (
       id               integer not null,
       prefix           varchar (10) not null,
       firstname        varchar(30) not null,
       middlename       varchar(30),
       lastname         varchar(30) not null,
       gender            char(1),
       nationality      varchar(20) not null,
       "TYPE"                   integer not null,
       preflang         CHAR(3) not null,
       advisor          integer,
       prefcontact      integer not null,
       homeStreet       varchar(30) not null,
       homeCity         varchar(20) not null,
       homeZip          varchar(10),
       homeCountry      varchar (20),
       workStreet       varchar(30) not null,
       workCity        varchar(20) not null,
       workZip         varchar(10),
       workCountry     varchar (20),
       homephone       varchar(15),
       workphone       varchar (20),
       cellphone       varchar(15),
       email           varchar (20),
       rating          char(5) not null,
       nabchkassets            decimal (9,2),
       nabsavassets            decimal (9,2),
       nabloanindicator        char(1) not null,
       nabloanamount           decimal (9,2),
       nabloanbalance          decimal (9,2),
       nabloanrate             decimal (6,3),
       ncbchkassets            decimal (9,2),
       ncbsavassets            decimal (9,2),
       ncbloanindicator        char(1) not null,
       ncbloanamount           decimal (9,2),
       ncbloanbalance          decimal (9,2),
       ncbloanrate             decimal (6,3),
       Brokindicator   char(1) not null,
       Brokassets      decimal (9,2),
       Brokmargin      decimal (9,2),
       CCindicator     char(1) not null,
       CClimit         integer,
       CCbalance                                       decimal (9,2),
       Carindicator    char(1) not null,
       Fullcoverind    char(1) not null,
       Carpremiums           decimal (6,2),
       Carenddate      date,
       primary key(id),
       foreign key (Advisor) references nabncb.employee (id),
       foreign key ("TYPE") references nabncb.contacttype (id),
       foreign key (preflang) references nabncb.iso_language (id),
       foreign key ("TYPE") references nabncb.customertype (id)
       );
-- Register relationship between customers; member of same household,
-- owner of business
create table nabncb.customerrelation (
       fromcustomer    integer not null,
       relationtype    integer not null,
       tocustomer      integer not null,
       primary key (fromcustomer,relationtype,tocustomer),
       foreign key (fromcustomer) references nabncb.customer (id),
       foreign key (relationtype) references nabncb.relationtype (id),
       foreign key (tocustomer) references nabncb.customer (id)
       );
-- Savings account, checkings account, car loan, home loan etc.
create table nabncb.product (
       id            integer not null,
       description   varchar(50) not null,
       business      integer not null,
       primary key(id),
       foreign key (business) references nabncb.lineofbusiness (id)
       );
-- Instance of a product related to one or more customers through
-- customerrole
create table nabncb.contract (
       id          integer not null,
       product     integer not null,
       status      integer not null,
       created     timestamp not null with default,
       updated     timestamp,
       primary key (id),
       foreign key (product) references nabncb. product (id)
       );
-- Additional information related to a specific contract e.g. currency
create table nabncb.contractitem (
       id          integer not null,
       contract    integer not null,
       item        integer not null,
       "value"       varchar(30) not null,
       primary key (id),
       foreign key (contract) references nabncb.contract (id),
       foreign key (item) references nabncb.item (id)
       );
-- Identifies the customer's role e.g. account owner, beneficiary,
-- stakeholder
create table nabncb.contractrole (
       id          integer not null,
       customer    integer not null,
       contract    integer not null,
       role        integer not null,
       primary key (id),
       foreign key (customer) references nabncb.customer (id),
       foreign key (contract) references nabncb.contract (id),
       foreign key (role) references nabncb.role (id)
       );
```

C

Additional material

This book refers to additional material that you can download from the Internet as described in this appendix.

Locating the Web material

The Web material associated with this book is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser at:

`ftp://www.redbooks.ibm.com/redbooks/SG247546`

Alternatively, you can go to the IBM Redbooks Web site at:

`ibm.com/redbooks`

Select the **Additional materials** and open the directory that corresponds with the IBM Redbooks form number, SG247546.

Using the Web material

The additional Web material that accompanies this book includes the following files:

File name	*Description*
SG247546.zip	Compressed Code Samples

How to use the Web material

Create a subdirectory (folder) on your workstation, and decompress the contents of the Web material zipped file into this folder.

Related publications

We consider the publications that we list in this section particularly suitable for a more detailed discussion of the topics that we cover in this book.

IBM Redbooks publications

For information about ordering these publications, see "How to get IBM Redbooks publications" on page 912. Note that some of the documents referenced here might be available in softcopy only.

- *SOA Solutions Using IBM Information Server*, SG24-7402
- *IBM WebSphere Information Analyzer and Data Quality Assessment*, SG24-7508
- *IBM InfoSphere DataStage Data Flow and Job Design*, SG24-7576

Other publications

These publications are also relevant as further information sources:

- *IBM WebSphere QualityStage Version 8 User Guide,* SC18-9922
- *IBM WebSphere QualityStage Version 8 WebSphere QualityStage Tutorial,* SC18-9925
- *IBM WebSphere QualityStage Version 8 Pattern-Action Reference,* SC18-9926
- *IBM WebSphere QualityStage Module for CASS Guide to IBM WebSphere QualityStage Module for CASS,* SC19-1225
- *IBM Information Server Version 8: Information Server Introduction,* SC19-1049
- *IBM Information Server - Delivering information you can trust,* IBM U.S. Announcement 206-308 dated 12 December 2006
- *IBM Information Server Version 8: IBM Information Server Planning, Installation, and Configuration Guide,* GC19-1048
- *IBM Information Server Version 8.0.1 IBM Information Server Administration Guide,* SC19-9929

- *IBM Information Server Version 8: Information Server Reporting Guide,* SC19-1162
- *IBM Information Server Quick Start Guide*
- *IBM Information Server — Delivers next generation data profiling analysis and monitoring through the new IBM WebSphere Information Analyzer module,* IBM U.S. Announcement 207-043 dated 13 March 2007
- *IBM Information Management Software Profiling: Take the first step toward assuring data quality,* December 2006, IMW11808-USEN-00.
- *WebSphere Information Analyzer Version 8.0.1 WebSphere Information Analyzer User Guide,* SC18-9902

Online resources

The following Web site is also relevant as a further information source:

- IBM Information Server information center

  `http://publib.boulder.ibm.com/infocenter/iisinfsv/v8r0/index.jsp`

How to get IBM Redbooks publications

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

IBM WebSphere QualityStage Methodologies, Standardization, and Matching

IBM

Redbooks

(1.5” spine)
1.5”<-> 1.998”
789 <->1051 pages

IBM WebSphere QualityStage Methodologies, Standardization, and Matching

IBM WebSphere QualityStage architecture

Merger and acquisition business scenario

IBM Information Server overview

IBM WebSphere QualityStage provides data cleansing capabilities to help ensure quality and consistency by standardizing, validating, matching, and merging information to create comprehensive and authoritative information for multiple uses.

IBM Information Server is a revolutionary software platform that helps organizations derive more value from the complex heterogeneous information that is spread across their systems. It enables organizations to integrate disparate data and deliver trusted information wherever and whenever needed, in line and in context, to specific people, applications, and processes.

This IBM Redbooks publication documents the procedures for implementing IBM WebSphere QualityStage and related technologies using a typical merger and acquisition financial services business scenario. The scenario covers all dimensions of data cleansing, standardization, and matching rules, job design and deployment through a data integration life cycle.

The book offers a step-by-step approach to implementing IBM WebSphere QualityStage on Red Hat Enterprise Linux 4.0 platform accessing information that is stored on IBM z/OS and IBM AIX platforms. If you are an IT architect, Information Management specialist, or Information Integration specialist who is responsible for developing IBM WebSphere QualityStage on a Red Hat Enterprise Linux 4.0 platform, you will find the information in this book helpful.

SG24-7546-00 ISBN 0738431206