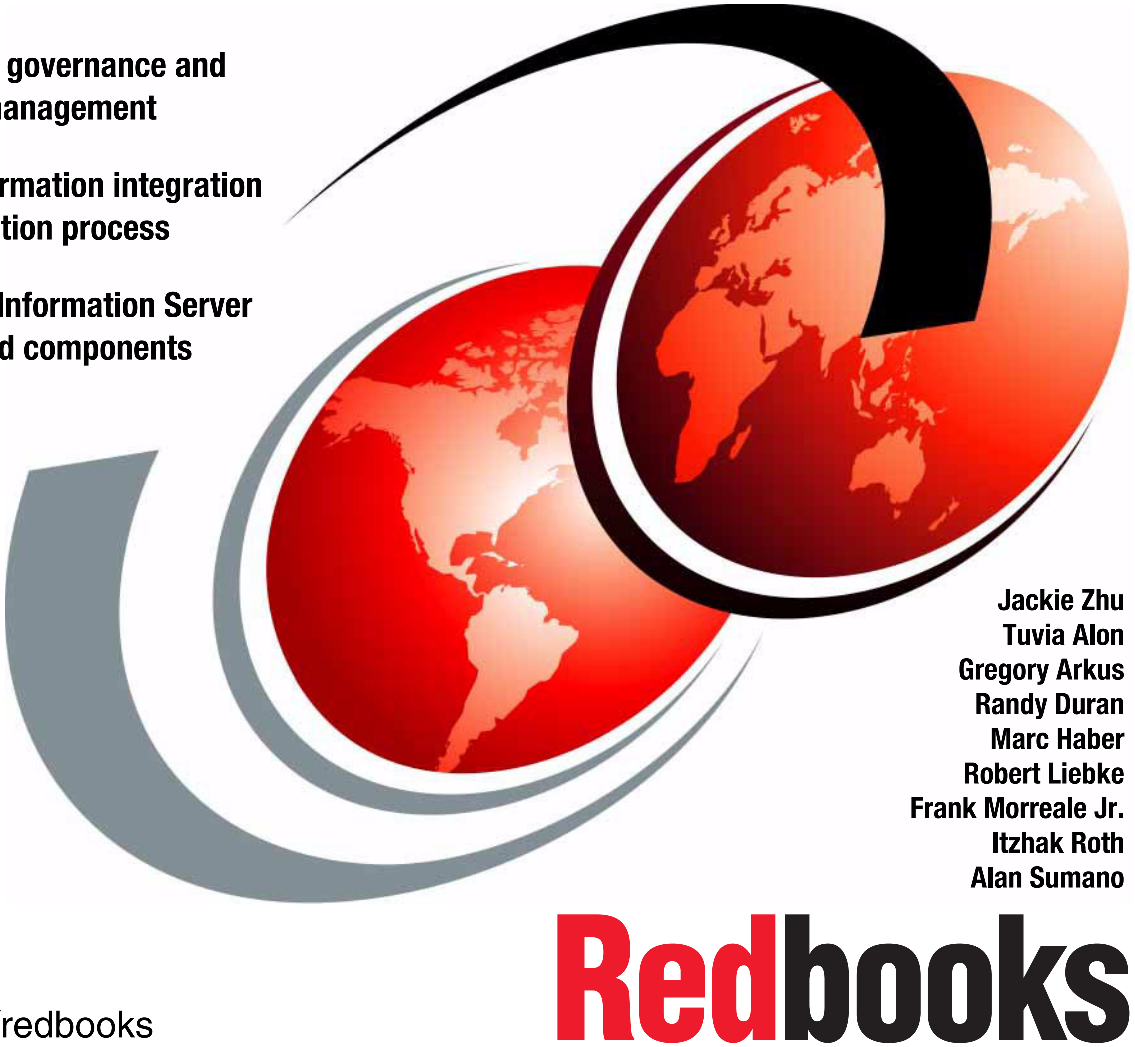
IBM® Information Management Software

Metadata Management with IBM InfoSphere Information Server

Information governance and metadata management

Typical information integration implementation process

InfoSphere Information Server modules and components



Jackie Zhu
Tuvia Alon
Gregory Arkus
Randy Duran
Marc Haber
Robert Liebke
Frank Morreale Jr.
Itzhak Roth
Alan Sumano

Redbooks

ibm.com/redbooks

International Technical Support Organization

Metadata Management with IBM InfoSphere Information Server

October 2011

SG24-7939-00

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

First Edition (October 2011)

This edition applies to Version 8, Release 7, of IBM InfoSphere Information Server.

Contents

Notices xi
Trademarks xii

Preface xiii
The team who wrote this book xiv
Now you can become a published author, too! xvii
Comments welcome xvii
Stay connected to IBM Redbooks xviii

Part 1. Overview and concepts 1

Chapter 1. Information governance and metadata management 3
1.1 Information governance 4
1.2 Defining metadata 5
1.3 Types of metadata 6
1.3.1 Business metadata 6
1.3.2 Technical metadata 7
1.3.3 Operational metadata 8
1.4 Why metadata is important 8
1.4.1 Risk avoidance 9
1.4.2 Regulatory compliance 9
1.4.3 IT productivity 9
1.5 Requirements for managing metadata 10
1.5.1 The information governance organization 12
1.5.2 Information governance operational teams 13
1.5.3 Standards, policies, and procedures 17
1.6 Business scenarios for metadata management 22
1.6.1 Metadata for compliance 22
1.6.2 Metadata for risk management 23
1.7 Where to start 24
1.8 Conclusion 27

Chapter 2. Solution planning and metadata management 29
2.1 Getting started with solution planning 30
2.1.1 Information integration solution 30
2.1.2 Information integration project 31
2.2 Stakeholders 31
2.3 Integrated solution data flow 33
2.4 Typical implementation process flow 34

2.4.1 Defining business requirements 36
2.4.2 Building business centric vocabulary 36
2.4.3 Developing data model 37
2.4.4 Documenting source data 37
2.4.5 Assessing and monitoring data quality 37
2.4.6 Building up the metadata repository 38
2.4.7 Transforming data 39
2.4.8 Developing BI solutions 39
2.4.9 Generating enterprise reports and lineage 39
2.5 Conclusion 40

Chapter 3. IBM InfoSphere Information Server approach 41
3.1 Overview of InfoSphere Information Server 42
3.2 Platform infrastructure 43
3.2.1 Services tier 44
3.2.2 Engine tier 45
3.2.3 Repository tier 46
3.3 Product modules and components 47
3.3.1 InfoSphere Blueprint Director 48
3.3.2 InfoSphere DataStage and InfoSphere QualityStage 49
3.3.3 InfoSphere Information Analyzer 49
3.3.4 InfoSphere Discovery 50
3.3.5 InfoSphere FastTrack 51
3.3.6 InfoSphere Business Glossary 52
3.3.7 InfoSphere Metadata Workbench 53
3.3.8 InfoSphere Information Server Manager, ISTools and InfoSphere Metadata Asset Manager 54
3.3.9 InfoSphere Data Architect 55
3.3.10 Cognos Business Intelligence software 57
3.4 Solution development: Mapping product modules and components to solution processes 58
3.4.1 Defining the business requirements 59
3.4.2 Building business centric vocabulary 59
3.4.3 Developing data model 60
3.4.4 Documenting source data 61
3.4.5 Assessing and monitoring data quality 63
3.4.6 Building up the metadata repository 64
3.4.7 Transforming data 64
3.4.8 Developing BI solutions 66
3.4.9 Generating enterprise reports and lineage 66
3.5 Deployment architecture and topologies 67
3.5.1 Overview of the topologies 67
3.5.2 Unified and shared metadata 68

3.5.3 Metadata portability 71
3.5.4 Alternative deployment 73
3.6 Conclusion. 73

Part 2. Implementation 75

Chapter 4. Use-case scenario 77
4.1 Scenario background 78
4.2 Current BI and data warehouse solution for Bank A 78
4.3 Project goals for the new solution 80
4.4 Using IBM InfoSphere Information Server for the new solution 80
4.4.1 Changes required 81
4.5 Additional challenges 82
4.5.1 The integration challenge and the governance problem 83
4.5.2 Additional business requirements 84
4.6 A customized plan 85
4.6.1 BI process development 87
4.6.2 Data Quality monitoring and subscription 88
4.6.3 Data lineage and reporting requirements and capabilities 88
4.7 Conclusion. 88

Chapter 5. Implementation planning 89
5.1 Introduction to InfoSphere Blueprint Director 90
5.2 InfoSphere Blueprint Director user interface basics 93
5.2.1 User interface 93
5.2.2 Palette 95
5.3 Creating a blueprint by using a template. 97
5.3.1 Customizing the template 101
5.3.2 Working with metadata repository. 105
5.3.3 Working with a business glossary 111
5.4 Working with milestones 118
5.5 Using methodology 128
5.6 Conclusion. 130

Chapter 6. Building a business-centric vocabulary 131
6.1 Introduction to InfoSphere Business Glossary 132
6.2 Business glossary and information governance 133
6.3 Creating the business glossary content 134
6.3.1 Taxonomy 135
6.3.2 The taxonomy development process 136
6.3.3 Controlled vocabulary 137
6.3.4 Term specification process and guidelines 138
6.3.5 Using external glossary sources 140
6.3.6 The vocabulary authoring process 141

6.4 Deploying a business glossary 142
6.4.1 InfoSphere Business Glossary environment 142
6.5 Managing the term authoring process with a workflow 146
6.5.1 Loading and populating the glossary 149
6.5.2 Creating and editing a term 150
6.5.3 Adding term relations and assigning assets 153
6.5.4 Reference by category 156
6.5.5 Custom attributes 158
6.5.6 Labels 160
6.5.7 Stewardship 161
6.5.8 URL links 162
6.5.9 Import glossary 163
6.6 Searching and exploring with InfoSphere Business Glossary 164
6.7 Multiple ways of accessing InfoSphere Business Glossary 168
6.7.1 InfoSphere Business Glossary Anywhere 168
6.7.2 REST API 169
6.7.3 Eclipse plug-in 170
6.8 Conclusion 173

Chapter 7. Source documentation 175
7.1 Process overview 176
7.2 Introduction to InfoSphere Metadata Asset Manager 176
7.3 Application systems 177
7.3.1 Extended data source types 179
7.3.2 Format 180
7.3.3 Loading the application system 182
7.3.4 Results 184
7.4 Sequential files 184
7.4.1 Loading a data file 185
7.4.2 Results 191
7.5 Staging database 192
7.5.1 Loading the staging database 193
7.5.2 Results 199
7.6 Data extraction 200
7.6.1 Input file format 202
7.6.2 Documenting the data extraction 203
7.6.3 Results 206
7.7 Conclusion 207

Chapter 8. Data relationship discovery 209
8.1 Introduction to InfoSphere Discovery 210
8.1.1 Planning equals saving 211
8.1.2 A step-by-step discovery guide 213

8.2 Creating a project 214
8.2.1 Pointing to the data requiring analysis 216
8.2.2 Importing the source data 217
8.2.3 Importing the target data 222
8.3 Performing column analysis 225
8.3.1 Monitoring tasks with the activity viewer 228
8.3.2 Reviewing the column analysis results 230
8.3.3 Metadata and statistical results. 231
8.3.4 Value, pattern, and length frequencies 233
8.4 Identifying and classifying sensitive data 238
8.4.1 Column classification view 239
8.4.2 Displaying hits for classification columns 241
8.4.3 Column classification algorithms. 242
8.5 Assigning InfoSphere Business Glossary terms to physical assets 243
8.5.1 Importing, mapping, and exporting term assignments 244
8.5.2 Mapping business glossary terms to physical assets 245
8.6 Reverse engineering a data model 250
8.6.1 Primary-foreign key candidates. 251
8.6.2 Discovering primary-foreign key candidates 251
8.6.3 Displaying the results 252
8.6.4 Data objects 252
8.6.5 Performing transformation discovery 253
8.7 Performing value overlap analysis 259
8.7.1 Running overlap analysis 260
8.7.2 Column Summary 265
8.7.3 Viewing value overlap details 266
8.8 Discovering transformation logic 268
8.8.1 Performing a transformation discovery 269
8.8.2 Reviewing maps 270
8.8.3 Exporting transformation results to InfoSphere FastTrack 280
8.9 Conclusion. 281

Chapter 9. Data quality assessment and monitoring. 283
9.1 Introduction to IBM InfoSphere Information Analyzer 284
9.1.1 InfoSphere Information Analyzer and information governance 284
9.1.2 InfoSphere Information Analyzer and InfoSphere Information Server 286
9.1.3 Metadata data repository 287
9.2 InfoSphere Information Analyzer data rules 289
9.2.1 Roles in data rules and data quality management 290
9.2.2 Properties of the InfoSphere Information Analyzer data rules 292
9.2.3 Data rules management 293
9.2.4 Rule definition guidelines for data quality 296

9.3 Creating a rule 297
9.3.1 Creating a rule definition 298
9.3.2 Testing a rule 302
9.3.3 Generating data rules 303
9.4 Data rule examples 304
9.4.1 Checking for duplicates 304
9.4.2 Generating a data rule 308
9.4.3 Use case: Creating a data rule to monitor high value customers . . 308
9.4.4 Creating rules to monitor gold customers 310
9.5 Data rules and performance consideration 315
9.5.1 Types of data rules 315
9.5.2 Using join tables in data quality rules 316
9.5.3 Cartesian products and how to avoid them. 318
9.5.4 Applying filtering in data quality rules 320
9.5.5 Filtering versus sampling 322
9.5.6 Virtual tables versus database views 322
9.5.7 Global variables 323
9.6 Rule sets 324
9.7 Metrics. 327
9.8 Monitoring data quality 329
9.9 Using HTTP/CLI API 331
9.10 Managing rules 332
9.11 Deploying rules, rule sets, and metrics 334
9.12 Rule stage for InfoSphere DataStage 335
9.13 Conclusion. 338

Chapter 10. Building up the metadata repository 339
10.1 Introduction to InfoSphere Metadata Workbench 340
10.2 Data storage systems 341
10.3 Data models 341
10.3.1 Loading the data models. 342
10.3.2 Results 347
10.4 Business intelligence reports. 348
10.4.1 Loading BI reports. 349
10.4.2 Results 355
10.5 Information asset enrichment 356
10.5.1 Business glossary terms 357
10.5.2 Business glossary labels. 360
10.5.3 Data stewardship 362
10.5.4 Asset descriptor and alias. 364
10.6 Conclusion. 373

Chapter 11. Data transformation 375

11.1 Introduction to InfoSphere FastTrack 376
11.1.1 Functionality and user interface 377
11.1.2 Administration 377
11.2 Basic mapping. 378
11.3 Advanced mapping 380
11.4 Mapping lifecycle management (job generation). 382
11.5 Metadata sharing (extension mappings) 385
11.6 InfoSphere DataStage job design 386
11.6.1 Job design details 388
11.7 Shared metadata. 390
11.7.1 Shared metadata that must be created. 390
11.8 Operational metadata 391
11.8.1 Creating operational metadata 391
11.9 Conclusion. 392

Chapter 12. Enterprise reports and lineage generation 393
12.1 Lineage administration 394
12.1.1 Business lineage. 395
12.1.2 Data lineage 396
12.1.3 Impact analysis 397
12.2 Support for InfoSphere DataStage and InfoSphere QualityStage jobs . 398
12.2.1 Design lineage 398
12.2.2 Operational lineage. 409
12.3 Support for external processes 412
12.4 Support for InfoSphere FastTrack mapping specifications 413
12.5 Configuring business lineage 416
12.6 Search and display 418
12.6.1 Information catalog 418
12.6.2 Find and search 422
12.7 Querying and reporting 424
12.7.1 Reports 424
12.7.2 Querying 429
12.8 Conclusion. 432

Related publications 433
IBM Redbooks 433
Online resources 433
Help from IBM 434

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

Preface

What do you know about your data? And how do you know what you know about your data?

Information governance initiatives address corporate concerns about the quality and reliability of information in planning and decision-making processes. *Metadata management* is cataloging information about data objects. Many organizations have data spread across disparate heterogeneous systems. Users who own, manage, and access these systems often do not communicate with each other. Metadata management refers to the tools, processes, and environment that are provided so that organizations can reliably and easily share, locate, and retrieve information from these systems.

Enterprise-wide information integration projects integrate data from these systems to one location to generate required reports and analysis. During this type of implementation process, it is important to provide metadata management along each step. Metadata management ensures that the final reports and analysis are from the right data sources, are complete, and have quality.

This IBM® Redbooks® publication introduces the information governance initiative and highlights the immediate needs for metadata management. It explains how *IBM InfoSphere™ Information Server* provides a single unified platform and a collection of product modules and components so that organizations can understand, cleanse, transform, and deliver trustworthy and context-rich information.

This book describes a typical implementation process. It explains how InfoSphere Information Server provides the functions that are required to implement such a solution and, more importantly, to achieve the metadata management goal.

This book addresses the following InfoSphere Information Server product modules and components (some in more detail than others):

- IBM InfoSphere Blueprint Director
- IBM InfoSphere Data Architect (overview only)
- IBM InfoSphere Business Glossary
- IBM InfoSphere Discovery
- IBM InfoSphere Information Analyzer
- IBM InfoSphere Metadata Asset Manager
- IBM InfoSphere Metadata Workbench
- IBM InfoSphere FastTrack

- IBM InfoSphere DataStage®
- IBM InfoSphere QualityStage™

This book is intended for business leaders and IT architects with an overview of metadata management in information integration solution space. The book also provides key technical details that IT professionals can use in a solution planning, design, and implementation process.

The team who wrote this book

This book was produced by a team of specialists from around the world working for the International Technical Support Organization (ITSO) at the IBM Jerusalem Lab in Israel.

Jackie Zhu is a Project Leader for the IBM ITSO in the US. Jackie joined IBM in 1996 and has more than 10 years of software development experience in accounting, image workflow processing, and digital media distribution. She is a Certified Solution Designer for IBM Content Manager. Currently, Jackie manages and leads the production of IBM Redbooks publications that focus on Enterprise Content Management. Jackie holds a Master of Science degree in Computer Science from the University of the Southern California.

Tuvia Alon is a senior consultant for the Lab Services Center of Excellence in Information Management for IBM Software Group. He provides services to Europe, Middle East, and Africa (EMEA) and other worldwide regions, operating from the IBM Israel Software Labs. Tuvia joined IBM in 2006 with the acquisition of Unicorn that specializes in metadata and semantic technologies. Tuvia has over 20 years of experience in developing, implementing, and supporting enterprise software. He is an expert in metadata technologies and a recognized leader in implementing enterprise data governance solutions. As a consultant in the center of excellence, Tuvia has served as a trusted advisor for IBM customers implementing data integration and data governance strategies. He has spoken at numerous conferences in the US and Europe about the use of metadata in corporate governance policy.

Gregory Arkus is a Program Manager for the IBM Information Management enablement team in the US. Previously, he was a Solution Architect for IBM InfoSphere Information Server. Greg has over 15 years of experience in Information Management and business intelligence systems. He began his IT career as a software engineer developing software in the data quality in structural and business form areas. He has written several courses on various metadata topics.

Randy Duran is a Solutions Architect for IBM Information Management Sales. Randy joined IBM in 2009 for a second time with the acquisition of Exeros. He offers more than 20 years of product management, technical sales, and consulting experience from prior roles at Oracle, Macromedia, and IBM Lotus®. He specializes in data discovery, data integration, and data privacy. He also develops and delivers enablement materials for the InfoSphere Discovery solution. Randy holds a degree in symbolic systems from Stanford University.

Marc Haber is the Functional Architect for IBM InfoSphere Information Server in IBM Israel. Marc joined IBM as part of the acquisition of Unicorn Software in 2006 and has worked in development, consultation, and product management roles. As the Functional Architect, Marc is responsible for working with customers to advance their understanding of the InfoSphere products and assist them in the planning, enablement, and rollout of metadata initiatives and strategies. Marc has authored and delivered many presentations and training at conferences and customer establishments.

Robert Liebke is a Senior Consultant and Technical Architect with IBM Lab Services in the US. He has 35 years of experience in information technology (IT). He was the key contributor in the development of a Decision Support System and Data Warehousing Resource Guide. He also has published articles in *DM Review Magazine*. Robert holds a Master of Telecommunications degree from Golden Gate University and a Bachelor of Computer Science degree from the University of Wisconsin.

Frank Morreale Jr. is a Senior Certified IT Specialist and Consultant for IBM Software Group working in Lab Services. Frank joined IBM in 2005 with the acquisition of Ascential® Software. He has more than 28 years experience in the Software Development field. His areas of experience include automated manufacturing systems, advanced user interface design, and user interface control development. He has extensive experience in the development and implementation of metadata solutions first with Ascential Software and now with IBM. Frank has authored and delivered many presentations at conferences and customer establishments.

Itzhak Roth is a Senior Consultant and Technical Architect with the practice team for IBM InfoSphere lab services in IBM US. He has more than 35 years of experience in systems analysis and process optimization across a range of industries from banking and services to medical sciences. Itzhak has developed financial simulation models and logistic optimization systems, rule-based diagnostic systems, and semantic models for financial institutions. He joined IBM in 2006 and assumed a lead consulting role in data quality, analytics, and metadata management. He has provided consulting and mentoring to customers in various industries from banking and insurance to oil, airlines, and medical diagnostics. Itzhak holds a Doctorate of Computer Science degree (Ph.D.) from Temple University in Philadelphia, PA, and a Master of Applied Statistics and Operations Research degree from Tel Aviv University in Tel Aviv, Israel.

Alan Sumano is an InfoSphere IT Specialist for IBM Mexico where he covers the Central America and Latin Caribbean regions. Alan joined IBM in 2009. He has 10 years of experience in designing and deploying Information Management Solutions in Mexico, US, Brazil, and Central America in multiple industries, both public and private. Alan holds a Master in Business Administration (MBA) degree from Thunderbird School of Global Management in Arizona. He also has a bachelor degree in Computing Systems for Management at Tecnológico de Monterrey in Mexico.

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

Yehuda Kossowsky
Michael Fankhauser
Joanne Friedman
Benny Halberstadt
Roger Hecker
Hayden Merchant
IBM Jerusalem Software Development Lab, Israel

Guenter Sauter
Alex Baryudin
Walter Crockett Jr.
Tony Curcio
Shuyan He
Pat Moffatt
Ernie Ostic
Patrick (Danny) Owen
Paula Sigmon
Harald Smith
Mark Tucker
Art Walker
LingLing Yan
IBM Software Group, US

Mandy Chessell
IBM Software Group, UK

Riccardo Tani
IBM Software Group, Italy

Leslie Parham
Jenifer Servais
Stephen Smith
IBM ITSO

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

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 publications in one of the following ways:

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

  ibm.com/redbooks

- Send your comments in an email 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

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

Part 1

Overview and concepts

This part introduces the following concepts:

- Information governance
- Metadata management
- Solution planning for a typical information integration project
- IBM InfoSphere Information Server product modules and components

With the product modules and components, organizations can understand, cleanse, transform, and deliver trustworthy and context-rich information to help fulfill information governance initiatives including metadata management.

This part includes the following chapters:

- Chapter 1, “Information governance and metadata management” on page 3
- Chapter 2, “Solution planning and metadata management” on page 29
- Chapter 3, “IBM InfoSphere Information Server approach” on page 41

1

Information governance and metadata management

The discipline of information governance helps organizations to better cope with the challenges of data explosion, fast-moving markets, elevated levels of uncertainty, and the increased burden of regulatory requirements. By practicing information governance, orchestrating people, processes, and technology ensures that information is understood, is of high quality, and is trusted.

Metadata management is a central tenet to information governance. *Information governance* is the ability of an organization to manage its information knowledge and to answer questions such as “What do we know about our information?” *Metadata management* refers to the tools, processes, and environment that are provided to enable an organization to answer the question, “How do we know what we know about our data?”

This chapter includes the following topics:

- Information governance
- Defining metadata
- Types of metadata
- Why metadata is important
- Requirements for managing metadata
- Business scenarios for metadata management

- Where to start
- Conclusion

1.1 Information governance

The world is experiencing an unprecedented explosion of data: An estimate of more than three exabytes of digital information are created everyday around the world. Increased levels of complexity and the need to keep pace with a fast-moving world are unforgiving to judgement errors or bad planning and decision making. Organizations realize that they need a better grip on their information and to better manage its creation, use, and distribution. *Information governance*, through people, processes, and technology, fills this gap to ensure that information is well understood, to improve its quality, and to instill trust in it.

An average organization handles thousands, often tens of thousands and more, of data assets. Organizations manage, maintain, execute, distribute, and consume databases, files, applications, and reports daily. Together, these assets, the people, and the processes that move and transform information comprise the *information supply chain* of the organization, as shown in Figure 1-1.

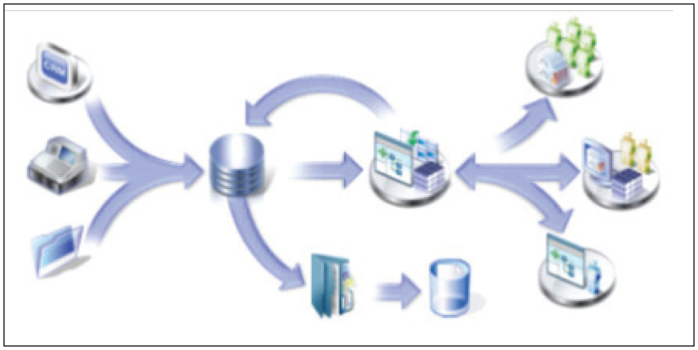


Figure 1-1 Information supply chain

Information governance initiatives address corporate concerns about the quality and reliability of information that is used by an organization in its planning and decision-making processes.

Information governance focuses on three areas:

Life-cycle management

Management of the creation, storage, retention, recovery, and destruction of information that might be required by the organization and regulatory authorities.

Protection Control of data use to provide privacy and security.

Trusted source of information
Assurance of the information quality, common understanding, timeliness, accuracy, and completeness of the information.

1.2 Defining metadata

If you search for a definition of metadata, you will find several of them. Definitions differ depending on the context domain in which the metadata is introduced.

For example, metadata in the context of library management emphasizes the cataloging aspect of library materials. Libraries use metadata to capture and describe library resources and various types of publications in a manner that users, librarians, and researchers can use it to locate, sort, and trace these publications. Researchers can search for publications by subject, name, or author, or they can look for related publications about the same subject, by the same author or in the same journal. The utilization of library resources increases many times while significantly reducing the effort involved.

In the field of art curation, metadata captures information about fine art artifacts. Curators use metadata to capture the information that describes the artifact, its history, location, and provenance. Museums and collectors use this information to manage their collections, validate their authenticity, and support research and education.

In the domain of information technology (IT), metadata is about data and other IT artifacts. The most common definition of metadata is “data about data.” However, because data without context is merely a meaningless sequence of symbols, the discussions about metadata quickly revert to ‘‘information about data, information about information.” If you have a large collection of information organized in a way that you can extract relationship and data flows and perform explorations and investigation so that it becomes actionable information, you can refer metadata as “knowledge about data.”

Maintaining a catalog of data artifacts serves a similar purpose of managing the inventory of publications in a library or fine art in a museum collection. You need to understand what the artifacts are, trace their origins, monitor their usage, and provide services to users and stakeholders.

In the domain of information processing, the objects are electronic records of data assets. Such data assets include data files, data columns in a database, business intelligence (BI) reports, or a transformation job that pours the content of one data store into another. These data assets are the type that an enterprise creates, maintains, and uses to conduct business. The information about these data assets is most valuable to separate users within the organization. Data analysts and developers want to know the physical characteristics of these assets.

Metadata includes technical attributes that describe the physical characteristics of a data element and, often, descriptive attributes that provide semantic and administrative information, such as meaning, usage, owners, and users. This information has broad usage at all levels of an organization and by users in various roles.

1.3 Types of metadata

Metadata has many sources, and many users find utility in various aspects of information about an object. You can classify metadata by content and usage in multiple ways. IBM InfoSphere Information Server classifies metadata into business, technical, and operational metadata types that are distinguished by their content and source.

1.3.1 Business metadata

Business metadata includes business terms and their definitions, examples of usage, business rules policies, and constraints. Altogether they define the semantics of a business concept and its realization in physical data assets.

Business metadata satisfies the needs of business people and the user community at large by answering the following types of questions:

- A report shows profit margin, but what does it mean?
- Where does the data for calculating the profit margin come from?
- What calculations went into the determination of the profit margin?
- Who (which *data steward*) owns this term?
- What are the business rules that are applied to profit margin?
- What other reports show profit margin?

Users of business metadata are primarily business users, but anyone can use it to understand what things mean. Examples include how, when, and by whom they are used, and what policies, rules, and restrictions might apply to the use of the data asset.

The source of business metadata is mostly the business. Subject matter experts (SMEs) and data stewards can be sources of business metadata. Internal and external business, operational, and policy manuals can be sources of business metadata.

You must express business metadata in a language that is spoken and understood by business people, which means spelling out terms fully and making sure that definitions are clear and unambiguous. Express business rules in plain language and not in a complex, formal manner with abbreviations, mathematical expressions, or functional expressions.

1.3.2 Technical metadata

Technical metadata consists of the technical description of data assets. Technical metadata includes the following descriptions:

- Schemas, tables, and file layouts
- Source and target datastore identification and physical attributes
- Data mappings
- Formal specifications of transformation jobs, business rules, and other processes

People and systems use technical metadata. IT technical staff, such as analysts, developers, and administrators, use technical metadata daily to perform jobs. For example, they might analyze requirements and write specifications for a new job, develop a method and write the code, or diagnose a problem and develop a fix for the problem.

Knowledge of the data and the processes that manipulate the data is valuable to businesses. Technical metadata helps to answer the following questions:

- What databases exist?
- What are the schemas, tables, views, and columns in a database?
- What are the keys of a given table?
- What do jobs read from a table?
- Can a column obtain a null value?
- What do jobs write to a table?
- What are the flat sequential files in the data stream?
- How is the data processed?
- What valid values are allowed for a column?

The source for technical metadata is usually the data assets themselves. Databases maintain data catalogs that contain all the information that a database management system needs to operate on the data that it stores. BI tools maintain data models, functions, specifications, and instructions to manipulate data and present it in the desired format.

Data integration tools maintain the details of the source, target, and flows of data; the transformations and parameters that are applied on the data; and all other details that are necessary to accomplish their designated function. All of this information that is used by the various tools to perform their tasks is technical metadata. Technical people create and use the technical metadata, and they usually store it in the individual tools. Technical people often enhance the technical data manually by annotating the data assets or by providing design and modeling documents.

1.3.3 Operational metadata

Operational metadata consists of information about the execution of an application or a job. Such information includes times and dates, counts of records processed and rejected, and statistics about processes, processors, and servers that are generated in the course of executing a job.

Operational metadata is used to answer the following questions:

- At what date and time was a job last executed?
- How many records were processed?
- How many records were rejected?
- How long did it take to process the job?

Users of operational metadata are primarily operations people who monitor the execution performance of various processes. IT and business management are often interested in the overall system performance and throughput as they consider adding more applications to existing resources or making decisions about additional computing resources. Business people might also be interested in the currency of the data that they use in their various business tasks. Therefore, they look for the last time that a job ran or that a data store was updated. Business people can obtain answers by reviewing the dates and times that a specific job was executed and that a data store was updated.

1.4 Why metadata is important

Metadata management is cataloging information about data objects. Metadata management provides the tools, processes, and environment to enable an organization to answer the question, “How do we know what we know about our data?” The capability to easily and conveniently locate and retrieve information about data objects, their meaning, physical location, characteristics, and usage is powerful and beneficial to the enterprise. This capability enhances the ability of the organization to deal with risk, meet regulatory requirements, and improve IT productivity.

1.4.1 Risk avoidance

Organizations face a multitude of risk types of which market, operational, and regulatory exposure are only a few. Information and knowledge are ways that an organization can mitigate risk. The capability to make plans and decisions based on reliable and trusted information does not eliminate risk. However, with this ability, businesses can more precisely evaluate the risk that they face and act accordingly. Metadata management provides the measure of trust that businesses need. Through data lineage and impact analysis, businesses can know the accuracy, completeness, and currency of the data used in their planning or decision making models.

1.4.2 Regulatory compliance

More than ever before, organizations are subject to regulatory requirements. They are required to submit periodical reports concerning their activities in their particular business domain. Certain regulatory directives are particular about data handling, specifying the security, privacy, and retention requirements on different types of data. Organizations are often subjected to audits where they are required to show and demonstrate that they comply with these requirements.

Metadata management conducted on a unified platform that provides stewardship, data lineage, and impact analysis services is the best assurance that an organization can validate and demonstrate that the data reported is true. It also validates and demonstrates that the data is correct and is handled according to regulations.

1.4.3 IT productivity

Many organizations are bogged down with the repetitive IT tasks of analyzing and reanalyzing the same data and processes whenever they are called to support a new service or reporting requirement. Data is abundant. Much of it comes from existing systems and data stores for which no documentation exists or, at best, the documentation does not reflect the many changes and updates of those systems and data stores. In many cases, this precious knowledge remained with the individuals who originally developed the system, but who might have left the company or moved on to roles in other departments or operations.

This lack of documentation translates to labor-intensive tasks of reverse engineering, research, and analysis to identify the elements that are required for the new tasks, their properties, and dependencies. You can avoid part of this effort if the metadata is captured and maintained in a repository that is accessible to developers, analysts, and other stakeholders.

1.5 Requirements for managing metadata

The key to metadata management requires a committed *partnership* between business and IT with joint ownership or guardianship in which both communities contribute time and effort and have input to the final product look and function.

Metadata management initiatives usually fall within the realm of the broader information governance program. Metadata management is an enabling practice, supporting management in its daily chores by providing trusted information for reporting, planning, and decision making. Metadata management helps users assert the meaning of information and determine its accuracy, completeness, and currency.

For a long time, information professionals and corporate management have recognized that data is a valuable asset and must be treated carefully. Although most organizations do not document the value of information assets, they are aware that they can extract value from the available information and might suffer damage if they fail to handle and secure it properly.

The discipline of information governance emerges as a primary condition for maintaining a healthy corporation. Presently, laws or regulations require many businesses and government entities to maintain specified levels of integrity, security, and privacy of their data and to be able to show compliance with these requirements. Beyond that, the ability of an organization to guarantee high-quality information or maintain the knowledge about its information is invaluable to its ability to compete successfully in today's markets.

Over the decades, information systems have evolved, creating a complex web of applications and data stores that feed each other. Transactional data systems perform monetary or material transactions, money transfers between accounts, reservations are made for trips, purchases are recorded, and shipping instructions are issued. These transactions represent a repository of valuable information that companies harvest and mine to extract patterns and identify trends.

Companies use BI applications to digest this data and present it in ways that managers and other knowledge workers can use to plan and make decisions. Businesses create information supply chains that consist of pipes, channels, and applications. IT uses various tools and utilities along these data routes to cleanse, integrate, consolidate, and reconcile data. These tools populate master data repositories, data warehouses, and data marts that are designed to meet various information needs.

At any given time, organizations deploy various information processing initiatives. Whether upgrading an application to support new offerings, products, and services, responding to compliance reporting requirements, or adopting new analytical and decision making disciplines, IT departments are busy moving information from one container to another, changing its format and representation. In the present information environment, these tasks have many risks.

A failure to deliver and possible exposure of the organization to legal and regulatory liability can result from the following issues:

- The failure to obtain complete and true specifications
- A lack of understanding the meaning and use of the data elements
- A lack of understanding the restriction and constraints that might apply to data elements

In traditional information management, knowledge resided in separate systems or with only individuals who had the knowledge of the subject or, if documented, on their personal workstation or in a drawer. Shared drives and directives to store all of these documents in a public area provide partial relief, but still much is to be desired. When companies deposit data in architectural designs, data dictionaries, system specifications, and other documents, these repositories do not lend themselves easily to search and exploration. Although information might be accessible to a broad community of users in a single location, the effort to identify the right documents, determine their currency and relevance, and retrieve the desired information might be significant.

The cost and risk that are associated with this style of operation are prohibitive, and organizations recognized this issue a long time ago. System development methodologies that have been introduced over the years have made significant strides in streamlining and rationalizing the software development process. However, not much was done in support of the preservation of the knowledge generated in the process post deployment.

Metadata management is the practice of managing knowledge about the information supply chain. Although many people refer to metadata as “data about data,” in reality, companies work with more than only data or information. *Metadata* refers to a rich structure of knowledge. This structure captures the meaning of a term or data asset, its relationships to other assets, and rules that might apply to it to determine the quality, policies, and regulations that specify its use.

Metadata management addresses many of the challenges that organizations face in the present reality of a fast-moving world. Transactions execute in a fraction of seconds, and decision making needs to match this speed. Trusted information in this world is invaluable, and the capability to trace and track the flow of data and access the information associated with it is critical.

1.5.1 The information governance organization

Managing metadata requires cooperation between the business and IT. Organization work with many types of metadata: business, technical, and operational. Any group of users might access any of these types of metadata. The power of metadata is the ability to place technical metadata into a business context and to derive from data in columns and tables the meaning and usage to interrelationships and dependencies.

As mentioned previously, metadata management is a committed *partnership* between business and IT. They have joint ownership or guardianship in which both communities contribute in time and effort and have input to the final product look and function. The term *management* implies proactive involvement in creating, monitoring, reporting, and making decisions about all issues concerning metadata.

The information governance organization has a layered structure that consists of three levels of management:

- Information governance steering committee. This group consists of senior representatives from business management and enterprise architecture. The team articulates the vision of the information governance initiatives, defines the strategic direction, and provides the oversight on its execution. With its executive power, the team can identify and allocate the resources needed for the execution of the program.

  The steering committee meets periodically to review progress reports, deliberate exceptions, and decide on budget and nominations for the various operational teams.

- Information governance center of excellence (CoE). This group consists of dedicated employees, including the information governance CoE lead. The information governance CoE provides logistic and organizational support to the operational teams. The CoE guides the separate operational teams in developing the processes and procedures for the various governance activities that are based on policies, principles, and guidelines provided by the steering committee. The CoE oversees the compliance of the operational teams.
- Information governance operational teams. Information governance operational teams work with various aspects of information governance and metadata management. Business and IT people staff the working groups as the subject might require. An operational team membership consists of a core team of individuals designated for a period of time, desirably not less than one year.

An organization typically has several operational teams with expertise in various subject areas, such as a BI metadata operational team, a data quality operational team, and a controlled vocabulary operational team. When necessary, the operational team can recruit additional members with the desired expertise and knowledge in a particular domain to assist in the development and maintenance tasks of the team. The operational teams are highlighted in more detail in the following section.

1.5.2 Information governance operational teams

The organization assembles operational teams to address various aspects of the collection, creation, and administration of metadata as needed and as available resources permit. Operation teams usually consist of the following teams:

- Controlled vocabulary operational team
- Data quality operational team
- Data modeling and business intelligence operational team
- Metadata administration operational team

Controlled vocabulary operational team

The controlled vocabulary operational team develops and maintains the taxonomy of terms and controlled vocabulary of the organization. It also promotes the use of the terms and controlled vocabulary throughout the organization. This team captures and represents the vocabulary that is used in the organization in a manner that enhances communication about, and the understanding of, matters concerning the creation, processing, and use of information. The team commits to adhere to the rules and guidelines that are accepted and approved by the information governance office.

Members of the teams train on the standards, conventions, and processes of the controlled vocabulary and business glossary development and maintenance. Team members commit to the content of the business glossary and its integrity.

In addition, the team might designate individuals across the organization to be authors or data stewards who are responsible for the integrity and accuracy of the sections of the business glossary that are assigned to them. The team lead can call on employees or external consultants to assist in the identification, definition, and enrichment of terms in the vocabulary in areas of their expertise.

The team consists of people with varied skills and experience to be involved in the design, creation, and maintenance of the taxonomy and vocabulary:

- SMEs, from a relevant business domain, understand the business use of the terms, their dependencies, and their relationships to other terms. They create and define new terms.
- Business analysts know the business definitions of the terms for each business entity. They work with SMEs to establish a list of the terms that represent the most common words that are used in reports, applications, and

general communication. They ensure that the term definition is consistent with the goals of the enterprise.

- Data architects understand the physical and structural aspects of the data sources to which the terms might be assigned. They establish the relationships between the terms and technical assets. They identify the terms or the data assets, such as database tables, columns, reports, and others, that need to be assigned to terms.
- Compliance officers are in charge of overseeing and managing compliance issues within an organization. They ensure that term definitions and relationships to other terms and assets conform to business policies and legal regulations.

Controlled vocabulary operational team roles

The controlled vocabulary operational team includes the following roles:

- The *controlled vocabulary team lead* is responsible for managing and coordinating all activities regarding the definition and creation of categories and terms. In addition, the team lead asserts that all entries comply with company standards. The team lead reports to the information governance CoE.
- The *controlled vocabulary author* is an SME in a particular domain who is in charge of identifying terms and formatting them according to standards and conventions that are established by the organization. These users have expertise in indexing and controlled vocabulary construction. They are likely to be experts in the subject domain of the controlled vocabulary.

  Controlled vocabulary authors must have access to all views of a controlled vocabulary and complete information about each term. They must have the ability to edit and manipulate term records, cross-references, classification notation, and hierarchies. They require glossary read/write access, which actual users of a controlled vocabulary do not need.
- The *controlled vocabulary advisory team member* is an SME in a particular domain in an advisory role to the team. These members validate and suggest vocabulary entries, their definitions, and their relationships to other terms in the vocabulary.

Data quality operational team

The data quality operational team assumes responsibility for the definition maintenance of data quality rules and standards and their monitoring. *Data quality rules* represent knowledge that the corporation has developed over time to meet its data quality objectives. Data rules can originate from various operational areas in the organization.

The data quality operational team translates the rules or quality metrics that are provided in natural language into formal expressions that can be executed and scheduled. The team also monitors the execution of these rules and reports to the owners.

Data quality operational team roles

The data quality operational team includes the following roles:

- The *data quality team lead* manages and coordinates all the activities concerning the creation, definition, and maintenance of the data quality rules. This person also monitors the execution these rules. In addition, the data quality team lead confirms that all entries comply with company standards. The team lead reports to management on the progress of data quality improvement. This person also helps to resolve data quality problems through appropriate process design strategies and by using error detection and correction tests and procedures.
- The *data quality analyst* helps the organization maintain data quality at the standards established by the information governance office. The data quality analyst defines the rules to evaluate data quality and create test plans that work. The data quality analyst also monitors the compliance of data flows against data quality standards. In addition, the data quality analyst develops, documents, and maintains data quality goals and standards.
- The *data quality developer* translates data quality rule specifications into executable rules. This person tests data rules against target data to determine whether the rule definitions were properly translated into formal expressions and the target data was properly selected. The data quality developer schedules the execution of the rules and the production of the data quality reports.

Data modeling and business intelligence operational teams

Data modeling and business intelligence operational teams exist within most IT departments under the auspices of one group or another. However, more than any other IT group, these teams are closer to the business. They bridge between business and IT, translating business requirements into data structures, models, and designs so that developers can turn requirements into operational systems and reports. Team members are in the front line of the IT operations taking business content to create IT specifications and artifacts. They generate metadata, placing details of the transformation of business concepts into data structures and specifications.

If not directly reporting to the information governance council part of the metadata landscape, data modeling and BI operations are subject to the rules, policies, and standards regarding the creation and maintenance of information about information.

Data modeling and business intelligence operational teams roles

The data modeling and business intelligence operational teams include the following roles:

- The *data modeling team lead* manages and coordinates the activities of the data modeling team. This person is responsible for the implementation of guidelines and adoption of rules and standards concerning data issued by the information governance steering committee. This person is also responsible for developing and applying data modeling best practices and the proficiency of team members to use the modeling tools employed by the organization. The activities of the group are driven by business priorities. However, if decided, the team lead also reports to the information governance CoE regarding activities of the data modeling group.
- The *data modeler* is professionally trained to capture and represent business concepts in data structures in a manner that functional requirements can be satisfied in the most efficient way. The data modeler interacts with the business to gather functional requirements and with users to gather their reporting requirements. This person abides by the organizational guidelines and applies established standards in developing data models. The data modeler conducts sessions with business people to understand the meaning for their requirements and the vocabulary they use. This person documents accurately the meaning of business terms and their data representation and realization.
- The *business intelligence team lead* manages and coordinates the activities of the business intelligence team. The activities of the group are driven by business priorities but are carried out according to the standards and guidelines established by the information governance steering committee. The BI team lead is responsible for developing and applying BI modeling and reporting best practices as they might apply to the BI tools used by the organization.
- The *business intelligence designer* is professionally trained on the creation of BI models and reports. The BI designer interacts with the business and with users to gather their requirements. This person creates BI models and design reports by using naming convention and terminology established by the information governance authorities.

Metadata administration operational team

In terms of managing metadata, people often think of a metadata repository, which is a database with the appropriate data structures capable of storing and retrieving information about data assets. The metadata administration team is responsible for all aspects of maintaining the metadata repository. Such aspects include loading metadata, establishing lineage between data elements, and performing all other tasks that are required to keep the repository current and secure. The workgroup consists of IT professionals with system and data administration skills who are trained on the metadata repository environment.

The team might have work groups or individuals that specialize in types of metadata or sources of metadata, such as the business vocabulary or glossary, operational metadata, or BI reports. The team monitors the use and performance of the metadata management system. The work of the team might often require collaboration with business analysts, data analysts, and SMEs to establish correct relationships among business metadata and technical and operational metadata.

Metadata administration team roles

The metadata administration team includes the following roles:

- The *metadata administration lead* is an IT manager who is responsible for the operation of the team. This person establishes guidelines and standards for operations and verifies the execution of tasks according to these standards. This person also reports to the information governance office lead.
- The *metadata administrator* is an IT person who is skilled at system and data administration and trained on the metadata management platform. This person is responsible for different aspects of the metadata repository maintenance including security and integrity. The administrator assigns access privileges and permissions to groups of users. This person is also responsible for promoting metadata from a development environment to a production environment.

1.5.3 Standards, policies, and procedures

Standards, policies, and procedures are the backbone of an information governance program. You establish standards, policies, and procedures; apply them to set goals for the program; and specify how to attain these goals.

Standards

Standards exist all around us. They permit the world to move forward in an orderly fashion and enable commerce to be conducted. Organizations create standards for the simplest of products, defining their measures and content, all the way up to building bridges and erecting skyscrapers. Standards enable power to flow along grid lines. They also determine the safety of the water that you drink, the medications that you might take, and the foods that you eat. In the domain of data processing, you use standards to enable processing and the exchange of information.

Data standards are documented agreements about the representation, formats, and definitions of data elements. Standardized data is more meaningful, more comparable, and easier to exchange and store.

The benefits of data standardization are major tenets of information governance:

- Improved quality
- Better compatibility
- Improved consistency and efficiency of data collection
- Reduced redundancy

Most IT departments apply a common naming standard. Because they use information component names as primary search keys, they must convey meaning so that users know what the key looks like and what it represents or is used for.

Naming standards

Naming standards or conventions are inherent in any IT methodology. Both large and small IT organizations develop or adopt naming standards to identify their IT assets. An IT organization defines how to construct names for servers and network nodes down to databases tables, columns, and report headers.

The *object/modifier/class* approach is a widespread naming standard that is commonly used to label data columns and other data artifacts. In this approach, the name of an attribute is driven from its definition or description. The name is constructed by using the following elements:

- Object: What is being defined?
- Modifier: Which fact about the object is being defined?
- Class: Which class of information is being defined?

For example, a column that captures the time of day that a flight is scheduled to arrive or the “scheduled flight arrival time” includes the following data:

- Object: Flight
- Modifiers: Scheduled, arrival
- Class: Time

The application of this approach to name tables, columns, or other data assets often resorts to the use of abbreviations. Whether to comply with length restrictions or to make life easier for the administrator, modeler, and developers, abbreviations are used to replace the full spelling of the word.

Abbreviations are also subject to standards, which are mostly developed by the company to reflect its own or industry terminology. Often to specify the class of words, people use `ID` for identifier, `CD` for code, `TM` for time, and `DT` for date. For the modifier, you might use `ARVL` for arrival, `DPRT` for departure, `SCHD` for scheduled, and `ACTL` for actual, and so on. By using these abbreviations, the scheduled flight arrival time might be displayed as `FLGHT_SCHD_ARVL_TM`.

Standard abbreviations

Organizations must adopt and publish its standard abbreviations list to avoid confusion or misinterpretation. Publishing this list is important given the large number of abbreviations and acronyms that organizations use and the multiple interpretations that a single abbreviation or acronym might have. Because these names represent an important part of the metadata that is captured and managed, the adoption of naming standards for all assets and data elements is critical to your success.

Term construction and definition standards

Other standards for names and definitions apply to the glossary (controlled vocabulary) terms and their definitions. Terms are essential to the ability of the company to find information in the metadata repository. Terms and their definitions must not be ambiguous and, thus, must follow clear rules of how to form a term and how to define it.

The following standards are common for vocabulary construction and metadata creation:

- National Information Standard Organization (NISO) Z39.19-2005 “Guidelines for the Construction, Format and Management of Monolingual Controlled Vocabularies”
- ISO 11179-4 Formulation of Data Definitions

A company might apply other standards that concern security, privacy, access, and communication. The information governance council and information governance center of excellence have jurisdiction over these issues and the development and application of these standards.

A long list is available of the metadata standards that apply to the content and structure of data messages that apply to some applications in certain industries. Specific companies are required to comply with these standards for reporting purposes or to conduct business with peers, vendors, or customers. The maintenance of, and compliance with, metadata standards also fall under the information governance council, which might create a workgroup to address the issues that relate to these standards.

Policies and procedures

By using policies and procedures, management can achieve its goals without constant intervention in daily operations. Employees have a clear understanding about what can and cannot be done and how to perform tasks. Policies and procedures are critical to instilling trust in the work products that employees create or need to work from and are the foundation to compliance.

Along with standards information, governance implies the creation and enactment of policies that direct all stakeholders' actions on all aspects of the creation, preservation, and use of information. The information governance steering committee oversees the development, review, and approval of these policies for its own management and conduct and for the various operational areas over which it has jurisdiction. You can delegate part of this responsibility to the office of the Chief Information Officer (CIO).

On matters of information governance and metadata management, the information governance CoE assumes the responsibility to develop, disseminate, and enforce the required policies and procedures. Policies and procedures guarantee that all activities are completed properly, on time, and with transparency. Policies also guarantee that all activities are completed by the people assigned with the proper authority.

Policies are descriptive and include the following information:

- Explain the purpose of the policy.
- Identify the governing rules.
- Explain in which instances the policy applies.
- Detail to whom or to what the policy applies.
- Describe how to enforce the policy.
- Describe the consequences.

The information governance team develops a set of policies and procedures for the various aspects of its operation. Every entity in the information governance hierarchy has a set of policies and procedures that defines its authority and the way it must conduct activities in relevant matters to its area of responsibility and operations.

Information governance and metadata management might require the following policies:

- Information governance council:
  - Member nomination
  - Meeting schedule and protocol
  - Decision making
  - Reporting
- Information governance CoE:
  - Membership nominations and tenure
  - Meeting schedule and protocol
  - Decision making
  - Reporting

- Controlled vocabulary workgroup:
  - Taxonomy construction and validation guidelines
  - Term naming and definition standards
  - Member nomination and tenure policy
  - Data steward appointment guidelines
  - Taxonomy creation and validation:
    - Naming conventions
    - Category definition guidelines
    - Taxonomy creation and validation guidelines
    - New category policy
    - Category splitting policy
    - Category deprecation policy
    - Publishing and reporting
  - Controlled vocabulary term creation and maintenance:
    - Term identification and approval
    - Naming conventions
    - Term definition standards
    - Custom variables, term relationships, and referencing guidelines
    - New term policy
    - Term deprecating policy
    - Asset assignment
    - Publishing and reporting
- Metadata repository administration workgroup:
  - Asset import policy and protocol
  - Asset linking and stitching
  - Data and business lineage generation
  - User definitions and permission assignment
  - Backup and recovery procedures

Procedures prescribe how and when to perform tasks. The procedures provide clarity to the employees who are performing the tasks and often protect the organization from security, safety, and liability risks. Along with the policies that specify who can take action, when they can take it, and what actions to take, the procedures that are defined along these lines specify how to perform these actions.

Metrics

Metrics are an essential management tool. Without the ability to measure input and output, management cannot evaluate the productivity and effectiveness of its actions, make plans, or chart a path for future development. Metrics are the means by which management can set goals and monitor progress.

Data quality is the highest concern of information governance. The accuracy, completeness, and timeliness of data are essential to generate the trusted information that management needs for its planning and decision making. Metrics for these aspects of quality are usually easy to set and measure. The information governance CoE or another data quality entity sets clear goals for the acceptable levels of errors. It also monitors progress toward achieving these goals, preventing the deterioration of data quality.

Metadata management might present additional metrics to determine the success of its programs or the scope of coverage:

- *Coverage* is the volume of metadata that is captured in the repository. Coverage can include a count of the data sources, applications, jobs, and other artifacts for the terms that are captured and represented in the repository.
- The success of a program is measured by the frequency of users accessing the metadata repository to explore and search for information. Each search on a metadata management system replaces a less efficient search, which is often done manually, in documents or other resources. The frequent use of a metadata system indicates that individuals from IT and business collaborate to identify an explanation or a solution to a data issue. It indicates that one side attempts to understand how the other side interprets or realizes things.

1.6 Business scenarios for metadata management

Metadata management plays a critical role in conducting business. It might be understanding what a report is about and what the figures mean or asserting that the organization complies with rules and regulations. The following sections present several business scenarios where the role of metadata management is indisputably critical.

1.6.1 Metadata for compliance

Government and institutional regulations require companies to store increasing amounts of data for compliance. Two such examples of regulations are the Sarbanes-Oxley Act requirements on financial reporting and the Health Insurance Portability and Accountability Act (HIPAA) requirements on medical record keeping. For each set of requirements, organizations must maintain information in a certain form and show that they comply. Furthermore, many of these regulations make executives accountable, and failure to comply can result in severe penalties and even a jail sentence.

In the modern information environment, multiple systems communicate and exchange information across networks and new systems and applications are added at an increasing pace. In this environment, the task of tracking the information flow, as required by many of these regulations, is massive. Without the proper technology and discipline, it is easy to see how things can be missed.

For example, financial services companies must retain certain records over a specified period of time. They must also maintain a system to show an audit trail for each of these records and verify that records have not been altered.

Medical institutions in the US, under HIPAA, are subject to strict regulations concerning patient information. The disclosure of information to unauthorized individuals or organizations can subject the institution to censure and penalties. Medical institutions are required to show that patient information is protected and to show the manner in which access is restricted.

Metadata management can help in the following cases:

- Financial institutions in the US, under SEC 17a-4, must retain certain records for a specified period of time. This information is part of the record metadata attributes that are retained in the metadata system. The systems that maintain and archive data can consult these values. The same information is also available to individuals, such as auditors, analysts, and developers, who need to work on or work with these records to access the necessary information to determine how to handle the information.
- For medical information, privacy and security are attributes of records or fields in a database. They are also elements of the metadata about these data elements. You can use privacy or security values as triggers to rules that can enable or prohibit access to the information, depending on the type of user access.

Similar regulations concerning privacy, security, data retention, and disclosure exist in the industrial world and in many developing countries.

1.6.2 Metadata for risk management

The explosive growth of information presents growing challenges to management to mitigate the risk that is associated with information. A considerable part of this data is subject to security, compliance, and retention requirements. Failure to meet these challenges often results in financial loss and often legal or regulatory liability.

Sensitive information about customers, products, partners, employees, and more is stored in diverse locations and shared across networks. Breaching security is common, resulting in the loss of confidence and customer goodwill and often in legal and regulatory liabilities. Classifying data to determine the levels or types of security, privacy, and compliance that need to be applied is an essential step toward securing the data and avoiding unnecessary exposure. Implementing selective access privileges based on the type of data and the role of the user must complement this initiative.

Risks in data exist in many places and are manifested in multiple ways. A failure in a remote transaction system might result in corrupt data that enters the information flow. A misunderstanding of a report specification might result in the inclusion of wrong or irrelevant data in the report. Changes in organizational structure might result in the omission or duplication of data in a report. These and other factors cause erroneous data to find its way into reports for planning and decision making or for compliance with regulatory requirements.

The capability to validate that the correct data is used exists in the following abilities:

- To trace data to its sources
- To understand the data and its use
- To identify the stewards who can provide answers or rectify a problem

In today's normal information environment, in which metadata management systems are not yet prevalent, detecting and addressing these problems are not easy. The lack of a central metadata repository, which can be interrogated for ownership, source, flow, and transformations, makes the task hard and prohibitively expensive. Answering these questions means exploring databases, data flows, and data integration processes and involving multiple individuals who might have parts of the information.

By having a metadata repository and management system, the organization can capture this "data about the data" and users can query it for the answers to these types of questions.

1.7 Where to start

A metadata management program is a significant undertaking. It requires an organization to make a commitment and provide resources. Because metadata spans broad arrays of applications, data stores, metadata types, and technologies, the organization must provide a clear statement of its goals and how to get there. A metadata management program requires standards, processes, and procedures that prescribe and dictate what metadata will be captured and maintained and how. The organization must dedicate resources from both business and IT to achieving the goals.

A cornerstone of the metadata management initiative is a *repository* where metadata is captured and maintained. This repository is open to all stakeholders, where they can search, browse, and query to find information about data artifacts, their meanings, their sources, and their relationships to other data artifacts. Building this repository entails creating a partnership between business and IT, often with conflicting interests and priorities.

Companies usually have large numbers of applications, data stores, and many other forms of information storage and delivery. An attempt to capture them all in one effort is destined to fail. Instead, you must form a strategy and chart a roadmap, detailing what metadata will be captured, in what order, and by whom.

Management must develop and articulate a vision that is accompanied by a commitment of financial resources. Beyond a statement of goals and a charter, careful planning must occur, stating clear objectives for every phase of the process. Planning must include the following tasks:

- Selecting and prioritizing operational domains
- Identifying metadata resources
- Creating the organization, roles, and processes to feed and maintain the metadata repository
- Educating stakeholders and users on the available tools to access and probe the repository

Corporate action that involves setting goals; designating resources; establishing standards, processes, and procedures; and selecting the technology on which the repository will be built must precede the creation of the metadata repository. By following these initial steps, a typical process of creating and populating the metadata repository is an iterative process.

Industries structure their operations differently to fit the ways they operate and the kinds of tasks that they can perform. In the banking industry, you might see the subject areas of customers, customer accounts, human resources, marketing, credit and risk, and so on. Other subject areas exist in the airline industry. The primary service in the airline industry is to operate flights from one airport to another using airplanes. These companies focus on the subject areas of reservations, flights, operations, assets, customers, employees, and so on.

Each iteration that involves subject areas contains the following steps:

1. Review and inventory metadata resources.

   Identify and define the resources in the subject area. These resources might include data models, glossaries, applications, data stores, reports, and the individual stewards who own or know about these artifacts.

2. Prioritize and line up partners from business and IT.

   Classify and prioritize the resources relative to their importance to the metadata management program objectives. Certain resources might be identified as immaterial for future directions, and other resources might have a low impact on the present management processes, thus being deemed low priority. For the select metadata sources, recruit the business and IT individuals who are knowledgeable about the subject to support the effort.

3. Populate the metadata repository.

   Population of the repository entails a long list of activities to be undertaken by the various operational teams:

   a. Create a vocabulary for the selected subject area.
   b. Load the technical metadata from data stores, BI tools, data modeling environments, data integration projects, external stores, and processes.
   c. Load the operational metadata.
   d. Establish links where applicable to enable search, navigation, and lineage reporting.

4. Test the metadata, and deploy it for general usage.

   The metadata in the repository must pass the test of users. Business and IT must validate that the business glossary contains the correct terms of the domain with the proper definitions, interrelationships to other terms, assignments of data assets, and links to reflect lineage. Only release the repository to the users after a period of testing and correcting information.

5. Establish a metadata maintenance and administration regimen.

   The metadata that is loaded into the repository has to be maintained and refreshed periodically, because changes in the subject areas occur frequently. Data store changes must be updated, new reports need to be developed, and operational metadata must continue to be produced as jobs execute according to schedules. The metadata administration team monitors the system to guarantee smooth operations. The various teams continue to develop and enrich the knowledge base.

6. Disseminate the metadata knowledge, and train users.

   The utility of the metadata management system grows as the amount of content grows and the number of users expands. The information governance council promotes and enacts training programs to broaden the use of the facilities that are provided by the metadata management system by a larger community of users.

1.8 Conclusion

In conclusion, this chapter explains what information governance and metadata are. It explains why it is important to manage metadata and what is required to manage metadata.

Chapter 2, “Solution planning and metadata management” on page 29, describes a common use case of an information integration project. It highlights the different phases of planning and design. In particular, it emphasizes the different artifacts that are created and passed downstream as a manner of sharing and distributing the project information. These artifacts, glossary terms, data models, table definitions, and so on are part of the metadata that is maintained in the repository and that is made it available to both IT and business.

2

Solution planning and metadata management

An enterprise-wide information integration project requires cooperation and committed partnership between the business and IT groups and personnel. To ensure a successful implemented solution, a well-planned and well-executed end-to-end solution plan must be in place. Key to successful implementation is open channels of communication and clear understanding of the requirements among the various groups involved in the project. Another key factor includes accurately translating requirements into specifications and from specifications into programs.

Information integration projects usually consist of four phases: understanding, cleansing, transforming, and delivering the information. These phases lead to a typical, more practical implementation process that serves for both general information integration and metadata management purposes.

This chapter includes the following sections:

- Getting started with solution planning
- Stakeholders
- Integrated solution data flow
- Typical implementation process flow
- Conclusion

2.1 Getting started with solution planning

Everything begins with a plan. A comprehensive plan determines and gauges the success of any project or task and is, therefore, required and critical. A plan further plots a course of action, leads to execution, and achieves the desired goals with a high level of success.

Plans are everywhere, not only in commerce and industry. They guide daily activities and aid in navigating often difficult or complex tasks. Plans include a strategy, an owner or an approval and acceptance process, and a measured set of goals for determining success. Plans must define a clear and attainable objective.

2.1.1 Information integration solution

Chapter 1, "Information governance and metadata management" on page 3, introduces information governance, metadata, and the need for *metadata management.* Organizations engage in various data processing initiatives. Often these initiatives involve data cleansing and integration. Data from multiple sources is cleansed and consolidated into a single data warehouse that can be used for an array of applications, from managing customer relations, to planning and decision making, to regulatory compliance.

During these processes, the organization harvests data sources for information about the data characteristics such as sources, format, type, and meaning. More information is created while developing the cleansing procedures and transformation processes to feed the target store. Ultimately, an information integration project creates and consumes information about data, processes, meaning, and business rules. This information, which for simplicity is called *metadata*, serves the project itself and many subsequent projects that will use the information for new purposes.

An information integration solution generally consists of four phases: understanding, cleansing, transforming, and delivering data. Figure 2-1 shows the four main phases of an information integration solution.

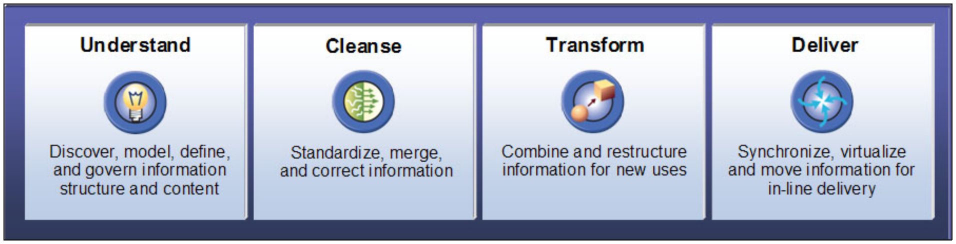

Understand
Discover, model, define, and govern information structure and content
Cleanse
Standardize, merge, and correct information
Transform
Combine and restructure information for new uses
Deliver
Synchronize, virtualize and move information for in-line delivery

Figure 2-1 Information integration solution: Four-phase approach

At the *understanding* phase, you identify data sources, capture enterprise definitions and requirements, and perform data relationship discovery. At the *cleansing* phase, you create data rules, perform quality assessment, ensure data accuracy, and eliminate duplicate data. At the *transforming* phase, you document the business rules, specify functions and data storage structures, and implement data development activities to support the transformation and movement of data. At the *delivering* phase, you deliver the necessary informational reports and a cross-enterprise vision of the data assets included for a given project. You also deliver the applied meaning and usages of such assets.

When you plan an information integration solution, you go through these four phases. To make the planning process more concrete and practical, you derive a typical implementation process as explained in 2.4, “Typical implementation process flow” on page 34.

2.1.2 Information integration project

An information integration project represents all the elements involved in realizing the information integration solution to achieve the desired goals of the project and to satisfy business requirements. After the initial information integration landscape is established, subsequent information integration projects can use a similar process for implementation, as explained in 2.4, “Typical implementation process flow” on page 34.

Regulatory agencies and internal auditing or corporate policies additionally provide input in the design of the project. Regulations dictate the rules that govern the storage or aggregation of information, the quality assessment criteria, and required identification methods. Corporate policies determine how data can be stored and accessed. They also define exact standards for data privacy and the means for monitoring data accuracy.

Strategic initiatives that drive adoption of a project plan must recognize the value and criticality of information to create a competitive advantage. Such initiatives must also commit the necessary resources to manage and deliver an information integration solution.

2.2 Stakeholders

When creating an information integration project, a key challenge is the acceptance, adoption, and implementation of the process. Stakeholders are a critical factor for the success of this type of the project, which explains why they are highlighted in this section before exploring the typical implementation process.

All stakeholders must be involved in the process for the following reasons:

- To ensure adherence to regulatory standards and compliance with corporate policies
- To address general concerns surrounding data quality, standardization and privacy
- To make data available for reporting and analytics

The primary stakeholders of an information integration project are *consumers*. Consumers can include executives, analysts, administrators, auditors, developers, and others who are interested in viewing and analyzing a catalog of information assets, their meaning, and their usage.

For example, an executive who receives a weekly summary report of high value customers might want to validate the enterprise definition of such a concept and the rules for calculating such information. This executive might also want to identify the originating sources of record contributing to the report and know when such sources where last updated.

Stakeholders are typically involved as either producers or consumers of information resources. They drive the strategic objective. Each stakeholder represents an individual or group within the enterprise with a specific task assignment or requirement. For a given information integration project, it is critical to identify the stakeholders, their given requirements, and their sponsors.

The following set of stakeholders and their responsibilities might be identified:

- Project leaders and owners:
  - Manage the data governance initiative.
  - Capture and align the business requirements and objectives.
  - Define and maps of the information integration project.
  - Illustrate and specify the subsequent process flow and supporting tasks.
  - Assign tasks, owners and responsibilities.
  - Monitor and respond to progress status.
- Business users:
  - Specify business requirements and objectives.
  - Review business intelligence (BI) reports for analysis.
- BI developer, data modeler, data developer, analyzer, and data administrator:
  - Develop and capture data model requirements.
  - Develop and capture database and data source requirements.
  - Develop and capture BI report requirements.
  - Develop data quality rules and quality assessment routines.
  - Maintain the integrated solution.

- Glossary author or subject matter expert
  - Author and structure business vocabulary capturing requirements.
  - Apply and enforce business requirements.
- Data quality steward
  - Apply and enforce quality standards and regulations.
  - Discover and document source data.
  - Profile and assess the quality of data stores.
  - Develop and monitor data rules.
- Metadata steward
  - Manage, load, and maintain a metadata repository.
  - Support lineage administration tasks and publication.
  - Support metadata reporting tasks and requirements.

2.3 Integrated solution data flow

In addition to involving the stakeholders in the process, another critical piece of solution planning is understanding the data flow within an integrated solution. Figure 2-2 shows the data flow of an integrated solution that guides a data integration implementation.

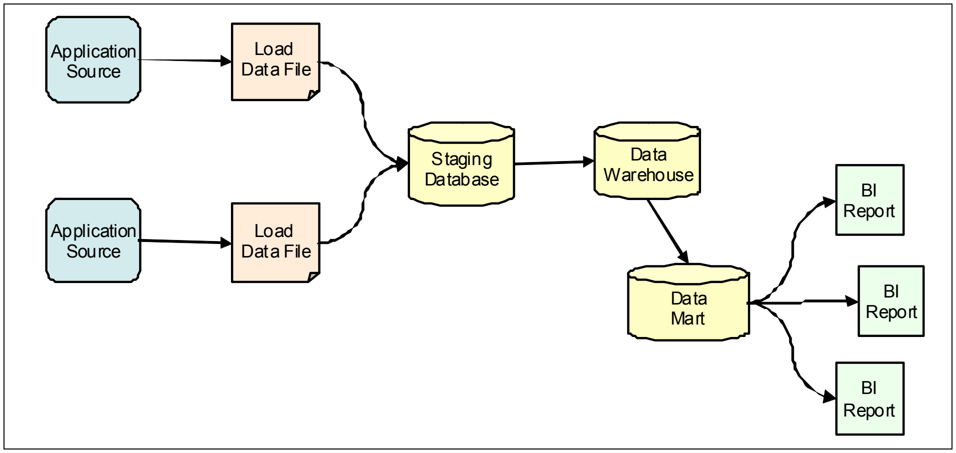


Figure 2-2 Data flow of an integrated solution

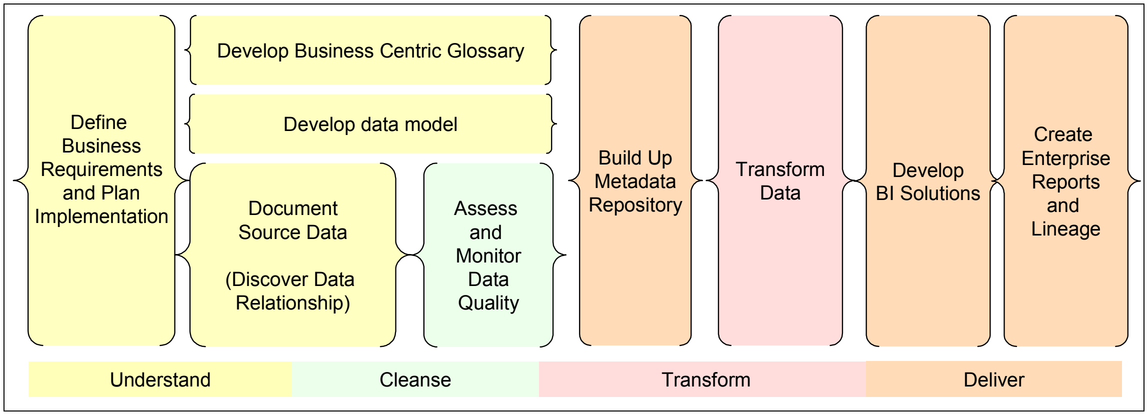
As illustrated in Figure 2-2, application source data is identified, documented, and loaded as data files. The content is then moved to the staging database. Transformation jobs and other data cleansing jobs then transform the data from the staging area to the data warehouse and data mart for final BI reports and analysis.

The metadata model that defines and maps all data sources, systems, and structures of the business is abstracted and referenced in the solution. This model forms a catalog or directory of informational assets that reference the specifications and logic that have been applied to each asset. The metadata model forms the basis of greater business understanding and trust of the information.

The metadata model and the understanding of the data flow facilitate the implementation of an information integration solution.

2.4 Typical implementation process flow

To implement an information integration solution, the four-phase approach shown in Figure 2-1 on page 30 (understanding, cleansing, transforming, and delivering) can be divided into more specific, practical implementation processes. Figure 2-3 illustrates a typical implementation process for an information integration solution with specific processes within the overall process.



Define Business Requirements and Plan Implementation
Develop Business Centric Glossary
Develop data model
Document Source Data
(Discover Data Relationship)
Assess and Monitor Data Quality
Build Up Metadata Repository
Transform Data
Develop BI Solutions
Create Enterprise Reports and Lineage
Understand
Cleanse
Transform
Deliver

Figure 2-3 Typical information integration implementation process

The typical implementation process collects information from disparate sources, transforms and consolidates the information, and makes it available for distribution, analysis, and reporting. The implementation includes a set of the specific processes that produce and consume data. Each process is critical to business understanding and decision making. Each process is also managed by a set of consumers who either input or extract data that contributes to the quality and understanding of the data results.

Defining data standards addresses the challenges found in data consolidation, such as inconsistent data formats and structures. Stakeholders must agree upon standards and apply them throughout the process. Further, data standardization benefits when a fully defined enterprise glossary exists, imparting the agreed upon meaning, structure, and format for the data assets.

Depending on your business requirements and system environment, the actual processes that you implement and the order of the processes you implement might vary. Some of the processes might occur in parallel. In almost all instances, the overall implementation process contains iterations. This book provides a typical implementation process, but the actual implementation might have many variations.

Implementation process in this book: The rest of the book centers around this typical implementation process. Chapter 3, “IBM InfoSphere Information Server approach” on page 41, explains which InfoSphere Information Server product modules and components are available to perform the individual steps. Chapter 4, “Use-case scenario” on page 77, and later chapters highlight the use-case scenario and its implementation steps.

Each process, within the overall process, calls for a distinct action. Each process must include an objective, specification, and a clearly identifiable owner who is responsible for implementing and documenting the associated tasks and for monitoring the results.

Multiple valid approaches likely exist for defining the implementation process and the associated process flow. The overall implementation process flow is unique according to the enterprise, industry, and varied objectives and requirements. It creates a clear and concise set of tasks and objectives with which to monitor success.

The typical implementation process flow includes the following steps (processes):

- Define the business requirements.
- Build a business-centric vocabulary.
- Develop a data model.
- Document the source data.
  - Discover the data relationship.
- Assess and monitor data quality.
- Build up a metadata repository.
- Transform data.
- Develop BI solutions.
- Generate enterprise reports and lineage.

2.4.1 Defining business requirements

Gathering and documenting business requirements is a crucial first step of any integration project. Such requirements capture the business objectives and define the expected result in addition to setting quality standards or metrics.

Business requirements account for the needs of the business to develop quality information and the corporate or regulatory standards of data storage, identification, and reporting. Requirements are further gathered from the various stakeholders, who represent different entities of the organization.

Requirements are best communicated through written documentation, where stakeholders can review, comment, and approve them. In addition, the processes and tasks required to support the requirements are further documented, specified, and reviewed.

The result of this step is the documented and approved requirements for the subsequent clearly defined tasks and processes, with their stated owners, scheduling, and expected achievement and result.

2.4.2 Building business centric vocabulary

A lack of a shared understanding of documented key performance indicators (KPI) can create problems with understanding information assets and, therefore, can hinder the development process. Building a business-centric vocabulary allows for the sharing of common definitions across physical or logical organizational boundaries.

Business definitions can reference industry standard definitions, requirements, or processes as a means of ensuring compliance and usage of such standards. All stakeholders must share this common definition of information as defined by the business and used in the process flow. All employees must share a common understanding of such information and have the ability to contribute to and collaborate on this information.

The result of the process is the published business-centric vocabulary (business glossary) and other business metadata that captures the enterprise definitions or requirements, their usages, and specifications in a structured manner.

2.4.3 Developing data model

Data modelling is the technique used to analyze and interpret the data storage requirements to support the project goal. A *data model* provides a template or blueprint for data storage systems, detailing how information is structured, mapped, and linked. Data models form the backbone of a centralized data warehouse that consolidates data for reporting and analytics.

Data modelling also helps to resolve issues of data incompatibility by providing the data structures that are used to store or access data. It defines how applications interface with data structures and applies business meaning or requirements to such data structures.

The result of this process is a data model that is used for final reporting and analysis of the integration project and metadata management.

2.4.4 Documenting source data

It is important to document source data that is required for data integration. Source systems represent the input (or pipe) of data from their point of origin to a staging area for data cleansing and transformation. This process is important because it provides a mechanism to identify, retrieve, and transfer data from disparate source systems and applications to a modelled data storage system, where data is staged. Data is typically transferred as is without transformation or change into the data storage system. Subsequent processes assess the data quality, apply data rules, and cleanse and transform the data into the required data warehouse and data marts.

Discovering data relationships

Before acting on the data, you must fully *understand* it. Understanding data means that you must understand the data content, data relationship, and data transformation within and across multiple heterogeneous sources. Understanding data also implies discovering business objects within and across data sources and identifying sensitive data within and across data sources.

The result of the process for documenting source data is a normalized data storage system that contains documented, mapped, and transferred source data.

2.4.5 Assessing and monitoring data quality

The *cleanse* phase follows the *understand* phase. In the cleanse phase, one of the steps is assessing data quality. In this step, you establish a set of data acceptance criteria and corrective measures to standardize data formats, ensure data quality, and achieve completeness. These measures help meet the corporate or regulatory requirements for information governance and support the health and growth of the business when using the data for reporting or analytics.

When data is collected, a broad range of data quality issues can arise, requiring a defined strategy for addressing data duplication, inconsistency, and invalidity. Source systems and applications store data in various structures, formats, and systems. This storage method causes concern about data content and quality of the data. You need to identify similar content from multiple sources, duplicating identical structure, consolidating content, and standardizing data format.

A proactive approach to data quality management with clearly stated requirements helps ensure high-quality, trusted information. The idea is to measure, monitor, and repair data issues before making the data available for analytics and reporting.

Ensuring data quality mandates continually tracking, managing, and monitoring data quality levels to improve business efficiency and transparency. Therefore, the ability to measure and monitor data quality throughout the development or project life cycle is essential. It is also important to compare the results over time.

Data monitoring is a framework that effectively monitors the results of data rules, data profiling, or other such procedures. Data monitoring is used to identify and distinguish invalid, incomplete, or misformatted data that might otherwise compromise the objective for reporting and analytics.

A formalized method for defining standards, measuring compliance, and effectively communicating and reacting to results is a required process for implementing the project. Quality metrics further provide a unified and consistent understanding of data quality. They act as a base to adhere to regulatory requirements for reporting and data storage.

This approach defines the process to effectively assess and monitor data quality, review the quality metric, and react to changes in requirements.

2.4.6 Building up the metadata repository

Collecting, documenting, and storing your core system data in the metadata repository is crucial for understanding the assets. When building up the metadata repository, it must also include such information as how the data relates to the business terminology and requirements and how it is used within the development or reporting systems. Such documentation describes the assets involved in the information integration project, their context, meaning, and specification.

In this process, you build up the metadata repository by documenting, collecting, and loading staged database to metadata repository for final reporting and analysis. The metadata repository serves as the point of communication between business users, stakeholders, developers, and IT owners. It allows them to benefit from a unified and representative system of record.

The result of this process is a central repository for the data, including the business vocabulary and requirements.

2.4.7 Transforming data

The process of transforming data between storage systems and data stores is a required step for centralized reporting, analytics, and metadata management. This process depends on the modelling and normalization of the data storage systems to which the data is transferred. This process also provides an understanding of the data structures, quality, and structures of the source data systems.

This process includes the specification of how data is extracted, transformed, or aggregated prior to loading into the target data store. It also includes implementing the defined requirements and regulations for data quality, format, measurement, regulation, and storage.

2.4.8 Developing BI solutions

The process for developing BI solutions includes developing reports, services, and applications that share and publish information. This process requires a clear specification and requirement that defines the data that is required and the expected format or structure. Reports represent the data that is aggregated from the data storage systems.

A BI solution encompasses an enterprise view of information, rather than a departmental or silo approach. The solution must include quality controls to ensure that the data represented is current and accurate. This data must be fully qualified in order to understand its intended meaning or usage and trace its pedigree to the source data systems.

2.4.9 Generating enterprise reports and lineage

Consumers want to trust the information they are reporting upon, developing against, or visualizing within the warehouse repository. Further, it is not sufficient to only understand the defined meaning of information, the data usage, structure and quality score. A clear understanding of the data provenance is also required.

By providing data lineage, stakeholders can inspect the source data systems of a BI report. This inspection includes the development specification and process that transformed the data and the quality assessment and business meaning of the data storage systems that are displayed. With data lineage, the stakeholders can also derive value and understanding from complex, heterogeneous information and can include this information within development initiatives or reports and analytics.

This step defines the processes to deliver and support the data lineage requirements of the stakeholders. It includes the capability to search and extract information from within the metadata repository.

2.5 Conclusion

In conclusion, this chapter introduced a typical implementation process for an information integration solution that also serves the purpose of metadata management.

Chapter 3, “IBM InfoSphere Information Server approach” on page 41, introduces the product modules and components that are offered by IBM InfoSphere Information Server. Specifically, 3.4, “Solution development: Mapping product modules and components to solution processes” on page 58, explains how each product module and component can be used in the typical implementation process.

3

IBM InfoSphere Information Server approach

IBM InfoSphere Information Server is a product family that provides a unified data integration platform so that companies can understand, cleanse, transform, and deliver trustworthy and context-rich information to critical business initiatives. InfoSphere Information Server offers various product modules and components that provide an integrated end-to-end information integration solutions.

The previous chapters focus on the need and a recommended approach for information integration solutions. This chapter further expands on the approach and maps individual implementation process with the product module or component that is provided by InfoSphere Information Server.

This chapter includes the following sections:

- Overview of InfoSphere Information Server
- Platform infrastructure
- Product modules and components
- Solution development: Mapping product modules and components to solution processes
- Deployment architecture and topologies
- Conclusion

3.1 Overview of InfoSphere Information Server

InfoSphere Information Server is built as a multi-tiered platform that includes a collection of product modules and components that primarily focus on different aspects of the information integration domain. Furthermore, it integrates with other third-party applications to use information, wherever it exists in the enterprise.

InfoSphere Information Server supports a range of initiatives, including business intelligence (BI), master data management, infrastructure rationalization, business transformation, and risk and compliance.

Business intelligence

InfoSphere Information Server makes it easier to develop a unified view of the business for better decisions. It helps in understanding existing data sources; cleansing, correcting, and standardizing information; and loading analytical views that can be reused throughout the enterprise.

Master data management

InfoSphere 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 across systems. This master record can be loaded into operational data stores, data warehouses, or master data applications. The record can also be assembled, completely or partially, on demand.

Infrastructure rationalization

InfoSphere Information Server aids in reducing operating costs by showing relationships among 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

InfoSphere 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

InfoSphere 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.

3.2 Platform infrastructure

InfoSphere Information Server consists of a robust, scalable, server architecture that is built on three distinct components as illustrated in Figure 3-1:

- A Java 2 Platform, Enterprise Edition (J2EE) application server
- A database repository
- A parallel processing runtime engine

Both the application server and the repository are standard server applications. Most enterprises already have the skills to manage and administer these server applications, particularly because the InfoSphere Information Server infrastructure is designed for minimal intervention.

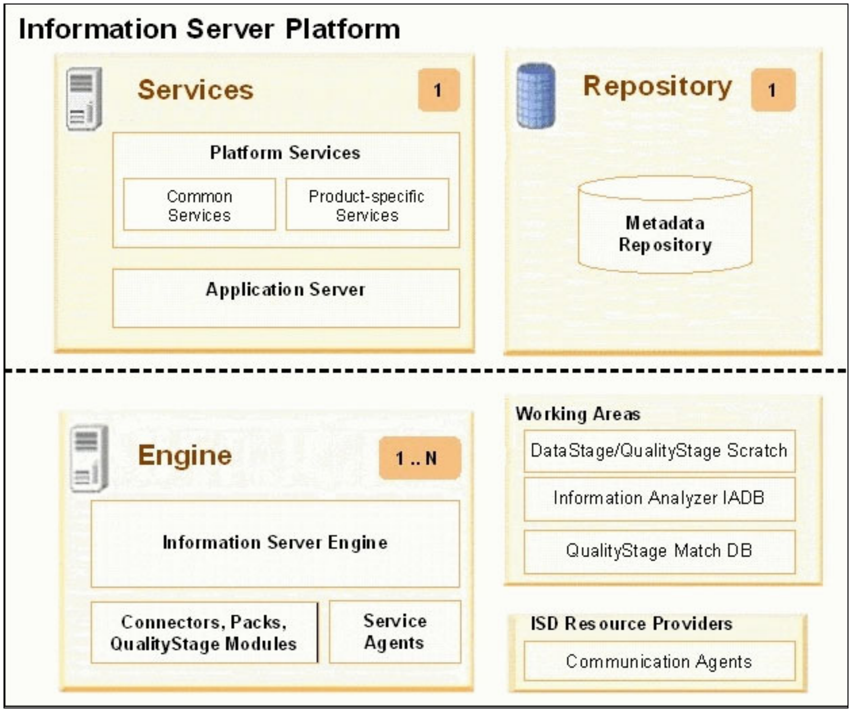


Figure 3-1 High-level architecture of InfoSphere Information Server

One of the advantages of InfoSphere Information Server is this shared infrastructure for all the individual InfoSphere Information Server product modules. This shared infrastructure enables integration of all these components, thereby, minimizing duplicate effort and maximizing reuse. In addition, InfoSphere Information Server integrates metadata from external applications with its metadata repository. This way, the InfoSphere Information Server product modules can use this external metadata to achieve business objectives.

3.2.1 Services tier

The InfoSphere Information Server services tier is built on IBM WebSphere® Application Server. WebSphere Application Server provides infrastructure for common services across all of the modules, such as authentication and repository access. It also provides infrastructure for the services, web applications, or both that are proprietary to the individual product modules and components. Figure 3-2 shows some of the product module-specific services and common services. Only the services of the installed product modules are available.

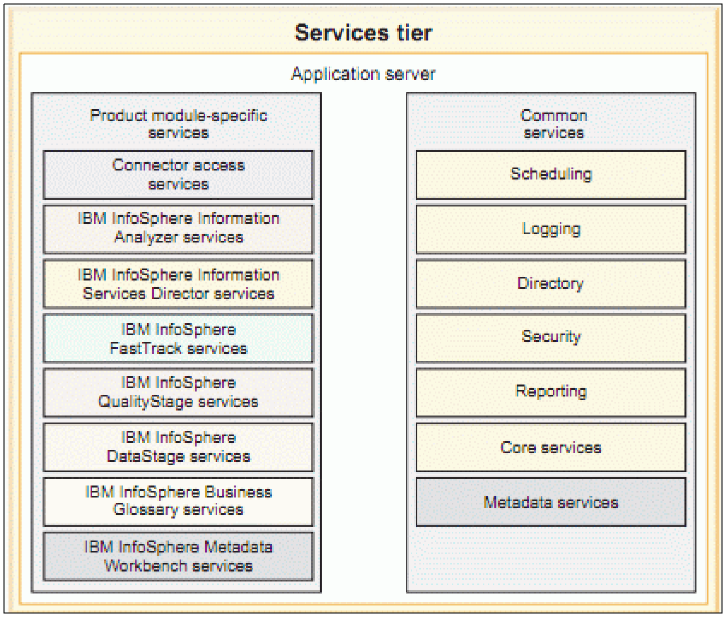

Services tier
Application server
Product module-specific services
Connector access services
IBM InfoSphere Information Analyzer services
IBM InfoSphere Information Services Director services
IBM InfoSphere FastTrack services
IBM InfoSphere QualityStage services
IBM InfoSphere DataStage services
IBM InfoSphere Business Glossary services
IBM InfoSphere Metadata Workbench services
Common services
Scheduling
Logging
Directory
Security
Reporting
Core services
Metadata services

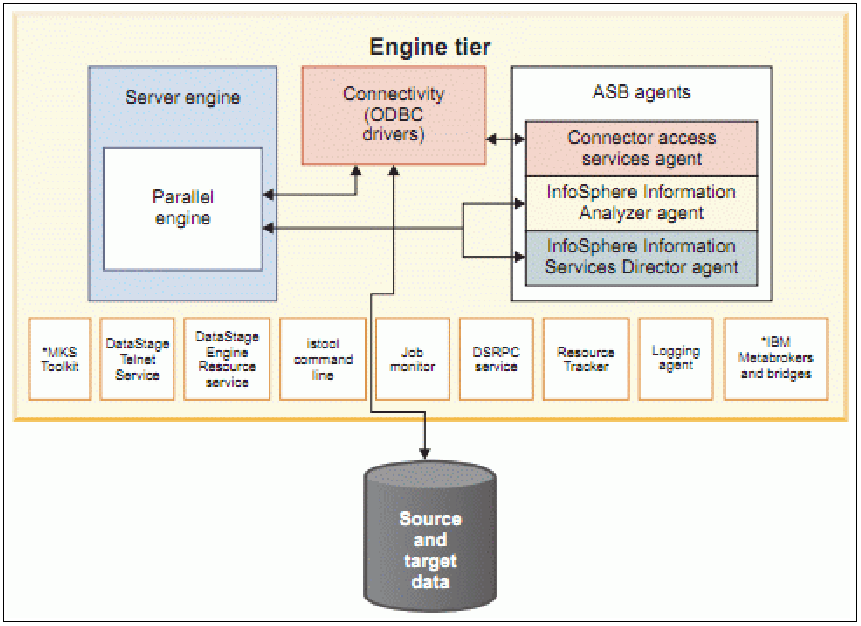
Figure 3-2 Services tier of InfoSphere Information Server

Security is one of the major functions managed by WebSphere Application Server. When InfoSphere Information Server is first installed, WebSphere Application Server automatically configures the InfoSphere Information Server internal user registry (persisted in the InfoSphere Information Server repository) as the default user registry. Then, it is possible to reconfigure WebSphere Application Server to authenticate through a different user registry, such as the operating system or Lightweight Directory Access Protocol (LDAP) (for example, Active Directory), to simplify the administration of users and groups. However, regardless of the user registry used, the InfoSphere Information Server roles and privileges are maintained in the InfoSphere Information Server repository. As such, WebSphere Application Server is the only application that has the authority and credentials to access the repository.

3.2.2 Engine tier

The InfoSphere Information Server engine tier provides parallel processing and runtime functionality for several InfoSphere Information Server product modules. This functionality is the basis for virtually unlimited scalability, because the only limit to processing capability is the available hardware.

The engine tier, illustrated in Figure 3-3, includes built-in data connectivity (Open Database Connectivity (ODBC) drivers) to external data sources whether for data cleansing, profiling, or transformation. In addition, InfoSphere Information Server includes native drivers for many data sources, taking advantage of better performance and functionality. Because it is often necessary to process millions of records, the parallel processing engine provides an efficient method for accessing and manipulating different data.



Engine tier
Server engine
Parallel engine
Connectivity (ODBC drivers)
ASB agents
Connector access services agent
InfoSphere Information Analyzer agent
InfoSphere Information Services Director agent
*MKS Toolkit
DataStage Telnet Service
DataStage Engine Resource service
istool command line
Job monitor
DSRPC service
Resource Tracker
Logging agent
*IBM Metabrokers and bridges
Source and target data

Figure 3-3 Engine tier of InfoSphere Information Server

3.2.3 Repository tier

The InfoSphere Information Server repository tier is built on a standard relational database management system (RDBMS, including IBM DB2®, Oracle, or MS SQL). The repository tier provides the persistence layer for all of the InfoSphere Information Server product modules and components.

The *metadata repository* (in the repository tier) of InfoSphere Information Server is a single repository. In this repository, information about data source, data integration jobs, business glossary, data warehouse and data mart, reports, and report models is brought into one shared location. This location can be used by InfoSphere Information Server product modules and components within the suite. Sections and chapters later in this book describe how each product module generates and uses the shared metadata repository.

Additionally, to ensure data integrity and processing performance and to provide temporary persistence, two InfoSphere Information Server product modules (IBM InfoSphere QualityStage and IBM InfoSphere Information Analyzer) also use their own schema as a workspace (Figure 3-4). When the work is done, the relevant metadata in the workspace is published to the shared metadata repository, at user designated intervals, to be used by other product modules.

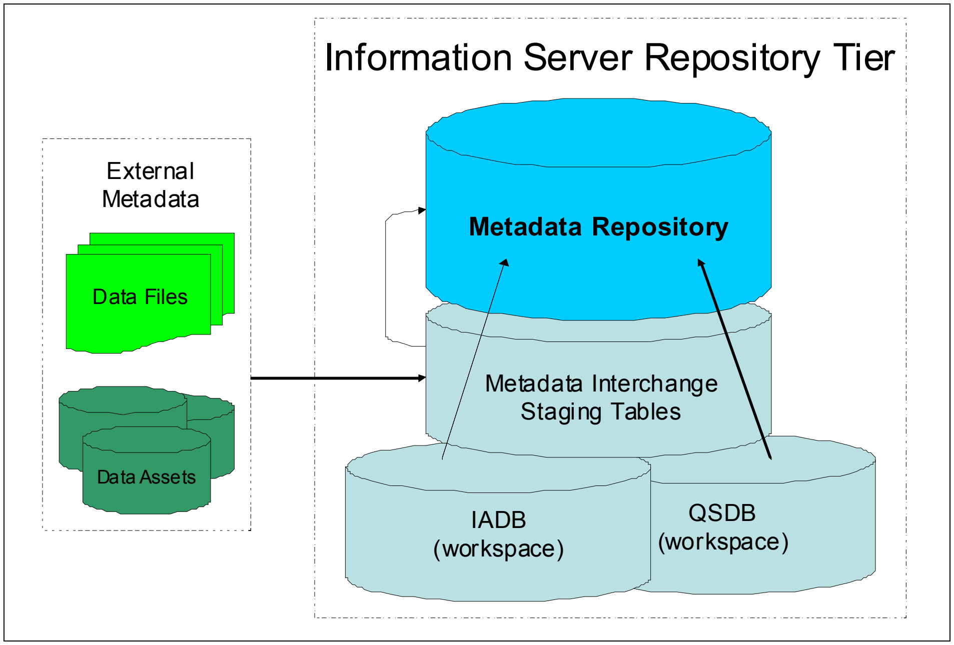


Figure 3-4 Repository tier of InfoSphere Information Server

The metadata in the metadata repository is usually internally generated data. It is also possible to import, load, and append metadata from external applications. Such applications includes data modelling tools, BI applications, and other sources of structured data that have relevance to one or more InfoSphere Information Server product modules. InfoSphere Information Server provides a mechanism (staging tables and application code) for users to import this third-party metadata into a staging area, referred to as the *Metadata Interchange Server*. The Metadata Interchange Server provides an interim load area for external metadata. Here, the new metadata can be compared to existing metadata (before the new metadata is loaded) in order to asses the impact and act upon it.

In a case where an external application does not provide a means to export or access its metadata, you can document these relevant external information assets in the repository, through controlled, manual processes and tools.

3.3 Product modules and components

InfoSphere Information Server offers a collection of product modules and components that work together to achieve business objectives within the information integration domain. The product modules provide business and technical functionality throughout the entire initiative from planning through design to implementation.

InfoSphere Information Server consists of the following product modules and components:

- IBM InfoSphere Blueprint Director
- IBM InfoSphere DataStage
- IBM InfoSphere QualityStage
- IBM InfoSphere Information Analyzer
- IBM InfoSphere Discovery
- IBM InfoSphere FastTrack
- IBM InfoSphere Business Glossary
- IBM InfoSphere Metadata Workbench

This section highlights each of the product modules along with their general functionality, the metadata associated with them, and the related application or product module that is integral to InfoSphere Information Server projects.

InfoSphere Information Server utilities consists of InfoSphere Information Server Manager, ISTools, and IBM InfoSphere Metadata Asset Manager, which are also highlighted in this section. This section also includes a brief introduction of IBM Cognos® Business Intelligence software, which is not part of the InfoSphere Information Server. However, by addressing this software in this section, you have a more comprehensive view of all the product modules, components, utilities, and products that can assist in an information integration project.

3.3.1 InfoSphere Blueprint Director

InfoSphere Blueprint Director is a graphical design tool that is used primarily for creating high-level plans for an InfoSphere Information Server based initiative. Such initiatives can be in information governance, information integration, BI, or any other information-based project. To make the task simpler, InfoSphere Blueprint Director comes bundled with several, ready-to-use, and project-type-based content templates that can be easily customized to fit the project. Alternatively, a new blueprint can be created from scratch as needed but is strongly discouraged.

InfoSphere Blueprint Director has a design canvas onto which standard graphical objects, representing processes, tasks, or anything else, are dragged and dropped. Objects can be connected one to the other, implying a sequential order to the events or dependencies between them. Each graphical object has a label that indicates its purpose. However, the object can optionally be linked to content that was produced and published in IBM Rational® Method Composer. When a single object represents several tasks or processes, the object can drill down to create or link to a more detailed layer of the blueprint diagram. This way, the final blueprint is likely to contain several hierarchical levels of processes (and subprocesses). The hierarchical blueprint diagram, combined with the methods (text descriptions), forms the basis of the project plan as built from top to bottom (high to low level).

InfoSphere Blueprint Director is a unique component among the InfoSphere Information Server product modules and components. It is a stand-alone, Eclipse-based, client-only application that does not have any dependencies on the InfoSphere Information Server infrastructure for persistence, authentication, or other shared services. This component provides useful flexibility for planning the project at an early stage before all of the infrastructure is ready and available.

InfoSphere Blueprint Director does not generate any metadata that is currently consumed by any of the InfoSphere Information Server product modules and components. It has its own persistence layer in an Extensible Markup Language (XML) format (`*.bpd`) file. However, it has the facility to link to InfoSphere Information Server repository objects. It can also launch InfoSphere Information Server product modules and components to display these linked objects in their native tool.

For more information about how InfoSphere Blueprint Director works, and for specific use cases, see Chapter 5, “Implementation planning” on page 89.

3.3.2 InfoSphere DataStage and InfoSphere QualityStage

InfoSphere DataStage and InfoSphere QualityStage provide essential integration functionality for a range of projects. InfoSphere DataStage integrates data across multiple source and target applications, collecting, transforming, and delivering high volumes of data. InfoSphere QualityStage provides data cleansing functionality from standardization, to deduplication, to establishing master data. Both product modules take advantage of the parallel processing runtime engine to execute their jobs efficiently, in a scalable manner that optimizes data flow and available system resources.

InfoSphere DataStage and InfoSphere QualityStage share the same rich user interface, the *Designer*. This application provides a straightforward method to design jobs that extract data from various data sources. It also processes (or transforms) the data (according to its functionality) and loads the resultant data into the relevant target data storage systems. InfoSphere DataStage and InfoSphere QualityStage share two additional rich client applications: the Administrator and the Director, for administering projects, testing, executing, and troubleshooting jobs. They also share a new web application, called the *Operations Console*, that provides an additional runtime job matrix that was not previously available in a GUI format.

InfoSphere DataStage and InfoSphere QualityStage projects are persisted in the metadata repository. This portion of the repository storage model contains job designs and other project artifacts. For example, it might contain shared table definitions, containers, parameter sets, and connector objects. The design metadata and sometimes the runtime operational metadata of InfoSphere DataStage and InfoSphere QualityStage are key components in calculating impact and lineage reports. Besides generating design and operational metadata, InfoSphere DataStage and InfoSphere QualityStage jobs consume physical metadata to populate table definitions that are used to design InfoSphere DataStage and InfoSphere QualityStage stage links within their jobs.

For more information about InfoSphere DataStage and InfoSphere QualityStage, see Chapter 11, “Data transformation” on page 375.

3.3.3 InfoSphere Information Analyzer

InfoSphere Information Analyzer helps organizations assess and monitor data quality, identify data quality concerns, demonstrate compliance, and maintain audit trails. It is a rich client that performs two functions: quality assessment (with data rules) and data profiling (with column analysis). InfoSphere Information Analyzer requires the metadata from these data sources to reside in the repository before performing these activities. Importing this technical metadata can be done from InfoSphere Information Analyzer or from an InfoSphere Information Server utility tool called *InfoSphere Metadata Asset Manager*. (For more information, see 3.3.8, “InfoSphere Information Server Manager, ISTools and InfoSphere Metadata Asset Manager” on page 54.)

InfoSphere Information Analyzer uses the connectivity and parallel processing functionality of the InfoSphere Information Server Parallel Engine to query the data sources and load the column analysis values in the InfoSphere Information Analyzer database workspace. InfoSphere Information Analyzer then performs the profiling analysis discretely on each column. All of the analysis results are persisted in the database workspace until it is refreshed by rerunning the process. However, a baseline can also be established for comparison. When the work is complete, the user can publish the results from the database workspace to the shared repository, where other product modules and components (such as InfoSphere DataStage and InfoSphere FastTrack) can use it.

The manner in which InfoSphere Information Analyzer assesses the level of data quality is by establishing constraints (data rules) to which the data should (or should not) adhere and then test whether the data complies with these rules. This functionality serves ongoing monitoring of the data (trend analysis), which is often part of a data quality and an information governance initiative.

For additional information about how InfoSphere Information Analyzer works, and for specific use cases, see Chapter 9, “Data quality assessment and monitoring” on page 283.

3.3.4 InfoSphere Discovery

InfoSphere Discovery is an automated data relationship discovery solution. It helps organizations to gain an understanding of data content, data relationships, and data transformations; to discover business objects; and to identify sensitive data within and across multiple heterogeneous data stores. The automated results derived from InfoSphere Discovery are actionable, accurate, and easy to generate, especially when compared to manual (non-automated) data analysis approaches that many organizations still use today.

InfoSphere Discovery works by *automatically* analyzing data sources and generating hypotheses about the data. Throughout the process, it interrogates the data and generates metadata that includes a *data profile* or *column analysis*.

In the context of an information integration project, this process provides an understanding of data and their relationships. It can be used for governance or to help in source-to-target mapping as a planning aid to data integration specification.

InfoSphere Discovery directly accesses the source data systems to get the data for data relationship discovery purpose. When the results are ready to be shared in the InfoSphere Information Server metadata repository, specifically for InfoSphere Business Glossary and InfoSphere FastTrack, its export and import utility can publish the results to the shared metadata repository. These results enhance the physical metadata, associate business terms with physical assets, and assist in designing mapping specifications for data integration.

For more information about how InfoSphere Discovery works and for specific use cases, see Chapter 8, “Data relationship discovery” on page 209.

3.3.5 InfoSphere FastTrack

InfoSphere FastTrack is a rich client that creates source-target (or technically target-source) mapping specifications to be used by developers of data integration jobs. The main window of the client application provides a spreadsheet-like, columnar structure. Here you can enter (by copying and dragging) objects from the repository display to cells for source columns, target columns, and assigned business terms. You can also enter manual descriptions of source to target transformations and, optionally, InfoSphere DataStage Transformer Stage-specific code.

The completed mapping specification can be output in several formats, including an annotated InfoSphere DataStage job generated directly by InfoSphere FastTrack. This format is useful for the InfoSphere DataStage developer because the specification is delivered in a manner in which the developer is familiar. In addition, this delivery format provides a job “template” that can be used as the basis for creating a new job, including design artifacts that can be copied to the new job as is.

Using InfoSphere FastTrack for mapping specification documentation includes the following additional advantages:

- Centrally stored and managed specifications
- Simple drag-and-drop functionality for specifying source and target columns
- Accuracy of source and target column names (that exist in the repository) with assured correct spelling
- Discovery of mappings, joins, and lookups assistance, based on published data profiling results, name recognition, and business-term assignment
- Use of a persistence layer in a shared repository in lineage reports

For more information about InfoSphere FastTrack, see Chapter 11, “Data transformation” on page 375.

3.3.6 InfoSphere Business Glossary

InfoSphere Business Glossary primarily uses a thin client browser approach and several interfaces to help users share information across the organization. The basic content is a glossary of terms with their definitions, organized into categories that provide containment, reference, and context. Both terms and categories include several descriptive attributes. They also include other attributes that define relationships to other InfoSphere Information Server repository objects (physical and business-related metadata), information stewards, development cycle statuses, and optional user-created custom attributes.

After the terms are published to the glossary, they are accessible to users through various search and browse features, available in the following interfaces:

- InfoSphere Business Glossary (web browser client)
- InfoSphere Business Glossary Anywhere pop-up client (rich client on workstation)
- IBM Cognos BI software context sensitive glossary search and display (menu option)
- Eclipse plug-in (using the Representational State Transfer (REST) application programming interface (API))
- REST API programmable access by using custom programming

In addition to the search and browse functionality available in the related interfaces for InfoSphere Business Glossary, InfoSphere Information Analyzer and InfoSphere FastTrack provide specific views of the business metadata contained in InfoSphere Business Glossary. InfoSphere Metadata Workbench provides specific views of the business metadata and provides for custom metadata queries that can include business metadata from InfoSphere Business Glossary.

InfoSphere Business Glossary provides the only graphical user interface (GUI) for authoring (read/write abilities including creating and modifying terms and categories) the glossary. It is also the main tool for searching and browsing (read only). The available functionality is based on roles, as defined by the InfoSphere Information Server administrator. However, additional, finer-grained, access control is managed by anyone with the InfoSphere Business Glossary administrator role.

InfoSphere Business Glossary provides a functional web browser interface for creating and modifying business glossary. However, an administrator can bulk load pre-existing glossary information from appropriately formatted comma separated values (CSV) and XML files, making it a useful facility to use glossary information that might be in external applications or spreadsheets. Furthermore, the REST API is available that can be accessed programmatically, through custom application development, for retrieving existing content or modifying the content.

InfoSphere Business Glossary persists all of its data (business metadata) in the InfoSphere Information Server repository. This approach is appropriate because business terms might have relationships with physical artifacts (tables, columns, report headers, and so on) that already exist in the repository. In addition to creating or modifying the glossary, you can use InfoSphere Business Glossary to view the published business metadata and other metadata objects in the repository, similar to InfoSphere Metadata Workbench, but from a different view perspective.

For more information about how InfoSphere Business Glossary works, and for specific use cases, see Chapter 6, “Building a business-centric vocabulary” on page 131.

3.3.7 InfoSphere Metadata Workbench

The primary function of InfoSphere Metadata Workbench is to provide a view of the content of the InfoSphere Information Server metadata repository, so that users can get answers to questions about the origin, history, and ownership of specific assets.

This “view” comes in a few different formats:

- Three pre-configured InfoSphere Metadata Workbench reports are available: impact analysis, business lineage, and data lineage. These metadata-based reports highlight the relationships and connections across metadata objects and the potential impact of changes to any object in the chain of metadata.
- Simple search and complex ad hoc and persisted custom queries are available, where the user specifies a list of metadata attributes to display for any given set of constraints.
- A browse feature is available that displays any category of metadata assets (hosts, tables, columns, terms, BI reports, and so on), so that the user can select the specific instance of the metadata for a drill-down display. All of these reports are displayed in a window, but most can also be exported to CSV format files for external use and further analysis.

In addition to the repository content views and reports, InfoSphere Metadata Workbench can enhance existing metadata with descriptions and by assigning related metadata assets, data stewards, and so on (that is provide read/write functionality). Furthermore, InfoSphere Metadata Workbench can create extended lineage objects. The purpose is to add metadata to the repository that is not generated automatically by InfoSphere Information Server product modules and components or that cannot be easily imported from external applications. This method provides documentation of external processes and enables extending a business or data lineage report beyond the boundaries of the client's use of InfoSphere Information Server.

Many of the reports that InfoSphere Metadata Workbench generates, particularly lineage, involve complex algorithms, which can place a heavy load on the system. To reduce this load and provide a more scalable environment, some of these algorithms are not running at all times and require manual execution. As a result, InfoSphere Metadata Workbench also provides several advanced repository services that are managed by the metadata administrator.

For information about how InfoSphere Metadata Workbench works to load source data, target data models, and creates data lineage and reporting, see the following chapters:

- Chapter 7, "Source documentation" on page 175
- Chapter 10, "Building up the metadata repository" on page 339
- Chapter 12, "Enterprise reports and lineage generation" on page 393

3.3.8 InfoSphere Information Server Manager, ISTools and InfoSphere Metadata Asset Manager

Each InfoSphere Information Server product module and component generates and consumes metadata to and from the InfoSphere Information Server repository. Therefore, it is necessary to provide a central mechanism for managing the repository. As such, InfoSphere Information Server has three main utilities for managing the InfoSphere Information Server metadata repository:

- InfoSphere Information Server Manager
- ISTools
- InfoSphere Metadata Asset Manager

InfoSphere Information Server Manager is a rich client-user interface. It connects to one or more instances of InfoSphere Information Server. It also allows the administrator to organize InfoSphere DataStage and InfoSphere QualityStage objects (and optionally, their dependent objects) from one or more InfoSphere DataStage/InfoSphere QualityStage project repositories, into packages. These packages can be exported as files for version control (and subsequent deployment and import). Alternatively, they can be deployed directly into another instance of InfoSphere Information Server, such as from Development to Test, Test to Pre-Production, and so on.

After packages are *defined* by using InfoSphere Information Server Manager, the file creation (export) and deployment (import) of the package files can also be performed by the command line utility *ISTools*. The ISTools command line interface (CLI) is installed on both the client workstation and the InfoSphere Information Server host. It can be initiated interactively by an administrator or scripted for standardized use. It is common for the ISTools CLI file creation and deployment scripts to be executed by the enterprise scheduler to automate the process and maximize security.

In addition, the ISTools CLI is used to export metadata from all other InfoSphere Information Server product modules and components into `*.ISX` archive files, such as the following examples:

- InfoSphere Business Glossary: Terms and categories
- InfoSphere FastTrack: Projects and mapping specification objects
- InfoSphere Information Analyzer: Projects and rules
- Shared (common) repository: Physical data resources (common metadata)

These archive files can then be imported into other instances of InfoSphere Information Server, whether for development cycle deployment (dev-test-prod) or migration to newer version of InfoSphere Information Server. This process is similar to the one for InfoSphere DataStage or InfoSphere QualityStage package files.

InfoSphere Metadata Asset Manager is a web-based component that provides several different functions to the InfoSphere Information Server suite:

- Imports metadata from external sources (RDBMS, business intelligence, modelling tools, and so on) into a staging area (Metadata Interchange Server) for comparison with existing metadata for manual conflict resolution
- Loads approved metadata from the staging area to the repository
- Manages duplicate metadata in the repository (merge or delete)

Underneath, *InfoSphere Metadata Asset Manager* uses InfoSphere Metadata Integration Bridges that can translate metadata from external sources into formats that can be loaded into, and used by, InfoSphere Information Server. However, it also uses the same connector functionality used by InfoSphere DataStage, InfoSphere Information Analyzer, and InfoSphere FastTrack to connect directly to compatible RDBMS and ODBC data sources. It also replaces most of the Import/Export manager functionality with an enhanced interface that provides the additional functionality described previously.

3.3.9 InfoSphere Data Architect

IBM InfoSphere Data Architect is an enterprise data modeling and integration design tool. You can use it to discover, model, visualize, relate, and standardize diverse and distributed data assets. Similar to InfoSphere Blueprint Director, it is a stand-alone, Eclipse-based, client-only product module with its own persistence layer (XML format files `*.dbm`, `*.ldm`, `*.ndm`, `*.ddm`), potentially with cross-file links (relationships) between the various models.

From a top-down approach, you can use InfoSphere Data Architect to design a logical model and automatically generate a physical data model from the logical source. Data definition language (DDL) scripts can be generated from the data model to create a database schema based on the design of the data model. Alternatively, InfoSphere Data Architect can connect to the RDBMS and instantiate the database schema directly from the InfoSphere Data Architect physical data model. This “generation” facility works both ways, in that you can also reverse engineer an existing database into an InfoSphere Data Architect data model for modification, reuse, versioning, and so on.

Rather than designing models from scratch, you can purchase one of the IBM Industry Models in a format that is consumable by InfoSphere Data Architect. In this manner, you can jump start the database design phase of the project and benefit from data modeling expertise in the specific industry. Standard practice is to scope the industry standard logical model to fit the customer’s requirements and build an appropriate data model that combines industry standards with customer specifics. An added advantage of the IBM Industry Models package for InfoSphere Data Architect is that it includes an Industry standard glossary model. This model populates the InfoSphere Business Glossary, complete with relationships (assigned assets) to the InfoSphere Data Architect logical model and generated physical data model.

InfoSphere Data Architect does not rely on the InfoSphere Information Server infrastructure for any shared services or persistence. However, you can import InfoSphere Data Architect models (logical, data, and glossary) into the InfoSphere Information Server repository by using the InfoSphere Metadata Asset Manager utility (see 3.3.8, “InfoSphere Information Server Manager, ISTools and InfoSphere Metadata Asset Manager” on page 54).

Furthermore, you can associate InfoSphere Business Glossary terms and categories directly with InfoSphere Data Architect logical model entities and attributes and data model tables and columns. You do this association with a drag-and-drop facility by using the InfoSphere Business Glossary Eclipse plug-in for InfoSphere Data Architect. With Business Glossary Eclipse, InfoSphere Business Glossary is downloaded to an offline XML format file for use with InfoSphere Data Architect. This offline glossary is manually synchronized with InfoSphere Business Glossary whenever desired. However, each time InfoSphere Data Architect is launched, the user is notified if the glossaries are out of synch.

3.3.10 Cognos Business Intelligence software

Cognos BI software is not part of the InfoSphere Information Server product modules and components. However, this section provides a comprehensive view of the product, product modules, and utilities that are available for your information integration projects.

Cognos BI software performs analytics and reporting, based on the specified requirements of the business, as an aid to decision making. It is an independent client-server application, with no shared services or infrastructure common to InfoSphere Information Server. However, its metadata can be extracted and loaded into the InfoSphere Information Server metadata repository by using InfoSphere Metadata Asset Manager.

The metadata that is related to Cognos BI software is loaded into the business intelligence subject area of the InfoSphere Information Server metadata repository. As such, it can be viewed, browsed and searched like any other InfoSphere Information Server metadata. The significance of imported Cognos BI software metadata is that it can be associated with relevant business terms generated by InfoSphere Business Glossary. It can also be included in business or data lineage reports, generated by InfoSphere Metadata Workbench. The Cognos BI software metadata helps fulfill the end-to-end vision of tracing data, from the source systems to the BI reports, and validating the reliability of these reports from a data governance perspective.

Aside from the metadata, Cognos BI software has additional integration points with InfoSphere Information Server product modules. From within the Cognos BI user interface, you can select a Cognos BI object (such as a report column heading) and search for the selected term in InfoSphere Business Glossary. InfoSphere Business Glossary returns a list of possible matches so that the user can select the correct one and drill down, with the same functionality as in InfoSphere Business Glossary. You can also invoke a data lineage report from a Cognos BI object, as though it were specified in the InfoSphere Metadata Workbench product module.

Although Cognos BI software is not technically part of the InfoSphere Information Server platform, the integration between the two products is quite useful.

3.4 Solution development: Mapping product modules and components to solution processes

By now, you should clearly understand the available functionality of InfoSphere Information Server, its product modules, and components. This section examines the approach that is used for implementing an information integration project with InfoSphere Information Server. To assist in this examination, by using the process flow in 2.4, “Typical implementation process flow” on page 34, this section maps InfoSphere Information Server product modules and components to each step within the process flow diagram.

Figure 2-3 on page 34 shows a typical information integration implementation process flow. For each step or process within the process flow, the following InfoSphere Information Server product modules and components (and Cognos BI software) are used:

1. Defining business requirements
   Uses: InfoSphere Blueprint Director
2. Building business centric vocabulary
   Uses: InfoSphere Business Glossary, InfoSphere Discovery
3. Developing data model
   Uses: InfoSphere Data Architect, InfoSphere Discovery
4. Documenting source data
   Uses: InfoSphere Metadata Workbench, InfoSphere Metadata Asset Manager
   - Discovering data relationship
     Uses: InfoSphere Discovery
5. Assessing and monitoring data quality
   Uses: InfoSphere Information Analyzer
6. Building up the metadata repository
   Uses: InfoSphere Metadata Workbench, InfoSphere Metadata Asset Manager
7. Transforming data
   Uses: InfoSphere FastTrack, InfoSphere DataStage, InfoSphere QualityStage
8. Developing BI solutions
   Uses: Cognos BI software
9. Generating enterprise reports and lineage
   Uses: InfoSphere Metadata Workbench and Cognos BI software

3.4.1 Defining the business requirements

The high-level business requirements define a *generic* project, its objectives, and usually the means for measuring its success. The term *generic* refers to the project template or repeatable process structure, established by the high-level business requirements, that can be used for most projects.

These requirements can be captured in an unstructured manner by using a typical word processor or spreadsheet application. However, to use this work later for consistency and reuse of the result from project to project, it is preferable to use an application that supports these objectives.

InfoSphere Blueprint Director is ideal for laying out or modelling business requirements in a graphical format without imposing any design constraints. It visually presents the process flow that represents the high-level business requirements. This format forms the basis for all business initiatives for a data integration or other related project.

The requirements are defined in a more detailed manner for each business initiative. As such, InfoSphere Blueprint Director provides drill-down capabilities from the generic template to the next level of granularity, eventually linking to related artifacts that are captured in the InfoSphere Information Server platform.

For more information, see Chapter 5, “Implementation planning” on page 89.

3.4.2 Building business centric vocabulary

After the business requirements are gathered or laid out in a way as to define the project and its process flow, you must specify the detailed business requirements. Many business initiatives are defined by a deliverable that answers the following questions:

- How is the business benefiting from this project?
- What metrics are used to determine success?
- How are these metrics defined and calculated?
- Are the correct data sources being used for measuring success?
- Is this data reliable?

When the business requirements reach the point of defining quality standards, report details, responsibility, metrics, and so on, *InfoSphere Business Glossary* is ideal for capturing and relating these key terms to associated definitions, responsible individuals, and related information assets.

Using InfoSphere Business Glossary helps to refine the business requirements. It also helps to establish an enterprise-wide, business-centric vocabulary of mutually agreed, centrally governed, shared business terminology. This building process is iterative and is not necessarily limited to the scope of a specific project. However, a specific project contributes to the development of an enterprise-wide, business-centric vocabulary. Furthermore, each successive business initiative adds content to the enterprise business vocabulary (glossary) until the majority of relevant business terms are defined and included in the glossary. At that point, most future business requirements will already be defined.

A business-centric vocabulary provides further benefits when glossary terms are mapped to their related physical assets, such as database tables. For example, a business term, such as *customer*, is more meaningful when it points to the database assets that store customer information.

You can also use *InfoSphere Discovery* in building a business centric vocabulary. It helps to map business terms to physical assets by using sophisticated pattern matching and value matching algorithms. After mapping the terms, InfoSphere Discovery can export the classifications to InfoSphere Business Glossary, enriching the metadata results.

For more information, see Chapter 6, “Building a business-centric vocabulary” on page 131.

3.4.3 Developing data model

Typically, business analysts and data analysts are subject-matter experts (SMEs) for much of the data that is generated by the various operational systems that help the business run. However, designing an appropriate data store or warehouse for loading data from these disparate data sources is not trivial. This task requires much expertise and significant design and planning. Using a data modelling tool for this purpose is standard operating procedure.

InfoSphere Data Architect is ideally suited for taking the business requirements and building them into a solid target data model. In addition to being a straightforward modeling tool, it integrates well with InfoSphere Information Server. For example, you can use it to associate the terms that define the business requirements and metrics with the logical entities of the target model. This term-entity association provides an additional layer of understanding to be used later, particularly at the stage of specifying source to target (or technically target to source) mapping. In addition, you can export to InfoSphere Data Architect by using Import Export Manager.

With InfoSphere Data Architect, it is also possible to take advantage of the industry expertise of IBM through the relevant IBM Industry Model. The Industry Model package provides the target model as a starting point and a well developed, industry standard business glossary, providing a significant jump start for two different steps in the design phase.

In situations where an industry model is not used, or when referential integrity is not defined in the database metadata layer, *InfoSphere Discovery* can help to reverse-engineer a data model. By directly interrogating the data, it constructs a graph of inferred relationships, which you can think of as *candidate keys*. After constructing the physical model, InfoSphere Discovery generates a logical model. Because the relationships are inferred by direct data analysis, some might be statistically correct, but not semantically meaningful. An analyst or SME needs to review the results and approve the correct relationships, after which it can be further manipulated or published in a data modeling product such as InfoSphere Data Architect.

Developing data model is beyond the scope of this book. Therefore, this process is not explained in later chapters.

3.4.4 Documenting source data

As mentioned earlier, often SMEs have a good understanding of the operational data on which the business relies, such as accounts, policies, and products. Unfortunately, this expertise is usually captured only in personnel, rather than in documentation, limiting access to this critical information. Often, these operational systems are hosted externally, by commercial vendors or remote data centers charged with managing these systems. As a result, the true SMEs are external to the business, which makes the business vulnerable, dependent, and open to significant risk.

Therefore, it is critical for internal personnel to understand these source systems and further document them, so that this knowledge is not restricted and can be used for the current project, on-going operations, and future projects. The first step of this documentation process is to identify the various source systems that support the new initiative:

- What data is required for making determinant business decisions?
- Where is the data located?
- In what format is it stored?
- How can it be accessed and by whom?
- How is it structured?
- Who owns the data?

- What information is stored in each attribute of the data, and what domains do they represent?
- How are data elements and objects related to each other?

You can easily construct a source metadata catalog in the InfoSphere Information Server metadata repository by using *InfoSphere Metadata Asset Manager*. Connecting to these data sources (usually by an ODBC connection) and loading their metadata into the InfoSphere Information Server metadata repository is the first step in documenting these source systems. This step also paves the way for subsequent InfoSphere Information Server product modules and components to use and reuse this metadata.

If source systems are not accessible to InfoSphere Metadata Asset Manager, you can use *InfoSphere Metadata Workbench* to create and load this source metadata manually.

In addition to loading the physical metadata from these source systems, the data analyst can further enhance the documentation with annotations, descriptive information, and other attributes by using InfoSphere Business Glossary or InfoSphere Metadata Workbench. These documentation enhancements help appropriate personnel understand the source data from a technical perspective. This entire documentation process is the backbone to extend InfoSphere Information Server functionality. It is used by other InfoSphere Information Server product modules and components.

For more information, see Chapter 7, “Source documentation” on page 175, and Chapter 8, “Data relationship discovery” on page 209.

Discovering data relationships

If you do not understand your source data fully, you need to gain a deeper understanding of your data, discover relationships among the data, and get a deeper level of insight. This way, you can derive meaningful metadata and correctly assess data quality.

SMEs who understand source data are usually constrained to a few subject areas. One individual cannot know everything about everything. Therefore, the organization relies on several SMEs to collectively keep the business running. The SMEs know their own subject areas, but they have limited knowledge of other subject areas. They might lack knowledge about how their subject areas overlap, and integrate, with others. They might not be aware of some business rules that are hidden within the data. Even if an SME knows a lot, upon leaving the organization, much of the knowledge also leaves. That is, the portion that was not accurately and sufficiently documented might be lost.

Organizations need is a way to scan the data automatically and infer the metadata, which they can store in robust, secure, and accessible repositories to be shared among other product modules and components.

To fully understand the data, you perform data relationship discovery with *InfoSphere Discovery*. InfoSphere Discovery can scan the data unattended in the background. When it finishes, the result is a rich data profile that includes detailed information about every attribute of every table it analyzed. It also includes potentially hidden relationships within and across the data structures. Data analysts or SMEs review and approve the results, and then export them to the metadata repository.

For more information about data relationship discovery, with illustrations and several use cases, see Chapter 8, “Data relationship discovery” on page 209.

3.4.5 Assessing and monitoring data quality

Performing a data quality assessment with *InfoSphere Information Analyzer* is the main method to gather accurate information about data content quality. The assessment reveals inconsistencies and anomalies in the data, determines whether it is normalized, and provides clear reports of the results. After publishing the results to the shared repository, they are immediately used by InfoSphere FastTrack, InfoSphere DataStage, and InfoSphere QualityStage.

InfoSphere FastTrack can use the published results of cross-table and domain relationships to identify potential tables that can be joined for lookups or straightforward mapping. The InfoSphere DataStage and InfoSphere QualityStage developers can view the published profiling results, including annotations, to determine the kind of data errors that exist and must be coded for in the error handling within the jobs. In addition to handling the errors, by using InfoSphere QualityStage, many of these errors can be corrected within the job even if there is no desire to fix the source data.

To ensure that the conclusions derived from final reports are trustworthy, the underlying data must be proven to adhere to the minimum quality standards. This proof can be obtained by first assessing and defining data quality to which the data must adhere using data quality rules. They must also be measured against these standards on a regular basis.

InfoSphere Information Analyzer has data quality rule functionality. With this functionality, the data analyst can define rules and then run all of the data against these rules to ensure that the data meets the requirements. The data quality rules can be run against millions of rows of data on a regular basis because the product module uses the InfoSphere Information Server parallel engine framework, which provides a scalable infrastructure that can process large volumes of data in a timely fashion.

For more information, see Chapter 9, “Data quality assessment and monitoring” on page 283.

3.4.6 Building up the metadata repository

This process follows the same process as described in 3.4.4, “Documenting source data” on page 61. However, this process refers to documenting and collecting target system metadata, such as those from a staging database, data warehouse, data mart, and business intelligence report, in the metadata repository.

Again, *InfoSphere Metadata Asset Manager* is used to connect to the external data source (repository or extract) and to load the metadata into the InfoSphere Information Server metadata repository. Depending on the type of data source to which it connects, it determines which subject area within the repository will be populated. For data sources, the physical data subject area is generally populated. For data modelling sources, the logical or physical data model subject area is populated. For BI data sources, BI subject area is populated.

Similar to the source data metadata catalog, the metadata repository can be enhanced with *InfoSphere Metadata Workbench* by adding descriptions and assigning stewards and other information assets to the relevant metadata. This process includes enhancing information assets in the metadata repository.

For more information, see Chapter 10, “Building up the metadata repository” on page 339.

3.4.7 Transforming data

To achieve the targeted business value of the initiative, whether generating revenue, reducing cost, managing risk, and so on, most initiatives require data delivered from the right source, at the right point, in the right way to the business process. The business requirements describe what is needed, but the data must still be delivered through the integrated solution to facilitate the business initiative.

As with any development project, it is necessary to plan before the implementation. Starting in 3.4.1, “Defining the business requirements” on page 59, until this point, the content has described the prerequisites to the design, development, and implementation phases.

The first steps in designing the data integration flow entail the following tasks:

- Identifying the target data store to populate
- Identifying which data source or sources will be used
- Determining the processing that needs to be done to the source or sources to load data into the target

InfoSphere FastTrack is fitted to provide exact design functionality. With source metadata documentation and target data modelling, users begin with a solid foundation to begin the specification process (with consultation of the SMEs as needed). Because the bulk of the metadata is already in the InfoSphere Information Server repository, InfoSphere FastTrack can use all of the associated relationships between related terms and their assigned assets. In addition, it can use the published cross-table relationships and domain associations that are revealed through the discovery process of InfoSphere Discovery and data quality assessment process of InfoSphere Information Analyzer.

At this stage, *InfoSphere Discovery* can provide additional automated methods for determining how to describe the transformations. One of the key features of discovery is to determine how one table relates to another. In instances where data from one table is derived from another, InfoSphere Discovery can reverse engineer the transformations that were executed on the source data to load it into the target table, on a column by column basis. These results can be exported from InfoSphere Discovery and imported directly into an InfoSphere FastTrack mapping specification, automating one of the more challenging tasks in the mapping specification process.

After the business or data analyst has documented the source-target mappings and described the transformations in the InfoSphere FastTrack mapping specification, the specification output can be generated as an InfoSphere DataStage template job. This artifact is saved in the same project where the job will be further developed.

Because the specification output is in the format of an annotated InfoSphere DataStage template job, you must develop the job with *InfoSphere DataStage*. Retain the template job as documentation, allowing a reliable audit trail to the specification, rather than developing directly from the automatically generated InfoSphere DataStage job. The job can (and must) be copied to another job and developed from that point onward. Also record the name of the InfoSphere FastTrack specification (and template job) from which this job is based, in the long description field of the job properties sheet for the new job. This further provides traceability back to the specification from which this job is designed.

When implementing the InfoSphere FastTrack mapping specification in the InfoSphere DataStage job, the InfoSphere DataStage developer can use artifacts that make sense to use. The developer can certainly change the job construction to meet the requirements as specified from the annotated to-do list of the InfoSphere FastTrack mapping specification. This list must include all of the “Rule” descriptions that were specified in InfoSphere FastTrack.

For more information, see Chapter 11, “Data transformation” on page 375.

3.4.8 Developing BI solutions

Although data is the core element of an information integration project, the bottom line results are the reports generated by the BI solutions for the business decision makers. These reports put into motion the entire data integration process, the business requirements that drive each business initiative. Therefore, the application used for this portion of the development process is critical.

Cognos BI software is the most appropriate software for BI process development. In addition to the standard features of report design and delivery, common to many other business intelligence software, Cognos BI software can integrate with and search the InfoSphere Business Glossary for definitions of Cognos BI report fields, metrics, and other BI concepts. This capability reinforces the use of InfoSphere Business Glossary for defining business requirements and initiatives, as dictated by the BI reports.

Additionally, you can import Cognos BI software metadata into the InfoSphere Information Server repository by using InfoSphere Metadata Asset Manager. This metadata is an aid to InfoSphere Metadata Workbench, because InfoSphere Metadata Workbench reports the full scope of business and data lineage from source to reports. Even though Cognos BI software is a solid BI application, the added advantages of its integration with InfoSphere Information Server make it by far the preferred choice.

Developing BI solutions is beyond the scope of this book. Therefore, this process is not explained in later chapters.

3.4.9 Generating enterprise reports and lineage

The core element of the information integration project is the data from which the BI reports are derived. You must validate the quality of the data to ensure that the report results are reliable. However, you must also validate that the report results are accurate based on the data and that the data is the correct data. For this purpose, *InfoSphere Metadata Workbench* has various metadata based reports to ensure that the high quality data is also the correct data that came from the right system of the record.

The lineage report that is generated by InfoSphere Metadata Workbench provides such oversight. For any given data, InfoSphere Metadata Workbench can derive all of the data sources that contribute to the specified data and all of the targets that used this data. This functionality is unique to InfoSphere Metadata Workbench.

InfoSphere Metadata Workbench can also create and load specific extended metadata when the InfoSphere Information Server metadata repository does not connect directly to metadata sources or when the sources or their metadata is not accessible. These extended metadata objects are further documentation that is added into the metadata catalog or in the InfoSphere Information Server repository. Their purpose is to enhance the lineage reports so that the entire Information Integrated solution can be represented in the full data flow.

For more information, see Chapter 12, “Enterprise reports and lineage generation” on page 393.

3.5 Deployment architecture and topologies

The following documentation describes most of the deployment architectures and topologies that are available for InfoSphere Information Server:

- IBM InfoSphere Information Server user documentation
- *Information Server: Installation and Configuration Guide*, REDP-4596 (describes the deployments that are appropriate under which circumstances)

This section describes the uniqueness of the metadata driven InfoSphere Information Server and provides possible deployment considerations.

3.5.1 Overview of the topologies

The standard InfoSphere Information Server topologies in the InfoSphere Information Server documentation include two-, three- and four-tier deployments, high availability active-passive failover, various clusters, grid, and so on.

Figure 3-5 on page 68 shows an example of the simplest, single server architecture: a two-tier topology that separates the client tier from the server tier. The server component encapsulates WebSphere Application Server, the services, metadata repository, and engine. A three-tier topology builds on the foundation of the two-tier topology but separates the engine piece from the server component. A four-tier topology distributes all components (the client, database, services, and engine) to separate pieces of hardware or nodes.

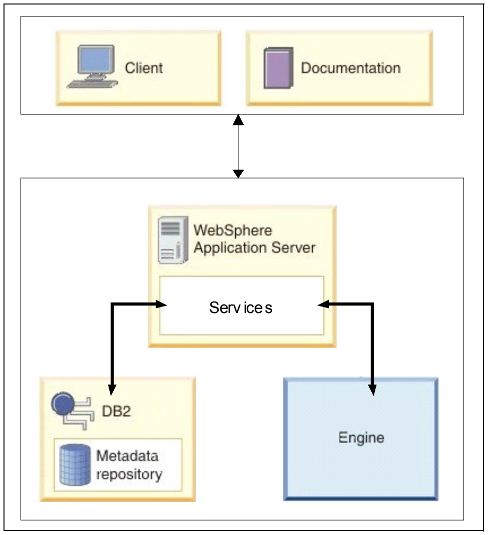


Figure 3-5 Two-tier deployment topology

The InfoSphere Information Server documentation and *Information Server: Installation and Configuration Guide*, REDP-4596, help to define the appropriate architecture based on a combination of factors and specific criteria. However, the common approach among all of these recommended topologies bases deployment on the typical development life cycle. It uses some or all of these separate environments for development, test, pre-production, and production, and perhaps separate sandbox environment, to test patches, fix packs and upgrades, before deploying to the "real" environments to minimize the impact on productivity.

Nonetheless, InfoSphere Information Server is not just a set of product modules and components for developers used by information technology (IT) personnel to support the needs of the business. It is also designed for business personnel and the traditional applications for the technology professionals. As such, you must consider additional factors for an InfoSphere Information Server deployment architecture. For example, you must consider the requirement of a larger user community and how to best facilitate their interactions across the enterprise. One of the main considerations is how to deploy shared metadata.

3.5.2 Unified and shared metadata

As referenced in 3.2.3, "Repository tier" on page 46, InfoSphere Information Server supports the concept of unified and shared metadata. The concept is that metadata is created, consumed, and used by various users. These users all want their efforts to result in a functional, consistent, well behaved system that is perceived and truly is reliable. However, each resource has a different area of focus, depending on their needs.

For example, the analyst wants the resultant data to be properly formed and accurate. The data steward or, potentially the CEO, wants all of the data assets to be verifiable and conform to regulatory constraints, if they exist. The developer wants to know how to efficiently and accurately develop applications that get the job done. All of these examples require distinct access to the InfoSphere Information Server.

Some might oppose any access to the production servers, often with good reason. Others, also for good reason, might demand access to the production data so that they can be confident in the data that is being produced. It is being implemented by various enterprises.

Disclaimer: The deployment options and topologies presented in this section are for your reference only. IBM does not provide support for the topologies provided here. This description is strictly for your information.

When you know the most crucial decision points to consider, you can set the policy to align with corporate concerns.

Start by focusing the following tasks that InfoSphere Information Server performs, such as the following examples:

- Moves data
- Cleanses or transforms data
- Analyzes data
- Stores business definitions
- Assigns owners to data assets

Then look at the users who are not just developers and operation staff in this environment. Many other groups of users in one way or another need to access the InfoSphere Information Server. Considering that they all have valid reasons, do you give them access to the production system? The answer depends on how you evaluate the needs of each group and weigh those needs against system performance.

A fundamental assumption is that you cannot let anyone impact the performance of the production system. What if you only process data at night and the loads are incremental and run in a relatively short time. Then is it acceptable to allow access to operational metadata on that system during the day by using InfoSphere Metadata Workbench? Again, the answer depends. Look at other options that, although require additional hardware and software, might be the answer you are looking for.

Unified metadata is not just for development, testing, or production environment. Analysts communicate with developers in many ways. Examples include through the business definitions and data analysis that are captured in InfoSphere Business Glossary or the mapping specifications that they create in InfoSphere FastTrack.

Because operational metadata is also of interest to many, the question is: How is this infrastructure going to be created that looks and feels like production, but that does not affect the production environment? What if you replicate the parts of production that satisfy the metadata needs? You can host the InfoSphere Business Glossary in a location that is acceptable to both the analyst and developer.

Design and operational metadata can also be included. InfoSphere Information Analyzer jobs that sample or perform complete table scans require a differently tuned database than the one optimized to handle transactions.

Consider adding it all in one environment as illustrated in Figure 3-6.

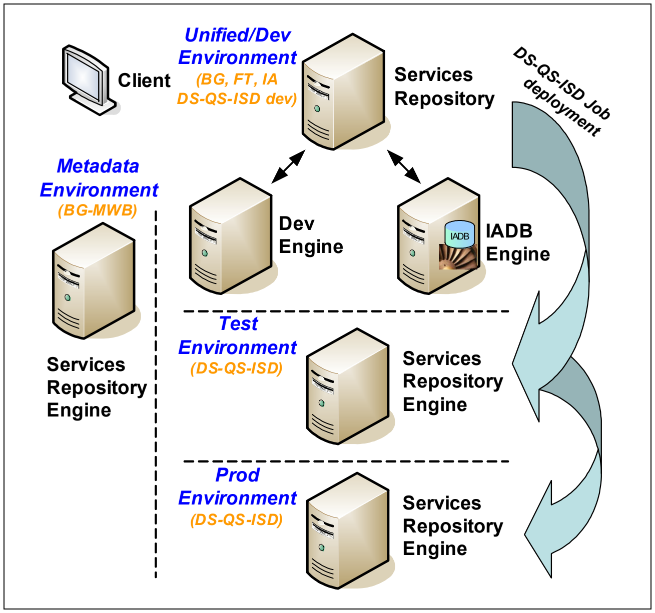


Figure 3-6 Unified metadata topology

The production environment is dedicated to production, and it has a two-tier topology. The same is true for the testing environment. The development environment has a three-tier topology because it works best for development. InfoSphere Information Analyzer has its own engine and database (IADB, the Information Analyzer database) workspace. The place to view and manage metadata of the production data is missing. This place is where the business and the technical users meet. It is the glue between the different resources. The following section describes this environment.

3.5.3 Metadata portability

This section describes the potential of having a dedicated metadata environment. The purpose of this dedicated environment is to provide all stakeholders full-time access to the various types of metadata that exist within InfoSphere Information Server.

All Information Server product modules generate and consume metadata. The metadata that they generate is persisted in the InfoSphere Information Server repository. This is true for all InfoSphere Information Server instances, such as development, test or QA, production, and other environments that a company might have.

Depending on business requirements, InfoSphere Information Server product modules might be deployed in some or all of the environments. For example, InfoSphere DataStage is always deployed in all environments because it is part of the development cycle to promote it through various environments. InfoSphere Metadata Workbench might not need to be installed (and used) in a development environment, but it might need to be installed in a production environment.

Figure 3-7 shows a composite of the topology that was previously described with the addition of the dedicated metadata environment.

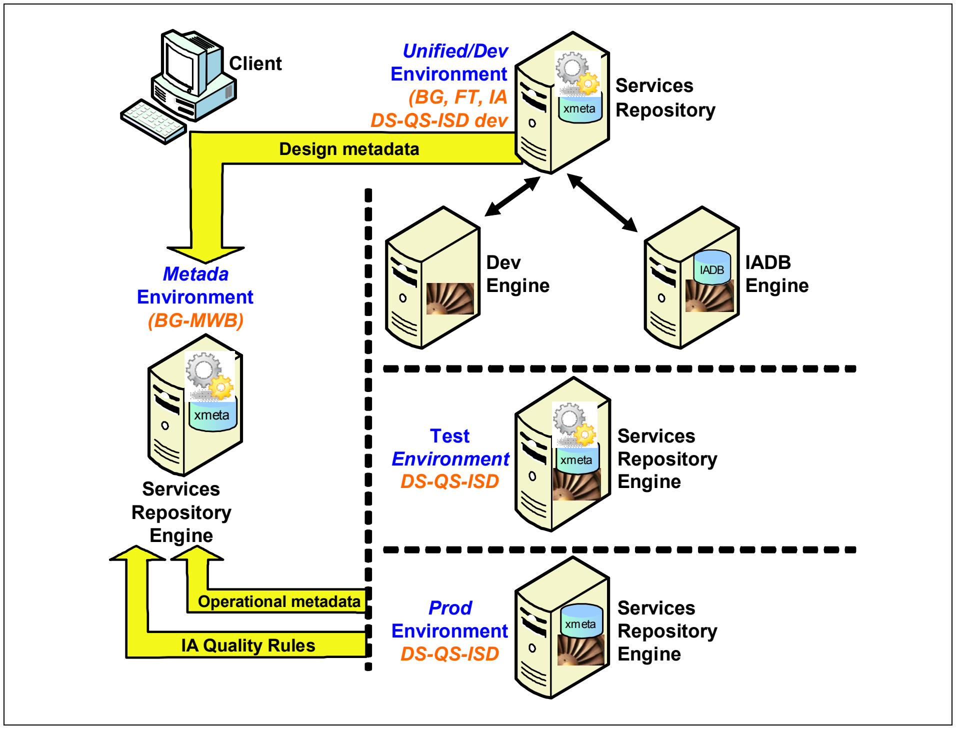


Figure 3-7 Dedicated metadata environment

In the dedicated environment, regardless of the environment in which the metadata is created and persisted, the relevant metadata is copied to this separate instance of InfoSphere Information Server that is dedicated to accumulating metadata. This dedicated environment becomes the place where all stakeholders can go and view the metadata they want.

Figure 3-7 shows four environments: Dev, Test, Prod, and Metadata. It also shows copying the design metadata (Information Analyzer Quality Rules) from the Dev to Metadata environment and copying the operational metadata from the Prod to Metadata environment. This way, the working environments can be minimally impacted by the various metadata needs. However, in the case of the operational metadata, some latency will occur from the time that the metadata is created to the time that it is available in the metadata environment. A well-understood change-management process must be in place for such an environment.

3.5.4 Alternative deployment

The previous deployment example is based on specific requirements that might or might not apply in a given customer environment. You must consider the objectives and understand all of the constraints when planning a deployment strategy. This approach is only one example, but it provides a way of thinking that can be applied to any environment.

The typical deployment scenario is to select one environment (usually the production environment) and ensure that all the necessary metadata from the development environment is promoted. This method requires establishing a disciplined process for promoting published business metadata with the physical metadata that corresponds with the production environment. Regardless of the deployment architecture that is selected, you must understand the considerations that go into this decision.

3.6 Conclusion

In conclusion, this chapter provided an overview of InfoSphere Information Server, the platform infrastructure, and deployment options. It also introduced the InfoSphere Information Server product modules and components that provide the functions for implementing an information integration solution. More importantly, this chapter mapped InfoSphere Information Server to the processes that are involved in a typical implementation process to illustrate the product modules or components that are involved and in which part of the process.

The rest of this book highlights the details of each process and the product modules and components that are used.

Part 2

Implementation

This part provides information about the implementation of a solution. It uses the process described in the previous part and includes a use case.

This part includes the following chapters:

- Chapter 4, “Use-case scenario” on page 77
- Chapter 5, “Implementation planning” on page 89
- Chapter 6, “Building a business-centric vocabulary” on page 131
- Chapter 7, “Source documentation” on page 175
- Chapter 8, “Data relationship discovery” on page 209
- Chapter 9, “Data quality assessment and monitoring” on page 283
- Chapter 10, “Building up the metadata repository” on page 339
- Chapter 11, “Data transformation” on page 375
- Chapter 12, “Enterprise reports and lineage generation” on page 393

4

Use-case scenario

This chapter presents a use-case scenario that is used for the remainder of this book. This use-case scenario facilitates your understanding of the concepts and procedures that are presented and the practical examples that are included.

The scenario is fictitious and is for illustrative purposes only. It explains in detail where an organization needs to expand the use of metadata across information integration and business intelligence (BI) projects. This scenario is not intended to describe the exact process or define an industry best practice.

In this scenario, a national bank, Bank A, with a mature and sophisticated BI and data warehouse solution acquires a regional bank, Bank B, which does not have many BI solutions. Bank A needs to integrate and consolidate data source systems from Bank B to create consolidated BI reports and analysis, which they have done currently.

This chapter includes the following sections:

- Scenario background
- Current BI and data warehouse solution for Bank A
- Project goals for the new solution
- Using IBM InfoSphere Information Server for the new solution
- Additional challenges
- A customized plan
- Conclusion

4.1 Scenario background

Bank A is a large corporation that provides financial services to customers from different industries across the globe. Bank A started operations in the US and expanded its presence in and expanded its presence in North America, South America, Europe, Asia, and Africa. The current product portfolio of Bank A includes savings accounts, checking accounts, mortgages, automotive financial services, and insurance. Bank A offers an extensive set of services to small-to-large sized companies.

Bank A has a large and mature BI infrastructure that provides reports and valuable information from all of its business units. The information is in the right format so that it can be used by business executives, stakeholders, and oversight authorities for different purposes.

Even though Bank A has a significant presence in most of the major financial markets in the world, in many countries, it lacks a presence on a regional basis. The representation of Bank A only includes major financial services provided to companies or governments. It does not reach individuals, and it limits its deployment of the full product portfolio.

To close this gap, Bank A has acquired local banks whose presence is established in the regional markets. Bank A gains access to the market through acquisition, which it might not have had otherwise.

Thus, in this use-case scenario, Bank A acquired Bank B.

4.2 Current BI and data warehouse solution for Bank A

The current BI solution of Bank A supports multiple BI reports, ad hoc queries, and form of advanced analytics. With their current solution, the business users define the list of reports required, their layout, the metrics, and dimensions to be used. They also generally define all of the business rules and exceptions that shape the analysis.

Figure 4-1 shows the layout for the BI solution of Bank A.

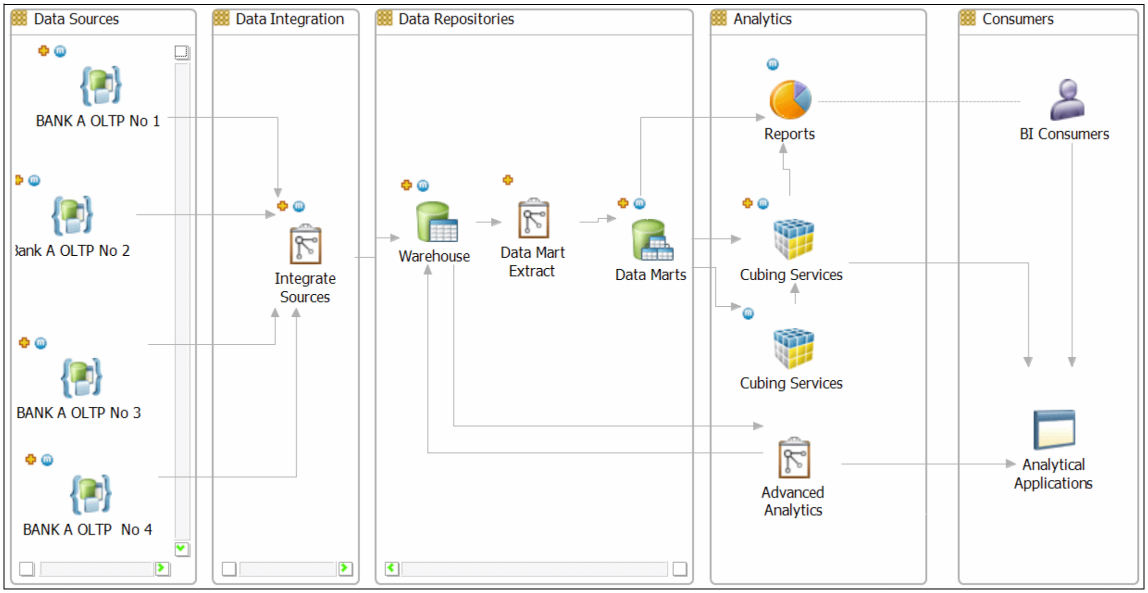


Figure 4-1 BI solution for Bank A

As shown in Figure 4-1, the BI solution of Bank has a standard approach to the BI initiatives. The BI landscape of Bank A includes the following areas:

- Data sources
- An information integration layer
- Data repositories
- Analytics
- Front-end services for consumers

With the current information flow and delivery process, the solution has several problems that affect the final BI reports and analytics:

- No reliable way of knowing that the BI reports are from accurate and complete source data
- Frequent misunderstanding and misusage of terms between business and technical teams
- No good way to maintain current business definitions
- Difficulty in consolidating new source data into the solution
- No easy way to achieve an enterprise architecture and to synchronize projects

4.3 Project goals for the new solution

From the senior management perspective for Bank A, one of the main goals for this project is to have a solution that treats information as a strategic enterprise asset, with insightful and actionable data. This solution must include capabilities such as flexibility for changes and development, agility to deploy, and compliance with regulation and directives. The new solution must be able to handle complex and constantly changing banking challenges, triggered by authorities, clients, and a fierce competition.

The new solution requires the following capabilities and characteristics:

- A common, centralized business vocabulary
- A way to discover, understand, and document source data
- The ability to design and publish data structures
- The ability to map discovered data to target structures
- The ability to develop data integration and data quality transformation
- The ability to analyze and monitor data quality
- The ability to discover new or hidden business rules within the data
- The ability to document, navigate, and analyze the metadata through:
  - Data lineage
  - Business lineage
- The ability to perform impact analysis to control changes and prevent errors
- The ability to control project deployment through multiple environments: development, test, and production
- The ability to reuse the components and artifacts to enhance productivity, reduce development time, and increase the quality of final products

4.4 Using IBM InfoSphere Information Server for the new solution

Bank A intends to use IBM InfoSphere Information Server as the platform where all of the information integration process takes place. This approach includes processes for understanding, cleansing, transforming, and delivering their data.

The current solution already uses some of the InfoSphere Information Server product modules for cleansing, transformation, and delivery. It includes IBM InfoSphere FastTrack, IBM InfoSphere DataStage, and IBM InfoSphere

QualityStage. However, it has not taken advantage of other product modules and components that provide information governance capabilities and the required capabilities mentioned earlier. These modules and components include IBM InfoSphere Blueprint Director for implementation planning and IBM InfoSphere Business Glossary for building centric business vocabulary. They also include IBM InfoSphere Metadata Workbench for housing source and target metadata for business and data lineage reports and impact analysis.

With the added InfoSphere Information Server product modules and components, Bank A plans to document the existing solution and then consolidate the data of Bank B into the existing solution. Bank A wants to find a fast and reliable way to achieve this goal.

4.4.1 Changes required

Several changes need to take place to the existing solution and to move the current solution toward the new one.

Bank A currently has multiple sets of business glossaries that contain definitions from various areas for different purpose. Bank A intends to build one business-centric vocabulary that is accurate, easy to use, and easy to share among all users.

Bank A plans to store all metadata in one central metadata repository for easy sharing and usage. The repository includes metadata of the source data systems and target reports. It also includes data quality rules, cleansing and transformation jobs, a delivery data model, and lineage information.

Bank A wants to assign this metadata (or other components required in the solution) to users who can be responsible for its content and structure. These users are *stewards* of the metadata or other required components.

Bank A wants to establish business and technical data stewards who link assets and update definitions. Whenever a user or developer needs to work with an asset (for example, a BI report needs an update within the business rules that might filter or expand the results), the steward will know who to contact. If the business inquiry is related to more than one asset, the business user can tell who the responsible parties. Then they can obtain detailed tracking without losing any perspective.

4.5 Additional challenges

Bank B has grown from a small business to achieve a regional presence. Although its market share is bigger in its own regional market, it is trailing behind other regional banks in the number of unique customers. Bank B is in this position because it lacks the infrastructure of their competitors. Bank B has not handled the wide spread of its customer basis under a single platform. Moreover, Bank B cannot offer a single product portfolio to its clients because its information resides in separate silos and they cannot integrate a single offering or a cross-selling campaign.

On the operations side, Bank B has failed to understand its client basis. This situation is common to bank executives who take too long to offer any product that can complement the clients' portfolio. The result is that the competition is making the offering and later cross-selling the product that Bank B had with this client. In summary, Bank B is failing to sell more to its actual clients and it is losing them to competitors. The reason for this status is that Bank B cannot create a trustworthy, single client database.

From the performance management perspective, Bank B did not have the tools to help the company foresee these problems and react to the market. For Bank B, analytics equal forensics. Reports are built with great effort and time consumption and are not delivered to executives in a timely manner.

The process of gathering information and feeding the data objects that eventually serve as source for the final reports is *not* a full data integration process. It sometimes lacks management and appropriate or updated business rules. Also if changes are required, it can be expected to shut down for long periods of time because almost everything must be created from the scratch. Maintaining this process is a nightmare for the development team.

In this scenario, Bank B business users do not trust the information they receive from the systems and often manually adjust the report data by using spreadsheets. Technical users need reports for different purposes. Sometimes the same reports are requested with no consistency in them and with different names and formulas being used.

The development team at Bank B is also in a complicated state. Managers do not have enough people on the team, and they have to push the developers to do more in less time. The result is almost no documentation is available to record the last updates and patches to systems. Regulations and reports delivered to authorities contain changes made as needed. Some events are not recorded at all because nobody remembers them. Crisis emerges if members of the team decide to leave the bank without providing sufficient knowledge transfer to their replacement.

To solve these challenges, the new solution uses the following tools:

- InfoSphere Business Glossary for a centrally shared glossary
- InfoSphere Discovery for discovering data relationships
- InfoSphere Information Analyzer to assess and monitor data quality
- InfoSphere FastTrack, InfoSphere DataStage, and InfoSphere QualityStage for data cleansing, transforming, and delivery
- InfoSphere Metadata Workbench for storing centrally shared metadata repository and for generating lineage reports, impact analysis, and more

4.5.1 The integration challenge and the governance problem

At the time that Bank A acquired Bank B and the merger was approved, the new management encountered several challenges when starting operations.

Management needed to start reporting the operation from Bank B according to the proven standards and methodology of Bank A, and they needed to do it quickly. The challenges were the traditional problems of having a different set of tools and infrastructure (both Banks A and B) and migration from the core systems from Bank B to Bank A.

At Bank B, transactional systems could keep working as they were, because by regulation, they were required to comply with a set of operational standards. Therefore, in this area, Bank A could rely on the current operation results in terms of service level agreements (SLAs) and quality of service (QoS). The problem was that the systems that were built to solve the BI requirements were not ready. For example, the level of detail provided by Bank B sources was unhelpful in calculating all the metrics and key performance indicators (KPIs) required by Bank A to measure business performance management (BPM).

In addition, both banks faced the problem that the reporting systems did not have current documentation. The latest version of the documentation was unusable because of major changes that were applied due to regulation requirements in both sources and data models. After a brief review, creating this documentation from scratch with the current team, for migration purposes and later for sending it to the trash, was determined to be too difficult and expensive. The time spent on creating the documentation might also distract resources that control the current operation in this critical phase of the acquisition. Bank A could not afford a shutdown on the service because it might compromise the acquisition and give a bad impression to customers.

4.5.2 Additional business requirements

Bank A decides to completely integrate reporting information for Bank B into its information-integration running environment and reporting platform. Bank A plans the following strategy to facilitate this business requirement. It uses the typical implementation process explained in Chapter 2, “Solution planning and metadata management” on page 29.

Bank A can decide to implement this flow in a different order. Remember that this process can be solved from many different approaches and is going to be affected by many variables such as the following examples:

- The maturity level of the BI and data warehouse projects of the organization
- The status of readiness that an organization must enter into this effort, which includes the following examples among others:
  - The amount of human resources with available time to be assigned for the project and incoming deployment from both technical and business sides
  - The capabilities of the transactional systems to provide the information required as input for further calculations without any modifications
  - A common set of skills that the development and operational teams need to have before the launching of the project.

Bank A has a vast amount of experience in these projects. Because Bank A has a running project that has been growing for years, it carefully reviewed how to approach this case. The decision to pass through all of the phases is consistent with the preferred practice. This practice indicates integrating the most possible pieces of metadata and inheriting the advantages of having a robust environment that uses them. A benefit that can be obtained includes assets related each other, showing dependencies between them. Another benefit includes having stewards who are responsible for providing maintenance for the assets and for preventing unexpected changes by having impact analysis.

The primary driver of the project is to complete all of the reports and get them running. Another main driver is to obtain as much knowledge as possible about the company that was acquired, its processes, KPIs, and business rules, among other information. Therefore, understanding and managing the Bank B metadata becomes critical for compliance with the business requirement. Managing metadata is a project driver. It can be a show stopper for many BI or data warehouse efforts if it is not given the attention it demands.

4.6 A customized plan

You have now reviewed the use-case scenario, the current solution of Bank A, its goals, and what and how InfoSphere Information Server product modules and components can be used for metadata management and consolidated integrated solution. Now Bank A is ready for a customized plan.

Bank A will use the typical implementation process described in Chapter 2, “Solution planning and metadata management” on page 29, and in Chapter 3, “IBM InfoSphere Information Server approach” on page 41. It will plan and implement the solution. This section briefly touches on each step in the implementation process.

Defining business requirements

The first step is to determine which information is required in the BI reports and the supporting definitions. For example, Bank A must determine which business rules apply, how often they apply, and the appropriate detail of information required for each report.

Building a business-centric vocabulary

Bank A needs to understand the business terminology of Bank B. Then Bank A must import it, use it, and create a common glossary or make the required adjustments to make it fit into the new and bigger business landscape.

A *business glossary* is valuable for an organization because it includes the vocabulary and business knowledge that members of the organization use to communicate more efficiently and precisely. It important for Bank A to know clearly the common terminology and its meanings for Bank B before using the information. By understanding this terminology, Bank A can confirm that it is using the appropriate information assets, such as when building a report for use by certain authorities or to help business executives make decisions.

In addition, among the divisions and users of Bank A, some have their own business glossaries that are not shared or exposed, and others do not have any formal glossaries. Bank A sees this situation as an opportunity to consolidate and publish one glossary to incorporate information from Bank B and the business terms and categories from Bank A.

Developing a data model

Bank A has a data model for its BI reports. However, it needs to ensure that the data model of Bank B fits in the existing data model. Therefore, they go through this step of developing a model by updating the existing model rather than starting from scratch.

Consider the following questions for developing the new consolidated data model:

- Is the current data model sufficient to fulfill the new or updated business requirements?
- Can the data model for Bank A be populated with information from Bank B? Does it provide the level of detail required for all analysis and reporting?
- Are any adjustments necessary for the new data model to fully integrate the two banks, such as fields or hierarchies with the existing information?

Documenting source data

Source systems represent the input or pipe of data from their point of origin or creation to a staging area.

To begin, Bank A must document metadata of its own source data systems that are used for its BI reports. Then Bank A must identify, understand, and document the source data systems that Bank B has, where they are, and what is required for the consolidated BI reports.

Discovering data relationships

To understand the data that Bank B has, Bank A also performs data relationship discovery to understand the data attributes and relationship among the data for Bank B. Finally, Bank A loads the metadata of the source data into one central metadata repository.

Assessing and monitoring data quality

Bank A must assess the quality of the data that Bank B has before adding it to the BI solution. For example, it is important that duplicate or inconsistent values are not added to the Bank A environment.

The development team expects an overlap in some of the dimensions used for analysis, such as clients and addresses. As ascertained, Bank B has problems in this area, with siloed environments for the different business lines, which is one part that prevented Bank B from understanding its common base of clients.

Adding to this, Bank A and Bank B might have shared clients before the acquisition. Such clients might include consumers or companies that are using the services of both banks. It is imperative to identify these clients in order to start treating them from a single point of view as business requirement demands.

From the metadata perspective, Bank A must discover and understand the dimensions, hierarchies, data objects, and processes that affect the data entities within the systems. By gaining this understanding, Bank A can start managing them and deciding which steps to take. This method might require running data lineage and performing impact analysis on the results for getting the appropriate inputs and starting the data quality assessment.

Documenting warehouse metadata

The current metadata repository from system A is expected to change as the project changes. If the process is well driven and executed, the metadata is expected to grow naturally, updating the context, meaning, and specification of the business terminology as the project evolves. Maintaining this communication is crucial for business users, technical users, or any stakeholder that reviews business terms or decides if the terms are still valid or need to be updated.

Transforming data

The data specification is set. Also process development is set to transform information that comes from data storage systems to the new target data stores.

Bank A has an operating process for its own sources, but needs to create new extensions for the Bank B sources. Within its new process, it must ensure that specifications and definitions are aligned with it.

The transformation and integration process from Bank B and Bank A sources are expected to converge at some point to populate the shared metadata repository that facilitates fulfilling business requirements. The development team must understand which assets to update, and it must monitor the impact of the updates.

The teams need to know the data lineage from the objects involved in the process. They must understand the traceability of these objects through the integration process and the overall environment to correctly update all dependant artifacts. To mitigate deployment risk, they must also be aware of unwanted changes before they apply them.

4.6.1 BI process development

BI process development is an area where new development is not a strong requirement. The reason is because BI reports, data collection, data marts, and other required components or artifacts are already in the Bank A infrastructure. Little modification is needed for this current project because no other business requirements are included other than integrating Bank B information.

The following additional functions might be required for the solution in future:

- Creating new Bank B users on the new platform who can consume the reports
- Enabling a current set of dashboards and reports to filter information for Bank A, Bank B, or both for an overall view, which might be helpful during transition
- Enabling personalized views to continue the business line monitoring for Bank B, but with the Bank A enterprise approach and KPIs

4.6.2 Data Quality monitoring and subscription

Bank A is likely to include the new data objects that are developed and, if required, include them into the scope of all of the data quality monitoring. These objects must be linked and visible through every metadata analysis and tracking. This way, every user who has the appropriate permissions will be aware of these dependent objects and the impact of changes made for follow-up and monitoring purposes.

4.6.3 Data lineage and reporting requirements and capabilities

Enabling data lineage capabilities as a feature of the complete solution takes more importance everyday, as benefits are perceived by the business and technical users. All of the solutions must provide components so that stakeholders across the project can review all of the artifacts and subprocesses that modify or transform the information in its life cycle.

The Bank A team knows that understanding the business meaning of an object or process and its dependency with other information assets will help to navigate across the heterogeneous environment. Understanding this information will also help to control the incorporation of new assets without affecting a productive system. Knowledge of these details will ensure project success as it facilitates and helps the search for information in the metadata repository.

4.7 Conclusion

In conclusion, this chapter presented a use-case scenario where a national bank (Bank A) acquires a regional bank (Bank B). It explains how Bank A needs to perform the consolidation and improve the existing data integration solution.

The remainder of this book uses this use case to provide practical examples. It goes through key steps in the implementation process. It also explains how individual InfoSphere Information Server product modules and components are used during the process.

5

Implementation planning

When planning for a solution implementation, you need to start with the process of creating a blueprint. You draw your approach on a canvas or on a sketch pad so that you can share, discuss, and gain consensus on the project that you are delivering. IBM InfoSphere Blueprint Director is that canvas or sketch pad and is for that purpose of drawing plans and sharing them with others. With InfoSphere Blueprint Director, you can take common processes or templates and evolve them as your business needs change.

By using InfoSphere Blueprint Director, you can design your information landscape with a visual approach from high-level to detailed design. With a set of readily available templates based on different project types, you can ensure an efficient planning process with a reusable approach and ensure collaboration on the design before action is taken. Guaranteed best practices are shared and implemented across teams and work projects.

This chapter describes how InfoSphere Blueprint Director can provide efficiency and consistency in your implementation planning. This chapter includes the following sections:

- Introduction to InfoSphere Blueprint Director
- InfoSphere Blueprint Director user interface basics
- Creating a blueprint by using a template
- Working with milestones
- Using methodology
- Conclusion

5.1 Introduction to InfoSphere Blueprint Director

InfoSphere Blueprint Director helps you to define the end-to-end architecture vision for your information projects *consistently* through the use of templates (reference architectures). It guides you and helps you understand how to achieve your vision through associated methods. You can start implementing your vision and maintaining controls with your blueprint.

Whether your initiative is for information governance, information integration, or business intelligence (BI), InfoSphere Blueprint Director makes your planning and implementation process simpler. InfoSphere Blueprint Director comes with several content templates by project types that are ready for immediate use. You can easily customize the templates to fit your project requirements.

The following templates are included:

- Master Data Management
- Business Driven BI Development
- Managed Data Cleansing
- Information Lifecycle Management

A benefit of using an existing template is time to value. You can quickly start your project planning process with an existing reference architecture and customize it as you require. Another important aspect is consistency. You define your blueprint according to a set of standards from the existing template. By using templates, you ensure process consistency throughout your project and among projects within your organization.

To draw the blueprint for your project, start with an existing template based on your project type. You then customize the template to fit your project requirements. The standard template guides you through your project phases and helps you understand what you need to accomplish in each phase.

Figure 5-1 on page 91 shows a blueprint that is created from the available template, the Business Driven BI Development template. It shows a Warehouse icon on which you can take action. Rather than drawing this icon on a board, with InfoSphere Blueprint Director, you drag the existing reusable icons.

The information flow illustrated in Figure 5-1 on page 91 shows four data sources that will go through a type of integration (which can be defined later). Then they will move the data into a warehouse, out to analytics, and finally to consumers and analytics applications.

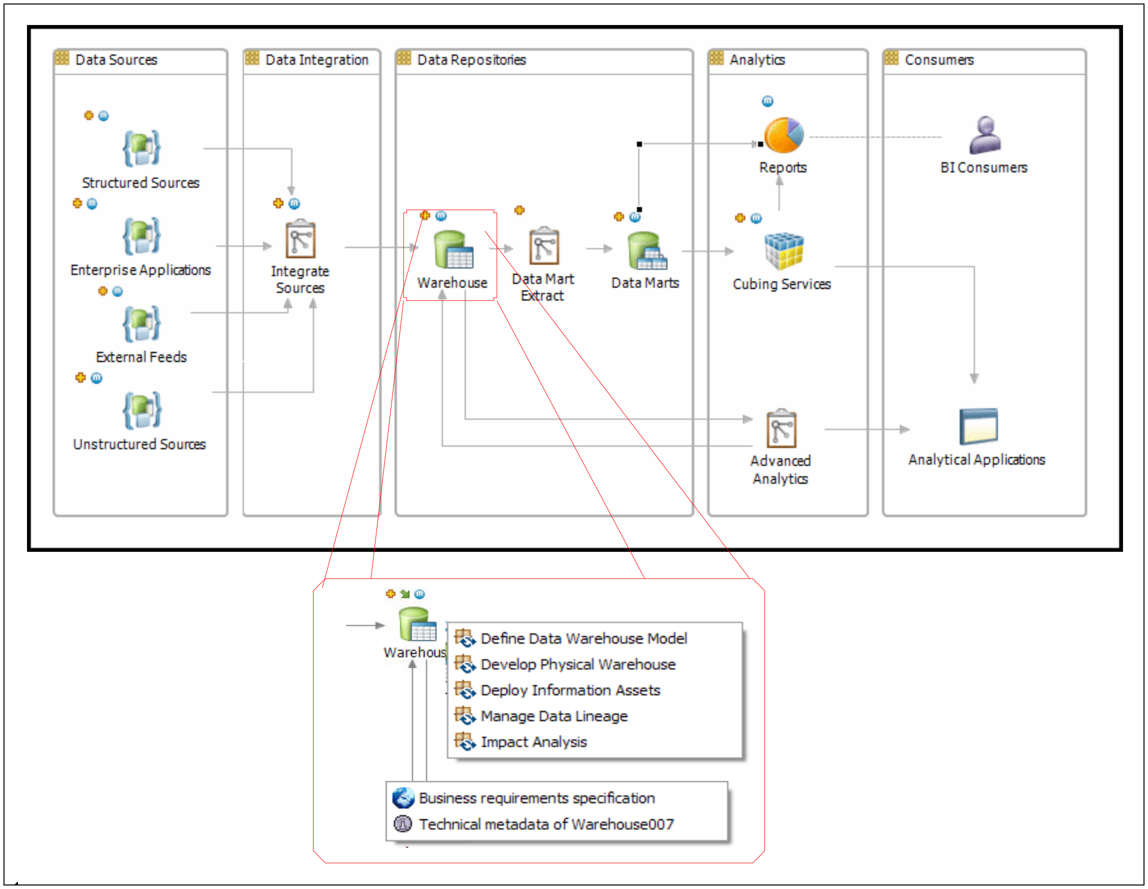


Figure 5-1 Blueprint template for business driven BI development

When you click the **Warehouse** icon, you see text associated with it that provides a list of options for this icon. In this example, the following options or potential tasks are associated with the Warehouse icon:

- Define the data warehouse model
- Develop the physical warehouse
- Deploy information assets
- Information governance functions such as manage data lineage and perform impact analysis

Also associated with the Warehouse icon are business requirements specification and technical metadata.

InfoSphere Blueprint Director provides this visual representation of the project that you are working on and the ability to navigate to specific functionality. With this approach, you can create a blueprint diagram, share it, tune it, and drive downstream processes.

To work with your blueprint, InfoSphere Blueprint Director provides a design canvas. On the canvas, you can drag graphical objects that represent processes, tasks, data assets, or other objects that are required for your project. You can connect the processes to establish a sequential order for which the processes should occur. In addition, you can connect the processes, assets, and other objects to show the dependencies among them. You can label each graphical object to indicate its purpose.

InfoSphere Blueprint Director supports complex multilayers of information details for your blueprint. You can use a single graphical object in the blueprint to represent various tasks, assets, or processes. When you drill down on the object, it shows its sublayer, with details of the tasks, assets, or processes that consist of the upper layer object. You can have several hierarchical levels that form the basis of your project plan from top to the bottom.

An important feature of InfoSphere Blueprint Director is the milestone feature. You can design and create a project road map based on milestones. You can specify the processes or phases of projects that will be completed at which milestone. By using the milestone feature, you can track your project easily. By selectively showing your blueprint based on the milestones, project stakeholders can quickly understand the overall end-to-end project plan and what will be accomplished at each stage.

InfoSphere Blueprint Director can link to InfoSphere Information Server repository objects. It can also launch InfoSphere Information Server product modules and components to display the linked objects in their native tool. This method ensures that the right resources are represented and used in the project plan.

In summary, by using InfoSphere Blueprint Director, you can use the readily available templates and use the best practices and methodology embedded in the them. You can also reduce the overall development costs and reduce the risk, oversights, and errors in implementing your projects.

5.2 InfoSphere Blueprint Director user interface basics

InfoSphere Blueprint Director provides a user interface and a palette where you can draw your diagram. This section highlights both features.

5.2.1 User interface

Figure 5-2 shows the user interface of InfoSphere Blueprint Director. You can navigate multiple frames to edit your projects and diagrams. InfoSphere Blueprint Director contains various views and frames that contain the components that you want to work with.

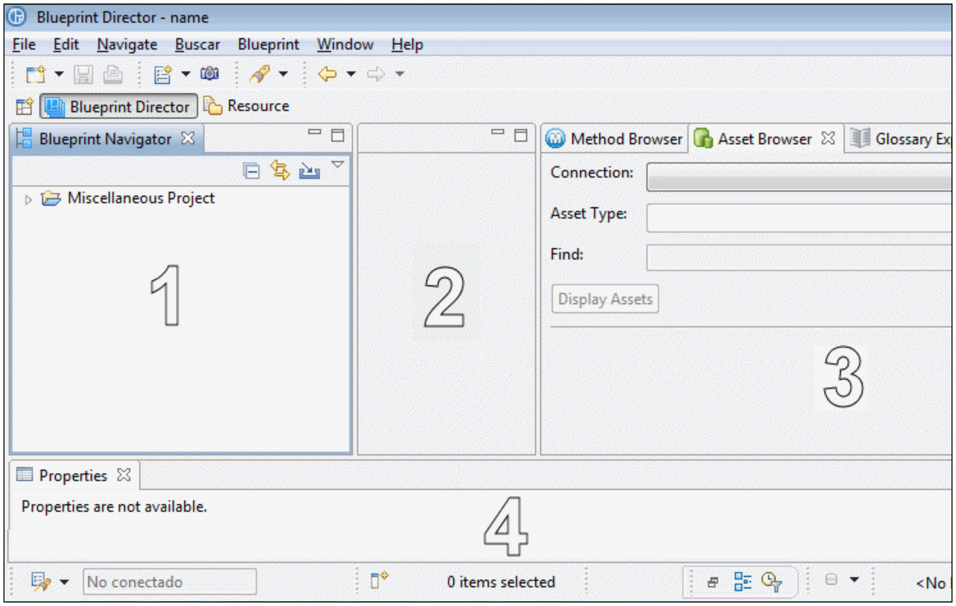


Figure 5-2 Sections of the user interface for InfoSphere Blueprint Director

To have a reference architecture with actionable information in the shortest time possible, the focus in this section is on the most used frames. Figure 5-2 contains the following sections and frames (where the numbers correspond to the numbers in the figure):

1. Blueprint Navigator
2. Editor area
3. Tab group
4. Properties editor

The sections that follow explain each area of the user interface.

Blueprint Navigator

Use Blueprint Navigator to browse blueprint projects and their content. A project contains a root diagram, which is the global view or main layout. From any diagram, you can navigate its domains and the artifacts, methods, or assets that each one contains. The navigator is a useful way to understand how the components are arranged within a given architecture.

Editor area

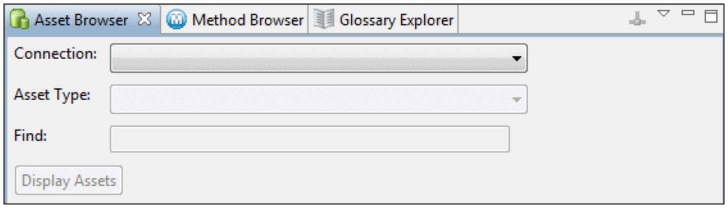
The editor area is the canvas that you use to draw diagrams and add components and artifacts to them. You can edit the content and properties to customize them as the project architecture requires.

Tab group

In the tab group, you can arrange different views to manage different components that are used in diagram design. This group has the following components:

- Method Browser
- Asset Browser
- Glossary Explorer

Figure 5-3 shows a tab group with the Asset Browser, Method Browser, and Glossary Explorer.



Connection:

Asset Type:

Find:

Figure 5-3 Tab group in InfoSphere Blueprint Director

Other views can be added to the working space. For example, you can add a view to a tab group or use the view detached from the frames, where it floats in the workspace at your convenience. To navigate through the available views, select **Window** → **Show View** (Figure 5-4).

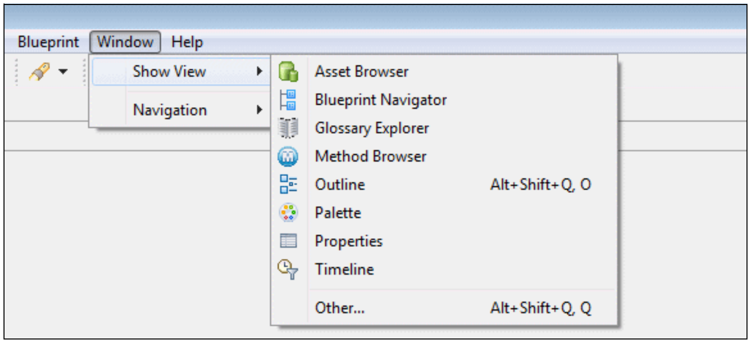


Figure 5-4 Navigating the views in InfoSphere Blueprint Director

Properties editor

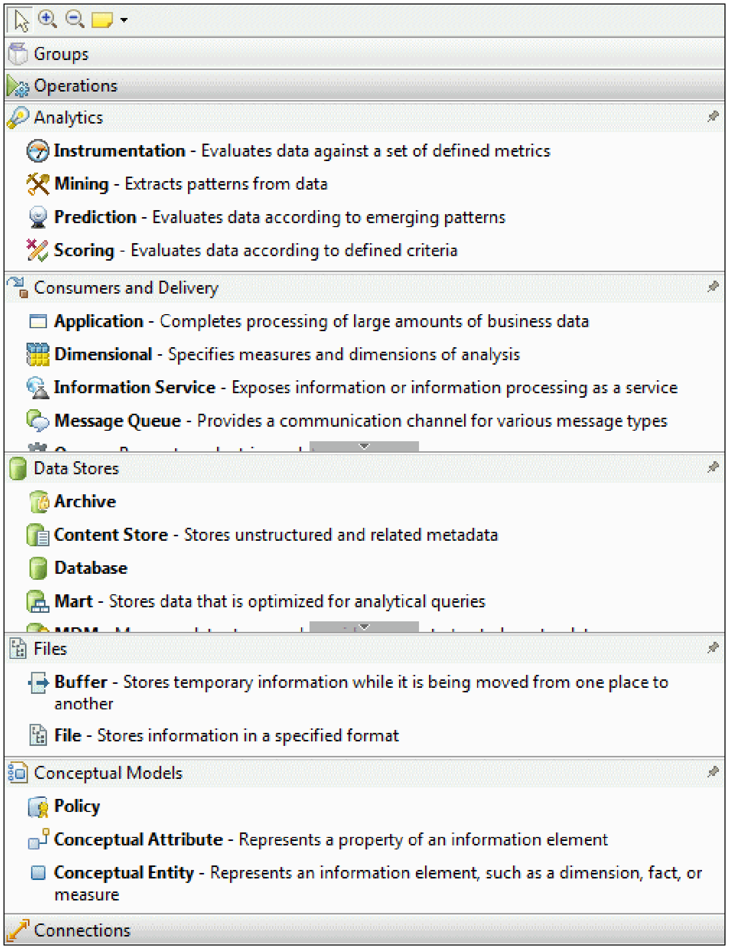
The properties editor at the bottom of the window is helpful for providing identification and format information about the specific component at use or the edition. For example, the information can include the name, the font used, a brief description, or owner. An advanced properties option exists if the brand, version, or steward is required.

Frame layout: You can modify and customize the InfoSphere Blueprint Director frame layout.

5.2.2 Palette

To draw a diagram, Blueprint provides a list of components that you can use to design your process or workflow. The palette has different tabs or areas that are divided by functionality and purpose. This flexibility gives you many options to create the Blueprint.

Figure 5-5 shows a standard palette.



Groups

Operations

Analytics

- **Instrumentation** - Evaluates data against a set of defined metrics
- **Mining** - Extracts patterns from data
- **Prediction** - Evaluates data according to emerging patterns
- **Scoring** - Evaluates data according to defined criteria

Consumers and Delivery

- **Application** - Completes processing of large amounts of business data
- **Dimensional** - Specifies measures and dimensions of analysis
- **Information Service** - Exposes information or information processing as a service
- **Message Queue** - Provides a communication channel for various message types

Data Stores

- **Archive**
- **Content Store** - Stores unstructured and related metadata
- **Database**
- **Mart** - Stores data that is optimized for analytical queries

Files

- **Buffer** - Stores temporary information while it is being moved from one place to another
- **File** - Stores information in a specified format

Conceptual Models

- **Policy**
- **Conceptual Attribute** - Represents a property of an information element
- **Conceptual Entity** - Represents an information element, such as a dimension, fact, or measure

Connections

Figure 5-5 Blueprint palette

The palette shows the following tabs:

Groups Specify domains, projects, or a set of assets.

Operations Specify the type of tasks that can be executed or within the information, such as a federated query, a data integration process by itself, a routine, a Change Data Capture (CDC) activity, and a correlation.

Analytics Specify analytics-specific tasks such as processes, mining, prediction, or instrumentation.

Consumers and delivery
Specify the deliverable (such as query, report, and dimension), users, and other components that show how data is finally delivered and used.

Data stores Specify where the information is stored, such as in Master Data Management (MDM), an operation data store (ODS), a database by itself, an archive file, or a data warehouse.

Files Indicate whether a physical file or a buffer is used to store information.

Conceptual models Represent the use of dimensions, facts, or measures that add context or properties to the flow or process.

Connections Specify the directional flow of information, whether a non-bidirectional relationship exists between elements, or if File Transfer Protocol (FTP) is used.

As you can see, a large set of components is available for you to draw your blueprint with different levels. The palette provides descriptions of almost every building component that you need. Go through the list of components, and understand the purpose of each component.

5.3 Creating a blueprint by using a template

You typically start creating a blueprint from one of the available templates and customize it to your system environment and solution requirement.

For the use case in this book, the project requires incorporating the components of Bank B into the current solution for Bank A. We use the “Business Driven BI Development” template and customize it first to create a blueprint that shows the current Bank A solution. We then customize it further to show the future solution with processes and assets of Bank B merged into the processes and assets of Bank A. We also use the milestone feature to show the different phases of the solution, one for the current solution and one for the future merged solution.

To create your blueprint from InfoSphere Blueprint Director, follow these steps:

1. Create a project:
   a. Select **File** → **New** → **Project** (Figure 5-6).
   b. Enter a project name. For this use case, we enter `Bank A BI`.

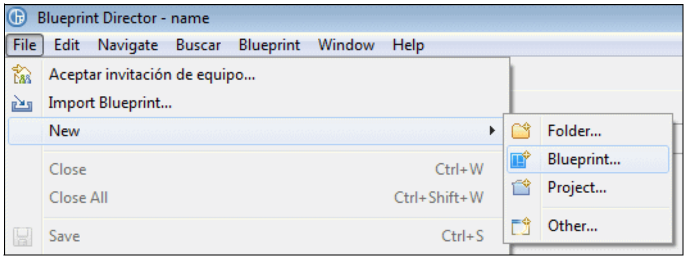


Figure 5-6 Creating a blueprint diagram from the File menu

You can also create a project by right-clicking **Blueprint Navigator** and selecting **New** → **Project** as shown in Figure 5-7.

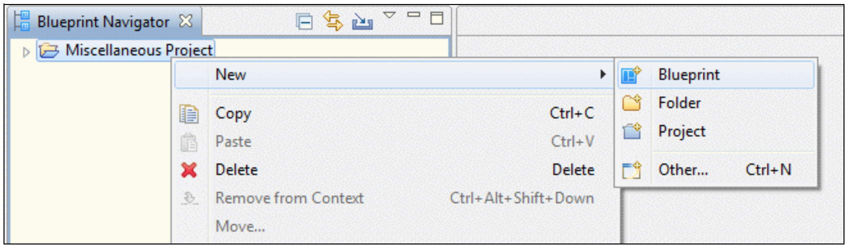


Figure 5-7 Creating a blueprint diagram from the Navigator tab

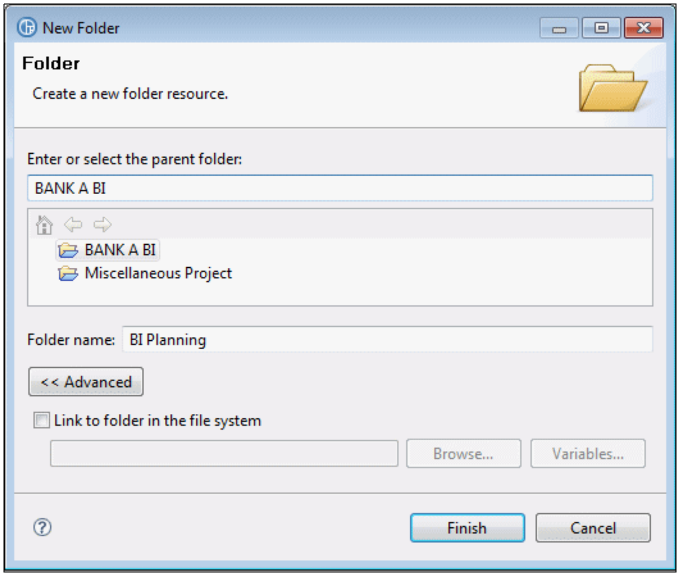
2. Create a folder for the project.

   To organize the blueprints, create a folder and a project that contain the blueprint. This way, you can export the blueprint completely and maintain control over the versions without mixing its content with other projects you might have.

   To create the folder, complete these steps:

   a. Select **File** → **New** → **Folder**. You can also create a folder by right-clicking **Blueprint Navigator** and selecting **New** → **Folder**.

   b. In the New Folder window (Figure 5-8 on page 99), enter a name for the new folder. In this scenario, we enter a name of `BI Planning`. Then place this folder inside the project that was just created. Click **Finish**.



New Folder

Folder

Create a new folder resource.

Enter or select the parent folder:

BANK A BI

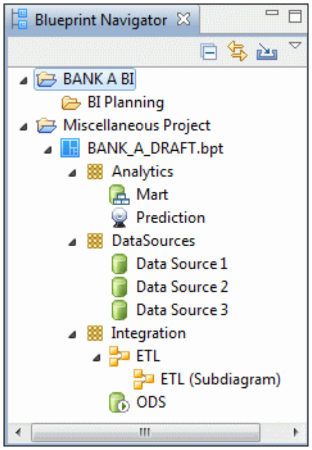
- BANK A BI
- Miscellaneous Project

Folder name: BI Planning

☐ Link to folder in the file system

Figure 5-8 Creating a folder

Figure 5-9 shows the new project, Bank A BI. The BI Planning folder is inside the corresponding project. Next you create a blueprint by using the appropriate template.



Blueprint Navigator

- BANK A BI
  - BI Planning
- Miscellaneous Project
  - BANK_A_DRAFT.bpt
    - Analytics
      - Mart
      - Prediction
    - DataSources
      - Data Source 1
      - Data Source 2
      - Data Source 3
    - Integration
      - ETL
        - ETL (Subdiagram)
      - ODS

Figure 5-9 New project resource

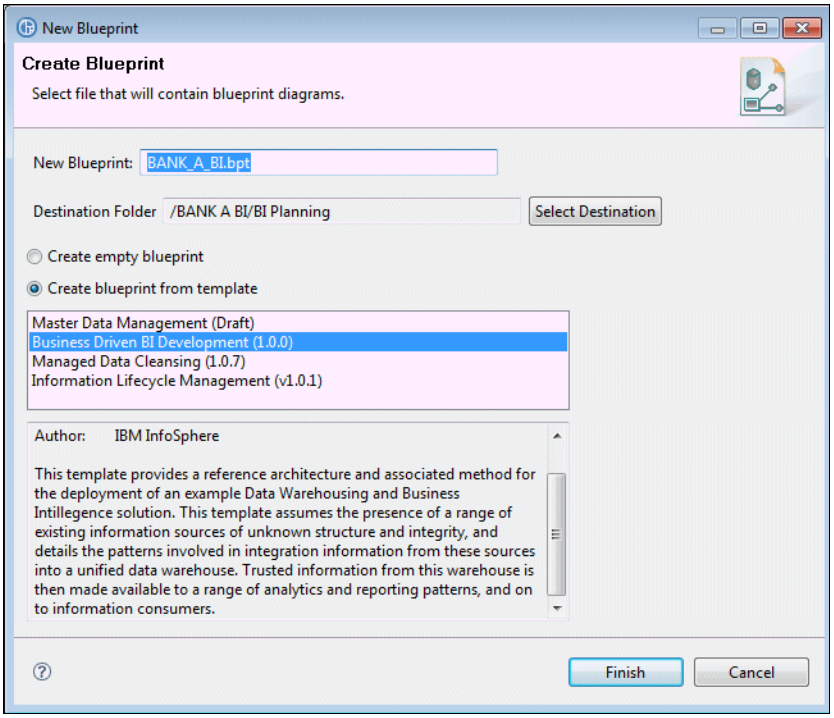
3. Create a blueprint:

   a. Select **File** → **New** → **Blueprint** to create the blueprint.

   b. In the New Blueprint window (Figure 5-10), under **Create blueprint from template** option (selected by default), select the **Business Driven BI Development** template from the list of available templates. Each template comes with a version number and a brief description that explains the use of methods and the overall processes and tasks covered by the template.

      In this window, notice that the blueprint is called `BANK_A_BI.bpt`. (All blueprint diagrams require the `.bpt` extension.) In addition, a destination folder is defined. In this scenario, because we started the creation blueprint in the BI Planning folder of the Bank A BI project, the destination folder is inherited into the creation. We can modify the folder here if we start from another position.

      Click **Finish**.



New Blueprint

Create Blueprint

Select file that will contain blueprint diagrams.

New Blueprint: BANK_A_BI.bpt

Destination Folder /BANK A BI/BI Planning

○ Create empty blueprint

● Create blueprint from template

Master Data Management (Draft)
Business Driven BI Development (1.0.0)
Managed Data Cleansing (1.0.7)
Information Lifecycle Management (v1.0.1)

Author: IBM InfoSphere

This template provides a reference architecture and associated method for the deployment of an example Data Warehousing and Business Intillegence solution. This template assumes the presence of a range of existing information sources of unknown structure and integrity, and details the patterns involved in integration information from these sources into a unified data warehouse. Trusted information from this warehouse is then made available to a range of analytics and reporting patterns, and on to information consumers.

Figure 5-10 Creating a blueprint

Blueprint Navigator shows the entire structure for the blueprint (Figure 5-11). In the editor area, the root diagram is now ready to be customized.

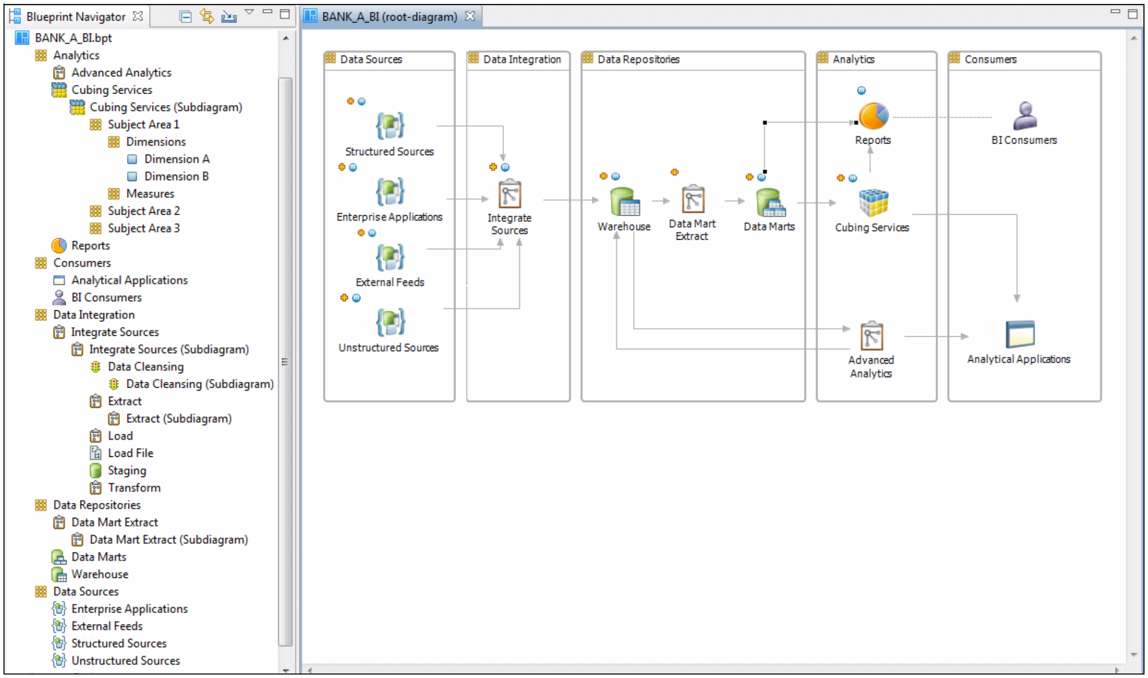


Figure 5-11 Blueprint structure

By using this template, you save time so that you can quickly start project planning, and you ensure consistency throughout the planning process. Now, you need to customize the blueprint and link it to the information assets in the metadata repository, including the business glossary.

5.3.1 Customizing the template

The template is divided into five major domains (or blocks) based on different stages that the data goes through to reach to the business requirements. This overview gives a clear idea about how information flows from the data sources and what processes or components are required and related in each one.

Figure 5-12 shows the template domains and its content. This diagram shows the following components in the first level:

- The Data Sources domain that shows all of the sources used in the solution
- The Data Integration domain that shows the tasks involved in the data transformation process
- The Data Repositories domain that shows where the information is stored
- The Analytics domain that shows the analytics tasks that need to be done
- The Consumers domain that shows the users and deliverable of the solution

All these domains are useful for Bank A because they are similar to the domains and the working areas in the project for Bank A.

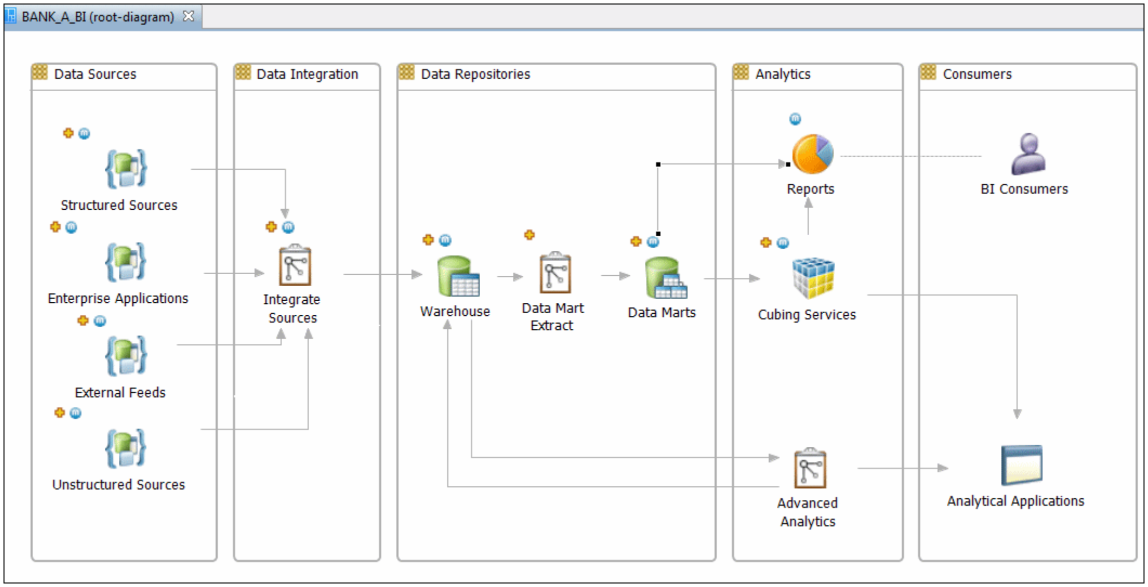


Figure 5-12 Template domains

You can update the database information in the Data Sources domain (Figure 5-13) by linking them to the assets that are available in the metadata repository.

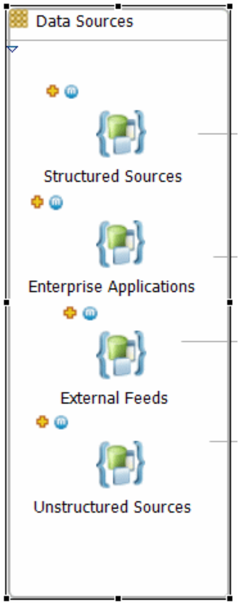


Figure 5-13 The Data Sources domain

The components in this domain are marked with two icons:

- The ✚ icon represents a subdiagram. You can drill down from this component into the subdiagram.
- The ⓜ icon represents a method. You can drill down to view more detail about the scope and objectives of the components.

The template provides the methods and subdiagrams. As needed, you can delete them or add more by right-clicking a specific component and selecting the appropriate action.

For this scenario, we go into one of the subdiagrams and update the properties of the data sources. We work in the Structured Sources diagram as shown in Figure 5-14.

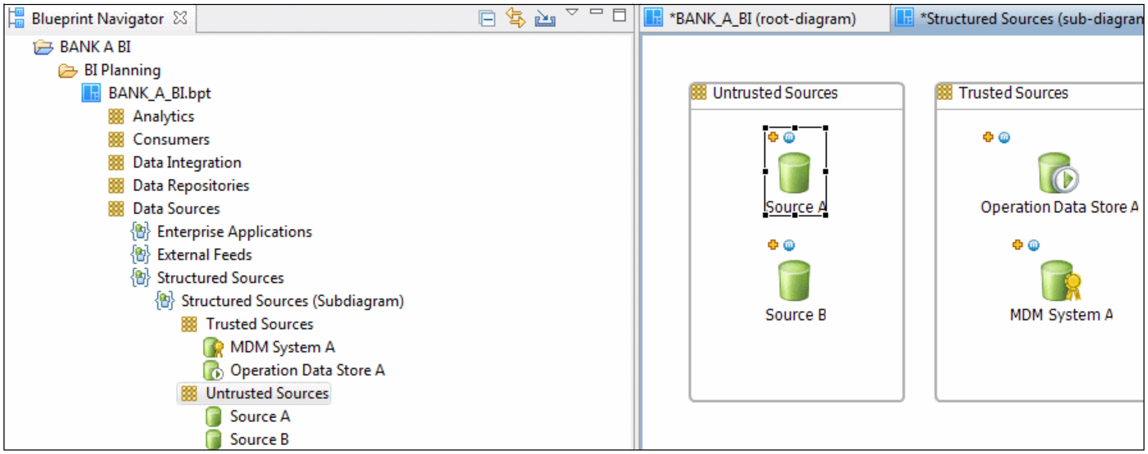


Figure 5-14 Structured Sources subdiagram

Deleting the unnecessary components

This subdiagram is divided into two domains: Untrusted Sources and Trusted Sources. The current Bank A solution has reached a maturity level where all their sources are trusted sources. Bank A does not have any untrusted sources as shown in the template. Therefore, we delete the untrusted domain:

1. Right-click every component in the untrusted domain, and select **Delete**.
2. After the domain is empty, delete the domain.

Figure 5-15 shows the subdiagram now.

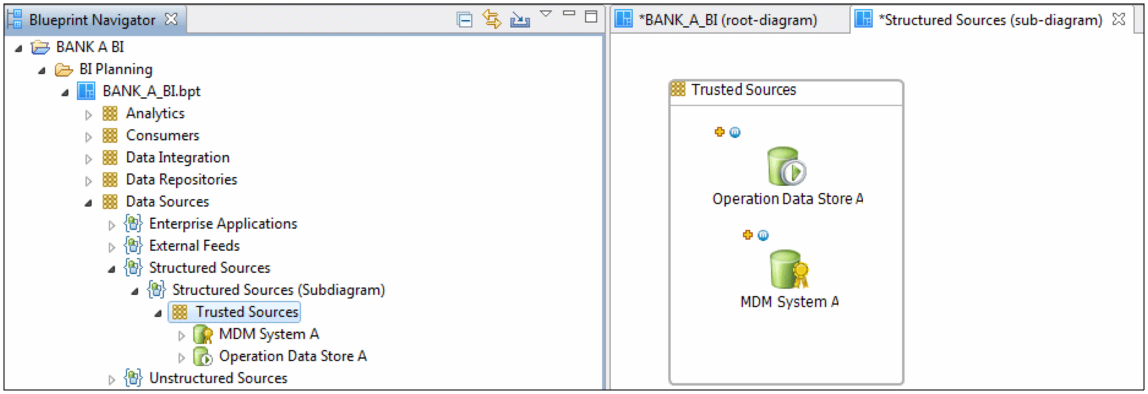


Figure 5-15 Deleting a subdomain

According to the infrastructure of Bank A, the following changes must be made to the standard template in the Data Sources domain for customization:

- Include a database to store the insurance information.
- Update the MDM system name.
- Use the ODS for saving accounts.

Adding and updating components

In this scenario, we need to add the Insurance component to the Trusted Sources and rename the two existing components. To add the new component, by using the palette, drag the required component to the Trusted Sources domain. To update a component name, modify it in the **Properties** tab.

Now the data sources reflect those data sources in the infrastructure of Bank A (Figure 5-16).

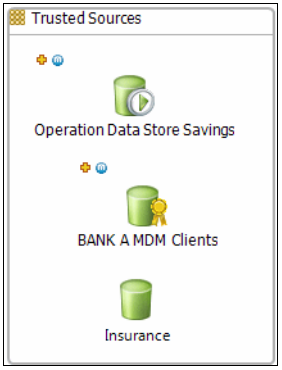


Figure 5-16 Updated data sources for Bank A

For this scenario, we keep the domain name Trusted Sources because we might need to include an Untrusted Sources domain to integrate and profile any database or ODS from Bank B. This method helps to maintain differences during the project.

Notice that the Blueprint Navigator shows the changes in the project in which we saved the subdiagram. The root diagram is also updated if it is affected.

5.3.2 Working with metadata repository

If you already loaded your source data in InfoSphere Metadata Workbench, you can connect InfoSphere Blueprint Director to InfoSphere Metadata Workbench. This way, your blueprint can point directly to the data sources loaded within the metadata repository.

If you have not yet loaded your source data in InfoSphere Metadata Workbench, you can skip this section for now. When you complete this step, come back to see this section and complete the connection piece.

For details about loading source data in InfoSphere Metadata Workbench, see Chapter 7, “Source documentation” on page 175.

Connecting to InfoSphere Metadata Workbench

To connect your blueprint to the metadata repository, complete these steps:

1. Select **Blueprint** → **Manage Server Connections** (Figure 5-17) to connect to a metadata repository.

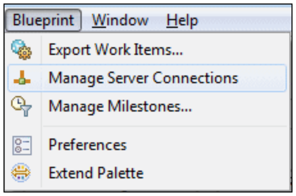
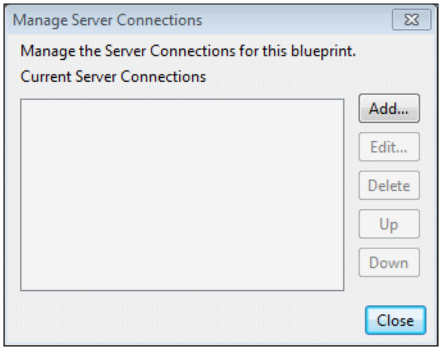


Figure 5-17 Selecting Manage Server Connections

2. In the Manage Server Connections window (Figure 5-18), click **Add**.

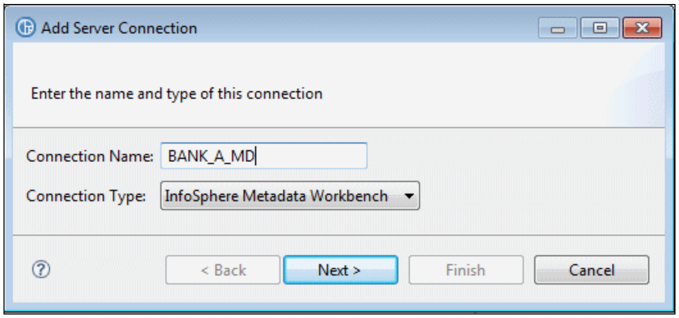

Manage Server Connections

Manage the Server Connections for this blueprint.

Current Server Connections

Figure 5-18 Manage Server Connections window

3. In the Add Server Connection window (Figure 5-19), enter the server connection name and the type. For this scenario, for Connection Type, we select **Infosphere Metadata Workbench**. Then click **Next**.



Add Server Connection

Enter the name and type of this connection

Connection Name: BANK_A_MD

Connection Type: InfoSphere Metadata Workbench

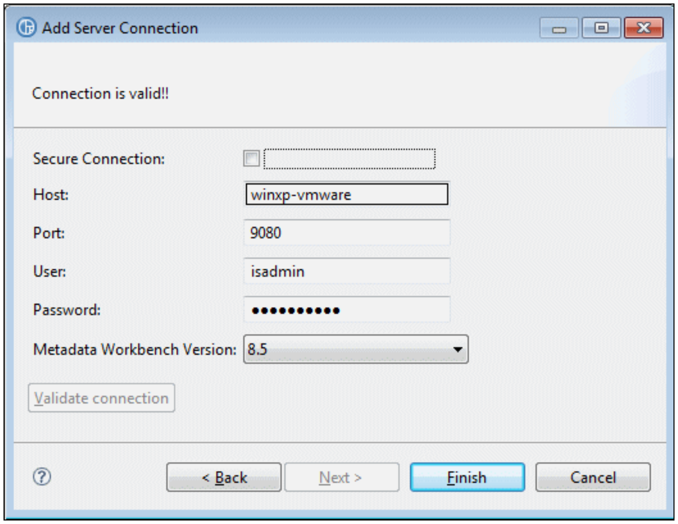
Figure 5-19 Entering the connection name and type

4. In the next window, complete these steps:

   a. Enter the host and authentication information.

   b. Select the appropriate version of the metadata repository to be connected. InfoSphere Blueprint Director can connect to InfoSphere Information Server Version 8.1.2 and later.

   c. To save time, before you apply the changes, test the connection. In the Add Server Connection window (Figure 5-20), click **Validate connection**.



Add Server Connection

Connection is valid!!

Secure Connection: ☐

Host: winxp-vmware

Port: 9080

User: isadmin

Password: ••••••••••

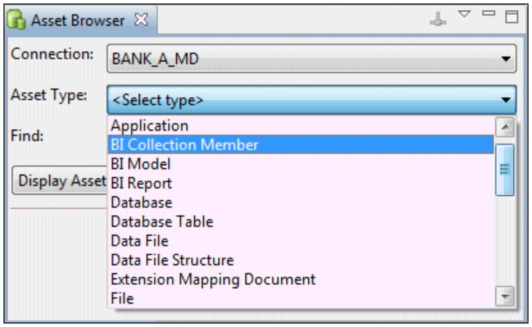
Metadata Workbench Version: 8.5

Figure 5-20 Testing the server connection

5. After the server is connected, in the Manage Server Connection window, look for the new entry. In this scenario, we do not need to use it for now. Therefore, close the window.

Exploring metadata

After connecting InfoSphere Blueprint Director with InfoSphere Metadata Workbench, you can explore the metadata through the **Asset Browser** tab. On this tab, the connection you created is available, and you can retrieve the content in the metadata repository. The type-based structure shown in Figure 5-21 is the same as the one used in InfoSphere Metadata Workbench for assets display.



Connection: BANK_A_MD

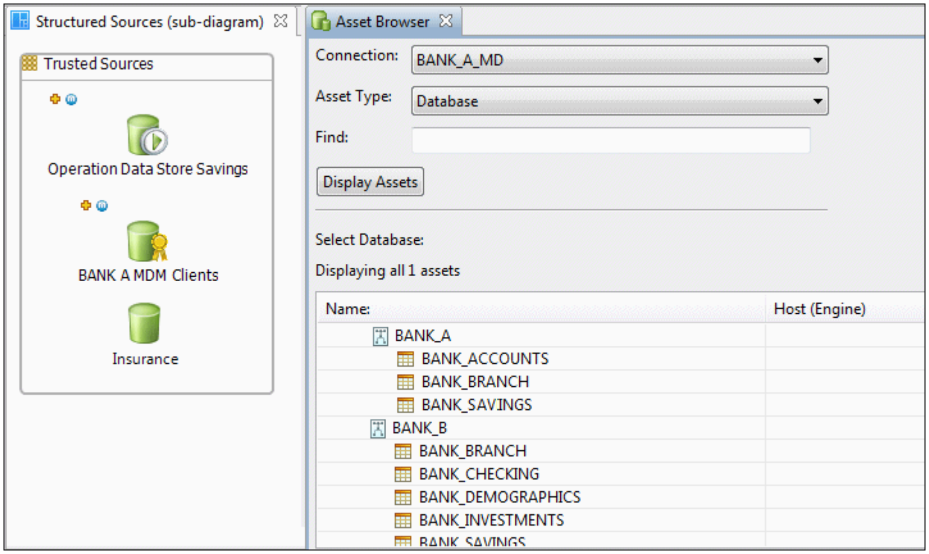
Asset Type: <Select type>

Find:

Figure 5-21 Asset Browser

Linking metadata to Blueprint objects (artifacts)

Now that you can browse the metadata, you can look for the information asset required in the blueprint. It is a table named *savings* that points the database component as shown in Figure 5-22.



Connection: BANK_A_MD

Asset Type: Database

Find:

Select Database:

Displaying all 1 assets

Name:	Host (Engine)
BANK_A	
BANK_ACCOUNTS	
BANK_BRANCH	
BANK_SAVINGS	
BANK_B	
BANK_BRANCH	
BANK_CHECKING	
BANK_DEMOGRAPHICS	
BANK_INVESTMENTS	
BANK_SAVINGS	

Figure 5-22 Displaying assets

You can see the asset identified, which is the BANK_SAVINGS table, and the artifact INSURANCE database in the blueprint subdiagram that will be linked.

To add an asset link, complete these steps:

1. In the editor area (Figure 5-23), right-click the artifact, and select **Add Asset Link**.

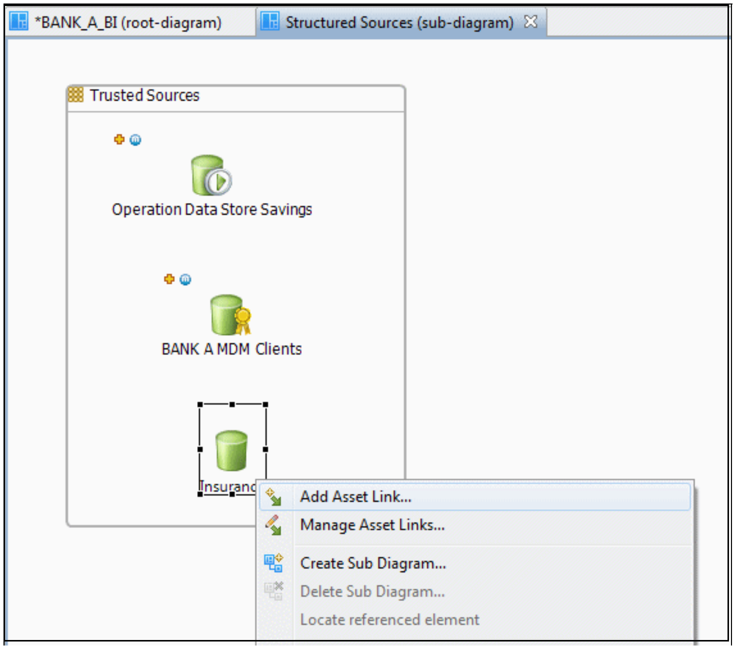


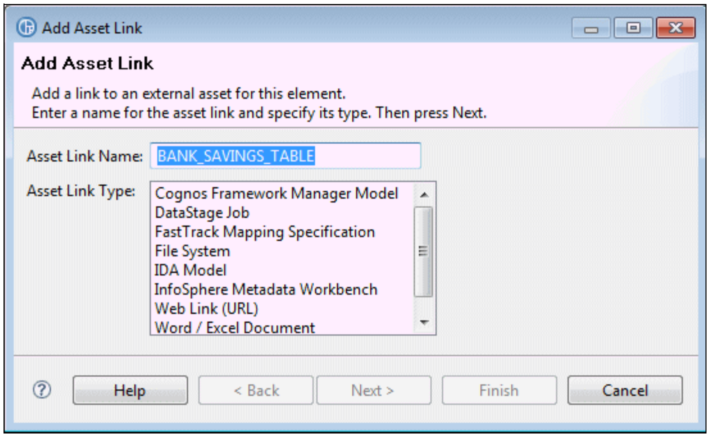
Figure 5-23 Adding an asset link

2. In the Add Asset Link window (Figure 5-24 on page 111), enter a meaningful name for the asset link and select the link type.

   For this scenario, we enter `BANK_SAVINGS_TABLE` as the name and select **InfoSphere Metadata Workbench** as the link type. The asset we want to link is already loaded in InfoSphere Metadata Workbench. If you have not uploaded InfoSphere Metadata Workbench with asset information, load the asset information there, and return to the blueprint to add the link.

   > **Tip:** Browse and review different options for link types. InfoSphere Blueprint Director provides a long list that you can use to link assets.

   Then click **Next**.


Add Asset Link

Add Asset Link

Add a link to an external asset for this element.
Enter a name for the asset link and specify its type. Then press Next.

Asset Link Name: BANK_SAVINGS_TABLE

Asset Link Type:
- Cognos Framework Manager Model
- DataStage Job
- FastTrack Mapping Specification
- File System
- IDA Model
- InfoSphere Metadata Workbench
- Web Link (URL)
- Word / Excel Document

Figure 5-24 Adding a link type

3. In the next window, complete these steps:

   a. Select a connection. Use the current `BANK_A_MD`.
   b. Choose the appropriate asset type, which is a table in this case.
   c. Retrieve all of the assets.
   d. Find and select the table to be associated.
   e. Click **Finish**.

   The asset is now linked to the component in the subdiagram.

4. To validate the asset link, or review if any other component is linked to an asset, click the green arrow that points to the component to obtain more information about the asset linked. Alternatively, with the pointer, hover over the artifact in the editor area, and a tooltip is displayed with this information.

5. In the editor area, review the asset information, and click the tooltip that contains the name of the asset is displayed. You then go to InfoSphere Metadata Workbench where you can browse the information asset in detail.

5.3.3 Working with a business glossary

In InfoSphere Blueprint Director, you can add business glossary terms to the components to give precise context about the information assets used in your blueprint. For this scenario, we add the terms to our blueprint.

InfoSphere Information Server provides a glossary of terms with their definitions, organized into categories that provide containment, reference, and context. You can use this glossary in the blueprint that links to the metadata repository. This way, the blueprint with diagrams that describe the project and components links to information assets and with the description or definition.

If you have not built a glossary or worked on InfoSphere Business Glossary, see Chapter 6, “Building a business-centric vocabulary” on page 131, to build one. Then, you can return to this section to connect your glossary with your blueprint.

Connecting to InfoSphere Business Glossary

First, you must connect and navigate to InfoSphere Business Glossary from InfoSphere Blueprint Director:

1. Go to the tab group, and open the **Glossary Explorer** tab (Figure 5-25).
2. Right-click **Glossary**, and select the **Preferences** option.

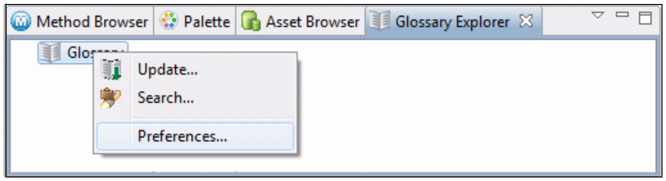
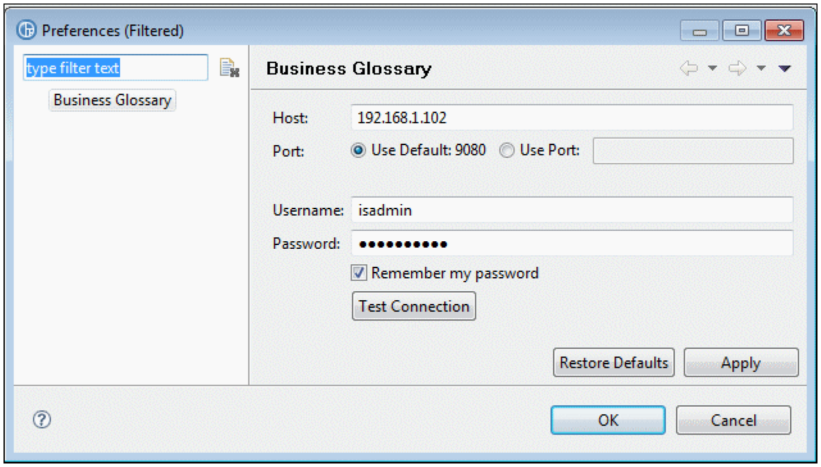


Figure 5-25 Selecting the Glossary Preference option

3. In the Preferences (Filtered) dialog box (Figure 5-26), add the appropriate information in the fields to connect InfoSphere Blueprint Director to InfoSphere Business Glossary. Enter a host, user name, and password. Click **Apply**. To ensure that the information that you entered is correct, click **Test Connection**. After the test is complete, click **OK**.



Host: 192.168.1.102

Port: ● Use Default: 9080 ○ Use Port:

Username: isadmin

Password: ••••••••••

☑ Remember my password

Figure 5-26 Connecting InfoSphere Blueprint Director to InfoSphere Business Glossary

4. After InfoSphere Blueprint Director is connected to InfoSphere Business Glossary, update the glossary, which you can choose to do now or later. To update the glossary, go back to this **Glossary Explorer** tab, right-click **Glossary**, and select **Update** (Figure 5-27).

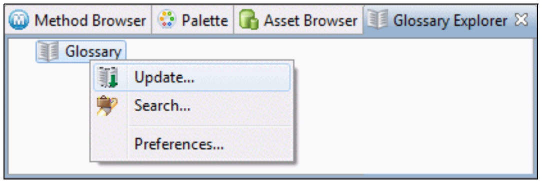
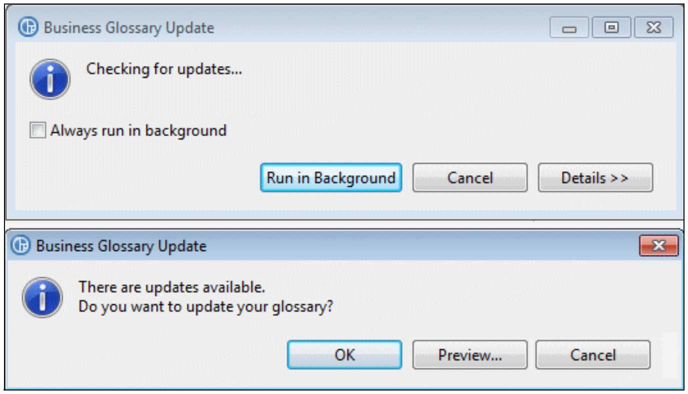


Figure 5-27 Updating the glossary in InfoSphere Blueprint Director

The system checks for any glossary update (upper window in Figure 5-28).

5. If the glossary has a change, when the system prompts you to accept the update, as indicated by the lower window in Figure 5-28, click **OK**. Then you receive important information about the components within the glossary that are changes. Alternatively click **Preview** to view the changes.



Checking for updates...

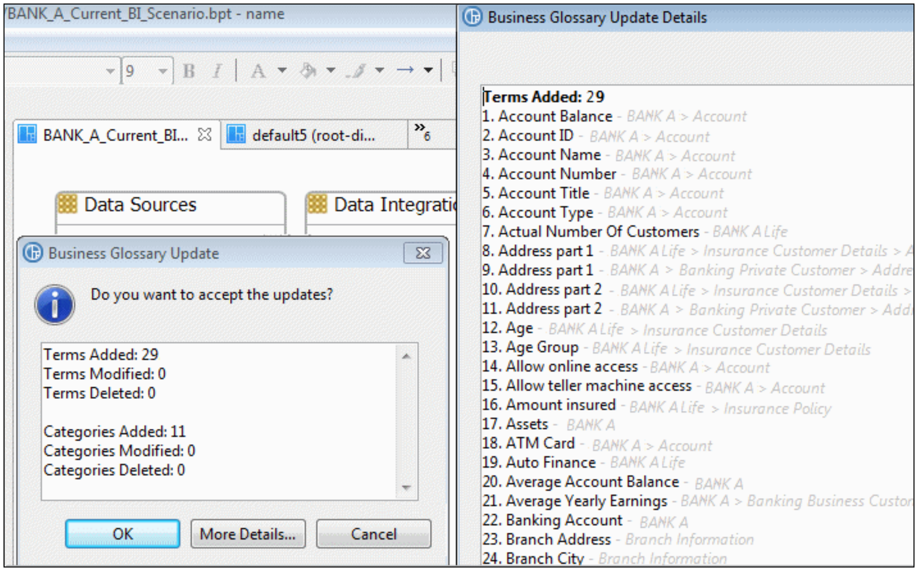
□ Always run in background

There are updates available.
Do you want to update your glossary?

Figure 5-28 Update glossary within InfoSphere Blueprint Director

6. In the summary window (Figure 5-29), review the details of the glossary updates. A full list of terms and categories that have been updated is show. You can also see a summary of the quantity of the components that have been added or deleted.

   Click **OK** to accept the update, or click **Cancel**. For this scenario, we add a glossary in Chapter 6, “Building a business-centric vocabulary” on page 131, and return to the blueprint to update the glossary. We click **OK** to accept the update of the glossary.



Do you want to accept the updates?

Terms Added: 29
Terms Modified: 0
Terms Deleted: 0

Categories Added: 11
Categories Modified: 0
Categories Deleted: 0

Terms Added: 29
1. Account Balance - BANK A > Account
2. Account ID - BANK A > Account
3. Account Name - BANK A > Account
4. Account Number - BANK A > Account
5. Account Title - BANK A > Account
6. Account Type - BANK A > Account
7. Actual Number Of Customers - BANK A Life
8. Address part 1 - BANK A Life > Insurance Customer Details > A
9. Address part 1 - BANK A > Banking Private Customer > Addre
10. Address part 2 - BANK A Life > Insurance Customer Details >
11. Address part 2 - BANK A > Banking Private Customer > Add
12. Age - BANK A Life > Insurance Customer Details
13. Age Group - BANK A Life > Insurance Customer Details
14. Allow online access - BANK A > Account
15. Allow teller machine access - BANK A > Account
16. Amount insured - BANK A Life > Insurance Policy
17. Assets - BANK A
18. ATM Card - BANK A > Account
19. Auto Finance - BANK A Life
20. Average Account Balance - BANK A
21. Average Yearly Earnings - BANK A > Banking Business Custor
22. Banking Account - BANK A
23. Branch Address - Branch Information
24. Branch City - Branch Information

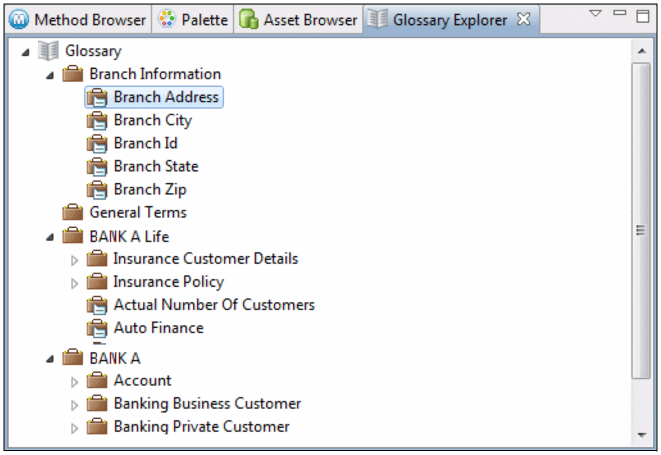
Figure 5-29 Changes in glossary

After the glossary is updated from the server, you can now apply business terms or categories to your blueprint.

Applying terms and categories to your blueprint

To apply terms and categories from InfoSphere Business Glossary to your blueprint in InfoSphere Blueprint Director, follow these steps:

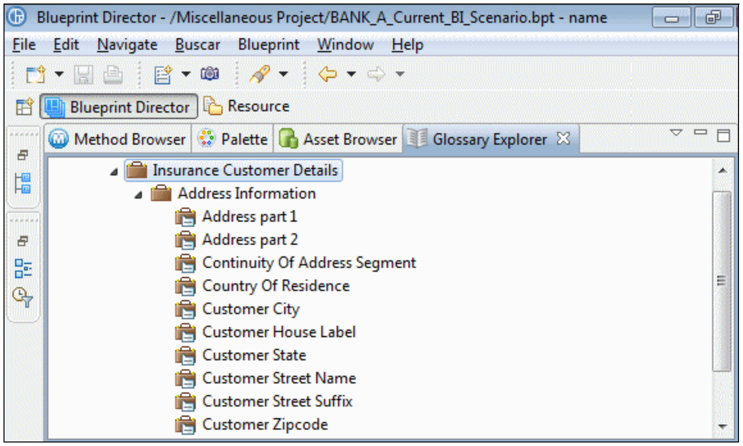
1. Navigate through the Glossary Explorer to see the list business terms and categories available as shown in Figure 5-30.



- Glossary
  - Branch Information
    - Branch Address
    - Branch City
    - Branch Id
    - Branch State
    - Branch Zip
  - General Terms
  - BANK A Life
    - Insurance Customer Details
    - Insurance Policy
    - Actual Number Of Customers
    - Auto Finance
  - BANK A
    - Account
    - Banking Business Customer
    - Banking Private Customer

Figure 5-30 Glossary Explorer tab showing the list of business terms and categories

To find more information about a term, click it, and then you see a description of that term as shown in Figure 5-31.



- Insurance Customer Details
  - Address Information
    - Address part 1
    - Address part 2
    - Continuity Of Address Segment
    - Country Of Residence
    - Customer City
    - Customer House Label
    - Customer State
    - Customer Street Name
    - Customer Street Suffix
    - Customer Zipcode

Figure 5-31 Detail about a business term

2. Add business terms in your diagram.

   As an example, we create a sample diagram and add three business terms to the diagram by using a drag-and-drop approach. A conceptual entity is created for each one (Figure 5-32).

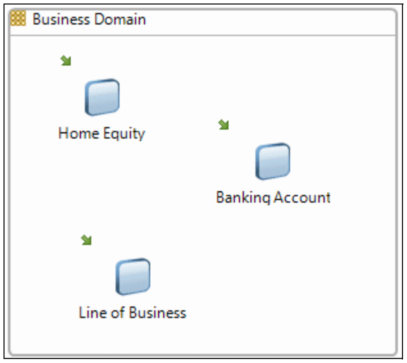


Figure 5-32 Adding business terms to a blueprint diagram

You can also apply a business glossary to your blueprint diagram by using these methods:

- You can use the business term with other components of the diagram. For this scenario, we link Business Group to Insurance as shown in Figure 5-33.

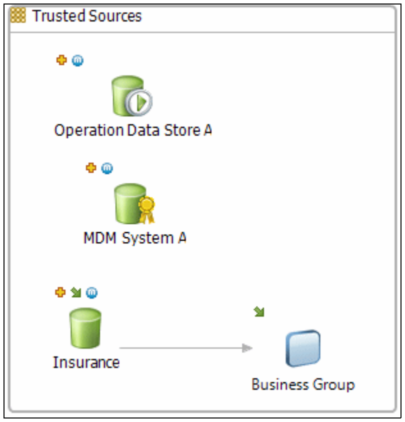


Figure 5-33 Linking a business terms to a component

- You can add a business term to an information asset instead of showing it as an entity by dragging the business term to the asset. A green arrow is displayed with the asset as shown in Figure 5-34.

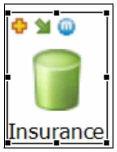


Figure 5-34 An information asset linked to a business term

The green arrow is also displayed when you link a component to an information asset. The green arrow indicates that a component is linked to the metadata repository, regardless whether the component is an information asset, business term, category, or steward. A component can be linked to many metadata repository assets at one time.

- After linking the asset to a business term, from this asset, you can navigate to the business term. To do this navigation, click the green arrow to see a tooltip that shows the business term (Figure 5-35).

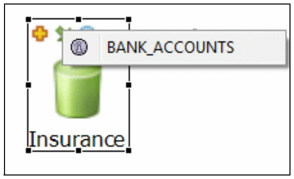
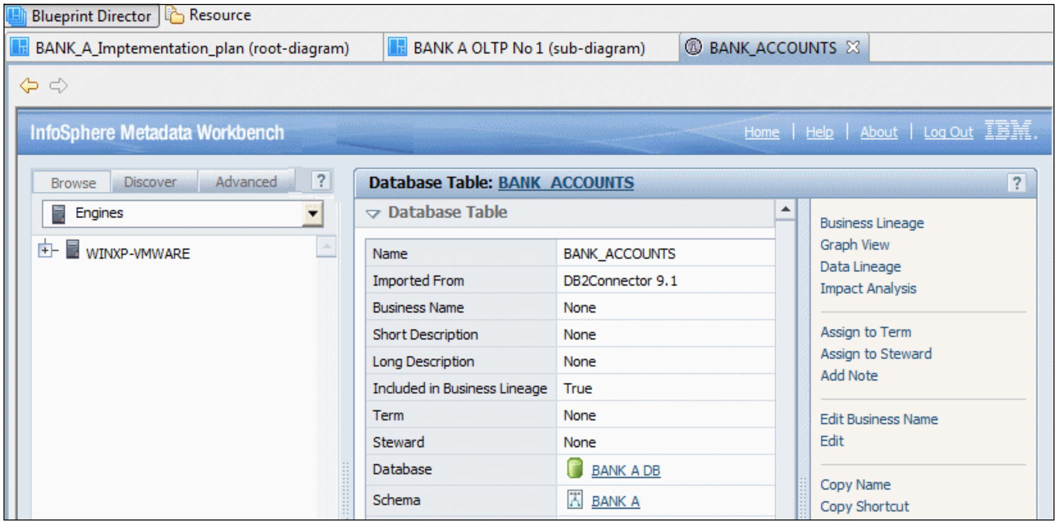


Figure 5-35 Navigating to a business term from an information asset

If you click the name, you connect to InfoSphere Business Glossary in InfoSphere Metadata Workbench as shown in Figure 5-36.



Database Table: BANK_ACCOUNTS

Database Table

Name	BANK_ACCOUNTS
Imported From	DB2Connector 9.1
Business Name	None
Short Description	None
Long Description	None
Included in Business Lineage	True
Term	None
Steward	None
Database	BANK A DB
Schema	BANK A

Figure 5-36 InfoSphere Business Glossary term in the editor area of InfoSphere Blueprint Director

You can continue the analysis with all the InfoSphere Business Glossary capabilities without leaving InfoSphere Blueprint Director. InfoSphere Business Glossary is held at a new tab opened in the editor area.

5.4 Working with milestones

A milestone is useful in cases where you want to show different snapshots of a project or blueprint. By using milestones, you can show how the blueprint of a project changes over time. If a project has multiple major phases, by putting each phase in a milestone, you can see what you have to do or accomplish in each phase. Milestones provide an overall view of the entire project and the fine detail of individual milestone.

For this scenario, in Bank A, the blueprint that we have built so far represents the current architecture of the bank before the acquisition. The objective is to set this initial state before adding the information assets of Bank B in the picture.

We use the blueprint customized in 5.3.1, “Customizing the template” on page 101, as a starting point (Figure 5-37).

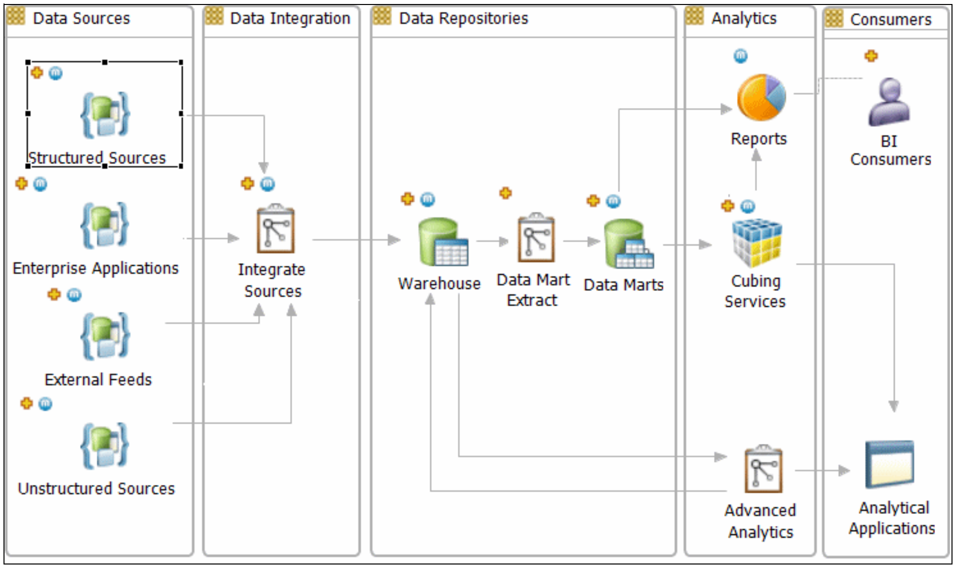


Figure 5-37 Initial blueprint of Bank A that reflects its current system

For this scenario, Bank A needs to add the information for Bank B that resides in the transactional system of Bank B to feed to the existent reporting solution. The main changes in the new system and the blueprint are in the Data Sources and Data Integration domains. These domains are where we need to connect the assets of Bank B and perform appropriate data integration.

We update the Data Sources and Data Integration domain by adding more details to their subdiagrams. For the simplicity of the demonstration, we define the change scope in the following three main goals:

1. Obtain an initial state with only Bank A data sources.
2. By using milestones, generate a second state where the trusted data sources of Bank B are included in the environment.
3. Create a third state where these sources are included in the Data Integration subdiagram.

Adding milestones to the project

To add milestones for the project, complete these steps:

1. Selecting **Window** → **Show View** → **Timeline** (Figure 5-38) to open the Timeline window.

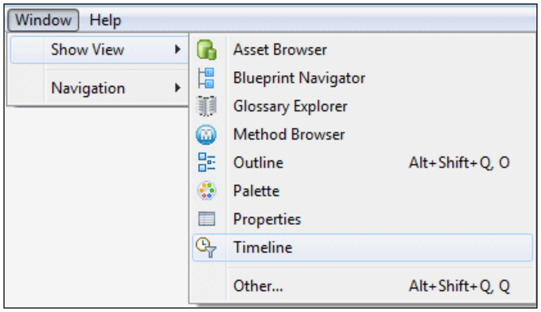
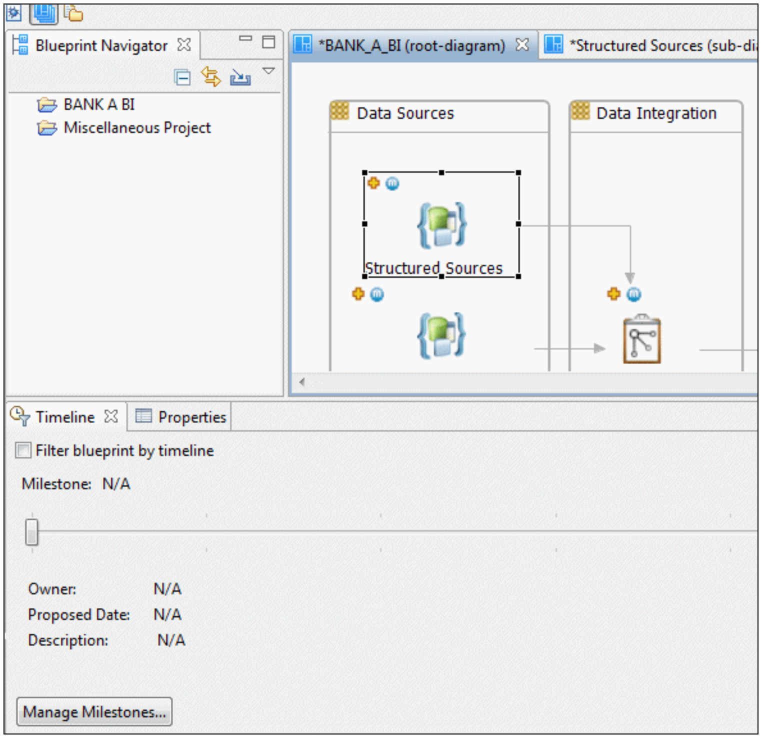


Figure 5-38 Selecting the Timeline view

The Timeline window is where the milestones reside. By default, the Timeline information is displayed in the bottom frame as highlighted in Figure 5-39.



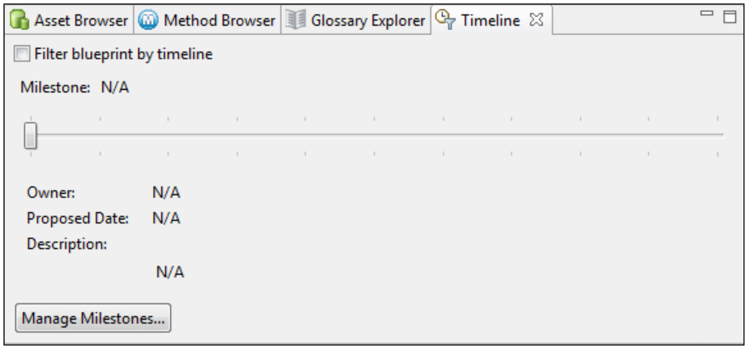
☐ Filter blueprint by timeline

Milestone: N/A

Owner: N/A
Proposed Date: N/A
Description: N/A

Figure 5-39 Timeline frame of a blueprint

2. As with any other window in InfoSphere Blueprint Director, work with the Timeline in this detached mode, or move it to the tab group (Figure 5-40).



☐ Filter blueprint by timeline

Milestone: N/A

Owner: N/A

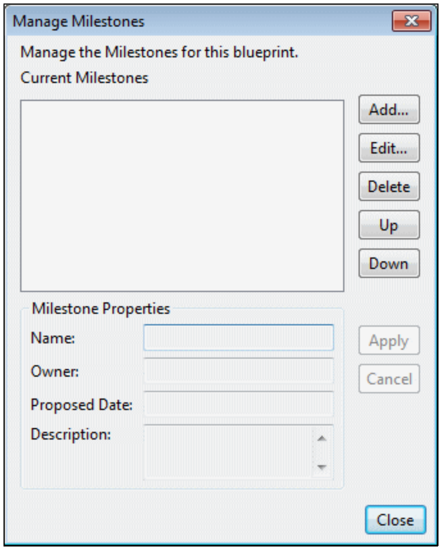
Proposed Date: N/A

Description:

N/A

Figure 5-40 Timeline moved to the tab group

3. To add a milestone or remove an existing one, click **Manage Milestones**.
4. In the Manage Milestones window (Figure 5-41), click **Add** to add the milestones you need. Under Milestone Properties, enter an owner and description to provide more detail to other users.

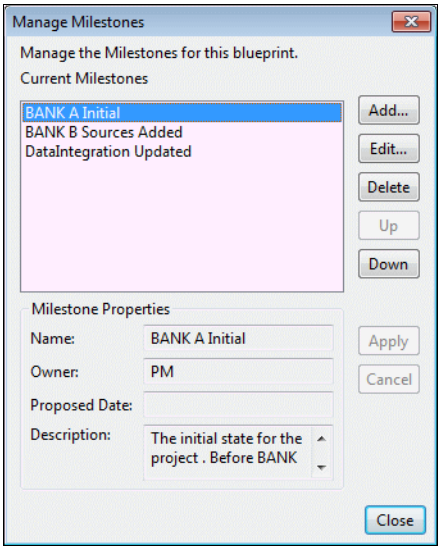

Manage Milestones

Manage the Milestones for this blueprint.

Current Milestones

Figure 5-41 Manage Milestone window

For this scenario, we create three milestones as shown in Figure 5-42.


Manage Milestones

Manage the Milestones for this blueprint.

Current Milestones

BANK A Initial
BANK B Sources Added
DataIntegration Updated

Milestone Properties

Name: BANK A Initial

Owner: PM

Proposed Date:

Description: The initial state for the project . Before BANK

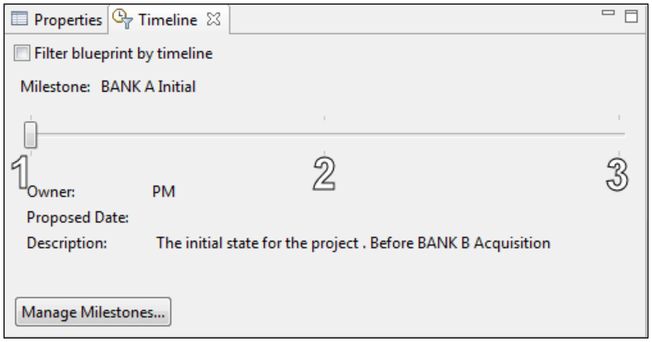
Figure 5-42 Three milestones for our use-case scenario

Then click **Close**.

5. Review the timeline. Figure 5-43 shows the Timeline for this scenario. The numbers show where each mark point is in the project. This timeline shows the following marked points indicating the number of milestones that we created:

   – Bank A Initial
   – Bank B Sources Added
   – Data Integration Updated

   The distribution of the mark point changes depending on the total amount of milestones used in the project.



□ Filter blueprint by timeline

Milestone: BANK A Initial

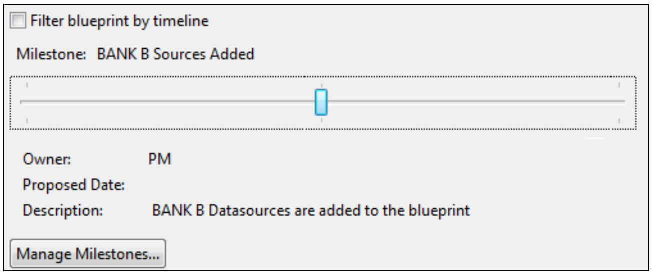
Owner: PM

Proposed Date:

Description: The initial state for the project . Before BANK B Acquisition

Figure 5-43 Timeline for Bank A Initial

Move the bar to the remaining marking points to see the information provided in the description. Milestone 2 looks similar to the example in Figure 5-44.



□ Filter blueprint by timeline

Milestone: BANK B Sources Added

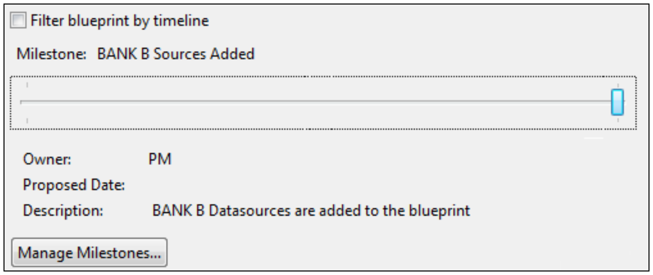
Owner: PM

Proposed Date:

Description: BANK B Datasources are added to the blueprint

Figure 5-44 Timeline for the second milestone

Milestone 3 looks similar to the example in Figure 5-45.



□ Filter blueprint by timeline

Milestone: BANK B Sources Added

Owner: PM

Proposed Date:

Description: BANK B Datasources are added to the blueprint

Figure 5-45 Timeline for the third milestone

Important: After you create your milestones, be careful about other changes that you make. Ensure that any updates in your blueprint reflect the milestones expected.

Modifying the milestones

With the milestones created, you can start modifying the blueprint according to your plan. For this scenario, we use the following steps:

1. Because the first milestone is ready, move it to the second milestone, Bank B Sources Added, where the first round of changes is held.
2. Open the Structured Sources subdiagram that was customized previously. Add another domain that includes Bank B data sources (Figure 5-46).

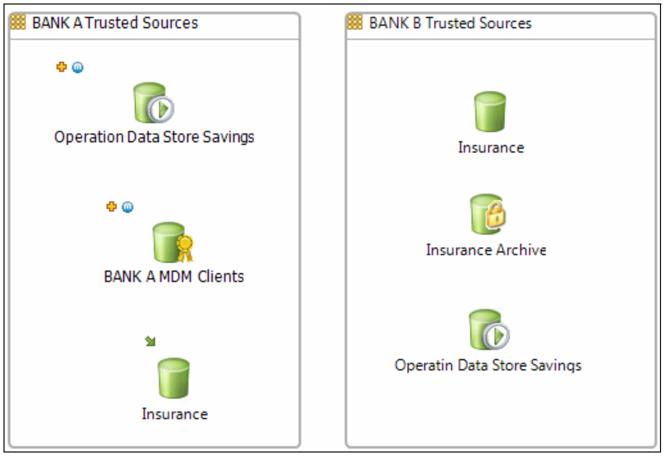
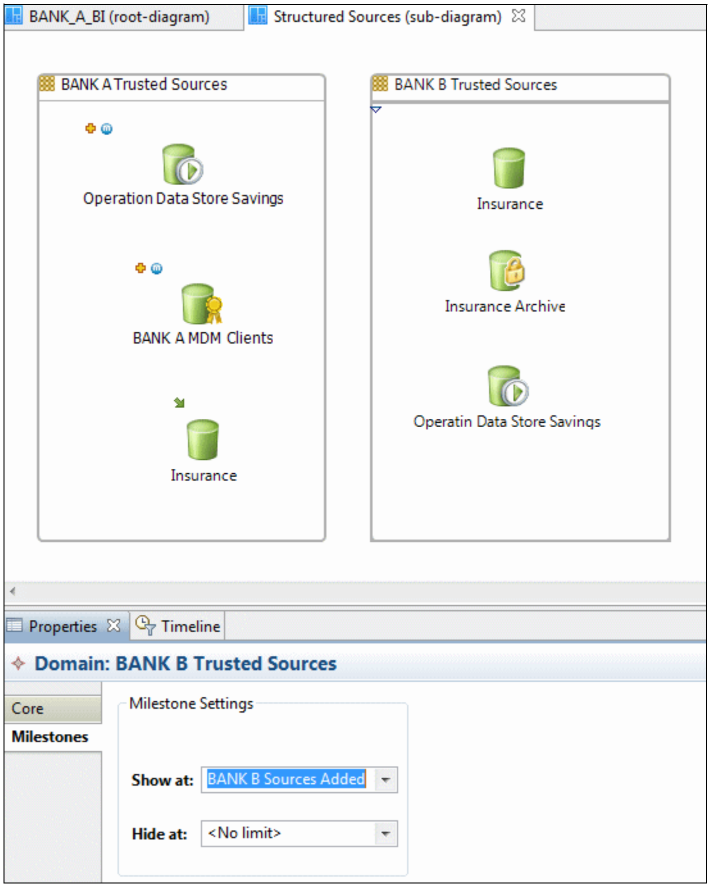


Figure 5-46 Adding Bank B trusted sources to the Structured Sources subdiagram

The subdiagram now includes the approved Bank B data sources to be used in the project.

3. Select the **Bank B Trusted Sources** domain.
4. Go to the Properties window, and configure the domain. In the Properties window (Figure 5-47), in the Milestones area, modify the Show at and Hide at properties.

   For this scenario, we want to show these domains at all times from this milestone forward. Therefore, set the Hide at field to **<No Limit>**.



Domain: BANK B Trusted Sources

Milestone Settings

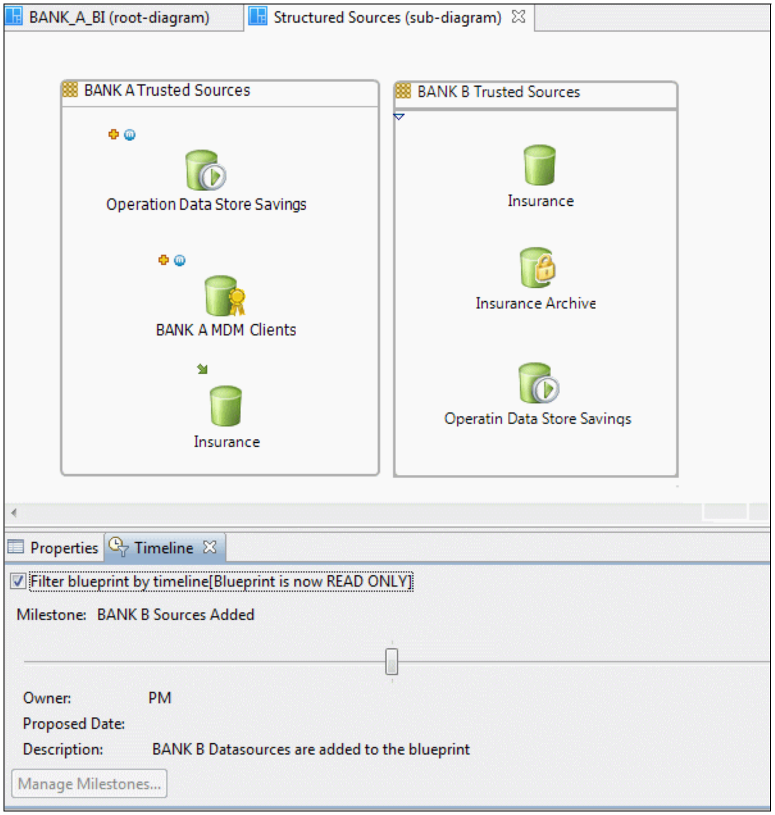
Show at: BANK B Sources Added

Hide at: <No limit>

Figure 5-47 Specifying the Show at and Hide at settings

5. To see how these settings work, go back to the **Timeline** tab, and select the **Filter Blueprint by timeline** option. When checking this option, the blueprint goes into read-only mode. You cannot manage milestones with this option.

   For this scenario, the timeline is displayed, but the **Manage Milestones** button is disabled as shown in Figure 5-48.



☑ Filter blueprint by timeline[Blueprint is now READ ONLY]

Milestone: BANK B Sources Added

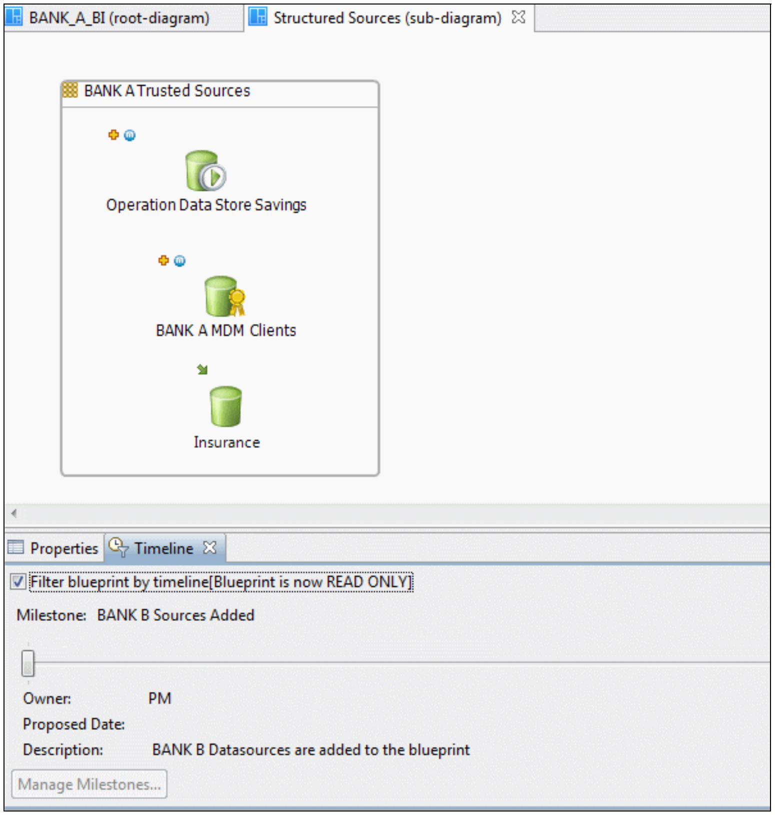
Owner: PM

Proposed Date:

Description: BANK B Datasources are added to the blueprint

Figure 5-48 Filter blueprint by timeline

If we move the bar starting from the initial milestone, the second domain from Bank B is not in the subdiagram as shown in Figure 5-49.



☑ Filter blueprint by timeline[Blueprint is now READ ONLY]

Milestone: BANK B Sources Added

Owner: PM

Proposed Date:

Description: BANK B Datasources are added to the blueprint

Figure 5-49 Structured Sources subdiagram for the initial milestone

You can make the required changes in the next milestone. A milestone shows how a blueprint can evolve according to business requirements and plan for each phase. It is also useful to maintain a blueprint, instead of having several versions of the same document.

You can add more milestones if subphases emerge. You can also use them to show temporary components that are useful during the development but that are not used when a project is finished. By using the show and hide features on the Properties window, you can show a component or take it out of the scope for a specific time period.

5.5 Using methodology

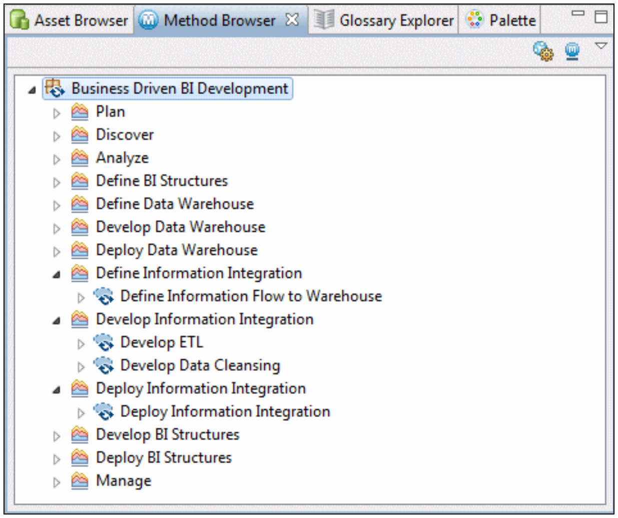
For this use case, we create a blueprint from an appropriate template. Many advantages can be obtained by working with existing templates. One advantage is working with method elements that provide contextual guidance about designing and development.

Every information integration project has specific steps or activities where methodology or best practices are available. Bank A stakeholders can access several method elements and link them to the artifacts in the blueprint. This approach gives context to the process and details expectations for this task.

The following steps explore the method browser and explain what is available in the current blueprint:

1. Go to the tab group, and select **Method Browser** (Figure 5-50).

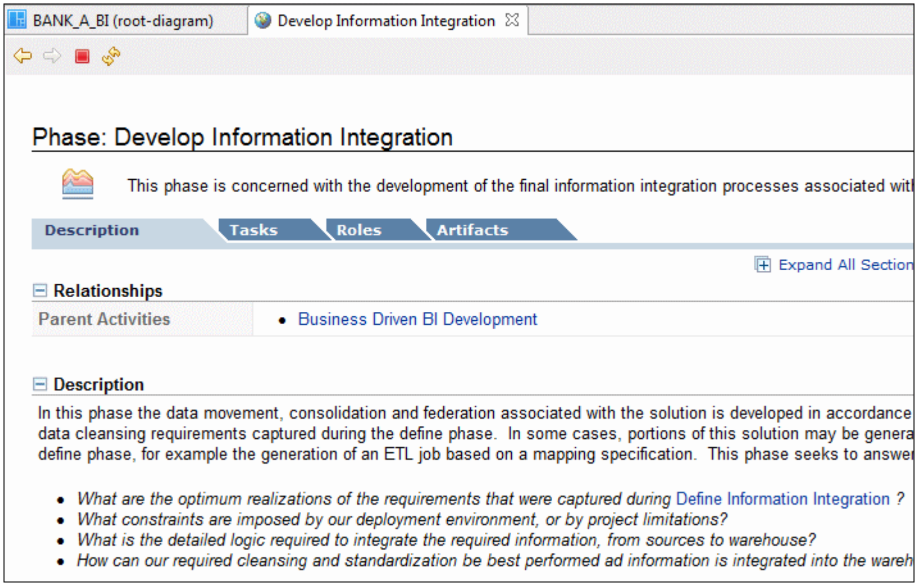
   As shown in Figure 5-50, the Business Driven BI Development method list is displayed because we use it as the template. A list of phases and nodes is also shown. These phases are steps that are expected to be accomplished in every project according to the best practices for each case.



- Business Driven BI Development
  - Plan
  - Discover
  - Analyze
  - Define BI Structures
  - Define Data Warehouse
  - Develop Data Warehouse
  - Deploy Data Warehouse
  - Define Information Integration
    - Define Information Flow to Warehouse
  - Develop Information Integration
    - Develop ETL
    - Develop Data Cleansing
  - Deploy Information Integration
    - Deploy Information Integration
  - Develop BI Structures
  - Deploy BI Structures
  - Manage

Figure 5-50 The Method Browser tab

2. Navigate through the Develop Information Integration node. Double-click the node to obtain more information as shown in Figure 5-51.



Phase: Develop Information Integration

This phase is concerned with the development of the final information integration processes associated wit

Relationships

Parent Activities • Business Driven BI Development

Description

In this phase the data movement, consolidation and federation associated with the solution is developed in accordance data cleansing requirements captured during the define phase. In some cases, portions of this solution may be genera define phase, for example the generation of an ETL job based on a mapping specification. This phase seeks to answe

- *What are the optimum realizations of the requirements that were captured during* Define Information Integration *?*
- *What constraints are imposed by our deployment environment, or by project limitations?*
- *What is the detailed logic required to integrate the required information, from sources to warehouse?*
- *How can our required cleansing and standardization be best performed ad information is integrated into the wareh*

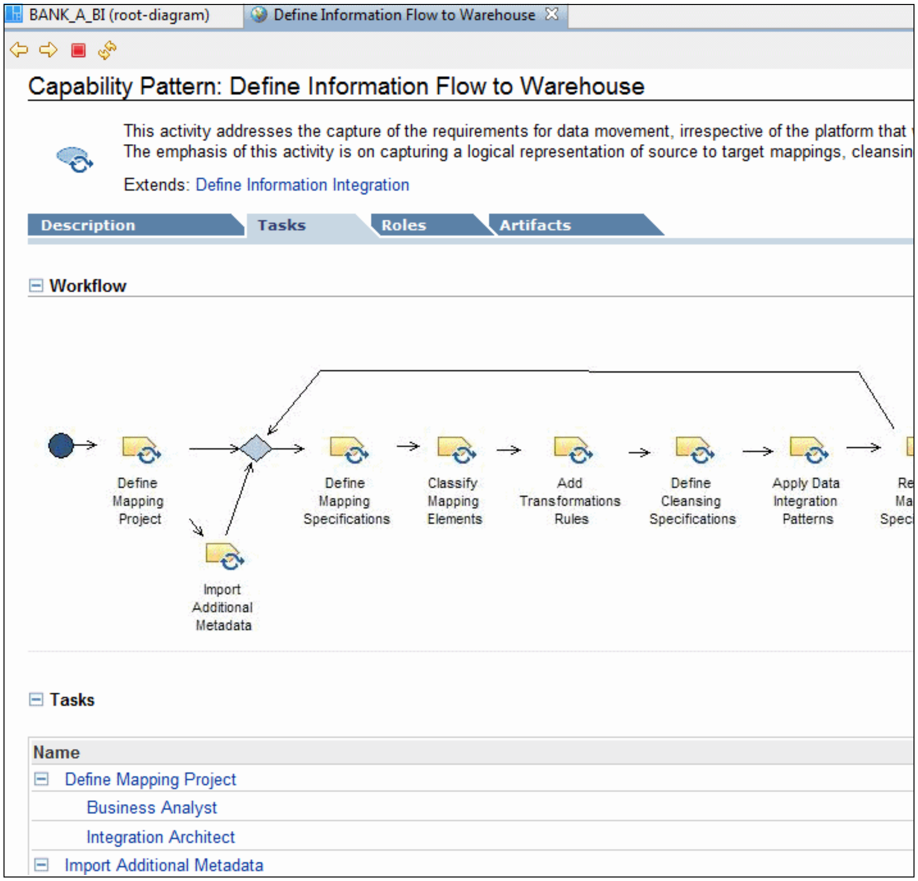
Figure 5-51 Information integration project phase description

The content includes tabs with information for each phase:

– Description
– Tasks
– Roles
– Artifacts

Notice that this tab layout is used in any level because InfoSphere Blueprint Director methodology is organized hierarchically, which is useful for obtaining more detail.

3. Click the **Tasks** tab to see all the tasks involved as shown in Figure 5-52.



BANK_A_BI (root-diagram) | Define Information Flow to Warehouse

Capability Pattern: Define Information Flow to Warehouse

This activity addresses the capture of the requirements for data movement, irrespective of the platform that
The emphasis of this activity is on capturing a logical representation of source to target mappings, cleansin

Extends: Define Information Integration

Description | Tasks | Roles | Artifacts

Workflow

Define Mapping Project
Import Additional Metadata
Define Mapping Specifications
Classify Mapping Elements
Add Transformations Rules
Define Cleansing Specifications
Apply Data Integration Patterns
Re
Ma
Spec

Tasks

Name
Define Mapping Project
Business Analyst
Integration Architect
Import Additional Metadata

Figure 5-52 Information integration project tasks

5.6 Conclusion

In conclusion, this chapter explained how InfoSphere Blueprint Director helps to standardize project planning with efficiency and consistency. It demonstrated how InfoSphere Blueprint Director provides ready-to-use templates, visual design canvas, and tools that you can use for your project needs.

The remaining chapters outline the specific steps of the implementation process for your project. These tasks include building a business-centric vocabulary, source documentation, data relationship discovery, data quality assessment and monitoring, and more.

6

Building a business-centric vocabulary

IBM InfoSphere Business Glossary, one of the IBM InfoSphere Information Server product modules, is the centerpiece of metadata management. It provides the entry point for users who are searching for information about assets in the repository.

This chapter provides an introduction to the process of creating and organizing a business vocabulary. It also explains how to populate InfoSphere Business Glossary with categories and terms and how to maintain those categories and terms.

The chapter includes the following sections:

- Introduction to InfoSphere Business Glossary
- Business glossary and information governance
- Creating the business glossary content
- Deploying a business glossary
- Managing the term authoring process with a workflow
- Searching and exploring with InfoSphere Business Glossary
- Multiple ways of accessing InfoSphere Business Glossary
- Conclusion

6.1 Introduction to InfoSphere Business Glossary

The practice of establishing a common vocabulary for an organization is widespread. A common vocabulary improves communication and removes ambiguities within the organization, leading to higher productivity and better utilization of resources. InfoSphere Business Glossary provides a framework where such a vocabulary can be created, nurtured, and promoted for the benefit of the organization.

Common shared business vocabulary is in the heart of information governance, data quality, and metadata management practices deployed by an organization. It is a vehicle of communication where business and IT are on the same plane with no gaps in understanding.

Consider the following questions:

- What is the other party talking about?
- What do the terms in a requirements document mean?
- Do the specifications accurately reflect the requirements?

By having a common vocabulary in InfoSphere Business Glossary, business and IT communities have access to a comprehensive body of information. They also have knowledge about the data the company generates, processes, stores, and uses to support its operations. Through a well-designed hierarchy (*taxonomy*) and carefully selected and properly formatted business terms (*business vocabulary*), users can navigate, browse, and search for information about business terms, their meaning and usage, and the IT assets used to realize them. The ability to retrieve information about data, its source, meaning, usage, and various aspects of processing it promotes the understanding of and trust in the data. It also enhances the efficiency of the processes concerning generation and use of the data.

In addition to providing an authoritative source for the terms and their meaning, the combination of InfoSphere Business Glossary and InfoSphere Metadata Workbench provides access to a rich store of knowledge and analytical capabilities. Business people have at their fingertips answers to data questions such as the following examples:

- What information is out there?
- What does it mean?
- Where is it stored?
- How is it being processed and used?
- When was this information last refreshed?
- Who owns this information?

6.2 Business glossary and information governance

Information governance initiatives are deployed to promote the following goals:

- Increase consistency and confidence in decision making.
- Decrease the risk of regulatory fines.
- Improve data security.
- Provide consistent information quality across the organization.
- Maximize the income generation potential of data.
- Designate accountability for information quality.

As explained in the following sections, a business glossary has many of the attributes that are required to support information governance. In essence, all of these attributes depend on the ability to create, preserve, and disseminate knowledge about information across the organization. This knowledge includes awareness about what the information is, how it is used, and who uses it. It also includes awareness about where the information is coming from, what happens to it along the way, and where it ends up.

A cornerstone for such knowledge is the *business glossary*. It starts with giving data names and definitions that are common and agreed upon by the community of users. By giving an object or concept a name, it can be located, tracked, assigned, and secured. Having these names created according to guidelines with a discipline of approved and chartered process creates consistency and instills confidence. Users who search for information rely on the authority of such a source to provide the correct and complete information.

A business glossary goes beyond just a list of terms. Linking terms to IT assets establishes a connection between business and IT and enhances collaboration between parties and users. General business users have fast access to commonly used vocabulary terms and their meaning often with additional information, such as any constraints and flags indicating special treatment. A business analyst has a better understanding of the terms used in business requirements, which translate to better and faster translation into technical requirements and specifications. By viewing the IT assets assigned to a term, data analysts and developers can be more precise in their job development or report design.

A business glossary provides an environment of sharing and collaboration so that you can achieve information governance goals. A business glossary helps achieve these goals in the following ways among others:

- Enables data governance
  - With a common language, supports compliance with regulations such as Basel II
  - Represents and exposes business relationships and lineage
  - Tracks a history of changes
- Accountability and responsibility
  - Assigns stewards as a single point of contact
- Improved productivity
  - Allows administrators to tailor the tool to the needs of business users
  - Provides access enterprise information when needed
  - Enables the use and reuse of information assets based on a common semantic hub
- Increased collaboration
  - Captures and shares annotations between team members
  - Offers a greater understanding of the context of information
  - Provides more prevalent use and reuse of trusted information

6.3 Creating the business glossary content

InfoSphere Business Glossary is an InfoSphere Information Server product module. It enables the capture, maintenance, search, and navigation of business terms and the physical assets associated with them. The business glossary is populated with terms that comprise the business vocabulary of the organization. The terms are organized in a hierarchical structure of categories, the taxonomy.

This section addresses the nature of the vocabulary and taxonomy and the process of their construction.

6.3.1 Taxonomy

The word *taxonomy* comes from the Greek words *taxis*, which means arrangement, and *nomia*, which means method. By combining these two words, taxonomy represents the science or method of classification.

Taxonomies provide authority control and facilitate browsing:

Authority control The process of verifying and establishing a preferred form of a proper name or subject term for use in a standardized list.

Browsing The ability to navigate, explore, and discover concepts in an organized structure of information. Classification schemes group related concepts together. This way, if you find an object or concept in a category, it is easy to find other related objects or concepts in the same category.

To achieve these objectives, a taxonomy must satisfy the following criteria:

- Strict classification rules. Categories must have a clear definition of what goes in and what does not. Adherence to this rule helps to keep the taxonomy relevant and useful.
- Mutually exclusive. A term can belong in one category only. A term might appear in more than one category if it has a different meaning in different contexts (to be avoided).
- Collectively exhaustive categories. Every term must belong to a category, and a category must exist for every term.
- No miscellaneous category. Avoid having a miscellaneous category for terms that do not fit all other categories. Use of this type of category promotes casualness in assigning terms to categories that will result in deterioration of the classification quality and usefulness.

Much of the success of the glossary and its usability depends on the structure of the category hierarchy that the team creates to contain the vocabulary. The hierarchy must reflect a world vision that is acceptable and agreeable by a majority of the users. Failure to meet this basic requirement frustrates and discourages users from using the glossary. The structure of the taxonomy provides a navigation path for search. It also provides a context in which the definition of a term is extended beyond the text that is in the definition field.

The creation of a taxonomy must not be done haphazardly, but through a careful process of planning, review, and validation. The users community must be able to review and validate the proposed structure and to assess its usability.

6.3.2 The taxonomy development process

Developing a taxonomy requires defining a scope and making development and design decisions.

Defining the scope

The scope is determined by the business glossary governance body for setting priorities and an approach to creating the business glossary.

Designing the taxonomy

Develop a business glossary that is scalable and flexible. You do not need to start with a well-established data dictionary to build a business glossary. If you have some categories and terms, you can build a simple glossary. Later, you can develop and expand the glossary contents. Start at a high level, such as the domain level, and then descend to lower-level details.

Important: Business people must participate in the process by expressing their view of the organization and their knowledge of it.

The initial design is done by the team based on preconceived notions of what a taxonomy should be. The team might follow an operational model, a data model, or a process model to create the initial breakdown of the domain that they decide to work on.

Consider using the following recommended approach:

1. Identify a small number (4–6) of major categories (subject areas). Starting with one subcategory, attempt to define subcategories that can further break drown the subdomain into smaller chunks.
2. For each category (major and subcategory), develop a rule or description for the content of the category and the terms that will go, or not go, into this category.
3. For each category, identify a subject matter expert (SME) to be the steward. The steward reviews and approves the structure and descriptions or proposes changes as required.

You can perform the initial work on a whiteboard, cards, a spreadsheet, or any other manner that supports a collaborative thought process.

After the initial design is complete, the taxonomy is presented and validated by business users. Validation involves reviewing the categories and their descriptions. Elicit critiques and suggestions for changes. Suggestions for a new category must be accompanied by proper changes or additions to the descriptions of the affected categories.

Using external third-party taxonomies

Third-party taxonomies are available for almost every domain of knowledge or field of human endeavor. Use of an external third-party taxonomy can have considerable advantages. Most are exhaustive and include thousands of terms with definitions that are constructed by professionals in the field.

However, use of a third-party taxonomy can have drawbacks:

- It might be too detailed for the current application.
- Many terms might be foreign to the users in the company.
- Some terms might have a different use in the company than what is indicated by the definition in the external taxonomy,
- Terms might be organized in a taxonomy that does not reflect the organization or processes of the company.

External taxonomies can be a good source for terms to start your own, but it comes with a cost. In addition to the cost of acquiring the taxonomy is the cost of adjusting and customizing it to your own needs.

6.3.3 Controlled vocabulary

The business glossary provides a flexible platform to capture information about different concepts and objects. You can build hierarchies of objects and capture their definition and usage as you find necessary. However, the primary purpose of the tool is to capture a vocabulary of terms. These terms help users to locate and retrieve information about data assets they are interested in. To support this ability and avoid ambiguities and misinterpretation, you must implement a level of control on the vocabulary creation process, hence a *controlled vocabulary*.

A controlled vocabulary is defined by the US Geological Survey as “A consistent collection of terms chosen for specific purposes with explicitly stated, logical constraints on their intended meanings and relationships.”[1]

Other definitions of controlled vocabulary emphasize the control element of creating a business vocabulary. The business terms must be predefined and agreed upon by the designers and stakeholders.

The emphasis in these definitions is consistency, design, and authorization. An introduction of terms into the vocabulary and classifying them into categories

[1] US Geological Survey: http://geo-nsdi.er.usgs.gov/talk/thesaurus/definition.html

must be defined through a disciplined framework that includes organization, process and procedure, standards, and guidelines.

Failure to implement vocabulary using this approach can have the following results:

- Multiple terms or their variants are used to represent a single concept.
- Technical assets are inconsistently mapped to business terms.
- Not knowing which term to use to uncover sought after information is a significant barrier to discovering assets.
- Some data and knowledge assets might never be reached through search or browsing, rendering them obsolete.
- Lost knowledge means lost productivity, which can also mean lost opportunity.

These pitfalls can jeopardize the adoption of the glossary and ultimately lead to diminished success of the metadata initiative.

6.3.4 Term specification process and guidelines

Developing terms includes selecting and defining the terms, defining the scope and definition of the terms, and eliminating any ambiguity.

Selecting and defining the terms

You must address the following issues when approaching a vocabulary construction task:

- Define the domain to which the vocabulary will be applied.
- Identify the source and authority of term: industry common terms, organizational formal terms, or common use terms.
- Specify the granularity of the term.
- Discover relationships with other related vocabularies.

Defining the scope and definition of the terms

The scope of terms is restricted to selected meanings within the domain of the vocabulary:

- Each term must be formulated so that it conveys the intended scope to any user of the vocabulary.
- Avoid as much as possible terms whose meanings overlap in general usage and homographs (words with identical spellings but different meanings) in the selection of terms.
- The use of homographs as terms in a vocabulary sometimes requires clarification of their meaning through a qualifier.

Eliminating ambiguity

Ambiguity occurs when a term has more than one meaning (a homograph or polysemy). Control vocabulary must compensate for confusion that might be caused by ambiguity. A term must have only one meaning in context.

If you need to maintain multiple meanings, add a qualifier as shown in the following examples:

- Example 1

  ```
  Account
  Customer Account
  General Ledger Account
  ```

- Example 2:

  ```
  Exchange Rate
  Buy Exchange Rate
  Sell Exchange Rate
  ```

Qualifiers must be standardized within a given vocabulary to the extent that is possible.

Principles for term expression

Each term in the vocabulary must represent a single concept or a unit of thought:

- The grammatical form of a term must be a noun or noun phrase.
- A concept might be represented by a single noun or by a multinoun phrase.
- Each term must be formulated in such a way that it conveys the intended scope to any user of the vocabulary.

Guidelines for term definitions

To guarantee uniformity and avoid ambiguity, adopt the following additional guidelines for term definitions:

- They must be stated in the singular.
- They must state what a concept *is* and not what it is not.
- They must be stated in a descriptive phrase.
- They must contain only commonly understood abbreviations.
- They must be expressed without embedding definitions of other underlying concepts.

6.3.5 Using external glossary sources

Often, the glossary team and management resort to the use of a logical data model as the source of a vocabulary to populate the business glossary. This approach is acceptable because the data modeler and analysts have already searched the domain. They have identified many terms and concepts that are needed to capture information in the underlying business domain. With data modeling tools, users can capture logical names and descriptions that can then be used to populate the glossary. Tables and columns are translated to categories and terms in the business glossary.

Using data model as a source for populating the business glossary capitalizes on the knowledge and experience of SMEs, data analysts, and data modelers. These people have studied the domain and identified the elements that are required to represent the domain objects, operations, and processes.

With all these advantages, keep in mind the drawbacks of using a data model-based taxonomy as the business glossary:

- Column labels and definitions created in a data modeling tool might not conform to the standards and guidelines established for the business glossary terms.
- The terms in the model might be too low level, and high-level terms are omitted. Technical people need to deal with the finest point of every object in a domain and capture it in databases. This way they can perform the various operations prescribed to the systems that they built. Business people have a broader view without explicitly addressing the finest details of every concept. The data model usually reflects the IT view and granularity. It might not have concepts that reflect the business view.
- The organization of terms into categories as derived from the table structure in the model might not reflect the business view of the domain.
- Data models contain repetition of columns that are used as foreign keys to enable the joins of tables and navigation of the database. This process translates into multiple identical instances of terms in different categories.

The import of any terms from an external source must be made available to the vocabulary team to review, assess, make corrections, and complete them before they are released for general consumption.

When imported with the workflow flag turned on, external vocabularies have a draft status. In this status, the vocabulary operations team has the opportunity to review, assess, edit, complete, approve, and publish the new categories. The team must also perform these tasks in accordance with the standards, policies, and procedures established for these matters.

6.3.6 The vocabulary authoring process

Chapter 1, “Information governance and metadata management” on page 3, addressed the organization, process, and policies that must be established and deployed to enable and support the creation and maintenance of the vocabulary. Term creation and approval can be as rigorous or as relaxed as the organizational culture and needs dictate.

Certain organizational environments have strict policies and rules about such matters. They often require levels of review and approvals. Others might place the burden on a single SME or author to pick up a term, define it, and publish it.

The process flow illustrated in Figure 6-1 leans toward the more rigorous approach with clear roles and responsibilities and the inclusion of external evaluators.

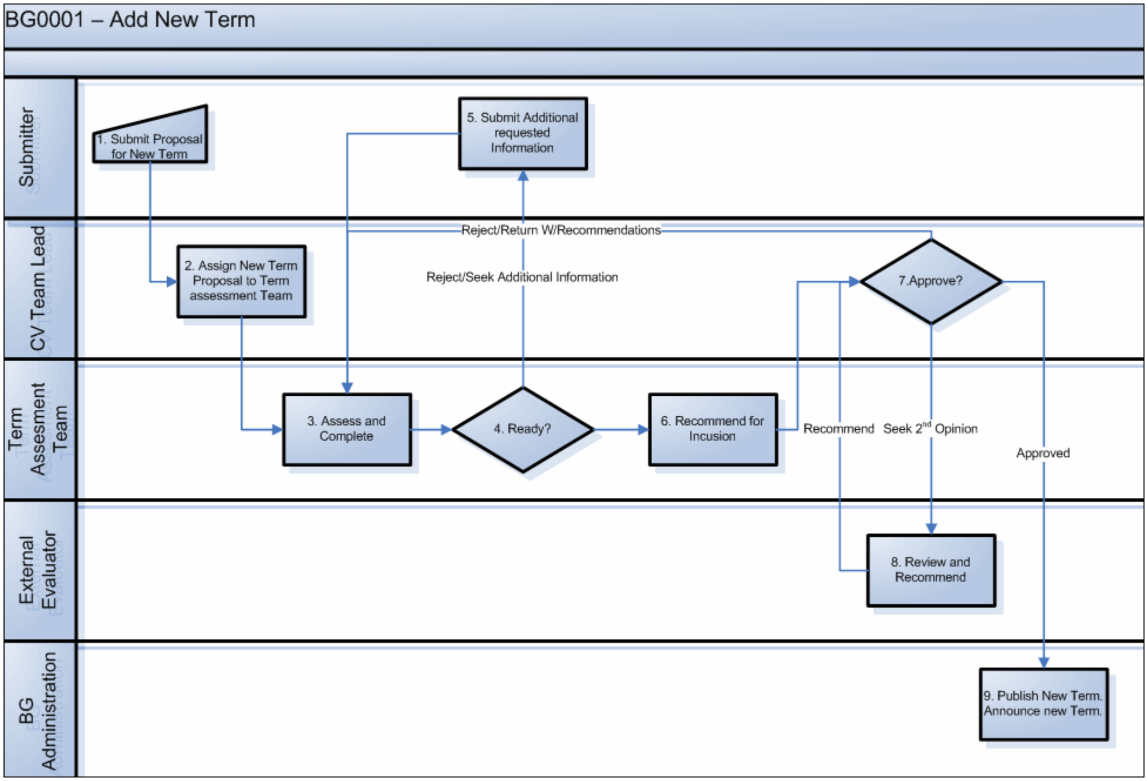

BG0001 – Add New Term
Submitter
CV Team Lead
Term Assesment Team
External Evaluator
BG Administration
1. Submit Proposal for New Term
2. Assign New Term Proposal to Term assessment Team
3. Assess and Complete
4. Ready?
5. Submit Additional requested Information
6. Recommend for Incusion
7.Approve?
8. Review and Recommend
9. Publish New Term. Announce new Term.
Reject/Return W/Recommendations
Reject/Seek Additional Information
Recommend
Seek 2nd Opinion
Approved

Figure 6-1 Process for adding a term

With an emphasis on control and the identification of roles and responsibilities within the vocabulary operations team, the process is tight, ensuring full accountability of the content.

6.4 Deploying a business glossary

The deployment of a business glossary can be a lengthy and arduous job that requires commitment and persistence by business and IT stakeholders. For a business glossary to expand and mature, and to allow the organization to realize the potential benefits early in the process, deploy the business glossary in the following phases. Each additional phase delivers more depth of knowledge and understanding than the previous phase and broader scope. In large organizations, a gradual approach entails prioritizing business domains to be incorporated in the glossary and the metadata repository. The expansion is done vertically by increasing the depth of knowledge through added relationships and documentation. It is done horizontally by adding business domains to the glossary and repository.

- Phase 1: Create a common vocabulary, definition, and shared concepts for a selected business subdomain.

  This phase has the following business benefits:

  - Improved communication between teams.
  - Helps to identify opportunities for common processes and capabilities by clarifying and correlating the details of local terminology across the organization.

- Phase 2: Use the glossary and assigned information assets to promote project efficiency.

  This phase has the business benefit of better documentation of tasks and processes.

- Phase 3: Promote comprehensive use of end-to-end understanding of information flow and processing for key systems.

This process repeats itself iteratively, expanding the circle of users by capturing the process and content of new areas of operations and expanding the opportunities for collaboration and sharing. This process is like a wave ripple phenomena: With a growing circle of users and utility, it enables a growing depth and integration of assets, processes, and resources.

6.4.1 InfoSphere Business Glossary environment

InfoSphere Information Server 8.7 offers a new InfoSphere Business Glossary environment that combines a glossary user view with author and administrator views in a single console.

Workflow management, which is a new feature to this release, separates published and development glossary. It enables enforcement of control over the authoring and approval process of a term. Turning on the workflow feature (a recommended practice) results in the creation of a development environment for the glossary team. You can introduce new terms or update existing terms without interfering with the published glossary (available to all users to browse and search).

The Development Glossary tab

The **Development Glossary** tab (Figure 6-2) provides a view into the development environment with the authoring and publishing functions.

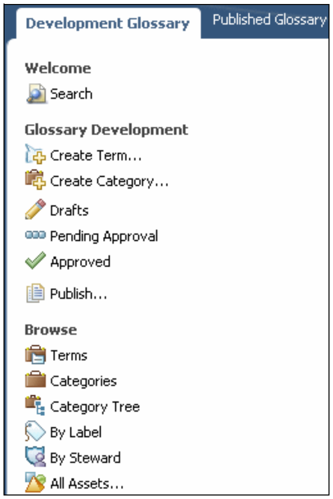


Figure 6-2 Development Glossary tab

In the **Development Glossary** tab, you can search, browse, and create new terms. You can also create, review, and approve terms. This tab is available to users with the *Author* or *Administrator* role. The browse section of the menu provides access to the published glossary and all other assets. A term or a category that was previously published can be returned to the editing desk for an update, if needed, without interfering with the published glossary and its users.

The Published Glossary tab

On the **Publish Glossary** tab (Figure 6-3 on page 144), users can view and search for the published terms and categories. A new glossary user type is added, *Basic User*. This type is unlike the original *User* role, where users can see terms and everything related or assigned to them, including data lineage. *Basic User* can only view and search the terms and their attributes that are published.

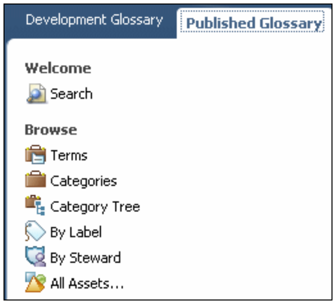
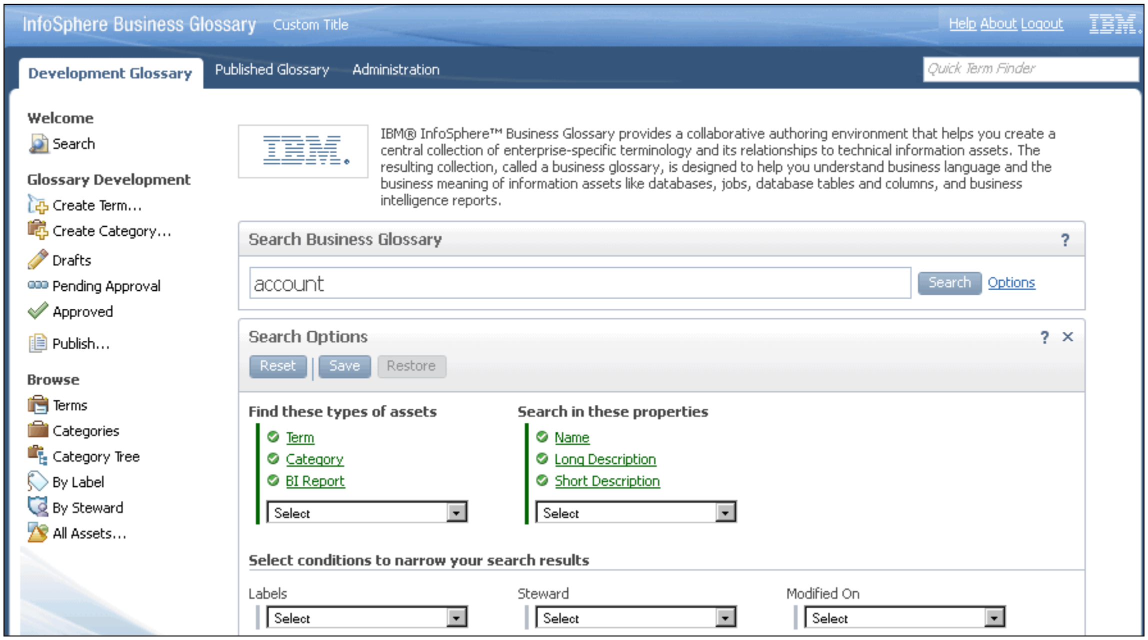


Figure 6-3 Published Glossary tab

The published glossary menu (Figure 6-4) provides a means to search and browse the glossary and the metadata repository as a whole. A search can be done on any asset or combination of assets and can be applied on any attributes of the selected assets. You can search by the type of assets, such as terms, categories, and business intelligence (BI) reports. You can also search by the properties of these assets, such as Name, Short Description, and Long Description.



IBM® InfoSphere™ Business Glossary provides a collaborative authoring environment that helps you create a central collection of enterprise-specific terminology and its relationships to technical information assets. The resulting collection, called a business glossary, is designed to help you understand business language and the business meaning of information assets like databases, jobs, database tables and columns, and business intelligence reports.

account

Figure 6-4 Search InfoSphere Business Glossary options

Browsing works in a different way than searching. You can open and browse an alphabetical list of terms or perform a quick filter to narrow the list to terms starting with a character or string that you provide. You can also open an alphabetical list of categories with similar browsing and filtering capabilities. Users might not know what they are looking for or want to explore the structure of the glossary. These users can open the category tree (Figure 6-5) and view the entire glossary. By using the category tree structure, users can drill down to a particular category and explore its content.

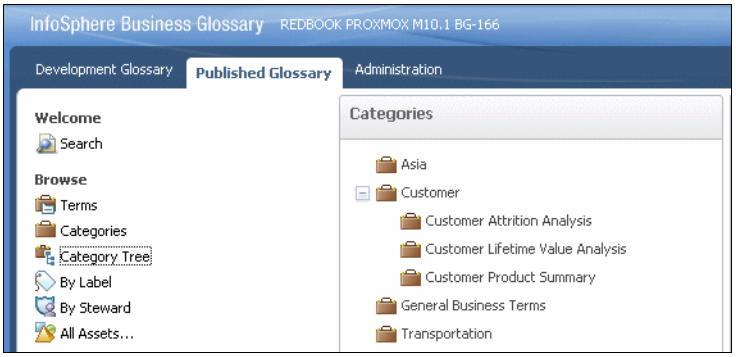


Figure 6-5 InfoSphere Business Glossary category tree

Another way to explore or browse a published glossary is to use labels that are assigned to particular terms. Alternatively, you can browse the terms by the stewards who are in charge of the terms.

The Administration tab

The **Administration** tab (Figure 6-6 on page 146) offers a menu of administrative tasks such as configuration, workflow control, import, and export.

An administrator can assign permissions to users that have Author access to edit or publish terms managed by the workflow process. Permissions are granted to individuals to work on terms in a particular category or subcategory. This way, the user can departmentalize work and limit the view that authors have only to terms in the categories for which they have permission.

Similarly, an administrator can set permission for users or user groups to view, browse, and search the published glossary, limiting their access only to certain categories.

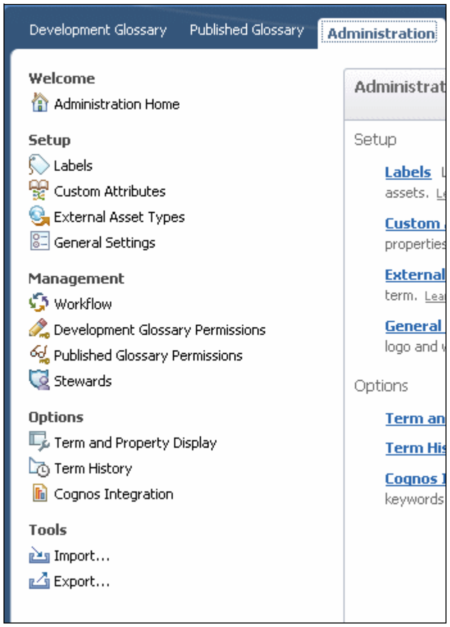
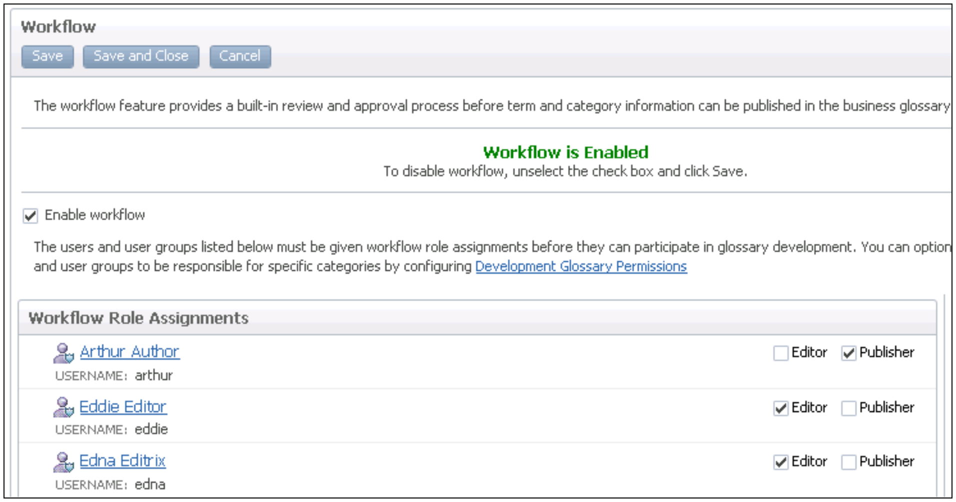


Figure 6-6 Administration tab

6.5 Managing the term authoring process with a workflow

The workflow management feature in InfoSphere Business Glossary supports a term creation and defining process with the assignment of roles and responsibilities. When the workflow feature is turned on for every new term, whether entered manually or imported from external sources using comma-separated value (CSV) or Extensible Markup Language (XML) files, it is called a *Draft* term. Draft terms are displayed on the **Development** tab. They must go through an approval and publishing process before they are available for consumption by general users.

Workflow is turned on by an administrator from the Workflow window (Figure 6-7).


Workflow

The workflow feature provides a built-in review and approval process before term and category information can be published in the business glossary

Workflow is Enabled

To disable workflow, unselect the check box and click Save.

☑ Enable workflow

The users and user groups listed below must be given workflow role assignments before they can participate in glossary development. You can option and user groups to be responsible for specific categories by configuring Development Glossary Permissions

Workflow Role Assignments

Arthur Author USERNAME: arthur	□ Editor	☑ Publisher
Eddie Editor USERNAME: eddie	☑ Editor	□ Publisher
Edna Editrix USERNAME: edna	☑ Editor	□ Publisher

Figure 6-7 Workflow setup window

An administrator can assign users with the Author role to the Editor or Publisher role. When workflow is enabled, users with the Editor role can change glossary content, and users with the Publisher role can approve and publish glossary content.

Publishing a glossary: Publishing a glossary can be done only after all terms and categories that are the responsibility of a particular publisher are approved for publishing.

The right to publish must be granted internally to a *super publisher*, which is a user with authority over the entire section that can coordinate the completion of work on open terms and categories.

The process flow in Figure 6-8 indicates a simplified, less rigorous, term authoring process. It is fully supported by the workflow functionality and involves individuals with InfoSphere Business Glossary access.

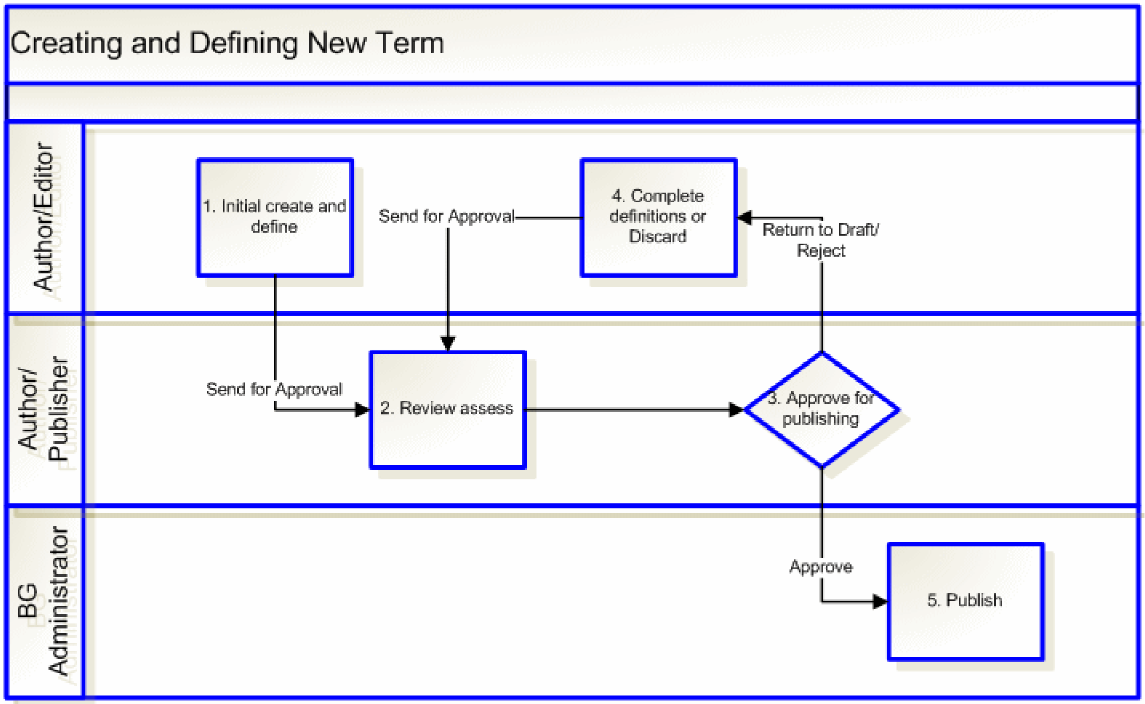


Figure 6-8 Process for creating and defining a term

To create and define a terms, follow these steps:

1. Create and define.

   An editor picks up a term from the terms listed in the Draft folder. The editor opens the term in edit mode by clicking the **Edit** button and makes all the necessary changes and additions. The editor completes this task and submits the edited term for approval by clicking the **Send for Approval** button. The term moves from the Draft folder to the Pending Approval folder.

2. Review and assess.

   A publisher picks up the term in the Pending Approval folder. The publisher opens a term and reviews the entry to determine that it is correct and conforms to the standards and guidelines.

3. Approve for publishing.

   If all aspects of the term definition meet the norms and standards of the organization, the publisher approves the term by clicking the **Approve** button.

At that time, the term is moved to the Approved folder. The publisher can decide that the definitions are not correct, not complete, or unsatisfactory for some other reason, denying the approval. Then the publisher can send the term back to the editor by clicking the **Reject** button and can add comments explaining the reasons for the rejections or recommendations for resubmittal.

4. Complete definition or discard.

   Rejected terms show up again in the Draft folder. The editor opens the term and reads the comments from the publisher. The editor completes the required attributes or change definitions suggested by the publisher and resends the term for approval with or without notes to the publisher. Alternatively, the editor can delete the term based on the recommendation of the publisher.

5. Publish.

   Approved terms waiting for publishing are in the Approved folder. Publishing is done by navigating to the Publish window and clicking the **Publish** button, in which case terms in the Approved folder are published. A publisher can only publish the terms and categories that are approved.

   In Figure 6-8 on page 148, we designate the publisher role to an administrator, who is a super user and has publish authority, to illustrate that this function must be coordinated and synchronized possibly with a new version.

6.5.1 Loading and populating the glossary

InfoSphere Business Glossary can be populated from various sources. In regard to the use case where Bank A acquires Bank B, a new combined glossary must be adjusted to accommodate business and IT workers in both banks.

Bank B, the acquired bank, has different processes and procedures to perform bank tasks than Bank A. Bank B also uses different set of terminology and vocabulary to communicate colloquially about their business. Bank B employees must acquire the vocabulary of Bank A and be able communicate and with their colleagues about the business they share. Both terminologies must be joined, consolidated, and reconciled so that employees of both banks can communicate with as few misunderstandings as possible.

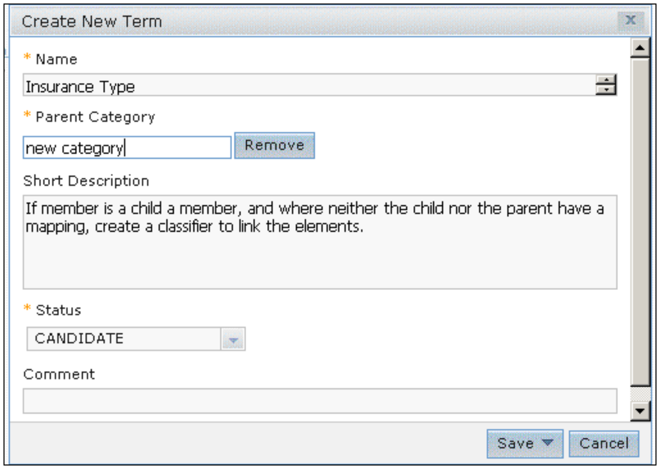
To load and populate a glossary, the following tasks might be required:

1. Load or enter the new bank vocabulary.
   a. Perform a manual data entry, or import the data from an external source.
   b. Adjust the definition to conform to the bank conventions and standards.
2. Adjust the taxonomy to accommodate the new terminology.

3. Establish new-to-old terminology relationships.
4. Link new terms to assets.

6.5.2 Creating and editing a term

In terms of the use case, the business glossary team wants to add categories and terms from Bank B to the glossary manually. In this case, an author with the Editor role needs to create a term in the glossary from the **Development Glossary** tab. Initial creation of a term requires minimal information. As shown in the Create New Term window (Figure 6-9), you must include the name of the term, a short description, and a parent category.



* Name
Insurance Type

* Parent Category
new category

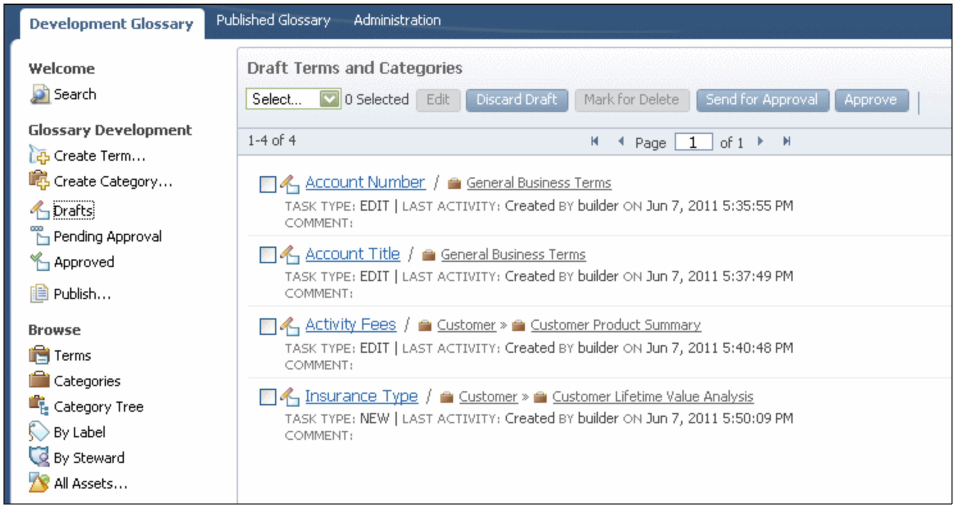
Short Description
If member is a child a member, and where neither the child nor the parent have a mapping, create a classifier to link the elements.

* Status
CANDIDATE

Comment

Figure 6-9 Create New Term window

After you save the initial term definition, the term shows up in the Draft folder, as shown in Figure 6-10. In this folder, a person designated as an editor can open the term for further editing.



Draft Terms and Categories

□ Account Number / General Business Terms
TASK TYPE: EDIT | LAST ACTIVITY: Created BY builder ON Jun 7, 2011 5:35:55 PM
COMMENT:

□ Account Title / General Business Terms
TASK TYPE: EDIT | LAST ACTIVITY: Created BY builder ON Jun 7, 2011 5:37:49 PM
COMMENT:

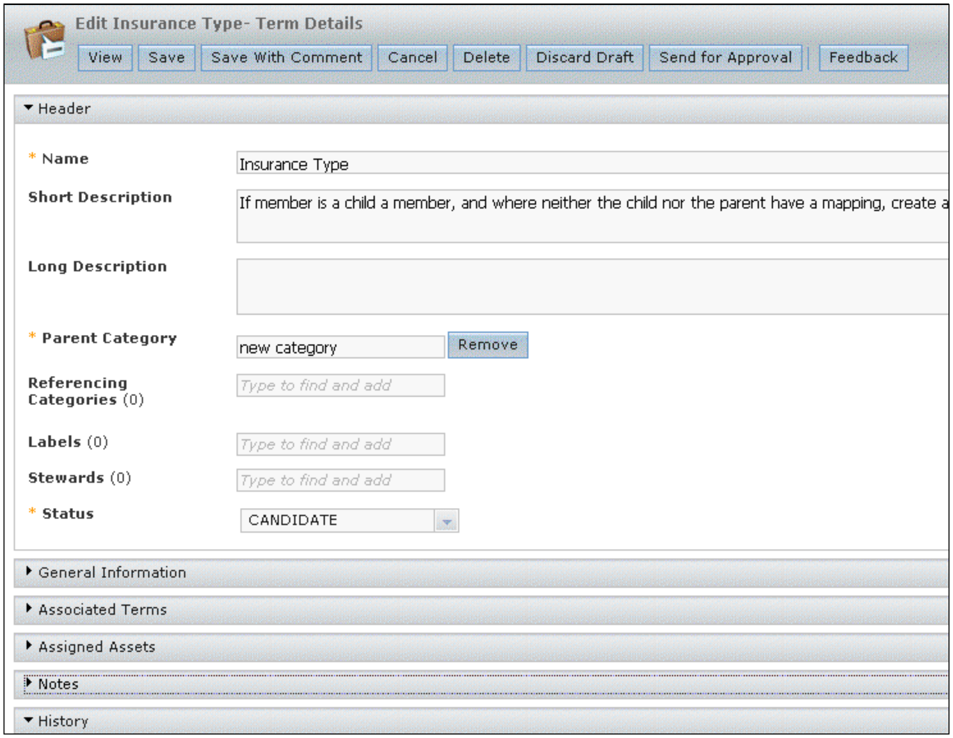
□ Activity Fees / Customer » Customer Product Summary
TASK TYPE: EDIT | LAST ACTIVITY: Created BY builder ON Jun 7, 2011 5:40:48 PM
COMMENT:

□ Insurance Type / Customer » Customer Lifetime Value Analysis
TASK TYPE: NEW | LAST ACTIVITY: Created BY builder ON Jun 7, 2011 5:50:09 PM
COMMENT:

Figure 6-10 Draft folder

Opening a term for editing provides access to all of the term attributes. The editor can go to any of the panes to add information or establish relationships to other terms and assets.

In the Header pane (Figure 6-11), you have access to the basic set of attributes including short and long descriptions, parent category, referencing terms, labels, stewards, and status.



* Name: Insurance Type

Short Description: If member is a child a member, and where neither the child nor the parent have a mapping, create a

Long Description:

* Parent Category: new category

Referencing Categories (0): Type to find and add

Labels (0): Type to find and add

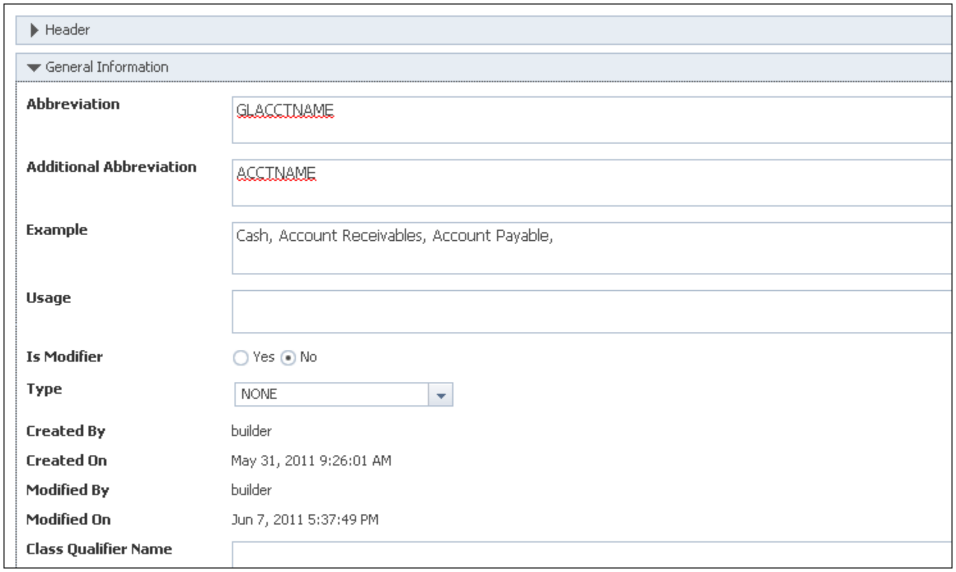
Stewards (0): Type to find and add

* Status: CANDIDATE

Figure 6-11 Header pane of the term edit window

The General pane (Figure 6-12) contains most of the remaining attributes including abbreviations, examples, usage, and custom attributes. It also contains audit information such as when the term was created and last modified and by whom.

These attributes can be useful for newcomers such as Bank B employees who are now bound to use the process, procedures, and codes of Bank A. Examples of the codes to be used and an explanation of how they can be used provide a confidence and productivity boost to new users.



▸ Header

▾ General Information

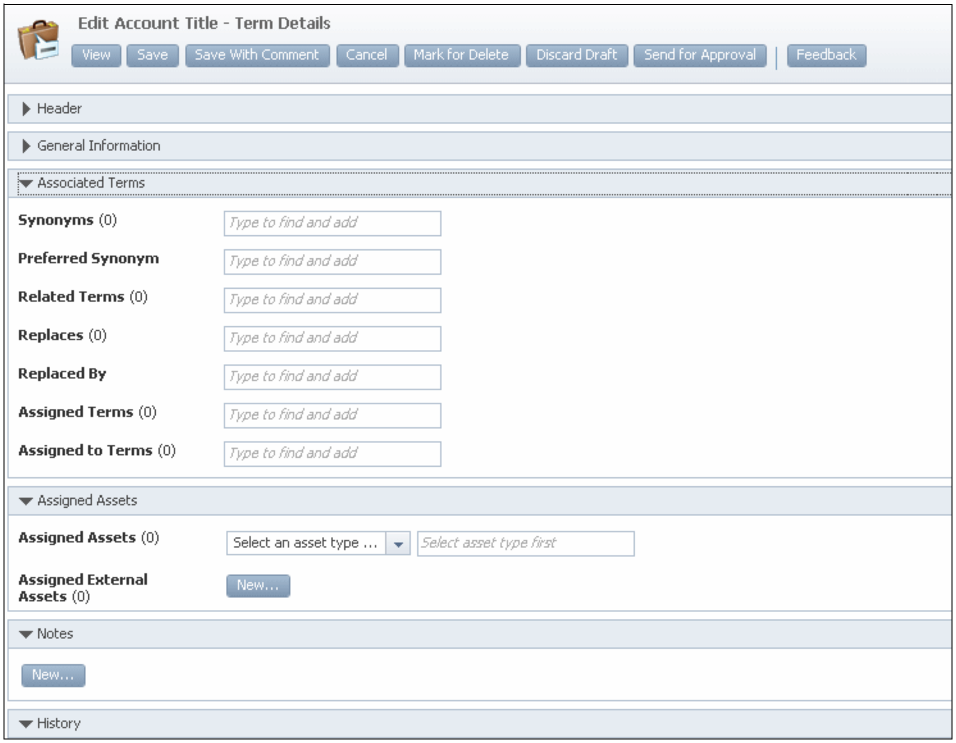
Abbreviation	GLACCTNAME
Additional Abbreviation	ACCTNAME
Example	Cash, Account Receivables, Account Payable,
Usage	
Is Modifier	○ Yes ● No
Type	NONE
Created By	builder
Created On	May 31, 2011 9:26:01 AM
Modified By	builder
Modified On	Jun 7, 2011 5:37:49 PM
Class Qualifier Name	

Figure 6-12 General pane of the Term edit window

6.5.3 Adding term relations and assigning assets

By using additional panes, you can specify synonyms, associate a term with other terms, or assign assets (Figure 6-13 on page 154). Synonyms are helpful to bridge between two vocabularies. For some time, Bank B employees will continue to use the old vocabulary. Linking old (Bank B) terms to equivalent new (Bank A) terms with all related information in them makes the transition to the new vocabulary much easier. Another option is to introduce the old terms in Bank B as deprecated terms replaced by the new terms from Bank A.

Associating terms and assigning assets are essential to building a body of knowledge as compared to a mere glossary of terms. Establishing relationships among terms and with assets enriches the understanding of the terms and provides a broader context to the terms. In general, a category provides context to the terms it contains. It groups terms that belong together and that overall represent the same bigger concept. Related and associated terms go outside of the category to expand the context and provide access to additional information.



Edit Account Title - Term Details

▶ Header

▶ General Information

▼ Associated Terms

Synonyms (0)	Type to find and add
Preferred Synonym	Type to find and add
Related Terms (0)	Type to find and add
Replaces (0)	Type to find and add
Replaced By	Type to find and add
Assigned Terms (0)	Type to find and add
Assigned to Terms (0)	Type to find and add

▼ Assigned Assets

Assigned Assets (0): Select an asset type ... | Select asset type first

Assigned External Assets (0)

▼ Notes

▼ History

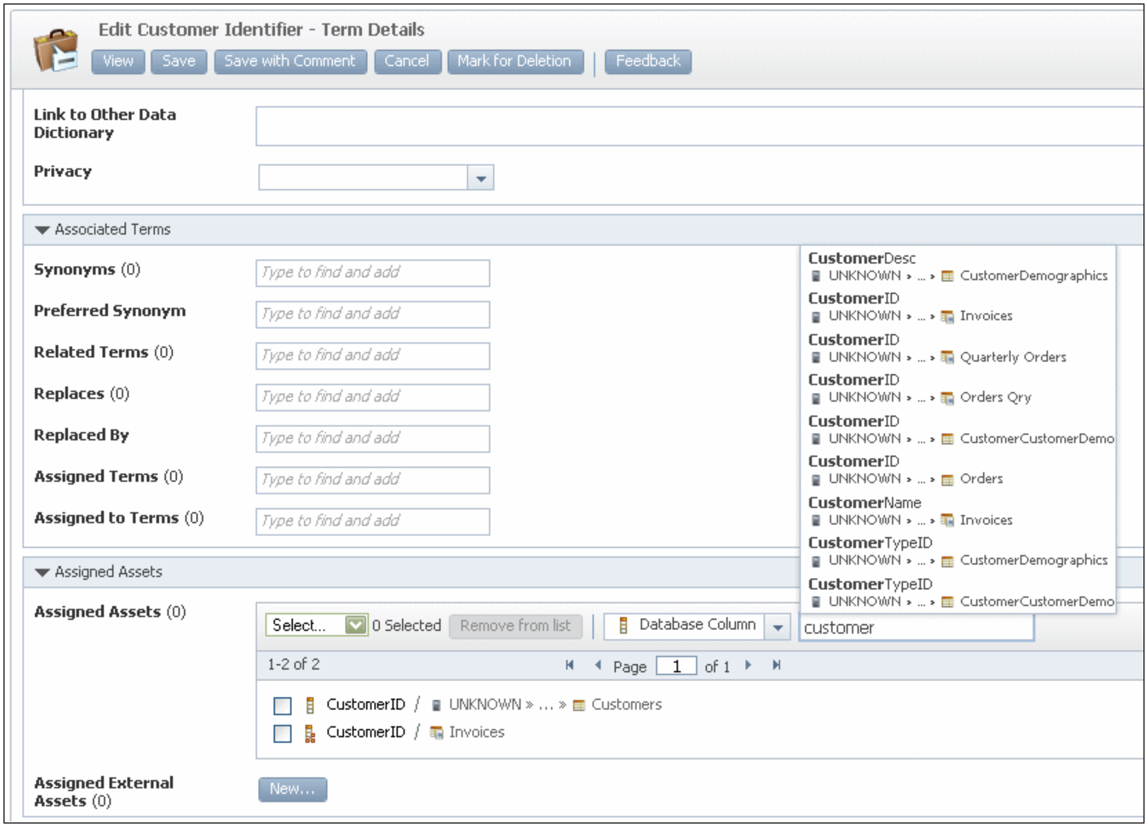
Figure 6-13 Associate Terms and Assign Assets panes

For example, in Figure 6-13, Account Title is grouped with other terms that describe an account. An account has a number, a title, a balance, and so on. However, if you want to learn more about the term Account Title, other categories might contain terms that are related to the Generally Accepted Accounting Principles (GAAP) standards that places terms in context of regulation and reporting principles. These terms might reside in a separate category of standards or regulations, different from the one that contains terms about accounts managed by the company.

By using the glossary, you can express and track the evolution of terms. Over time, new terms go into usage, replacing existing terms that are deprecated. If you are still using the old, deprecated term, the Replace By field indicates the new term to use and the information associated with it.

Assigned terms is another type of association that you can use to help expand knowledge while maintaining the consistency and integrity of the glossary. For example, report headers are created by the report developer or designer, and they are intended to reflect what is in the report column or cell. Reports are often a starting point for a glossary search that a user initiates. However, these headers do not necessarily comply with the term naming standards. A report header can still be placed in the glossary as a term, in a category dedicated for that term. Yet, it can be assigned to the real term that represents the concept and is properly formed and defined. This way, you maintain consistency and compliance in the glossary with the ability to accommodate nonstandard expressions.

Asset assignment is done in a similar fashion. You select the type of asset you want to assign from a list of assets. Then you provide a letter or a character string to search for the asset by name. A list of assets with matching names is displayed, as shown in Figure 6-14.



Edit Customer Identifier - Term Details

Link to Other Data Dictionary

Privacy

Associated Terms

Synonyms (0) Type to find and add

Preferred Synonym Type to find and add

Related Terms (0) Type to find and add

Replaces (0) Type to find and add

Replaced By Type to find and add

Assigned Terms (0) Type to find and add

Assigned to Terms (0) Type to find and add

CustomerDesc
UNKNOWN › ... › CustomerDemographics
CustomerID
UNKNOWN › ... › Invoices
CustomerID
UNKNOWN › ... › Quarterly Orders
CustomerID
UNKNOWN › ... › Orders Qry
CustomerID
UNKNOWN › ... › CustomerCustomerDemo
CustomerID
UNKNOWN › ... › Orders
CustomerName
UNKNOWN › ... › Invoices
CustomerTypeID
UNKNOWN › ... › CustomerDemographics
CustomerTypeID
UNKNOWN › ... › CustomerCustomerDemo

Assigned Assets

Assigned Assets (0) Select... 0 Selected Remove from list Database Column customer

1-2 of 2 Page 1 of 1

□ CustomerID / UNKNOWN » ... » Customers
□ CustomerID / Invoices

Assigned External Assets (0)

Figure 6-14 Asset assignment

When you select an asset from the list, it is added to the term assigned assets. Multiple assets of different types can be assigned to a term.

Assigned assets provide links to different data assets: database columns, file fields, BI reports, or anything else that might realize the concept. By assigning assets to a term, users can more easily explore and answer questions such as the following examples:

- What is the asset that realizes the concept?
- What is the source of the information?
- When was the last time it was updated?
- In what other reports does this value also appear?

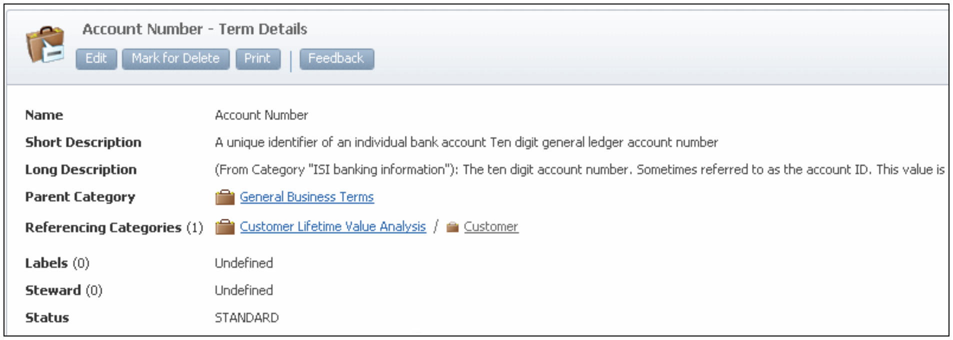
6.5.4 Reference by category

Every term must have a parent category, and every term must have only one parent category. If the same term appears in more than one category, it must have different meanings in the context of the different categories.

Having the same term with the same meaning in two separate categories violates the consistency directive. Each instance of the term might have slightly different content or be assigned to different assets, potentially providing users with conflicting answers to a query or a search.

Occasionally, you might want or need to have the same term in various separate categories. You can achieve this goal by establishing a reference link from any category to the term instance in its parent category. This relationship is called *Reference by Category*. The term has a single instance that is maintained in one place only, while showing up in various separate categories as *referenced term*. Checking the term from anywhere shows the same content and relationships.

The term with a referencing category is displayed in the header pane of the referencing category, as shown in Figure 6-15.

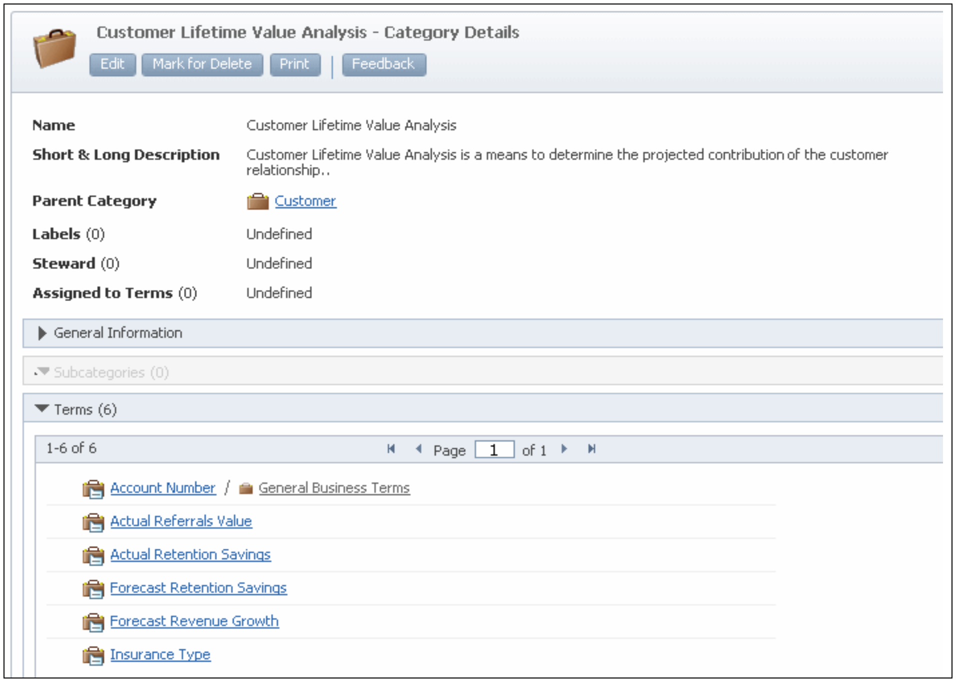


Account Number - Term Details

Name	Account Number
Short Description	A unique identifier of an individual bank account Ten digit general ledger account number
Long Description	(From Category "ISI banking information"): The ten digit account number. Sometimes referred to as the account ID. This value is
Parent Category	General Business Terms
Referencing Categories (1)	Customer Lifetime Value Analysis / Customer
Labels (0)	Undefined
Steward (0)	Undefined
Status	STANDARD

Figure 6-15 Term with a referencing category

At the same time, the referencing category (Customer Life Time Analysis in this case), in its list of terms, shows referenced terms with an indication of the parent category. As shown in Figure 6-16, Account Number is displayed with its parent category General Business Terms.



Customer Lifetime Value Analysis - Category Details

Name	Customer Lifetime Value Analysis
Short & Long Description	Customer Lifetime Value Analysis is a means to determine the projected contribution of the customer relationship..
Parent Category	Customer
Labels (0)	Undefined
Steward (0)	Undefined
Assigned to Terms (0)	Undefined

- Account Number / General Business Terms
- Actual Referrals Value
- Actual Retention Savings
- Forecast Retention Savings
- Forecast Revenue Growth
- Insurance Type

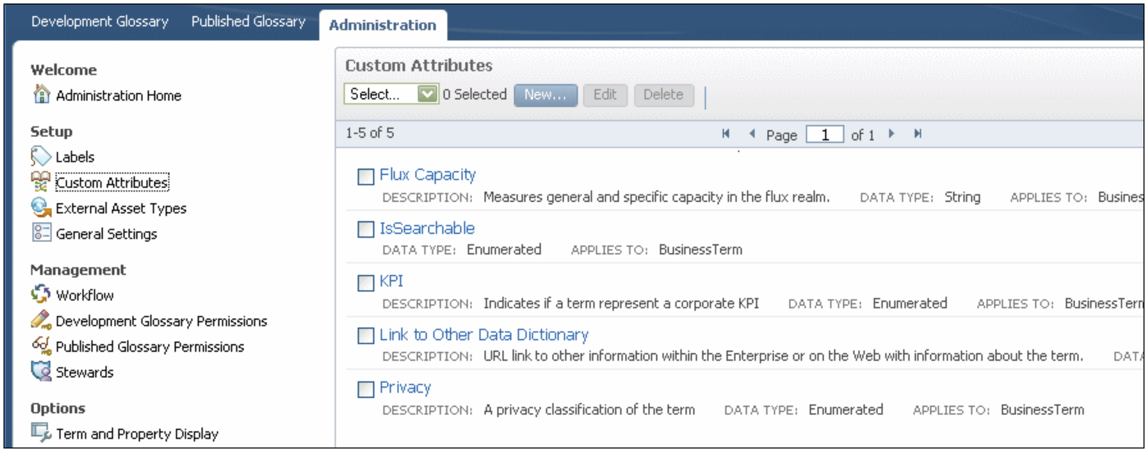
Figure 6-16 Category with a referenced term

With this feature, the glossary designer has the flexibility to express different points of view. Often, constituencies in an organization have different and competing points of view. All of them want to see terms organized, grouped, or structured to accommodate their view. Whether it is business versus technical or operations versus accounting, using the reference by category feature can help maintain the integrity and consistency of the glossary while accommodating multiple communities.

This feature can further help bridge two glossaries with two different structures. Consider that Bank B has the same set of terms but they are organized in a different category hierarchy. Bank B employees can be accommodated with a hierarchy structure they are used to by taking advantage of this feature. By using Reference by Category, each bank employee can access and browse terms in the structure they are used to. Yet the InfoSphere Business Glossary administrator maintains only one instance of the terms.

6.5.5 Custom attributes

By using custom attributes, you can expand the content of a term definition with additional classifiers, relationships, or any other content that you want to associate with a term or category. They are defined on the **Administration** tab (Figure 6-17) and are available for all terms or categories in the glossary.



□ Flux Capacity
DESCRIPTION: Measures general and specific capacity in the flux realm. DATA TYPE: String APPLIES TO: Busines

□ IsSearchable
DATA TYPE: Enumerated APPLIES TO: BusinessTerm

□ KPI
DESCRIPTION: Indicates if a term represent a corporate KPI DATA TYPE: Enumerated APPLIES TO: BusinessTern

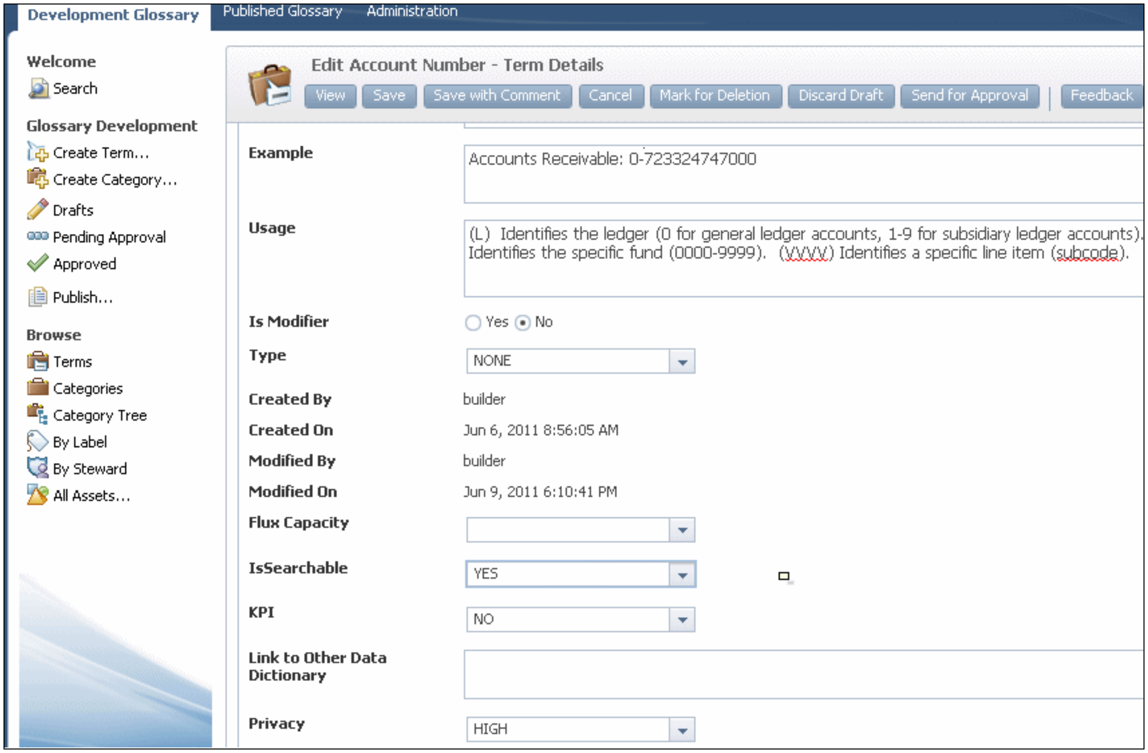
□ Link to Other Data Dictionary
DESCRIPTION: URL link to other information within the Enterprise or on the Web with information about the term. DATA

□ Privacy
DESCRIPTION: A privacy classification of the term DATA TYPE: Enumerated APPLIES TO: BusinessTerm

Figure 6-17 Custom attribute administration page

Custom attributes are enterprise wide. When created in the Administration console, they show up on the edit panels of all terms and categories as being designated by the glossary designers.

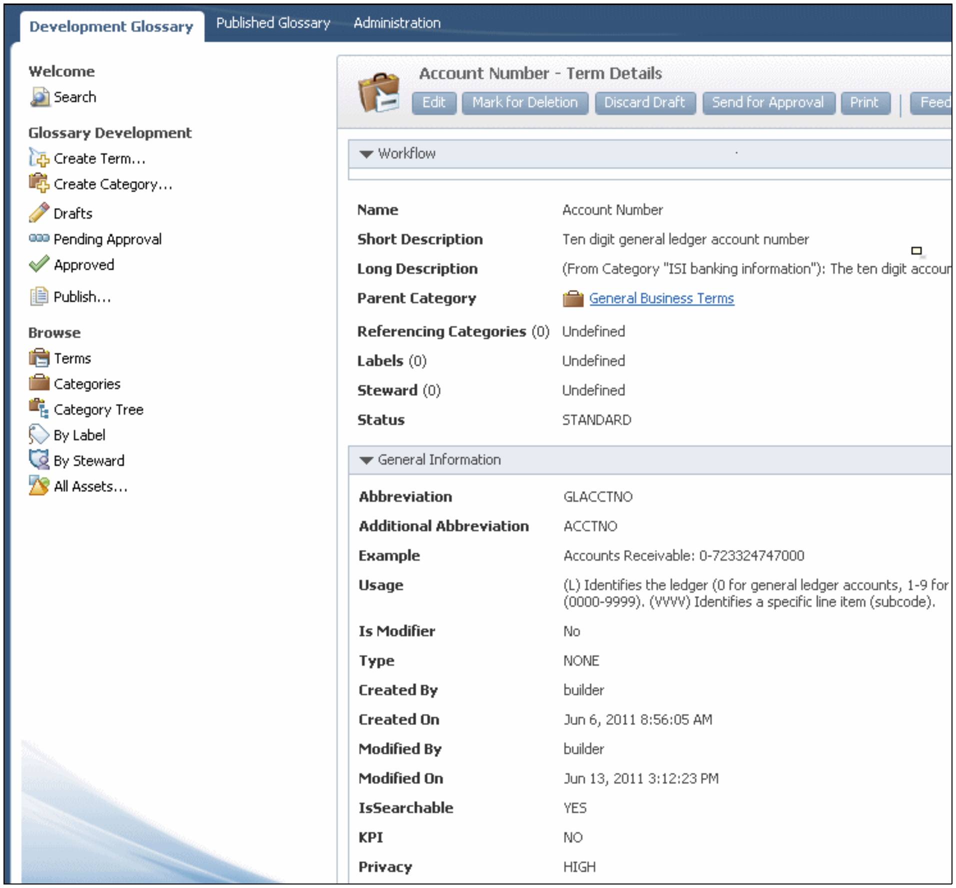
Custom attributes are applied to business terms, and others are applied to categories. When opening an edit panel for a term, in the General Information panel of the term, you see custom attributes in the list. Each custom attribute is displayed with an appropriate edit field. Custom attributes can accept two types of data: enumerated lists or text fields. For an enumerated list, you provide the list of values, which can be short, such as *yes* or *no*, or long, such as state codes or other code lists that might apply (Figure 6-18).



Example	Accounts Receivable: 0-723324747000
Usage	(L) Identifies the ledger (0 for general ledger accounts, 1-9 for subsidiary ledger accounts). Identifies the specific fund (0000-9999). (VVVV) Identifies a specific line item (subcode).
Is Modifier	○ Yes ● No
Type	NONE
Created By	builder
Created On	Jun 6, 2011 8:56:05 AM
Modified By	builder
Modified On	Jun 9, 2011 6:10:41 PM
Flux Capacity	
IsSearchable	YES
KPI	NO
Link to Other Data Dictionary	
Privacy	HIGH

Figure 6-18 Term custom attributes in edit mode

However, when you open a published term for viewing, you see only the custom attributes that were populated for the term. In Figure 6-19. the attributes IsSearchable, KPI, and Privacy shown in the General Information pane are custom attributes, and they are the only ones that are shown.



Name	Account Number
Short Description	Ten digit general ledger account number
Long Description	(From Category "ISI banking information"): The ten digit accour
Parent Category	General Business Terms
Referencing Categories (0)	Undefined
Labels (0)	Undefined
Steward (0)	Undefined
Status	STANDARD

Abbreviation	GLACCTNO
Additional Abbreviation	ACCTNO
Example	Accounts Receivable: 0-723324747000
Usage	(L) Identifies the ledger (0 for general ledger accounts, 1-9 for (0000-9999). (VVVV) Identifies a specific line item (subcode).
Is Modifier	No
Type	NONE
Created By	builder
Created On	Jun 6, 2011 8:56:05 AM
Modified By	builder
Modified On	Jun 13, 2011 3:12:23 PM
IsSearchable	YES
KPI	NO
Privacy	HIGH

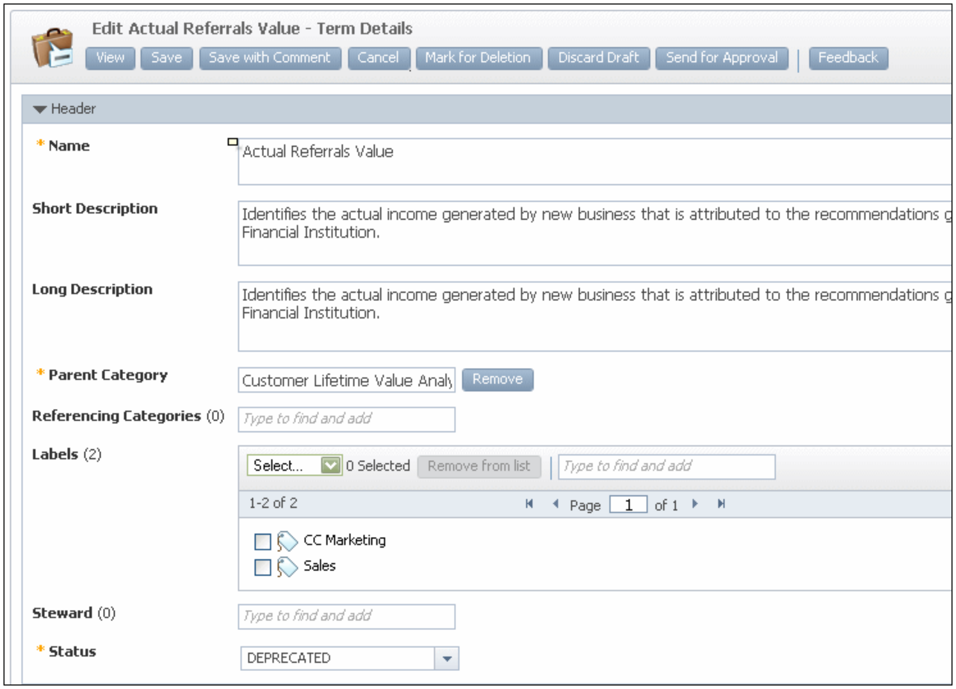
Figure 6-19 Term view with custom attributes

6.5.6 Labels

Labels are another way to help the organization use InfoSphere Business Glossary. Labels are used to create new context to terms and other IT assets. Whether you have a marketing campaign, a project, or a cross-domain initiative, you can create labels to group and sort terms and assets to provide easy access to users in the particular domain.

Labels are created by an administrator from the **Administration** tab. When you create a label and provide a description, you are ready to start tagging terms and other assets with the label.

Assigning a label to a term is done by a user with the Author credential from the header pane of the term editing window. You can tag a term with more than one label, as shown in Figure 6-20. In this example, Actual Referral Value is tagged for two lists, CC Marketing and Sales.



* Name: Actual Referrals Value

Short Description: Identifies the actual income generated by new business that is attributed to the recommendations g Financial Institution.

Long Description: Identifies the actual income generated by new business that is attributed to the recommendations g Financial Institution.

* Parent Category: Customer Lifetime Value Analy

Referencing Categories (0): Type to find and add

Labels (2):

□ CC Marketing

□ Sales

Steward (0): Type to find and add

* Status: DEPRECATED

Figure 6-20 Term tagging with labels

6.5.7 Stewardship

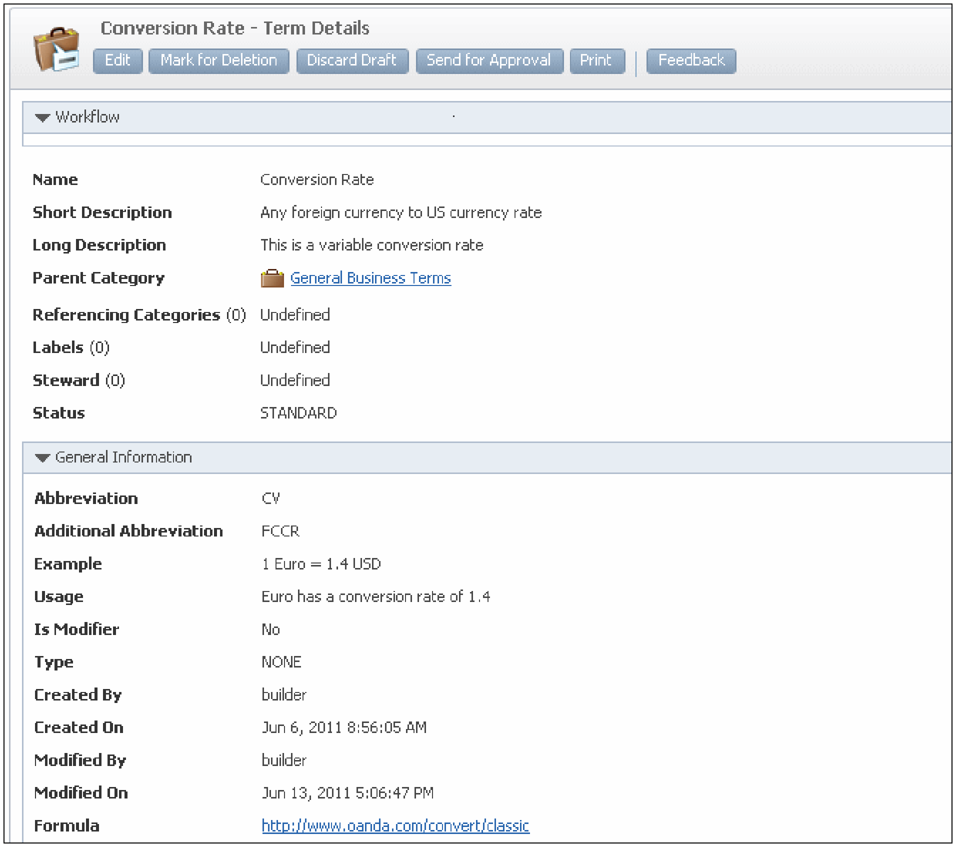
Data stewards are assigned to terms and categories. They are not necessarily SMEs or term authors but rather someone who knows about the object and the business. The name and contact information of the data steward is available with the term information. If users have further questions about the term, its definition, usage, or any other issue, they contact the data steward first. In a way, data stewards own the objects to which they are assigned. They are familiar with the content of the assets assigned to them and how and where they are used.

Stewards are assigned by the administrator from among users with InfoSphere Business Glossary credentials. Stewards are normally granted author credentials to allow them access to terms in the development environment.

6.5.8 URL links

InfoSphere Business Glossary supports live URL links in any of the text fields. These links can be used to further extend the reach of the glossary to additional external resources. You can click the URL to access reference data stored on internal or external locations, documents, policies, operational manuals, and more related to the subject conveyed by the term.

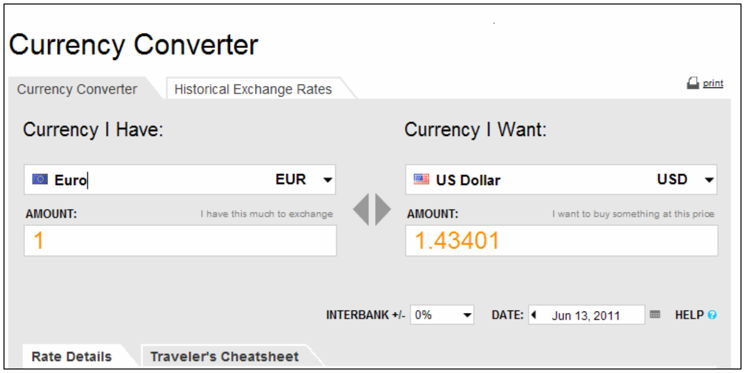
Whether locating a branch address, a ZIP code, or a formula behind the calculation, even links outside of the system are accessible to the InfoSphere Business Glossary users. Figure 6-21 shows a link to a URL that has a conversion engine.



Name	Conversion Rate
Short Description	Any foreign currency to US currency rate
Long Description	This is a variable conversion rate
Parent Category	General Business Terms
Referencing Categories (0)	Undefined
Labels (0)	Undefined
Steward (0)	Undefined
Status	STANDARD
Abbreviation	CV
Additional Abbreviation	FCCR
Example	1 Euro = 1.4 USD
Usage	Euro has a conversion rate of 1.4
Is Modifier	No
Type	NONE
Created By	builder
Created On	Jun 6, 2011 8:56:05 AM
Modified By	builder
Modified On	Jun 13, 2011 5:06:47 PM
Formula	http://www.oanda.com/convert/classic

Figure 6-21 Term attributes with a URL in the Formula field

The link leads to a public page (Figure 6-22) that provides currency conversion calculations between major currencies.



Currency Converter

Currency I Have: Euro EUR

AMOUNT: 1 (I have this much to exchange)

Currency I Want: US Dollar USD

AMOUNT: 1.43401 (I want to buy something at this price)

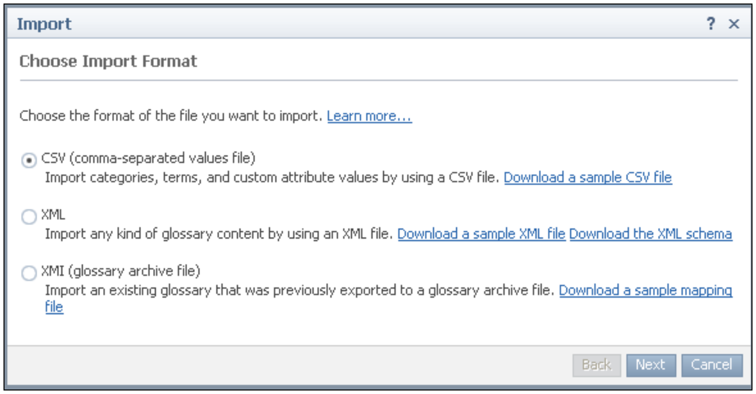
INTERBANK +/-: 0% DATE: Jun 13, 2011 HELP

Figure 6-22 Currency Converter service page

6.5.9 Import glossary

Bank B might already have a vocabulary in some form used across the bank. It might be in a tool for vocabulary management or in a form of a document or a spreadsheet that people maintain on their desk for reference. In this case, several options are possible for importing vocabulary from the external sources.

You can use CSV and XML file formats to create a vocabulary externally and import them into InfoSphere Business Glossary. The import is done from the **Administration** tab. You click the **Import** task in the menu, and then the Import window (Figure 6-23) opens showing the available options.



Import

Choose Import Format

Choose the format of the file you want to import. Learn more...

● CSV (comma-separated values file)
Import categories, terms, and custom attribute values by using a CSV file. Download a sample CSV file

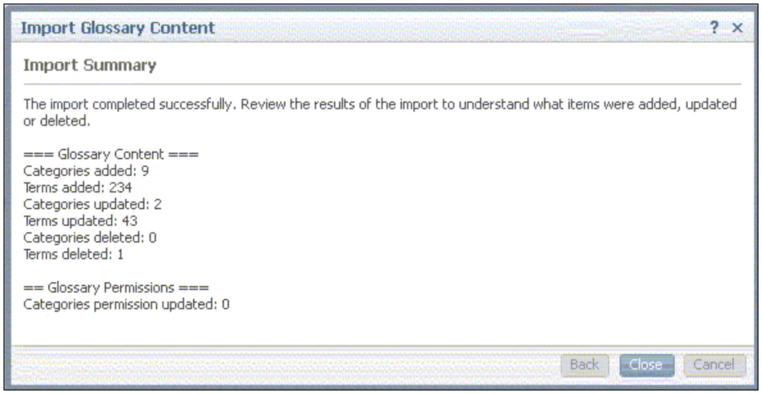
○ XML
Import any kind of glossary content by using an XML file. Download a sample XML file Download the XML schema

○ XMI (glossary archive file)
Import an existing glossary that was previously exported to a glossary archive file. Download a sample mapping file

Figure 6-23 InfoSphere Business Glossary Import window

For both file formats, you are provided with templates and examples that you can build on to create your vocabulary import file. The XML format also offers options on how to consolidate new and old terms.

After the import is completed, the system reports on the results of the import, including the number of categories and terms added to the glossary as shown in Figure 6-24. All terms are imported into the Draft folder.



Import Summary

The import completed successfully. Review the results of the import to understand what items were added, updated or deleted.

```
=== Glossary Content ===
Categories added: 9
Terms added: 234
Categories updated: 2
Terms updated: 43
Categories deleted: 0
Terms deleted: 1

== Glossary Permissions ===
Categories permission updated: 0
```

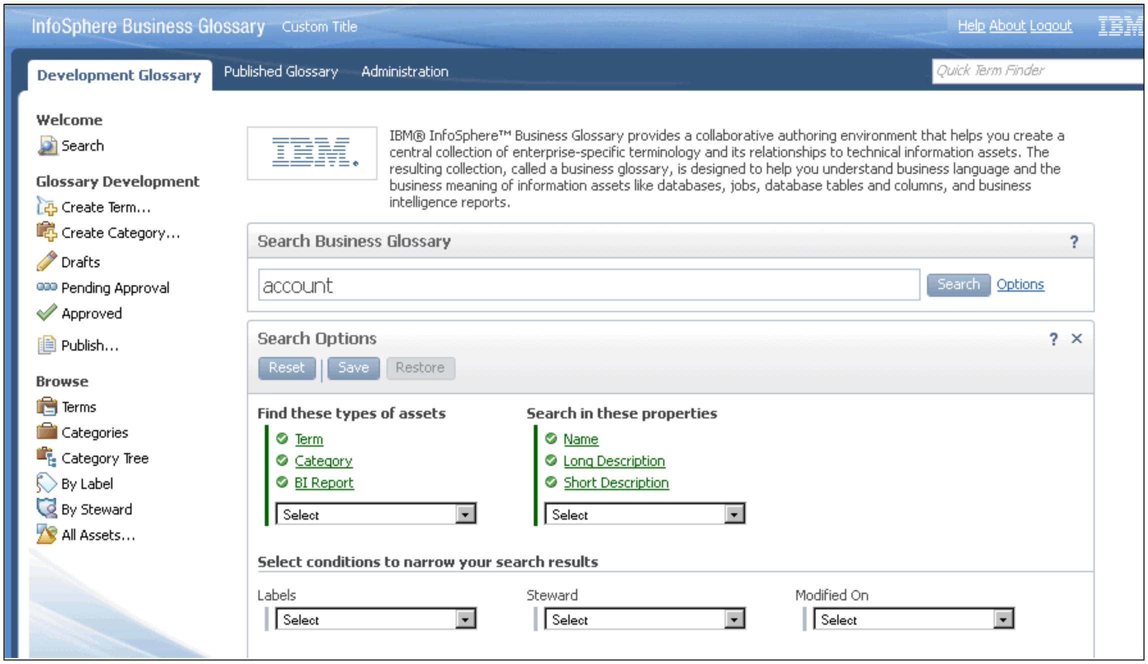
Figure 6-24 Glossary import summary

After the new categories and terms are imported and are displayed in the Draft folder, they are available for editing. Editors can use the options mentioned earlier to fully define the new items and to consolidate and the reconcile glossary elements of the two banks.

6.6 Searching and exploring with InfoSphere Business Glossary

The published glossary menu provides the means to search and browse the glossary and all other assets in the metadata repository. The search can be done on any type of asset or combination of assets and can be applied on any of the attributes of the selected assets.

As shown in Figure 6-25, the search command can search for terms, categories, and BI reports that have the string `'Account'` in the attributes, name, short description, or long description of the assets.



IBM® InfoSphere™ Business Glossary provides a collaborative authoring environment that helps you create a central collection of enterprise-specific terminology and its relationships to technical information assets. The resulting collection, called a business glossary, is designed to help you understand business language and the business meaning of information assets like databases, jobs, database tables and columns, and business intelligence reports.

Search Business Glossary: account

Figure 6-25 Search window

A *term* is the entry point for searches into the metadata repository. Opening a term for viewing from the browser or InfoSphere Business Glossary Anywhere shows the full content of the term, including the definition, examples, relationships, and assignment. Now you can explore any available thread from any related term or assigned asset.

In most cases, the casual user is satisfied with the information provided in the glossary attribute fields. The more advanced user digs deeper into the information associated with a term. Other than the static information of an asset such as name, description, status, and owner or steward, the advanced user is interested in knowing where it fits into the bigger scheme. For example, the advanced user wants to where it is positioned in a data flow, where the data comes from, and where the data goes. Data lineage to and from an assigned asset can provide answers to these questions and more.

Figure 6-26 shows the entire flow from the source on the left side to the BI reports on the right side.

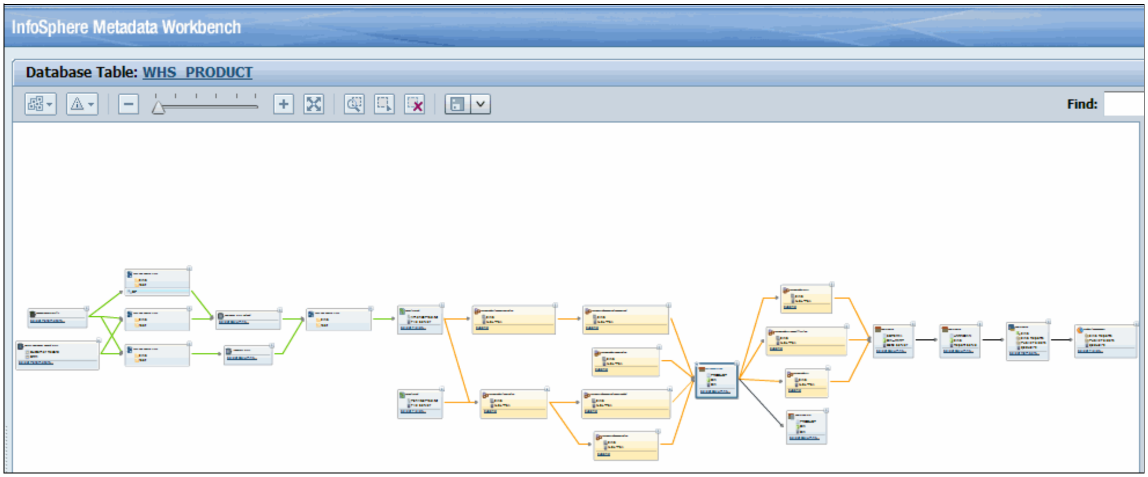
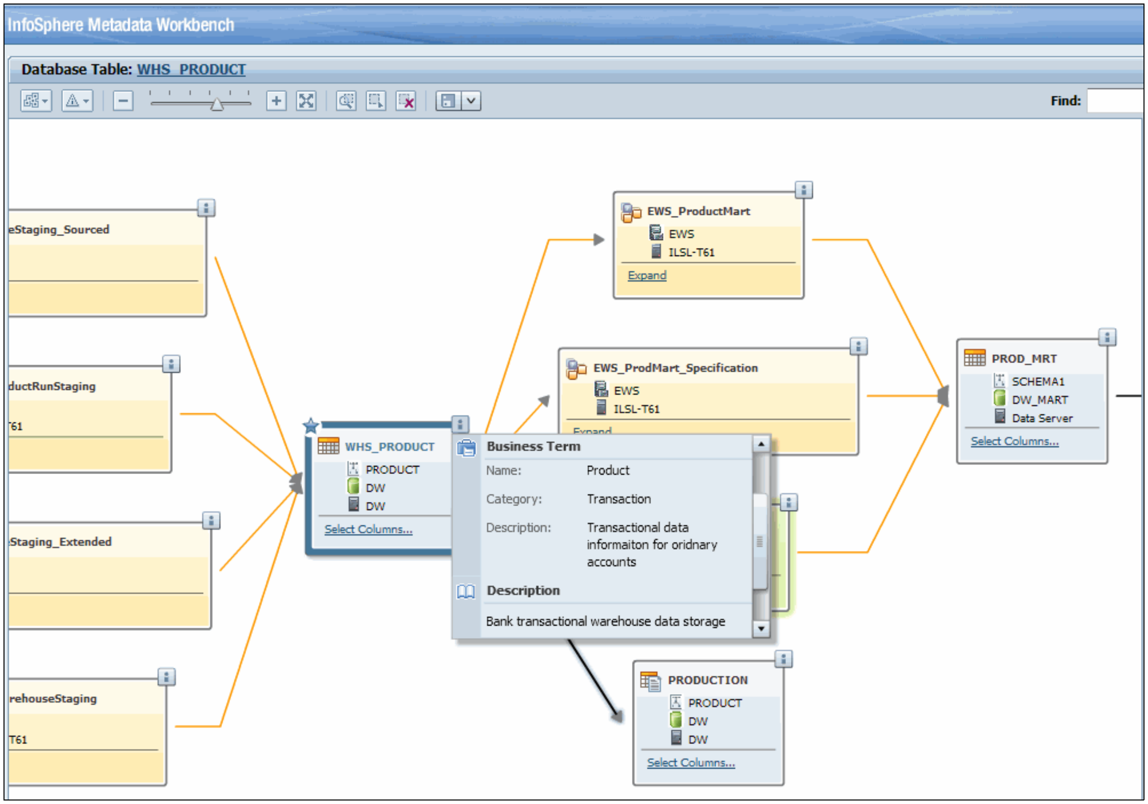


Figure 6-26 Data lineage

Each box in the lineage represents an asset of a certain type. You can view additional details of assets (Figure 6-27) by clicking the **Information** icon (ℹ) in the upper right corner of each box.



Business Term

Name: Product

Category: Transaction

Description: Transactional data informaiton for oridnary accounts

Description

Bank transactional warehouse data storage

Figure 6-27 Data lineage: Asset details

In Figure 6-27, the term *Bank Account Number* is assigned to the WHS_PRODUCT asset. It is the core information of the data lineage report that is generated to display the complete data flow, all of the assets involved before and after this point. Full details are provided in the window in the pane on the right. You can trace how the value of PRODUCT was created, where it originated, and where it is used. If operational metadata is available for the jobs, you can tell when it was last updated. You can also tell if any issues existed with the jobs that might raise questions about the reliability of the data, such as rejected records or job failure.

Other users, business analysts, developers, and other IT professionals might use the glossary as the entry point to expand their knowledge of data assets and their usage. A data analyst is given business requirements using business terms.

The analyst searches for a term to find its meaning, how it is used, and where it is used. The analyst can also explore the category to find what other terms populate the same category to gain a better understanding of the business domain. The analyst can explore neighboring categories to understand, from a broader perspective, possible relationships to other operational domains.

Throughout the organization, this information enhances understanding and improves communication about the creation, processing, and use of data.

6.7 Multiple ways of accessing InfoSphere Business Glossary

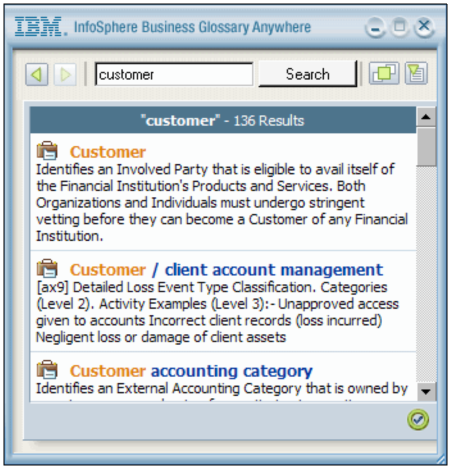
The business glossary must be the most accessible aspect of the metadata vocabulary. Your organization wants to take advantage of the full benefits of the information and knowledge stored in the repository and to improve the overall productivity in all business processes. To realize these objectives, your organization must be open to a large community of users, that is to everyone in an organization that wants to have access to the respository.

Various access methods are available to InfoSphere Business Glossary for searching and viewing. For example, with InfoSphere Business Glossary Anywhere, users can search and browse the glossary and other assets within a web browser. By using Representational State Transfer (REST) application programming interface (API), you can integrate the search and view capabilities of InfoSphere Business Glossary from other applications.

6.7.1 InfoSphere Business Glossary Anywhere

InfoSphere Business Glossary Anywhere, a small desktop client, provides viewing and searching access to InfoSphere Business Glossary from anywhere. For example, you might be reading a report, an email, or a spreadsheet and need an explanation or a definition of a term. In this case, you highlight the term or phrase and press Ctrl+Shift to start the InfoSphere Business Glossary Anywhere client, which instantly provides a result.

For example, you can highlight the term **Consumer** and then press Ctrl+Shift. The InfoSphere Business Glossary Anywhere window then opens, showing a list of terms with “Customer” in them (Figure 6-28 on page 169). With InfoSphere Business Glossary Anywhere, you can set the server, user preference, and the key combinations to start the product for words or phrases.



"customer" - 136 Results

Customer
Identifies an Involved Party that is eligible to avail itself of the Financial Institution's Products and Services. Both Organizations and Individuals must undergo stringent vetting before they can become a Customer of any Financial Institution.

Customer / client account management
[ax9] Detailed Loss Event Type Classification. Categories (Level 2). Activity Examples (Level 3):- Unapproved access given to accounts Incorrect client records (loss incurred) Negligent loss or damage of client assets

Customer accounting category
Identifies an External Accounting Category that is owned by

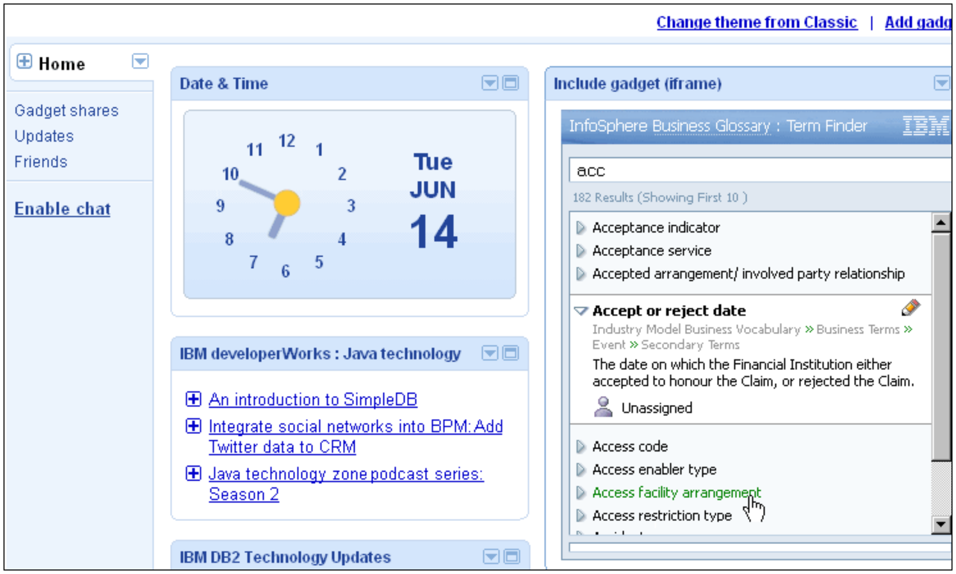
Figure 6-28 InfoSphere Business Glossary Anywhere window

Selecting a term opens the full range of attributes. You can navigate to the full browser and, from there, to any other asset associated with the term.

6.7.2 REST API

By using InfoSphere Business Glossary, you can create, manage, and share the vocabulary in your own web-based applications. The product exposes rich functionality through an API that uses a REST-based service.

REST is a standard web services protocol for reading and writing to a web server. It provides read/write access to the business glossary with proper authentication for accessing the business glossary content. The glossary content can be easily integrated into your custom web application and services to provide instant access to terms and their attributes. The content can be integrated in your corporate home page or any other application, as shown in Figure 6-29 on page 170.


Change theme from Classic | Add gadg
Home
Gadget shares
Updates
Friends
Enable chat
Date & Time
12
1
2
3
4
5
6
7
8
9
10
11
Tue
JUN
14
IBM developerWorks : Java technology
IBM DB2 Technology Updates
Include gadget (iframe)
InfoSphere Business Glossary : Term Finder
IBM

An introduction to SimpleDB
Integrate social networks into BPM: Add Twitter data to CRM
Java technology zone podcast series: Season 2

acc

182 Results (Showing First 10)

- Acceptance indicator
- Acceptance service
- Accepted arrangement/ involved party relationship
- **Accept or reject date**
  Industry Model Business Vocabulary » Business Terms » Event » Secondary Terms
  The date on which the Financial Institution either accepted to honour the Claim, or rejected the Claim.
  Unassigned
- Access code
- Access enabler type
- Access facility arrangement
- Access restriction type

Figure 6-29 InfoSphere Business Glossary integration on the home page with the REST API

6.7.3 Eclipse plug-in

InfoSphere Business Glossary Client for Eclipse is available to enable integration of business glossary terms in the data modeling and software design processes. By using the InfoSphere Business Glossary Client for Eclipse, you can view terms that are used within your enterprise and view details about them from within your Eclipse-based application.

For example, with InfoSphere Business Glossary Client for Eclipse, you can view glossary content while you develop software models of business processes with IBM Rational Software Architect products. Similarly, you can view glossary content while you work with logical data models and physical data models in IBM InfoSphere Data Architect. You can view glossary content while you work with physical data models in IBM InfoSphere Information Server metadata repository. You can easily choose the correct terms from the glossary to associate with logical and physical data model elements.

InfoSphere Business Glossary Client for Eclipse consists of these Eclipse features: core, Unified Model Language (UML) Profile, UML Integration, and data modeling integration.

Core features of InfoSphere Business Glossary

You can view a navigation tree of the terms and categories in the glossary from the Glossary Explorer view. You can perform text searches for terms and categories and view more in-depth information about them in the Properties view.

InfoSphere Business Glossary includes the following core features:

- Enables an in-context view of business vocabulary from any Eclipse client
- Enables read-only access to the full, updated content of InfoSphere Business Glossary
- Provides an updated view of InfoSphere Business Glossary content from within Eclipse
- Supports an offline workspace

InfoSphere Business Glossary UML Profile

The InfoSphere Business Glossary UML Profile feature provides the InfoSphere Business Glossary profile. The InfoSphere Business Glossary profile is applied to a UML model element when you assign a term to the element by using this feature. The InfoSphere Business Glossary profile includes a stereotype, «glossaryAssigned», that stores information about the assigned terms.

You can view the terms that have been associated with particular UML model elements. In addition, you can remove assigned terms from model elements, even if you have not installed the UML Integration feature.

The UML Profile feature has the following characteristics:

- Is based on InfoSphere Business Glossary Standard Eclipse
- Supports the dragging of terms to an empty canvas for the creation of new classes and attributes based on InfoSphere Business Glossary terms
- Supports the dragging of terms to existing classes and attributes for creation of local assigned relationships
- Storage of local assigned relationships in the UML model

  The relationships are available with that model when the same model is opened from any InfoSphere Business Glossary-enabled client. Local assignments are not stored in the Information Server.
- In IBM Rational Software Architect, notification by the validation mechanism of inherent models when changes occur in the InfoSphere Business Glossary content that affect assigned model elements

InfoSphere Business Glossary UML integration

You can associate glossary terms with UML model elements by using the drag-and-drop interface features. You can name model elements after terms, assign terms to model elements, and include term descriptions in the model element. When you assign terms to model elements, you apply the InfoSphere Business Glossary profile to the model elements.

If you have installed the InfoSphere Business Glossary Client for Eclipse UML Integration feature, you can incorporate business glossary terms into UML model elements in the following ways:

- Use a term name as the name of a UML model element.
- Assign terms to model elements. This assignment applies the «glossaryAssigned» stereotype to the model element. This stereotype is part of the InfoSphere Business Glossary profile and has the property Assigned Terms. The repository identifier (RID), name, and context of the assigned terms are stored in the Assigned Terms property.
- Use the term description in the documentation of the UML model element.

InfoSphere Business Glossary data model integration

You can assign terms to logical and physical data model elements and export these term assignments to the business glossary. You can also import term assignments that exist in the metadata repository to your physical and logical data model.

You can integrate the business glossary with InfoSphere Data Architect. You can also navigate the InfoSphere Business Glossary tree in InfoSphere Data Architect.

The Data Model Integration feature has the following characteristics:

- Is based on InfoSphere Business Glossary Standard Eclipse
- Supports the dragging of terms to an empty canvas for the creation of new entities, attributes, tables, and columns based on InfoSphere Business Glossary terms
- Supports the dragging of terms to existing entities, attributes, tables, and columns for the creation of assigned relationships.
- Interchange of assigned relationships between the InfoSphere Data Architect canvas and the physical schemas in the Information Server
  - Upload assignments from InfoSphere Data Architect to InfoSphere Information Server.
  - Copy assignments of the Information Server to the InfoSphere Data Architect canvas.

- Assigns terms to model elements

  The RID, name, and context of the assigned terms are stored as hidden annotations of model elements.

- Uses the term description in the documentation of the model element

  You import existing term assignments from the metadata repository to your local model.

- Exports term assignments from your local model to the metadata repository

6.8 Conclusion

In conclusion, this chapter explained how to use InfoSphere Business Glossary to define a business-centric glossary. A business glossary serves as the centerpiece for metadata management. This chapter illustrated how to search and explore the business glossary for business needs and introduces many options that are available to implement the business glossary.

Chapter 7, “Source documentation” on page 175, addresses the initial tasks that support a data integration project. These tasks include identifying, understanding, and documenting all source data that is required for the data integration project and for metadata management. Subsequent processes, addressed in other chapters, assess data quality, apply data rules, cleanse, and transform the data into the required data warehouse and data marts.

7

Source documentation

This chapter addresses documenting source data that is used in an information integration project for metadata management. Here, the term *documenting* refers to identifying, documenting, and loading the source data systems and applications into the metadata repository for IBM InfoSphere Information Server. IBM InfoSphere Metadata Asset Manager and IBM InfoSphere Metadata Workbench are used for this process. In addition, this chapter describes the intermediate data storage systems that support an integration project.

This chapter includes the following sections:

- Process overview
- Introduction to InfoSphere Metadata Asset Manager
- Application systems
- Sequential files
- Staging database
- Data extraction
- Conclusion

To gain a deeper understanding of source data, such as relationships among data, see Chapter 8, “Data relationship discovery” on page 209.

7.1 Process overview

To manage metadata to fully support data quality and regulatory requirements in any information integrated solution, you must document the source data systems and applications where information is created and generated. The documentation of these systems allows for the application of enterprise definitions, in the form of glossary terms. It also allows for the inclusion of these systems in data lineage and analysis, representing the origin of information for business intelligence (BI) reports and data storage tables.

Source information is created or generated at the point of origin, for example, at the point of a transaction, production, or sale. Furthermore, a single customer might generate multiple transactions, and each transaction includes a referenced point of production.

Metadata management requires the identification of all source systems, specifically understanding the information that is represented and how data is structured and referenced in those systems. After the data is identified, the data is often loaded into an intermediate data source that supports multiple and separate data sources. The intermediate data storage center typically offers a well-defined and normalized data structure and, therefore, might differ from the original source system. You can use data quality rules, privacy regulations, or quality assessment tasks to ensure the validity and deduplication of information. Additionally, the intermediate data storage center provides the continued development requirement for data warehousing and reporting.

7.2 Introduction to InfoSphere Metadata Asset Manager

IBM InfoSphere Metadata Asset Manager (Figure 7-2 on page 178) loads and manages information assets (in this case, metadata of the source data systems) that are used in an integrated solution. You can import assets into a staging area before you share them with the metadata repository. In the metadata repository, you can browse and search for common metadata assets, set implementation relationships between them, and merge duplicates.

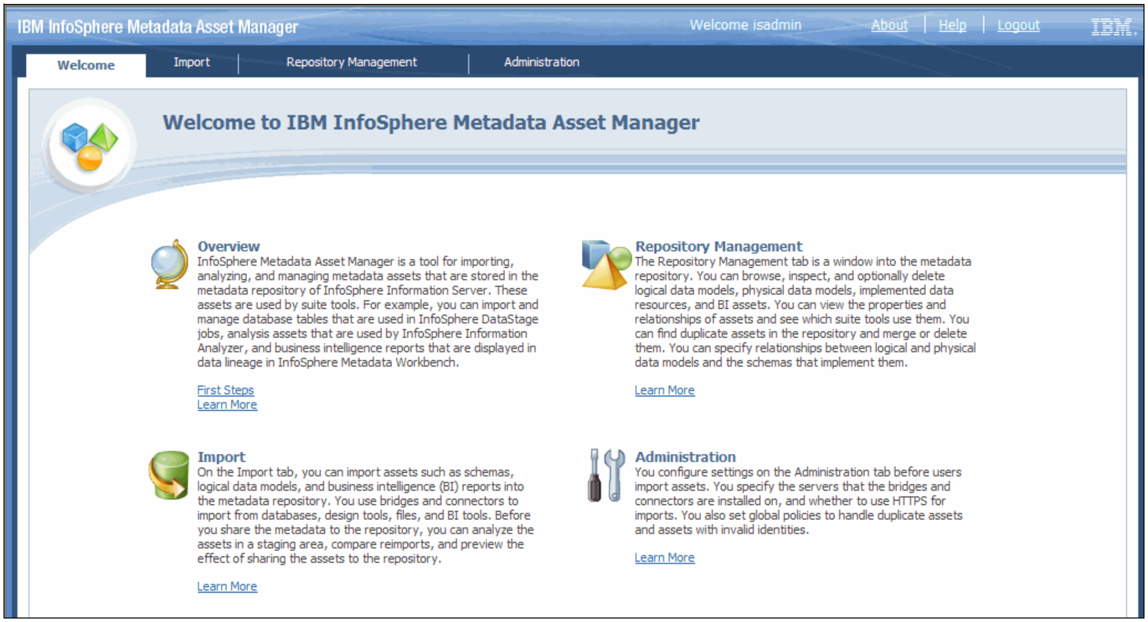
When you share imports to the InfoSphere Information Server metadata repository, you can use the imported assets to analyze the assets, use them in jobs, assign them to terms, or designate stewards for the assets. Until you share the import, the assets are not visible in the metadata repository and cannot be used other InfoSphere Information Server product modules and components.

Information assets include relational database management systems (RDBMS) and model, file, and BI systems.

You install InfoSphere Metadata Asset Manager as part of the InfoSphere Information Server and access it using a web browser at the following default URL:

`http://`*`servername`*`:9080/ibm/imam/console`

The import process includes defining a connection to the source asset, filtering the asset content, previewing the loaded metadata, and publishing to the repository. InfoSphere Metadata Asset Manager plays an important role in satisfying development, data quality, and business requirements.



Welcome to IBM InfoSphere Metadata Asset Manager

Overview
InfoSphere Metadata Asset Manager is a tool for importing, analyzing, and managing metadata assets that are stored in the metadata repository of InfoSphere Information Server. These assets are used by suite tools. For example, you can import and manage database tables that are used in InfoSphere DataStage jobs, analysis assets that are used by InfoSphere Information Analyzer, and business intelligence reports that are displayed in data lineage in InfoSphere Metadata Workbench.

First Steps
Learn More

Repository Management
The Repository Management tab is a window into the metadata repository. You can browse, inspect, and optionally delete logical data models, physical data models, implemented data resources, and BI assets. You can view the properties and relationships of assets and see which suite tools use them. You can find duplicate assets in the repository and merge or delete them. You can specify relationships between logical and physical data models and the schemas that implement them.

Learn More

Import
On the Import tab, you can import assets such as schemas, logical data models, and business intelligence (BI) reports into the metadata repository. You use bridges and connectors to import from databases, design tools, files, and BI tools. Before you share the metadata to the repository, you can analyze the assets in a staging area, compare reimports, and preview the effect of sharing the assets to the repository.

Learn More

Administration
You configure settings on the Administration tab before users import assets. You specify the servers that the bridges and connectors are installed on, and whether to use HTTPS for imports. You also set global policies to handle duplicate assets and assets with invalid identities.

Learn More

Figure 7-1 Welcome window for InfoSphere Metadata Asset Manager

7.3 Application systems

The objective in the source documentation process is to document and include metadata from all disparate source data systems that are used in an integration project. These sources might come from conventional data models, databases, or data files. They can be directly loaded, by using InfoSphere Metadata Asset Manager, into the shared metadata repository. Sometimes these sources represent user applications, transactional systems, or existing data storage centers. These external systems can be represented in a generic format in the metadata repository.

InfoSphere Metadata Workbench allows for the import, management, and mapping of these external sources, which are referenced as *extended data* *sources*. The purpose of documenting these external data sources is to enable the generation of lineage reports and impact analysis that include the originating extended source data.

Additionally, InfoSphere Metadata Workbench can extend business definitions and the requirements of these assets from IBM InfoSphere Business Glossary. This way, business and IT users have a complete association from the business term to information assets.

InfoSphere Metadata Workbench fully creates and manages the extended data sources from within. InfoSphere Metadata Workbench supports three asset types, which might represent the varied sources to be documented and represented. In the bank scenario for this book, we load an extended data source of the application type, which is the application source of Bank A and Bank B (Figure 7-2).

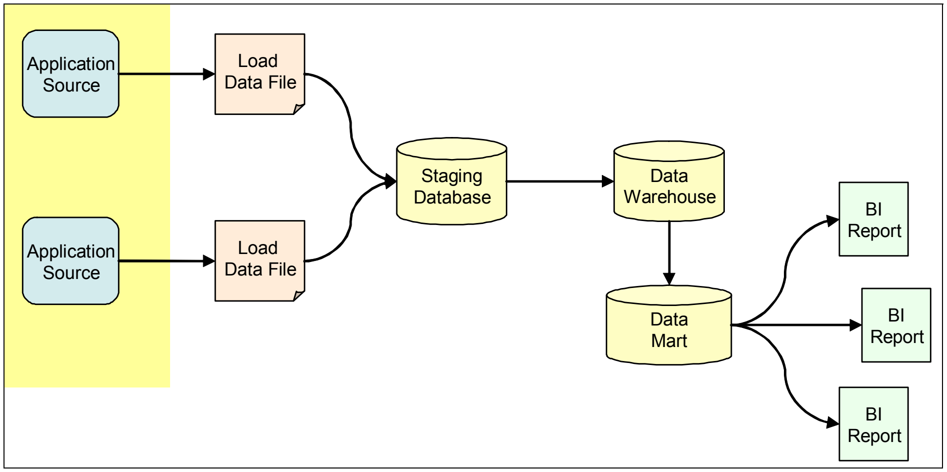


Figure 7-2 Application source for Bank A and Bank B in a data flow

Figure 7-3 shows the sample customer terminal application of Bank A with two tables, Account and Customer.

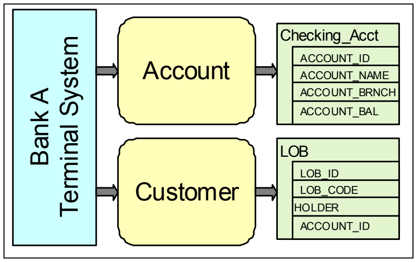


Figure 7-3 Bank A sample customer terminal application

7.3.1 Extended data source types

You can create and manage the following extended data source types in InfoSphere Metadata Workbench:

- Application assets
- Stored procedure definition assets
- File asset

Application assets

Application assets might represent web services, source applications, or point-of-sale transactional systems. Applications include the following hierarchy of information:

Application ( ) Alternate data structures, such as web services, programs, or scripts that contain information or perform specific actions. Applications contain object types.

Object type ( ) A grouping of methods that characterizes the input and output structures of an application, representing a common feature or business process in an application. Object types contain methods and belong to a single application.

Method ( ) Functions or procedures that are defined in an object and perform a specific operation. Operations send or receive information as parameters or values. Methods contain parameters or values and belong to a single object type.

Input parameter ( ) The input of information from a host application to a method, allowing the method to perform its intended function. Input parameters belong to a single method.

Output value ( ) The results of a method, processing, or creation of data that is returned to the host application. Output values belong to a single method.

Stored procedure definition assets

Stored procedure definition assets represent stored procedures or native scripts, which extract, transform, and load (ETL) data. A stored procedure includes the following hierarchy of information:

Stored procedure definition ( )

Routines that access, transform, or consolidate information and can update, append, or retrieve data. Typically, stored procedures are created and stored in a database system and are used to control data processing as condition handlers or programs. A stored procedure definition can have multiple in parameters, out parameters, inout parameters, and result columns.

In parameter ( ) Carry information that is required for the stored procedure to perform its intended function, for example, variables that are passed to the stored procedure. In parameters belong to a single stored procedure definition.

Out parameter ( ) The value or variable that is returned when a stored procedure executes. For example, a field that is included in the result set of the stored procedure can be an out parameter. Out parameters belong to a single stored procedure definition.

Inout parameter ( ) Information that is passed to the stored procedure that is used to perform its intended function and then processes the information, returning an output value by using the same parameter. Inout parameters belong to a single stored procedure definition.

Result column ( ) The returned data values of a stored procedure when processing or querying information. Result columns belong to a single stored procedure definition.

File asset

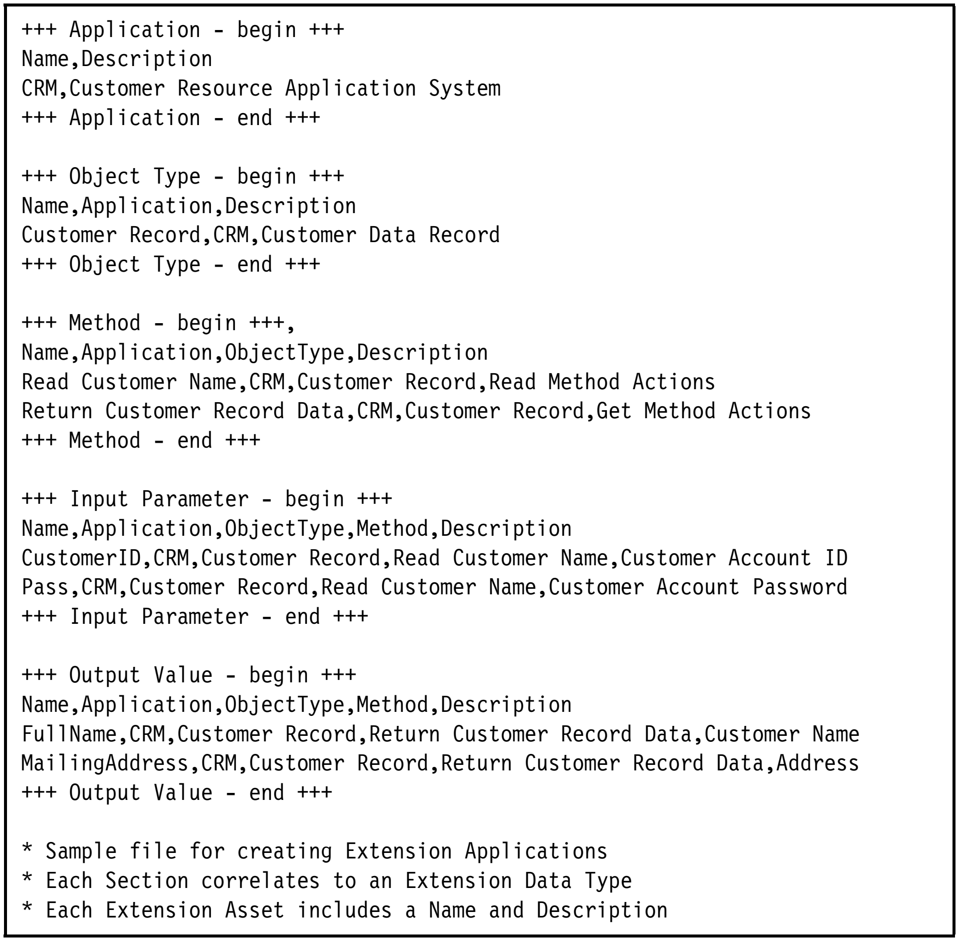
A file asset represents a single unit of data, such as a file or closed system. A file does not includes a hierarchy of information. A file ( ) represents a storage medium for data, for capturing, transferring, or otherwise reading information. Files are featured in the information supply chain, used in data integration, or used for data lookup. Data file assets are similar in definition to a file. However, data file assets are loaded directly into InfoSphere Information Sever and contain data file fields. Files represent the documentation of this type of a storage medium and do not include fields.

7.3.2 Format

Extended data sources are documented in an input file and loaded into InfoSphere Metadata Workbench. The input file is a comma-separated value (CSV) file, delineating the name, description, and containment of each information asset.

The format and process allow the necessary representation of data sources and systems, providing a better understanding of the enterprise assets and the validation of the assets. In particular, the format and process provide the capability to assign glossary terms to an asset and to include the asset in the data lineage analysis as the source of information.

Figure 7-4 shows an example of a CSV input file that represents an extended data source of the application type.



```
+++ Application - begin +++
Name,Description
CRM,Customer Resource Application System
+++ Application - end +++

+++ Object Type - begin +++
Name,Application,Description
Customer Record,CRM,Customer Data Record
+++ Object Type - end +++

+++ Method - begin +++,
Name,Application,ObjectType,Description
Read Customer Name,CRM,Customer Record,Read Method Actions
Return Customer Record Data,CRM,Customer Record,Get Method Actions
+++ Method - end +++

+++ Input Parameter - begin +++
Name,Application,ObjectType,Method,Description
CustomerID,CRM,Customer Record,Read Customer Name,Customer Account ID
Pass,CRM,Customer Record,Read Customer Name,Customer Account Password
+++ Input Parameter - end +++

+++ Output Value - begin +++
Name,Application,ObjectType,Method,Description
FullName,CRM,Customer Record,Return Customer Record Data,Customer Name
MailingAddress,CRM,Customer Record,Return Customer Record Data,Address
+++ Output Value - end +++

* Sample file for creating Extension Applications
* Each Section correlates to an Extension Data Type
* Each Extension Asset includes a Name and Description
```

Figure 7-4 Sample input file representing extended source application

The following information describes the sample input file that is shown in Figure 7-4:

- Each hierarchy is defined in a unique section representing a single asset type.
- Each section includes a header, footer, and column header.
- The header and footer are predetermined and include the expected name of the asset type.

- The column header is predetermined for the specific asset type.
- Each row of the section represents a specific asset and includes a name.

7.3.3 Loading the application system

In this example, we import an extended data source of the application type from InfoSphere Metadata Workbench by using the following steps:

1. Prepare the input file to represent an extended data source application.

   The input file includes five sections, representing `Application`, `Object Type`, `Method`, `In Parameter`, and `Output Value`. The input file generates a single application that contains a single object type having two methods. Each method further includes two parameters or values. The application represents the Bank A data center.

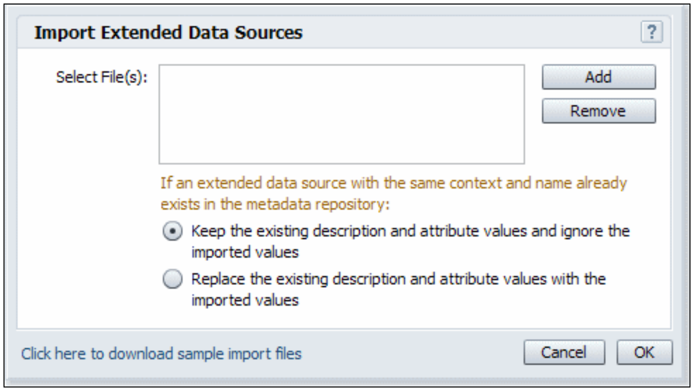
2. Save the input file in a CSV text format.
3. Log on to InfoSphere Metadata Workbench as an InfoSphere Metadata Workbench Administrator user.
4. Click **Advanced** from the left navigation pane, and select **Import Extended Data Sources**.
5. In the Import Extended Data Sources window (Figure 7-5 on page 183), complete these steps:

   a. Click **Add** to browse and select one or more of the input files. The input files can represent various extended data source types, such as application and file.

   b. Optional: Select **Keep the existing description and attribute values and ignore the imported values** to retain the authoring of the existing extended data sources that are in InfoSphere Metadata Workbench. This selection does not overwrite the asset description with the asset description that was authored in the input file.

   c. Optional: Select **Replace the existing description and attribute values with the imported values** to remove the authoring of the existing extended data sources that are already in InfoSphere Metadata Workbench. This selection overwrites the asset description with the asset description that was authored in the input file.

   d. Click **OK** to proceed to import the input files and create the extended data sources.



Import Extended Data Sources

Select File(s):

If an extended data source with the same context and name already exists in the metadata repository:

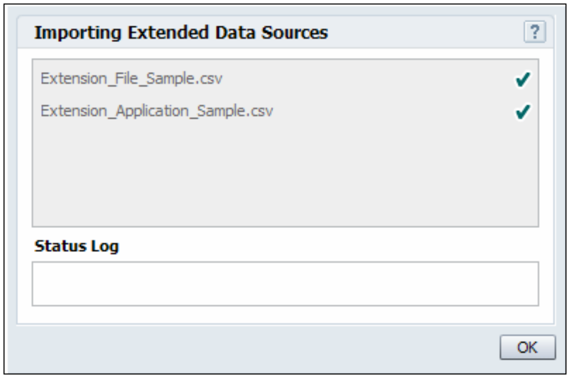
● Keep the existing description and attribute values and ignore the imported values

○ Replace the existing description and attribute values with the imported values

Click here to download sample import files

Figure 7-5 Importing extended data sources using InfoSphere Metadata Workbench

6. In the Importing Extended Data Sources window (Figure 7-6), review the message that indicates the progress of the import and any generated warnings. Then click **OK**.



Importing Extended Data Sources

Extension_File_Sample.csv ✓

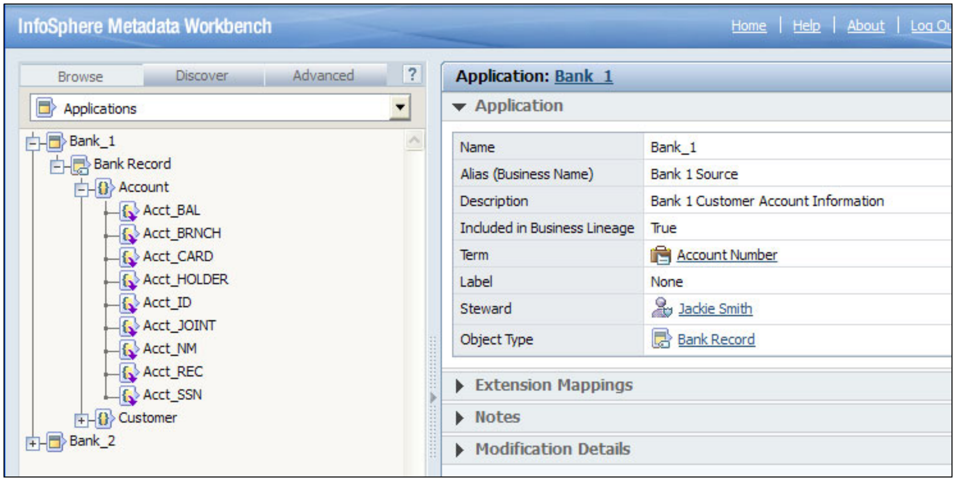
Extension_Application_Sample.csv ✓

Status Log

Figure 7-6 Importing Extended Data Sources progress window

7.3.4 Results

Extended data sources represent data structures or data information that is defined in the data flow. You can preview the application, stored procedure definition, or file assets in InfoSphere Metadata Workbench or InfoSphere Business Glossary, as shown in Figure 7-7.



Name	Bank_1
Alias (Business Name)	Bank 1 Source
Description	Bank 1 Customer Account Information
Included in Business Lineage	True
Term	Account Number
Label	None
Steward	Jackie Smith
Object Type	Bank Record

Figure 7-7 View of extended data source application in InfoSphere Metadata Workbench

7.4 Sequential files

Sequential files represent data source or lookup information. Alternatively, they serve as the intermediary data storage requirements of the integration project, when gathering and consolidating source data. With the representation of these files in InfoSphere Information Server, stakeholders can reference information while developing, understanding the meaning of, or inspecting data flow analysis.

Sequential files are part of the data lineage analysis reports from InfoSphere Metadata Workbench. They represent a step in the data flow. Furthermore, you can extend sequential file definitions to reference business definitions and requirements from InfoSphere Business Glossary. Sequential file definitions include the name and path location of the file, in addition to its defined columns with their associated data types and lengths.

You load sequential files by using InfoSphere DataStage and InfoSphere QualityStage Designer, by invoking the sequential file definitions import. When complete, this import process creates a table definition that represents the structure of the sequential file, including its columns and their types. Publish the created table definition to InfoSphere Information Server. This way, you can preview and use the data file assets in all their components, including data lineage reports from InfoSphere Metadata Workbench.

In this example, we import a sequential file definition and publish it to InfoSphere Information Server, as shown in Figure 7-8.

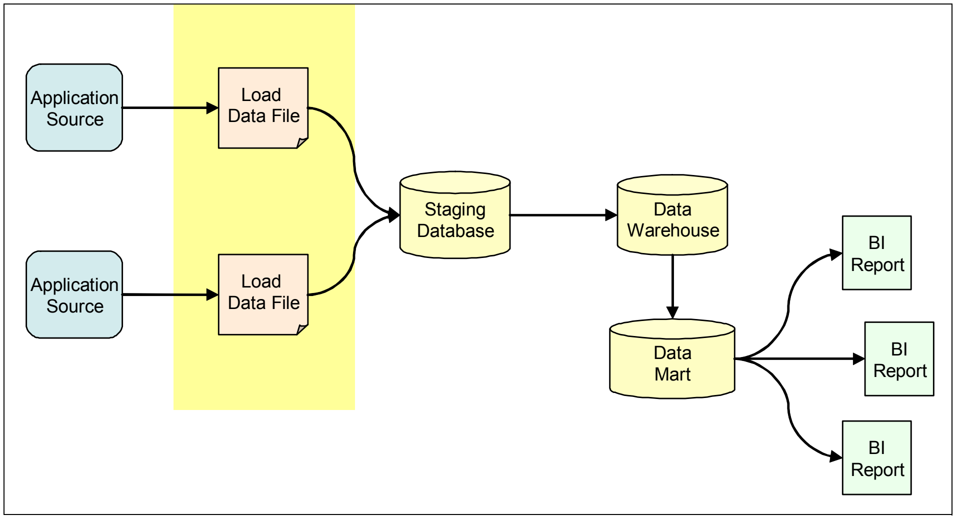


Figure 7-8 Data flow of an integrated solution

7.4.1 Loading a data file

To load a data file, complete these steps:

1. Log on to the InfoSphere DataStage and InfoSphere QualityStage Client for the data integration project.
2. Select **Import** → **Table Definitions** → **Sequential File Definitions**.

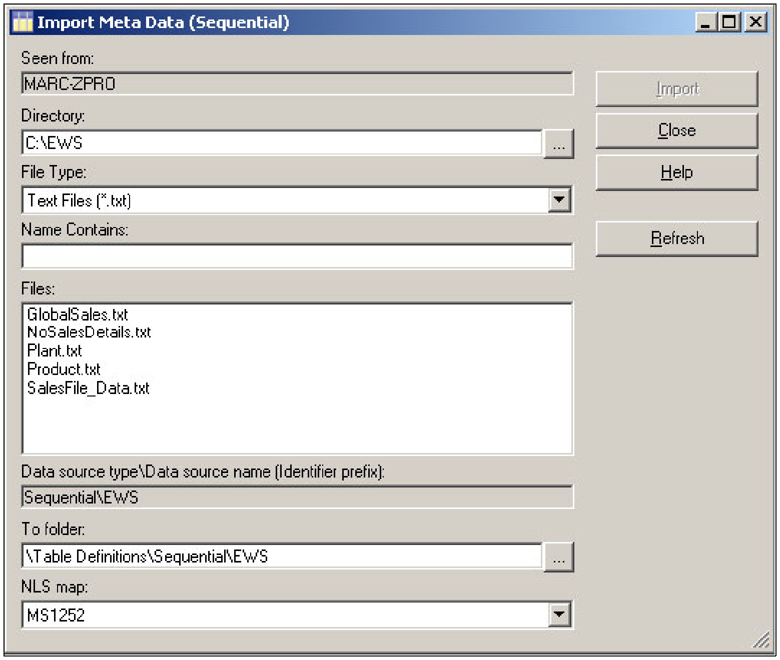
3. In the Import metadata (Sequential) window (Figure 7-9), complete these steps:

   a. Select the directory that contains the file to import. Click the ellipsis (**...**) button to browse to select a directory.

   b. Select the file from the list of displayed files for import.

   c. Set the InfoSphere DataStage project folder to contain the table definition to be created by the import process. Click the **...** button to browse the folder to select the InfoSphere DataStage project folder.

   d. Click **Import**.



Seen from: MARC-ZPRO

Directory: C:\EWS

File Type: Text Files (*.txt)

Name Contains:

Files:
GlobalSales.txt
NoSalesDetails.txt
Plant.txt
Product.txt
SalesFile_Data.txt

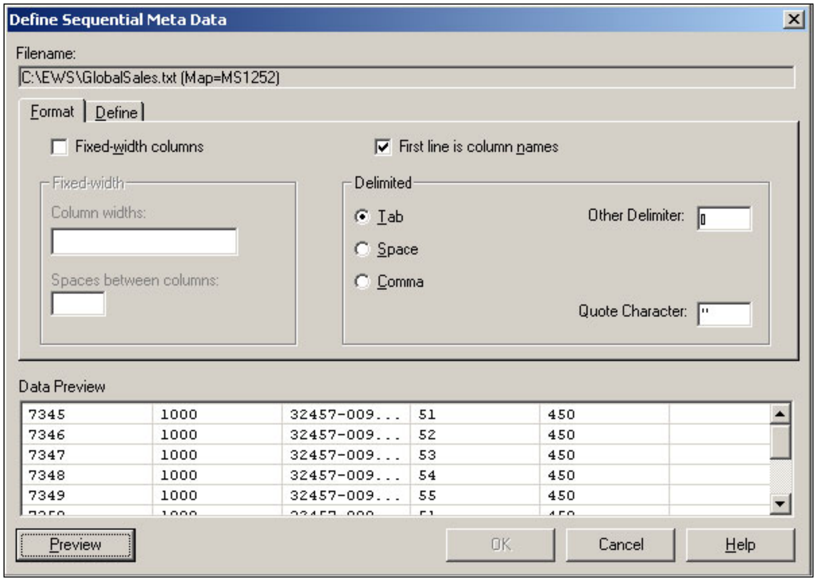
Data source type\Data source name (Identifier prefix): Sequential\EWS

To folder: \Table Definitions\Sequential\EWS

NLS map: MS1252

Figure 7-9 Importing the data file from InfoSphere DataStage

4. In the Define Sequential Metadata window, complete these tasks:
   a. On the **Format** tab (Figure 7-10), complete these steps:
      i. Select the correct delimiter for the file (tab, space, or comma).
      ii. Select **First line is column names** if the first line of the file contains the column names.
      iii. If the width of each column of the file is fixed, select **Fixed-width columns**.
      iv. Click **Preview**. Ensure that the data preview shows the columns of the file correctly as shown in Figure 7-10.



Define Sequential Meta Data

Filename: C:\EWS\GlobalSales.txt (Map=MS1252)

□ Fixed-width columns ☑ First line is column names

Fixed-width

Column widths:

Spaces between columns:

Delimited

● Tab
○ Space
○ Comma

Other Delimiter:

Quote Character: "

Data Preview

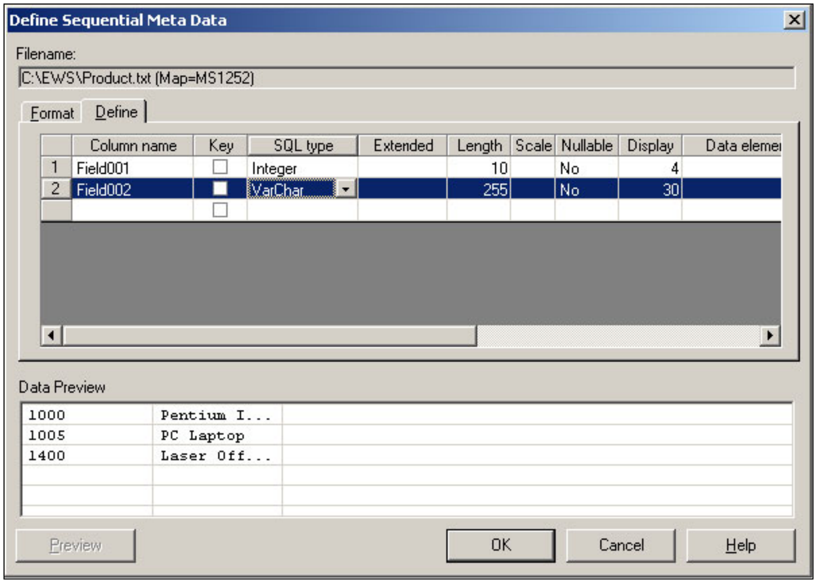
7345	1000	32457-009...	51	450	
7346	1000	32457-009...	52	450	
7347	1000	32457-009...	53	450	
7348	1000	32457-009...	54	450	
7349	1000	32457-009...	55	450	

Figure 7-10 Previewing the sequential file structure

b. On the **Define** tab (Figure 7-11), complete these steps:

i. Enter the names of the columns for the file. If the first line of the sequential file contained the column names, the list is prepopulated.

ii. Optional: Set the column properties, including Key, SQL type, Length, Nullability, and Description.



Define Sequential Meta Data

Filename:
C:\EWS\Product.txt (Map=MS1252)

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	Display	Data elemei
1	Field001	□	Integer		10		No	4	
2	Field002	□	VarChar		255		No	30	
		□							

Data Preview

1000	Pentium I...
1005	PC Laptop
1400	Laser Off...

Figure 7-11 Previewing and defining the sequential file column name and type

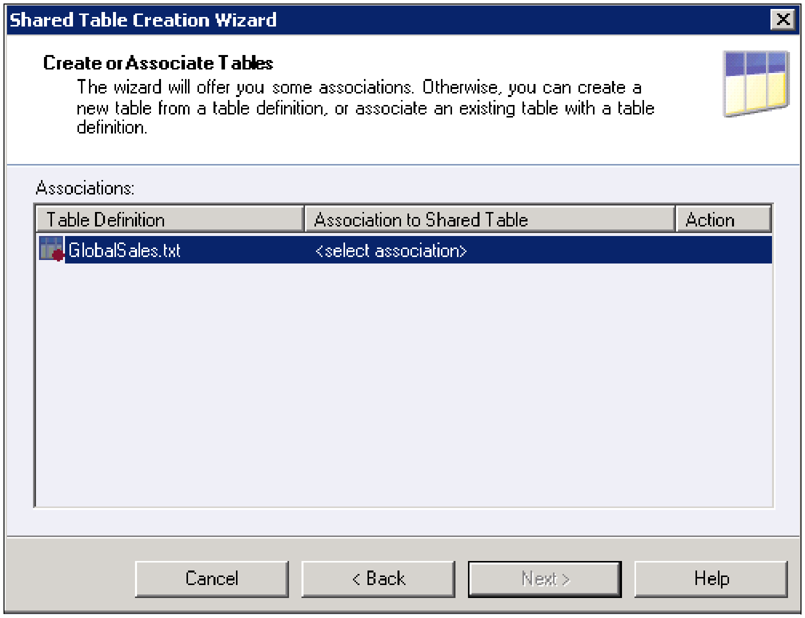
c. Click **OK**.

5. Select another file for import, or click **Close** to close the Import Sequential Metadata window.

You have now created a table definition in the InfoSphere DataStage project that represents the imported sequential file. Developers can use this table definition in the project when developing jobs to read from the data file and load the staging database. However, you must publish these files to InfoSphere Information Server metadata repository. This way, the repository can be used by the components of InfoSphere Information Server, including InfoSphere Metadata Workbench and InfoSphere Business Glossary.

To publish the files to InfoSphere Information Server metadata repository, complete the following steps:

1. Browse to select the newly created table definition in the InfoSphere DataStage Client folder repository view.
2. Right-click the table definition, and then select **Shared Table Creation Wizard**.
3. Select the table definition, and then click **Next**.
4. In the Create or Associate Tables window (Figure 7-12), select the table definition, and then click **<select association>** to select an association to an existing data file. Select **Create New** from the list box.



Shared Table Creation Wizard

Create or Associate Tables

The wizard will offer you some associations. Otherwise, you can create a new table from a table definition, or associate an existing table with a table definition.

Associations:

Table Definition	Association to Shared Table	Action
GlobalSales.txt	<select association>	

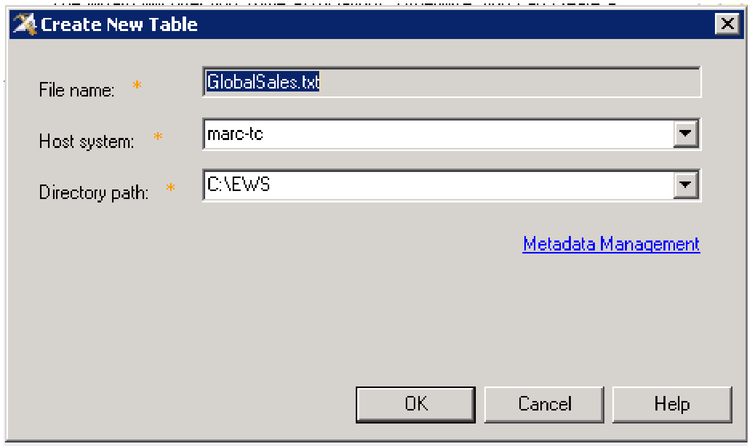
Figure 7-12 Create or Associate Tables window

5. In the Create New Table dialog box (Figure 7-13), complete these steps:

   a. Select the name of an existing host system, or click the **Metadata Management** link to create a host system from within the InfoSphere DataStage Client. Host systems define the container of a data file.

   b. Select the Directory path, or type another path location where the sequential file is stored. This location must match the actual location of the file, as referenced in InfoSphere DataStage and InfoSphere QualityStage job, to ensure accurate data lineage results.

   c. Click **OK**.



Create New Table

File name: * GlobalSales.txt

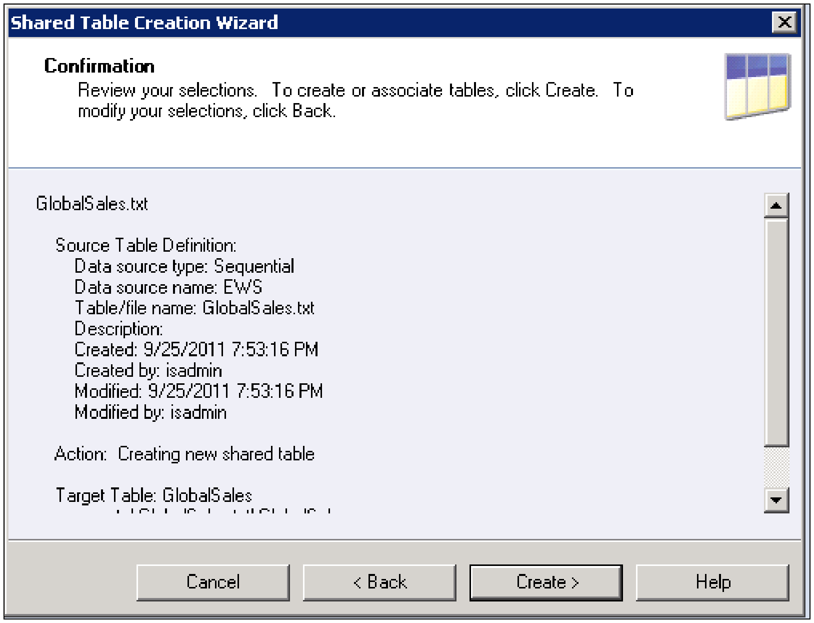
Host system: * marc-tc

Directory path: * C:\EWS

Figure 7-13 Publish data file to InfoSphere Information Server

6. Back in the Create or Associate Tables window, click **Next**.

7. In the Confirmation window (Figure 7-14), click **Create** to publish the table definition as a data file asset.



Shared Table Creation Wizard

Confirmation

Review your selections. To create or associate tables, click Create. To modify your selections, click Back.

GlobalSales.txt

Source Table Definition:
Data source type: Sequential
Data source name: EWS
Table/file name: GlobalSales.txt
Description:
Created: 9/25/2011 7:53:16 PM
Created by: isadmin
Modified: 9/25/2011 7:53:16 PM
Modified by: isadmin

Action: Creating new shared table

Target Table: GlobalSales

Figure 7-14 Clicking Create in the Confirmation window

7.4.2 Results

Data files represent the data sources or lookups that are part of the integrated solution. The following information assets are available in InfoSphere Information Server upon import of a data file:

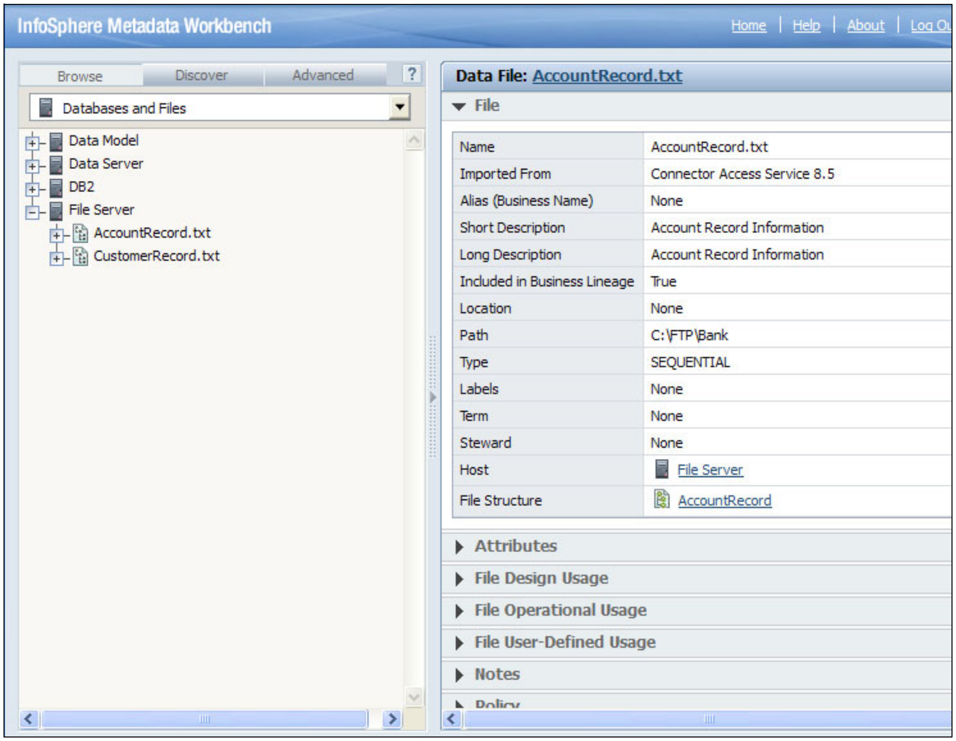
Host system ( ) The computer system in which the file resides. A host system is a user-defined structure that is created during import.

Data file ( ) The name of the data file.

Data file structure ( ) The component or root structure of the file, which is defined during publication.

Data file field ( ) The data fields that are in a file, including their defined types and lengths.

You can search, browse, and view the details of data file assets from the components of InfoSphere Information Server, including InfoSphere Metadata Workbench, as shown in Figure 7-15.



Data File: AccountRecord.txt

Name	AccountRecord.txt
Imported From	Connector Access Service 8.5
Alias (Business Name)	None
Short Description	Account Record Information
Long Description	Account Record Information
Included in Business Lineage	True
Location	None
Path	C:\FTP\Bank
Type	SEQUENTIAL
Labels	None
Term	None
Steward	None
Host	File Server
File Structure	AccountRecord

Figure 7-15 Display of a loaded data file from InfoSphere Metadata Workbench

7.5 Staging database

Data storage systems, such as relational databases, represent the information data at the center of all business transaction and decision making. By loading and including these database assets in InfoSphere Information Server, stakeholders can reference this information while developing and viewing their enterprise definitions or business requirements. Stakeholders can also profile data, discover data, and apply data quality rules. These database assets are crucial when inspecting data lineage from generated BI reports or data marts that depict the steps in the data flow.

You load and manage database assets by using InfoSphere Metadata Asset Manager. In the bank example in this book, we import an RDBMS by using a pre-existing Open Database Connectivity (ODBC) connection to the database source, as shown in Figure 7-16.

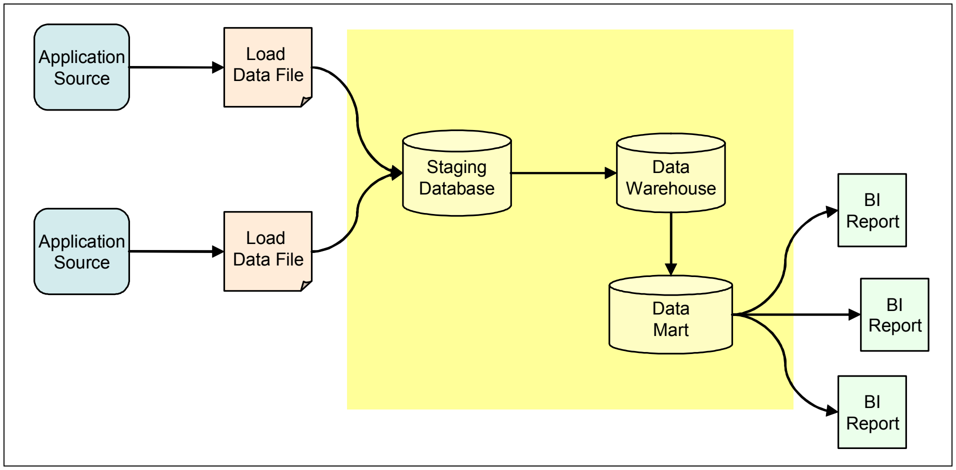


Figure 7-16 Loading the staging database

7.5.1 Loading the staging database

To load the staging database, complete these steps:

1. Log on to InfoSphere Metadata Asset Manager.
2. Click **New Import Area** to define a new area to import the database system. The import area allows subsequent reimporting and management of the imported metadata.

3. In the New Import Area panel (Figure 7-17), complete these steps:

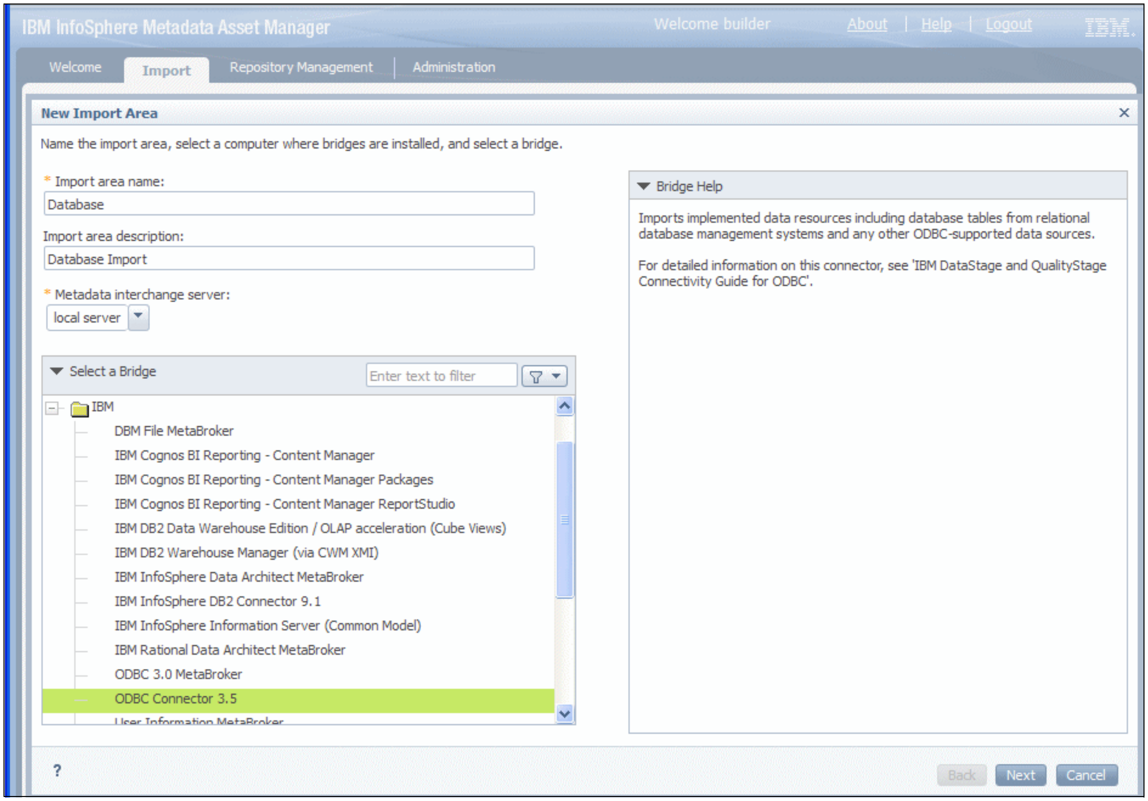
   a. Enter a name for the Import Area that uniquely identifies the import process for future reimporting or administration.

   b. Optional: Enter a description for the Import Area that describes the database asset to import or the process.

   c. Select a metadata interchange server. This server defines the connectivity to the IBM InfoSphere DataStage Engine or IBM InfoSphere Metadata Interchange Agent where the bridges or connectors for import are defined.

   d. Browse for and select a bridge, which provides the connection parameters to the load information from the source system. In this example, we select the **ODBC Connector 3.5** bridge.

   e. Click **Next** to proceed.



New Import Area

Name the import area, select a computer where bridges are installed, and select a bridge.

* Import area name: Database

Import area description: Database Import

* Metadata interchange server: local server

Select a Bridge

Enter text to filter

IBM
- DBM File MetaBroker
- IBM Cognos BI Reporting - Content Manager
- IBM Cognos BI Reporting - Content Manager Packages
- IBM Cognos BI Reporting - Content Manager ReportStudio
- IBM DB2 Data Warehouse Edition / OLAP acceleration (Cube Views)
- IBM DB2 Warehouse Manager (via CWM XMI)
- IBM InfoSphere Data Architect MetaBroker
- IBM InfoSphere DB2 Connector 9.1
- IBM InfoSphere Information Server (Common Model)
- IBM Rational Data Architect MetaBroker
- ODBC 3.0 MetaBroker
- ODBC Connector 3.5
- User Information MetaBroker

Bridge Help

Imports implemented data resources including database tables from relational database management systems and any other ODBC-supported data sources.

For detailed information on this connector, see 'IBM DataStage and QualityStage Connectivity Guide for ODBC'.

Figure 7-17 Creating an import area in InfoSphere Metadata Asset Manager

4. Specify the values for the bridge import parameters (Figure 7-18). The parameters identify the connection to the database system to be imported.

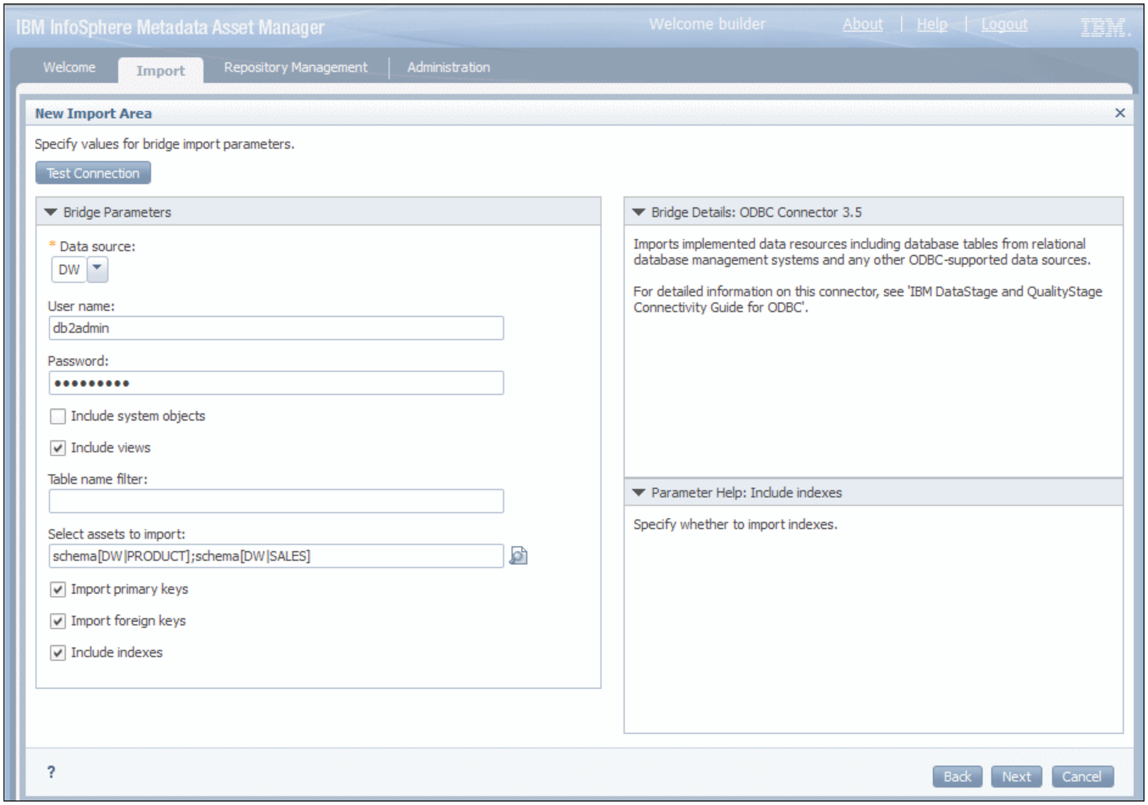
   a. Browse for and then select a previously configured ODBC data source. The ODBC Connection represents the data connectivity between InfoSphere Information Server and the database server.

   b. Enter the user name to connect to the database system. The user must have privileges to read from the database catalog.

   c. Enter the password for this user.

   d. Optional: Select **Include system objects**, which represent the system tables.

   e. Optional: Select **Include views** that are defined in the database system.



New Import Area

Specify values for bridge import parameters.

Bridge Parameters

* Data source: DW

User name: db2admin

Password: •••••••••

☐ Include system objects

☑ Include views

Table name filter:

Select assets to import: schema[DW|PRODUCT];schema[DW|SALES]

☑ Import primary keys

☑ Import foreign keys

☑ Include indexes

Bridge Details: ODBC Connector 3.5

Imports implemented data resources including database tables from relational database management systems and any other ODBC-supported data sources.

For detailed information on this connector, see 'IBM DataStage and QualityStage Connectivity Guide for ODBC'.

Parameter Help: Include indexes

Specify whether to import indexes.

Figure 7-18 Specifying the values for the bridge import parameters

   f. For Select assets to import field, select the database objects for import. Click the **Search** icon (magnifying glass) to browse for and select the database schema or tables to import. Identify and load the key database objects that are required in the integrated solution.

g. Click **Import primary keys** to capture the keys that are defined in the database system.

h. Click **Import foreign keys** to capture the reference keys that are defined in the database system.

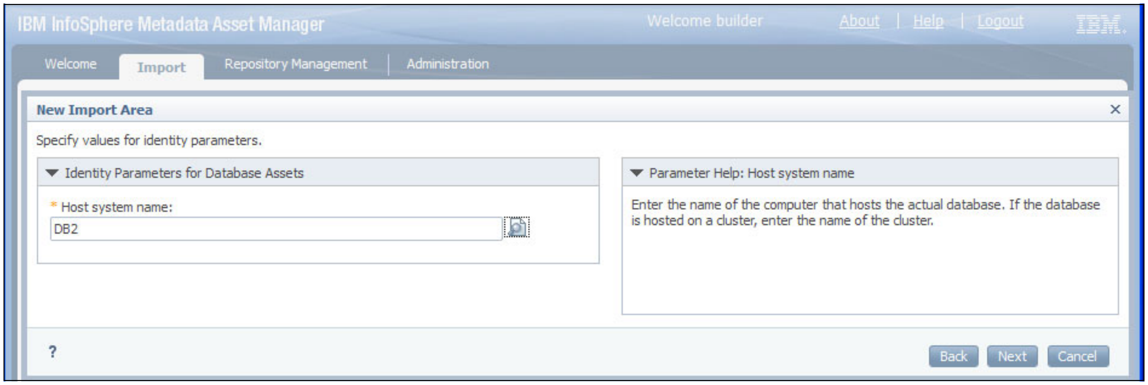
i. Click **Include indexes** to capture the indexes that are defined in the database system.

j. Click **Next**.

5. Specify the values for the identity parameters (Figure 7-19). The identity parameters include the host system, which helps a user identify and classify information in InfoSphere Information Server.

   a. Browse for and then select an existing host system or enter the name of the host system. The host system must reflect the server on which the database system has been deployed.

   b. Click **Next**.



Specify values for identity parameters.

* Host system name: DB2

Enter the name of the computer that hosts the actual database. If the database is hosted on a cluster, enter the name of the cluster.

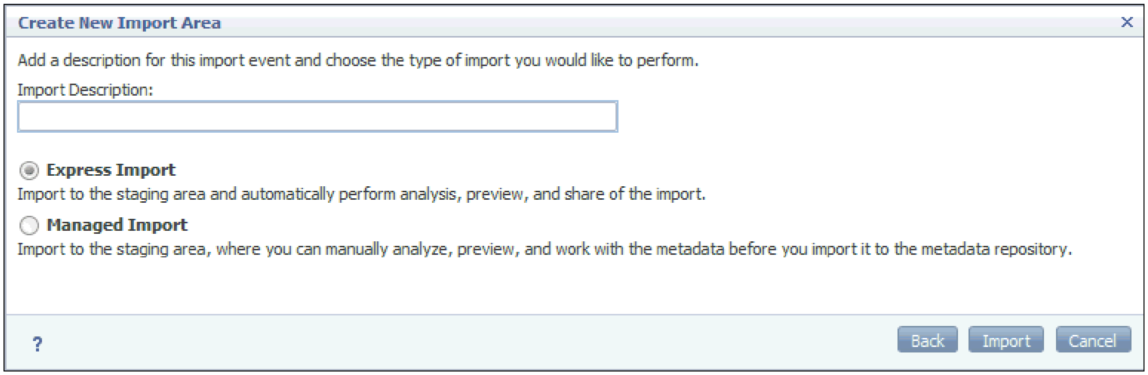
Figure 7-19 Specifying the values for the identity parameters

6. Complete the import event (Figure 7-20) so that you can preview the database asset before publishing it to InfoSphere Information Server:

    a. Select **Express Import** to automatically publish the database to the metadata repository.

    b. Select **Managed Import** to review the database before its publication to the metadata server.

    c. Click **Import** to complete the process and publish the database system to the metadata repository.



Create New Import Area

Add a description for this import event and choose the type of import you would like to perform.

Import Description:

● Express Import

Import to the staging area and automatically perform analysis, preview, and share of the import.

○ Managed Import

Import to the staging area, where you can manually analyze, preview, and work with the metadata before you import it to the metadata repository.

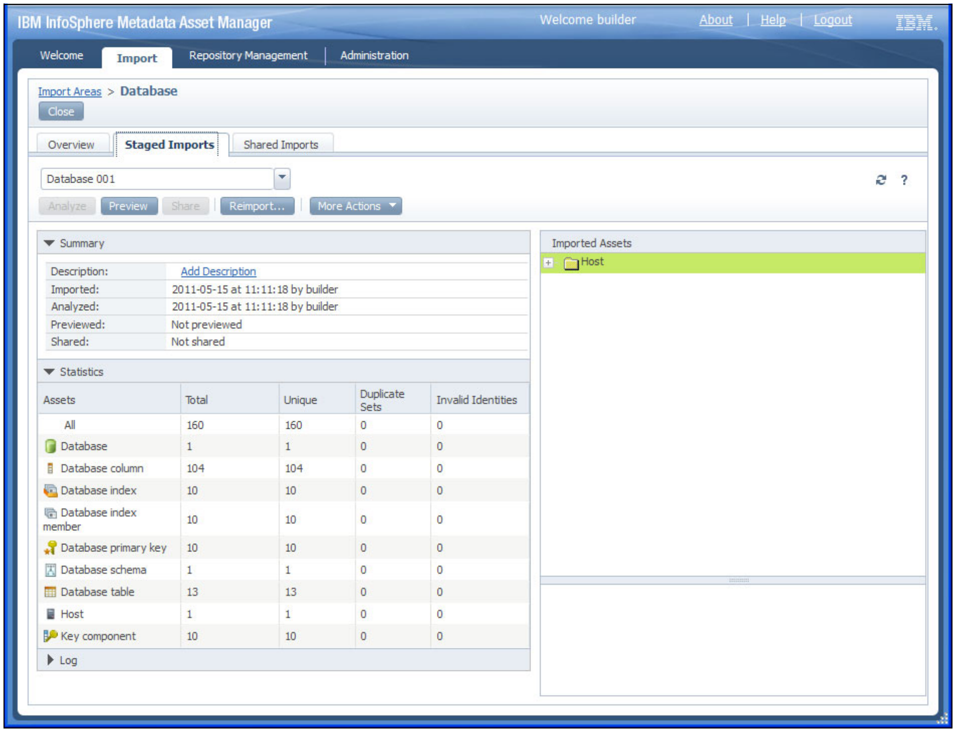
Figure 7-20   Completing the import of the database asset

7. Optional: If you selected **Managed Import**, publish the database asset to InfoSphere Information Server. By publishing the database asset, you can preview the asset to be imported, so that you can compare it with existing assets that will be merged or otherwise updated.

   a. Select and open the import area that contains the previously imported database table.

   b. Click the **Staged Imports** tab (Figure 7-21).

   c. Click **Preview** to analyze the database assets to be shared and the effect on the existing assets in InfoSphere Information Server.

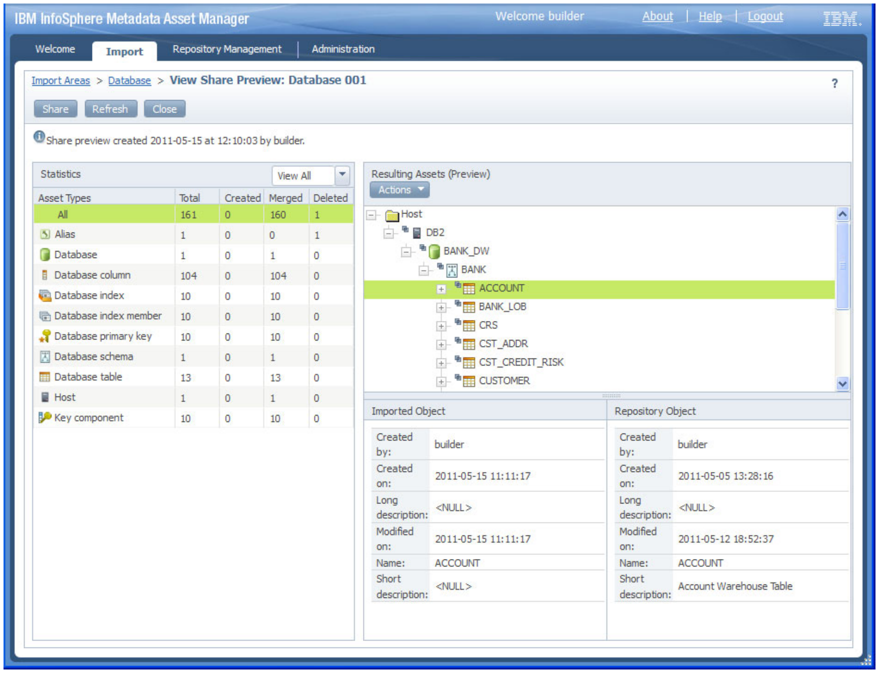


Description:	Add Description
Imported:	2011-05-15 at 11:11:18 by builder
Analyzed:	2011-05-15 at 11:11:18 by builder
Previewed:	Not previewed
Shared:	Not shared

Assets	Total	Unique	Duplicate Sets	Invalid Identities
All	160	160	0	0
Database	1	1	0	0
Database column	104	104	0	0
Database index	10	10	0	0
Database index member	10	10	0	0
Database primary key	10	10	0	0
Database schema	1	1	0	0
Database table	13	13	0	0
Host	1	1	0	0
Key component	10	10	0	0

Figure 7-21 Preview import area

As shown in Figure 7-22, you can preview the type and number of assets to be created, deleted, or merged upon sharing. Browse to select a particular database asset object, so that you can view the pending changes after it is shared.



Share preview created 2011-05-15 at 12:10:03 by builder.

Asset Types	Total	Created	Merged	Deleted
All	161	0	160	1
Alias	1	0	0	1
Database	1	0	1	0
Database column	104	0	104	0
Database index	10	0	10	0
Database index member	10	0	10	0
Database primary key	10	0	10	0
Database schema	1	0	1	0
Database table	13	0	13	0
Host	1	0	1	0
Key component	10	0	10	0

```
Host
  DB2
    BANK_DW
      BANK
        ACCOUNT
        BANK_LOB
        CRS
        CST_ADDR
        CST_CREDIT_RISK
        CUSTOMER
```

Imported Object	
Created by:	builder
Created on:	2011-05-15 11:11:17
Long description:	<NULL>
Modified on:	2011-05-15 11:11:17
Name:	ACCOUNT
Short description:	<NULL>

Repository Object	
Created by:	builder
Created on:	2011-05-05 13:28:16
Long description:	<NULL>
Modified on:	2011-05-12 18:52:37
Name:	ACCOUNT
Short description:	Account Warehouse Table

Figure 7-22 Sharing database assets with InfoSphere Information Server

d. Click **Share** to publish the database asset to InfoSphere Information Server, making the loaded database asset available to all components.

7.5.2 Results

Database assets represent the data storage systems that are defined in the integrated solution. You can browse and view information assets to understand their usage and meaning.

The following information assets are available in InfoSphere Information Server upon import of a data model:

Host system (▮) The computer system on which the database resides. A host system is a user-defined structure that is created during import.

Database (🛢) A structured collection of records or data that is stored on the host system.

Database schema (⊞) An organization or structure of the content in a database, often used for access control.

Database table (▦) The organization of information into a group that is structured by rows and columns. A table and its description are loaded from the source and contained in a schema.

Database view (▤) A virtual table, which dynamically defines and computes a subset of database tables. A view and underlying SQL syntax are loaded from the source and contained in a schema.

Database field (▯) A set of data values of a particular type for each row of the database table or database view. A field and its defined data type and length are loaded from the source and contained in a table or view.

You can search for, browse, and view the details of database assets from the components of InfoSphere Information Server, including InfoSphere Metadata Workbench.

7.6 Data extraction

The documentation and representation of the prescribed data flow between a given source and target asset are crucial to support the business and regulatory requirements. The documentation and representation of the prescribed data flow between a given source and target asset affect the overall development process and data quality assessment. This representation provides the generation of data lineage analysis reports and the capability to inspect the data flow from assets (such as BI reports) to the originating data source systems. It also provides the required auditing or governance. In addition, it enables the trust or quality that is associated with a report.

An integrated solution has a place where the extraction and load process for data between separate source systems and data storage systems occurs. As the development of the data integration initiative progresses, and when the final BI reports are brought online, you must enable the facility to trace the developed or runtime processes that extract and load the data.

The extraction and load process is a facility of InfoSphere DataStage and InfoSphere QualityStage, but you can extract and load data in an external script, procedure, or application. For example, a stored procedure can natively load a flat file into a database table, or an application can extract data from a data center, generating a load file.

InfoSphere Metadata Workbench delivers and administers the data lineage. However, the data lineage is built upon the documented and loaded data source and data storage systems.

It is possible that a data flow does not result from InfoSphere DataStage and InfoSphere QualityStage. In this case, you must document the data flow in InfoSphere Metadata Workbench as an extension mapping document or in IBM InfoSphere FastTrack as a mapping specification.

The extension mapping documents of InfoSphere Metadata Workbench are a prescribed mapping between a source and target asset, such as database columns. Additionally, this mapping can include a business rule that defined the transformation, the function for implementing the transformation, or a description of the transformation or aggregation action. Furthermore, with these mappings, you can include custom properties to document and track the specific requirements, for example, organization, run time, or author.

In the bank scenario for this book, we create an extension mapping document in InfoSphere Metadata Workbench to document the flow of data from the parameters of the source application and the subsequent load data file that is generated. Figure 7-23 illustrates the process flow.

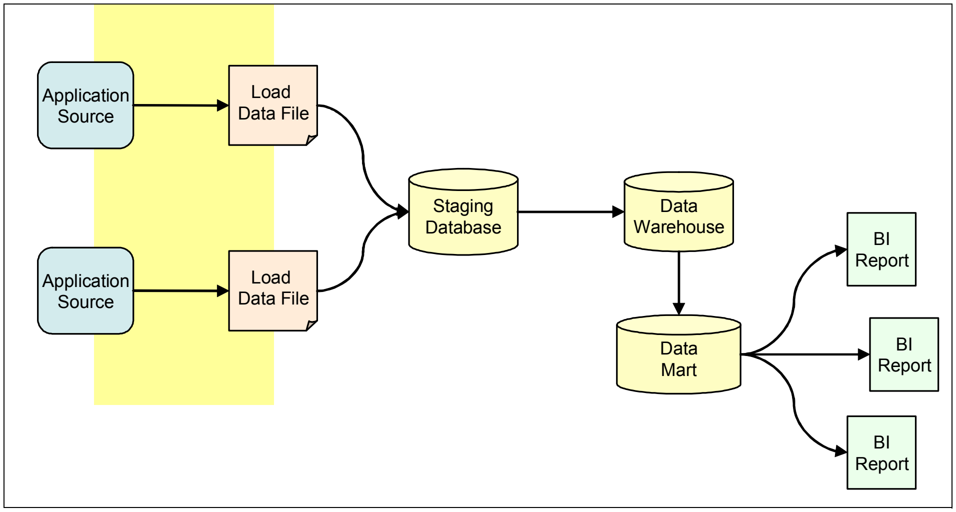


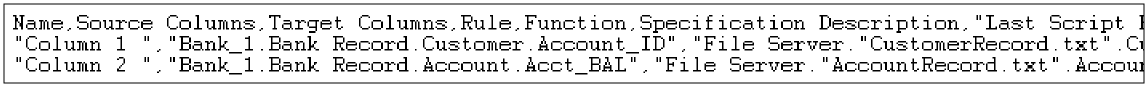
Figure 7-23 Data extraction from data source assets

7.6.1 Input file format

You can either create the extended mapping documents in InfoSphere Metadata Workbench or define them in an input file, which can be loaded into InfoSphere Metadata Workbench. Each mapping document relates or describes a single process or flow segment, rather than an entire data flow diagram. In this scenario, the mapping document describes the specific extraction of the data source representing Bank A, rather than the complete flow starting from the report.

Mapping documents contain individual mappings, which represent the extraction of data from a source and the transforming or loading of the data to a target. The source and target do not need to be the same asset type. However, it is preferred that they reference the same database, file, or asset level. For example, map a database column to a database field or application parameter, rather than mapping a database column to a database table.

The input file (Figure 7-24) is a CSV file, delineating the individual mappings of a mapping document. Each mapping includes a name, source assets, target assets, rule, function, description, and any available custom properties. The input file and ensuing import process are meant to fully document the extraction of data from the source system and the loading of that data to the target system. You can import multiple input files into InfoSphere Metadata Workbench.



```
Name,Source Columns,Target Columns,Rule,Function,Specification Description,"Last Script
"Column 1 ","Bank_1.Bank Record.Customer.Account_ID","File Server."CustomerRecord.txt".C
"Column 2 ","Bank_1.Bank Record.Account.Acct_BAL","File Server."AccountRecord.txt".Accou
```

Figure 7-24 Input file representing an extension mapping document

The following characteristics describe the input file:

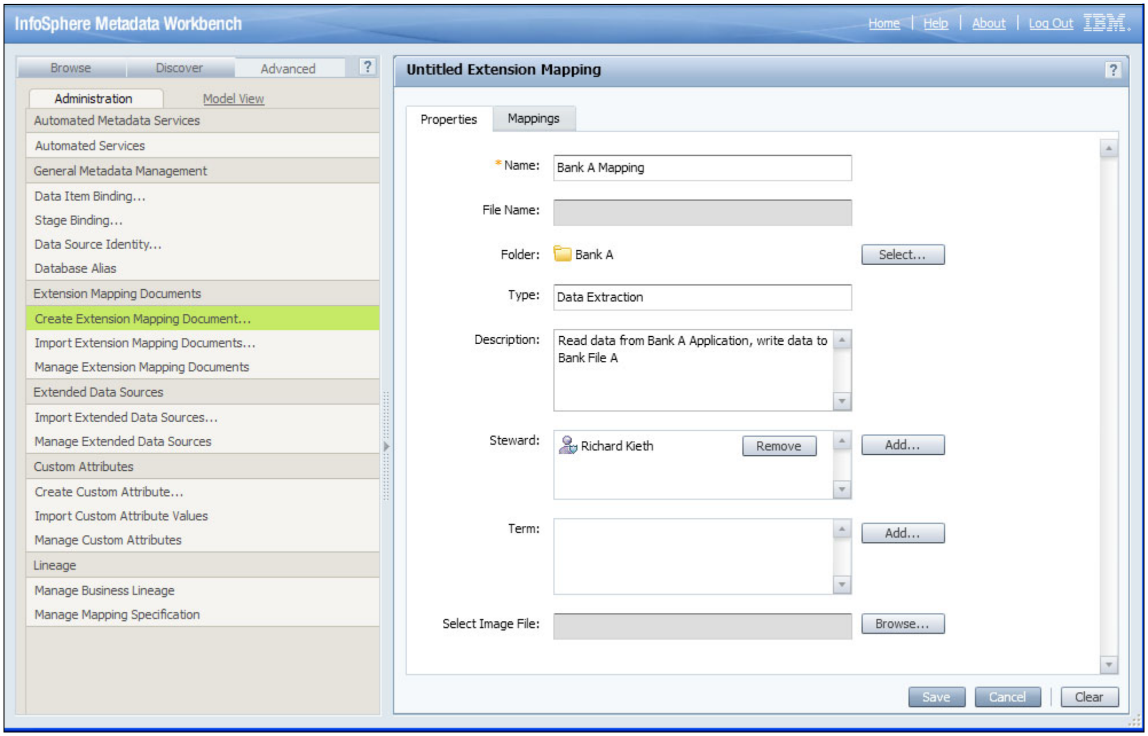
- The input file represents a spreadsheet that contains rows and columns. The rows represent individual point-to-point mapping, and the columns represent the properties of the mapping.
- The column header is required and must include source columns and target columns.
- The input file supports these asset types as valid source or target columns:
  - Database, database schema, database table, and database column
  - Data file, data file structure, and data file field
  - BI report, BI report field, BI model collection, and BI collection member
  - Extension application, stored procedure, and file
  - InfoSphere DataStage job, stage, and stage column
- Each row represents an individual mapping, for example, source to target column mapping.

7.6.2 Documenting the data extraction

In the bank scenario, we create an extension mapping document from within InfoSphere Metadata Workbench that demonstrates the extraction of data from the application data source for Bank A to populating the data file for Bank A.

To document the data extraction, complete the following steps:

1. Log on to InfoSphere Metadata Workbench as an InfoSphere Metadata Workbench Administrator user.
2. Click **Advanced** from the left navigation pane and select **Create Extension Mapping Document**.
3. In the Untitled Extension Mapping window (Figure 7-25 on page 204), import the extended data sources:
   a. On the **Properties** tab, complete these steps:
      i. Enter a name for the mapping document to be created. The name must reflect and indicate the process, script, or application that the mapping depicts, so that stakeholders can identify its purpose or usage.
      ii. Browse for and select a folder to contain the created mapping document. Folders represent the containers for mapping documents, so that the documents can be grouped and managed easily according to the project or commonality. If necessary, create a folder for the specific data integration project that the mapping document supports.
      iii. Optional: Enter a type for the mapping document. The mapping type describes the general purpose or action, for example, merge or transform.
      iv. Optional: Enter a description for the mapping document. Descriptions help to document the intent and design of the mapping for stakeholders, in particular, relating additional information about the procedure, script, or application that the document represents.
      v. Optional: Browse for and select a data steward to associate with the mapping. *Data stewards* are the technical owners or subject matter experts, with in-depth knowledge of the mapping.
      vi. Optional: Browse for and select a business term to associate with the mapping. *Business terms* extend the business definitions and requirements to technical assets, so that stakeholders can impart a greater understanding of their expected usage, structure, and content.
      vii. Optional: Browse for and select a static image that represents the process flow or specification, so that stakeholders can have a visual cue of understanding of the intended data flow process.


InfoSphere Metadata Workbench
Home | Help | About | Log Out
Browse
Discover
Advanced
Administration
Model View
Automated Metadata Services
Automated Services
General Metadata Management
Data Item Binding...
Stage Binding...
Data Source Identity...
Database Alias
Extension Mapping Documents
Create Extension Mapping Document...
Import Extension Mapping Documents...
Manage Extension Mapping Documents
Extended Data Sources
Import Extended Data Sources...
Manage Extended Data Sources
Custom Attributes
Create Custom Attribute...
Import Custom Attribute Values
Manage Custom Attributes
Lineage
Manage Business Lineage
Manage Mapping Specification
Untitled Extension Mapping
Properties
Mappings
Select...
Remove
Add...
Add...
Browse...
Save
Cancel
Clear

*Name: Bank A Mapping

File Name:

Folder: Bank A

Type: Data Extraction

Description: Read data from Bank A Application, write data to Bank File A

Steward: Richard Kieth

Term:

Select Image File:

Figure 7-25 Create extension mapping document

b. On the **Mappings** tab (Figure 7-26 on page 205), define the individual point-to-point mapping that is in the document to represent the extraction of data from the source system and its transform or load into the target system:

   i. Select the **Name** column, and optionally, enter a name for the mapping. The name identifies the specific transaction of reading from and writing to the specified source and target asset, so that stakeholders can further relate to the mappings. The name must be succinct and informative, for example, `Data_Merge` rather than `Row_1`.

   ii. Select the **Source** column, and then click **Add** to browse for and select an asset. In the Asset selection dialog box, select the asset type of the source column, and enter a partial name on which to search. From the list of results, select the asset to include in the mapping as the source column.

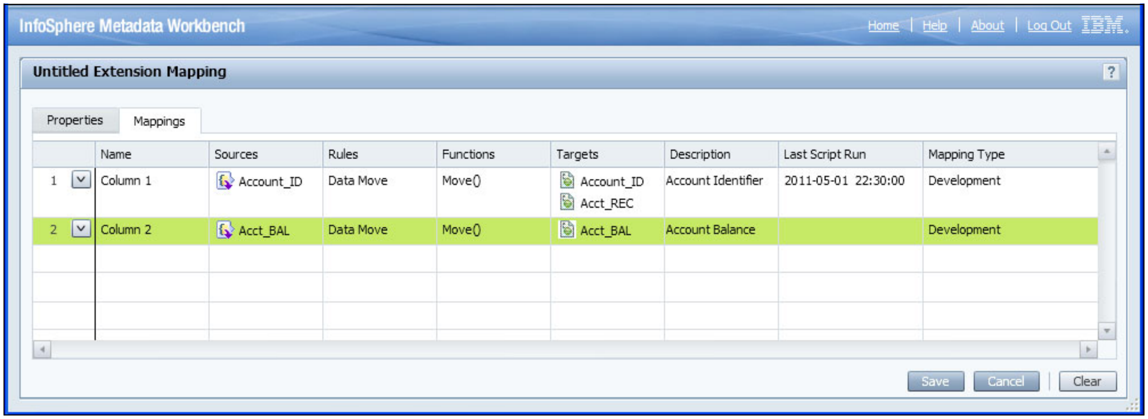
   iii. Select the **Rule** column, and optionally, enter a defined business rule or logic that defines the transform of data between the source and target.

iv. Select the **Function** column, and optionally, enter a function or process definition that structures the transform of data between the source and target.

v. Select the **Target** column, and then click **Add** to browse for and select an asset. In the Asset selection dialog box, select the asset type of the target column, and enter a partial name on which to search. From the list of results, select the asset to include in the mapping as the target column.

vi. Select the **Description** column, and optionally, enter a description that documents the transform of data between the source to target, so that stakeholders can search upon and understand the mapping in greater detail.

vii. Select any of the available custom properties, and optionally, enter the desired value to represent them.


InfoSphere Metadata Workbench
Home | Help | About | Log Out
IBM.
Untitled Extension Mapping
?
Properties
Mappings
Save
Cancel
Clear

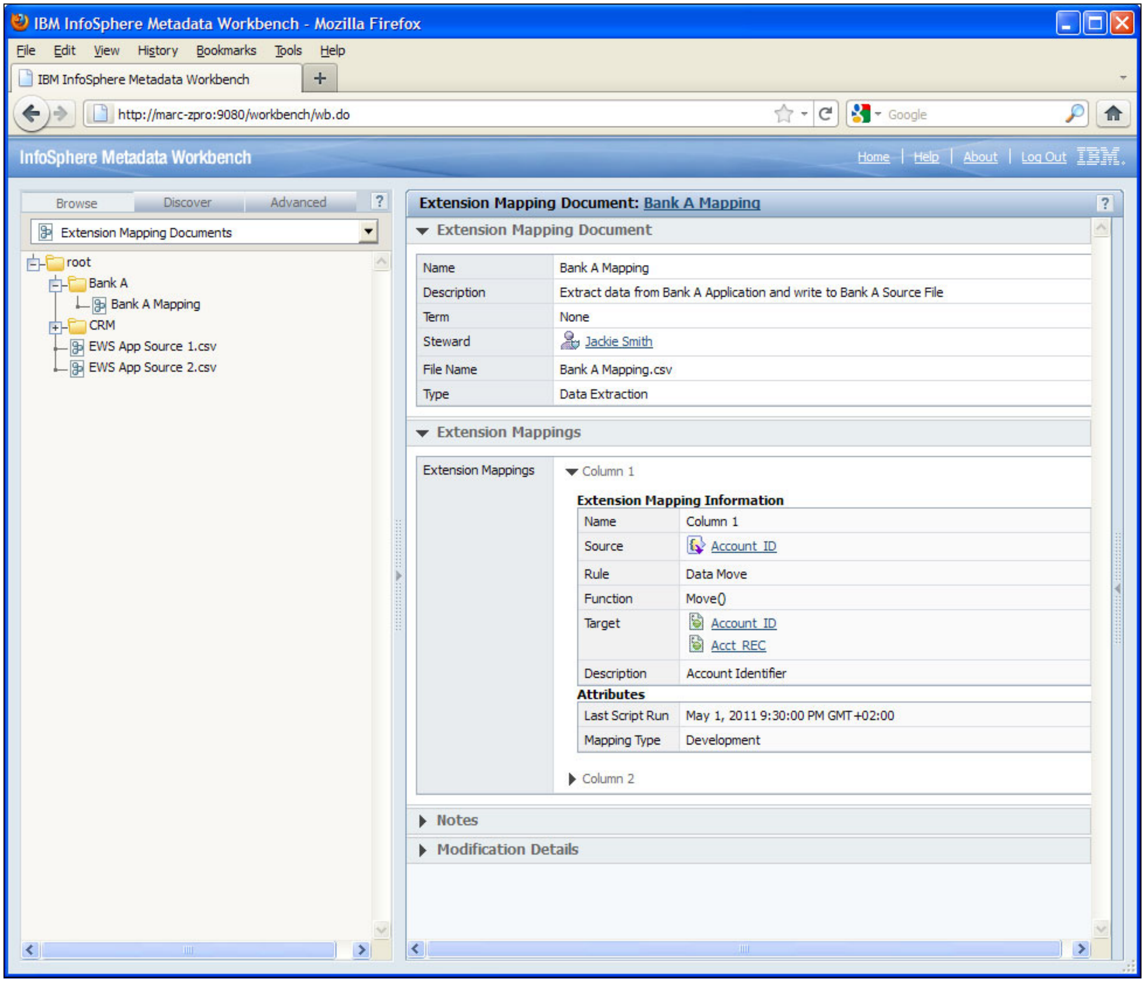
	Name	Sources	Rules	Functions	Targets	Description	Last Script Run	Mapping Type
1	Column 1	Account_ID	Data Move	Move()	Account_ID Acct_REC	Account Identifier	2011-05-01 22:30:00	Development
2	Column 2	Acct_BAL	Data Move	Move()	Acct_BAL	Account Balance		Development

Figure 7-26 Import extended data source completed successfully

c. Click **Save** when finished to complete the process.

7.6.3 Results

Extension mapping documents (Figure 7-27) represent the defined mapping between a source and a target asset. They are necessary to provide support for the required data flow and lineage reporting requirements.


IBM InfoSphere Metadata Workbench - Mozilla Firefox
File Edit View History Bookmarks Tools Help
IBM InfoSphere Metadata Workbench
http://marc-zpro:9080/workbench/wb.do
Google
InfoSphere Metadata Workbench
Home | Help | About | Log Out IBM.
Browse Discover Advanced ?
Extension Mapping Documents
root
Bank A
Bank A Mapping
CRM
EWS App Source 1.csv
EWS App Source 2.csv

Extension Mapping Document: Bank A Mapping

Extension Mapping Document

Name	Bank A Mapping
Description	Extract data from Bank A Application and write to Bank A Source File
Term	None
Steward	Jackie Smith
File Name	Bank A Mapping.csv
Type	Data Extraction

Extension Mappings

Extension Mappings

Column 1

Extension Mapping Information

Name	Column 1
Source	Account_ID
Rule	Data Move
Function	Move()
Target	Account_ID Acct_REC
Description	Account Identifier

Attributes

Last Script Run	May 1, 2011 9:30:00 PM GMT+02:00
Mapping Type	Development

Column 2

Notes

Modification Details

Figure 7-27 Extension Mapping Document in InfoSphere Metadata Workbench

7.7 Conclusion

In conclusion, this chapter explained how to identify and document source data systems. It also described how to load them into the InfoSphere Information Server metadata repository. This chapter included a scenario in which you document data extraction.

To fully understand source data, Chapter 8, “Data relationship discovery” on page 209, addresses how to use IBM InfoSphere Discovery to discover data attributes and relationships among the data.

To ensure source data quality, Chapter 9, “Data quality assessment and monitoring” on page 283, explains how to use IBM InfoSphere Information Analyzer to assess and monitor source data quality.

8

Data relationship discovery

Really understanding information assets that are used in an information integrated solution is an important factor in metadata management. This chapter focuses on data relationship discovery, which is a technique that organizations can use to gain a deep understanding of their information assets.

Recent technological advancements have paved the way for modern data relationship discovery. A new breed of data analysis software interrogates structured data sources and uses statistical analysis to infer important business relationships. These business relationships, which span both physical and logical domains, provide an essential view of the *business objects* of an organization. The most mature data relationship discovery solution in the market is IBM InfoSphere Discovery, which is one of the IBM InfoSphere Information Server product modules.

This chapter describes InfoSphere Discovery and its usage scenarios. It includes the following sections:

- Introduction to InfoSphere Discovery
- Creating a project
- Performing column analysis
- Identifying and classifying sensitive data
- Assigning InfoSphere Business Glossary terms to physical assets
- Reverse engineering a data model
- Performing value overlap analysis
- Discovering transformation logic
- Conclusion

8.1 Introduction to InfoSphere Discovery

Each year, organizations tend to store more data. Over time, it becomes harder to understand and to manage. For example, if Bank A buys Bank B, we can assume that Bank A will suddenly have more customers, more transactions, more everything.

As Figure 8-1 illustrates, you cannot manage what you do not understand. When data proliferates, it becomes harder to understand and manage, both within and across systems. Much of the data is connected, but how? Out-of-date documentation, complex data relationships, inadequate corporate memory all impede your ability to enforce strong governance measures.

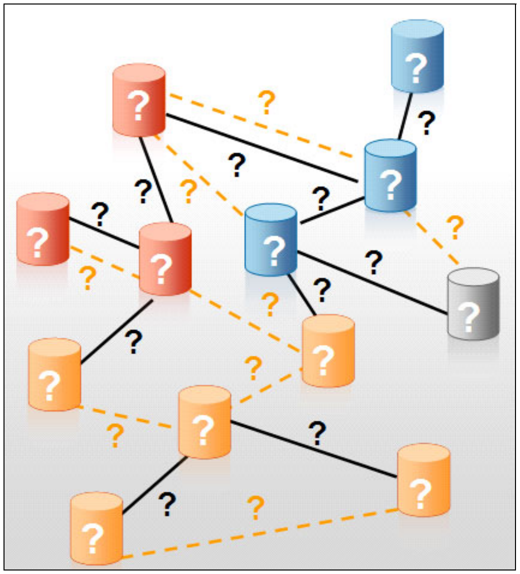


Figure 8-1 Visual representation of our inability to manage what we do not understand

InfoSphere Discovery provides a powerful solution for automated data relationship discovery. InfoSphere Discovery comes with a full range of capabilities to identify many data relationships. The automated results derived from InfoSphere Discovery are actionable, accurate, and easy to generate, especially when compared to manual data analysis approaches that many organizations still use today. One common manual approach involves handwriting SQL queries and storing results in spreadsheets. Does this practice sound familiar to you?

Depending on the type of project or stage of a project, different discovery might be more useful than others. In the example of Bank A buying Bank B, we have the following potential use cases, among others, for InfoSphere Discovery:

- Data model discovery based on primary-foreign key pairs and logical relationships
- Sensitive data discovery
- InfoSphere Business Glossary term mapping and refinement
- Advanced unified schema prototyping for data consolidation projects
- Single-source and cross-source data redundancy analysis
- Transformation discovery to reverse-engineer complex business rules used in the data integration processes

InfoSphere Discovery works by *automatically* analyzing data sources and generating hypotheses about the data. Throughout the process, it interrogates the data and generates metadata that includes a *data profile* or *column analysis*. A data analyst or subject matter expert (SME) interprets the results, leading to further analysis or other action.

Imagine that after buying Bank B, Bank A initiates a Master Data Management (MDM) initiative to create a consistent view of its customers. Bank A acquires an MDM solution, but is it ready to start moving data? The answer is no, not until the employees of Bank A learn more about the customer data, such as where its customer data is located, the number of customers that are duplicated, and which systems are more trusted.

Another important factor is organizational memory and the effects when knowledge workers come and go. When Bank A buys Bank B, some employees will change roles, and others might not have jobs anymore. What happened to the information they knew? Did they document it well and maintain it over time? Hopefully, they did, but it is unlikely and rather rare.

8.1.1 Planning equals saving

Always include data relationship discovery and data quality in your project plans when you initiate an information initiative. The earlier you start automating these steps, the more benefits you will obtain.

Figure 8-2 on page 212 illustrates a common problem with manual data analysis. Without the power and predictability of an automated solution, you cannot easily estimate the length of the discovery phase of a project. Even when you have an estimate, manual analysis is more unpredictable than automated analysis, often leading to project plan changes later.

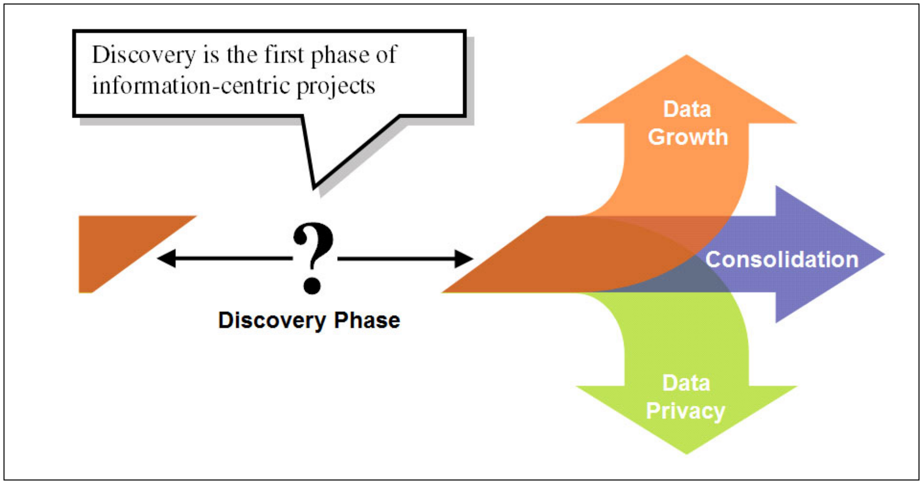
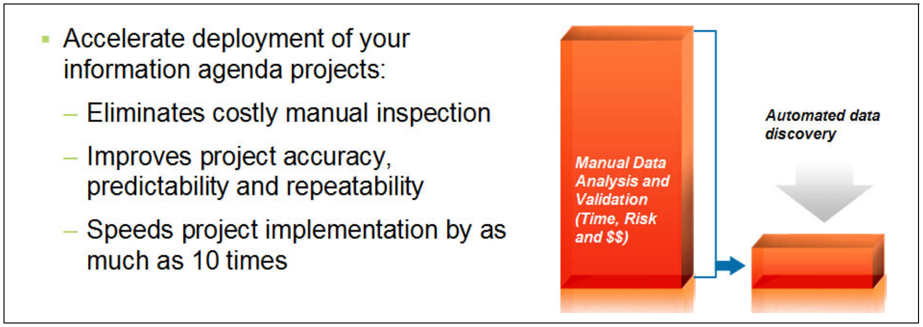

Discovery is the first phase of information-centric projects
?
Discovery Phase
Data Growth
Consolidation
Data Privacy

Figure 8-2 InfoSphere Discovery accelerates data analysis

You can accelerate the completion of your data analysis tasks up to ten times when you use InfoSphere Discovery. You can get to the next phase of a project faster, and with more reliable results, when you understand your data and the underlying relationships within and across systems.

Figure 8-3 graphically illustrates how much time and effort you can save with InfoSphere Discovery. The savings can be dramatic.



- Accelerate deployment of your information agenda projects:
  - Eliminates costly manual inspection
  - Improves project accuracy, predictability and repeatability
  - Speeds project implementation by as much as 10 times

Figure 8-3 Time and cost savings, in addition to reduce risk, with InfoSphere Discovery

Time savings lead to cost savings, both of which are hard and soft costs. The hard costs include such items as labor savings. The soft costs include items that are harder to quantify, but that are still real. How much value does Bank A realize by harmonizing its customer data weeks or months faster because of automated discovery?

While one group at Bank A initiates its customer harmonization program, a different group decides to build a business glossary with IBM InfoSphere Business Glossary. The glossary will help the bank manage business terms, so that employees across the enterprise consistently use the same terms with the same definitions. Bank A can build an initial glossary without knowing much about its data assets.

At first, the glossary might contain terms that have not been mapped to the physical assets of the bank. For example, the bank might define a business term, such as customer number, without knowing where the customer number is stored in its databases. The bank will obtain more value by mapping its business terms to its physical data assets, providing a direct link between business assets and data assets.

InfoSphere Discovery provides an automated way to understand your data. The result is a rich data landscape that describes the relationships within and across tables, databases, and systems.

After gaining an understanding of your data and the often-hidden relationships within it, you assess data quality and enforce policies to monitor and validate data quality. For more information, see Chapter 9, “Data quality assessment and monitoring” on page 283.

8.1.2 A step-by-step discovery guide

Bank A buys Bank B. That event triggers changes at every level of the new organization. The bank makes legal, marketing, and administrative changes. Information technology (IT) is no exception. When you log in to see your checking account online, when you receive a monthly statement, or when you pay a bill, the underlying data must be accurate. If the data is wrong, the bank will likely face many challenges, from legal battles, to penalties, to lost customers.

The remaining sections in this chapter highlight the step-by-step process, which entails the following tasks, to analyze a small banking system:

1. Creating a project
2. Performing column analysis
3. Identifying and classifying sensitive data
4. Assigning InfoSphere Business Glossary terms to physical assets
5. Reverse engineering a data model
6. Performing value overlap analysis
7. Discovering transformation logic

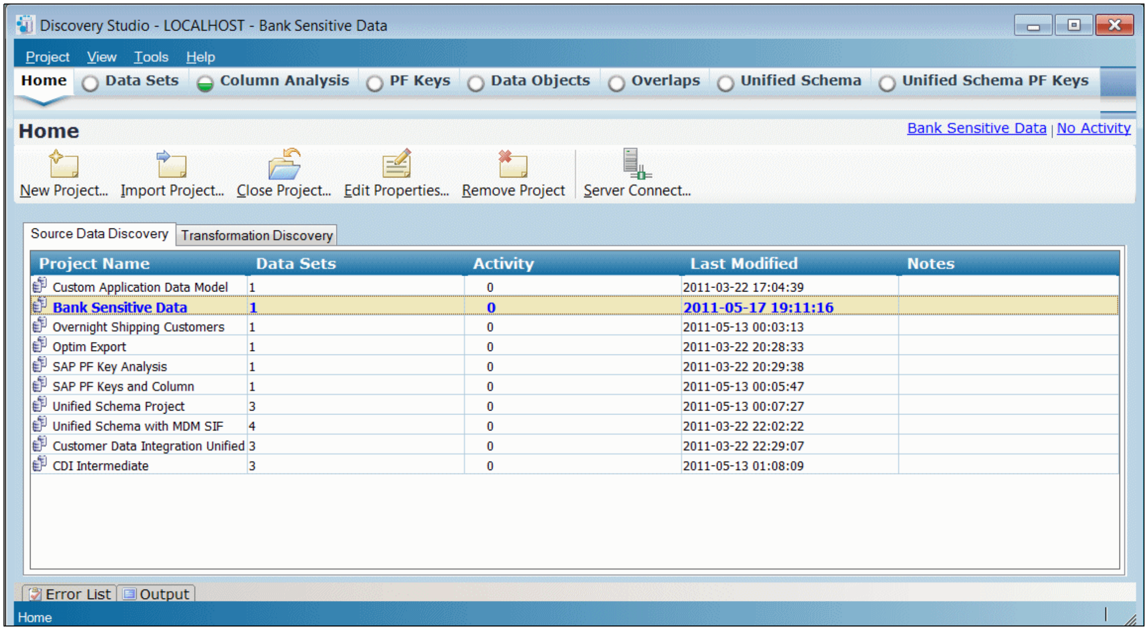
Your goal is to automate as much of the data analysis effort as possible and to gain a thorough understanding of your data assets. The first step is to create a project.

8.2 Creating a project

To create a project, complete these steps:

1. Open the InfoSphere Discovery Studio application.

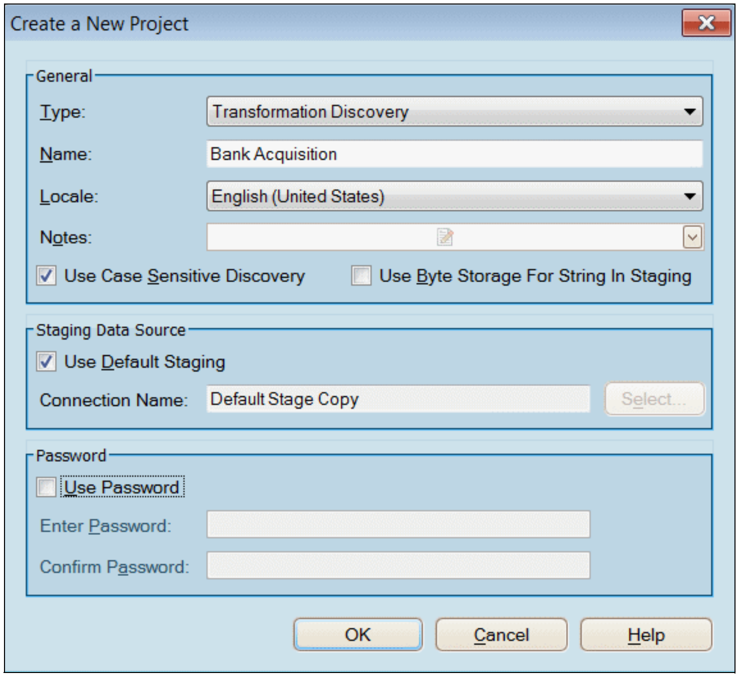
   You see the **Home** tab of the Discovery Studio application similar to the one shown in Figure 8-4, although it might not have any existing projects. The user interface is divided into several tabs that define a workflow. Typically, you move from left to right. For example, you create a project on the **Home** tab, then you define the data sets in the **Data Sets** tab, and so on.


Discovery Studio - LOCALHOST - Bank Sensitive Data
Project View Tools Help
Home Data Sets Column Analysis PF Keys Data Objects Overlaps Unified Schema Unified Schema PF Keys
Home
Bank Sensitive Data | No Activity
New Project... Import Project... Close Project... Edit Properties... Remove Project Server Connect...
Source Data Discovery Transformation Discovery
Error List Output
Home

Project Name	Data Sets	Activity	Last Modified	Notes
Custom Application Data Model	1	0	2011-03-22 17:04:39	
Bank Sensitive Data	**1**	**0**	**2011-05-17 19:11:16**	
Overnight Shipping Customers	1	0	2011-05-13 00:03:13	
Optim Export	1	0	2011-03-22 20:28:33	
SAP PF Key Analysis	1	0	2011-03-22 20:29:38	
SAP PF Keys and Column	1	0	2011-05-13 00:05:47	
Unified Schema Project	3	0	2011-05-13 00:07:27	
Unified Schema with MDM SIF	4	0	2011-03-22 22:02:22	
Customer Data Integration Unified	3	0	2011-03-22 22:29:07	
CDI Intermediate	3	0	2011-05-13 01:08:09	

Figure 8-4 List of InfoSphere Discovery projects

2. Click the **New Project** button in the workflow bar.
3. In the Create a New Project dialog box (Figure 8-5), complete these steps:
   a. For Type, select **Transformation Discovery**.
   b. For the name, type a project name, such as `Bank Acquisition`.
   c. At the bottom of the General group box, clear the **Use Byte Storage For String in Staging** check box.
   d. In the Password group box, clear the **Use Password** check box.
   e. Click **OK**.



Create a New Project

General

Type: Transformation Discovery

Name: Bank Acquisition

Locale: English (United States)

Notes:

☑ Use Case Sensitive Discovery ☐ Use Byte Storage For String In Staging

Staging Data Source

☑ Use Default Staging

Connection Name: Default Stage Copy

Password

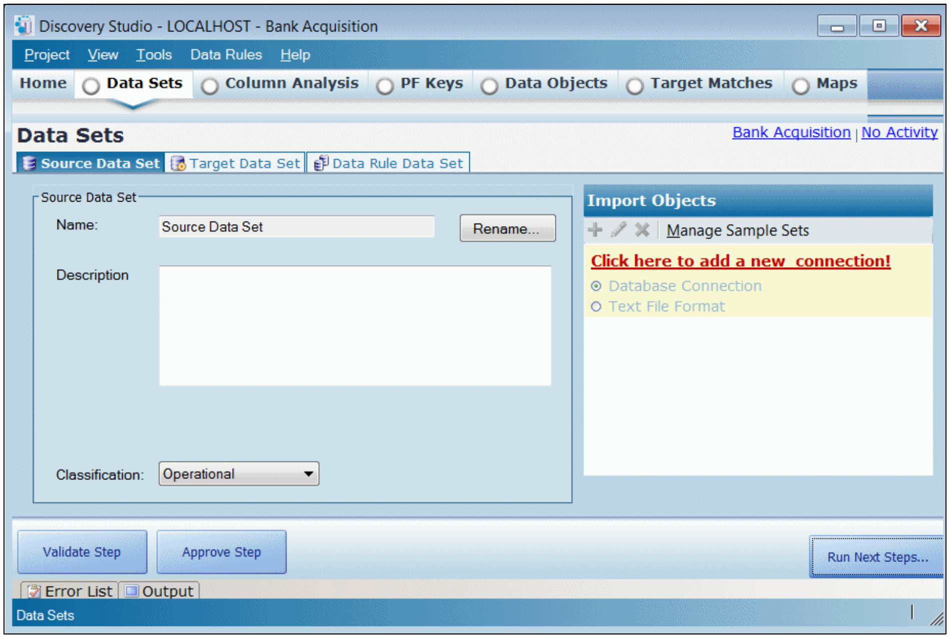
☐ Use Password

Enter Password:

Confirm Password:

Figure 8-5 Creating a project with InfoSphere Discovery

You automatically move to the **Data Sets** tab (Figure 8-6).



Source Data Set

Name: Source Data Set

Description

Classification: Operational

● Database Connection

○ Text File Format

Figure 8-6 Data Sets area of InfoSphere Discovery

8.2.1 Pointing to the data requiring analysis

Next, point to the data you want InfoSphere Discovery to analyze. Whether you want to analyze just one table, or all of the tables in a schema, you must point to each table individually.

InfoSphere Discovery operates against structured data, which can be database tables or structured flat files. You use the radio button to choose whether to define a database connection or to import a flat file. The data can be from one homogenous database system or from several disparate systems.

Pointing to the data categorizes it into one or more data sets. A data set is like a bucket where we place data. In some situations, such as performing cross-system overlap analysis, you place each data source into a separate data set. In other situations, such as performing transformation discovery, you place the data into source and target data sets.

InfoSphere Discovery supports various database platforms and versions. You can find the most recent InfoSphere Discovery system requirements and platform support matrix at:

http://www.ibm.com/software/data/infosphere/discovery/requirements.html

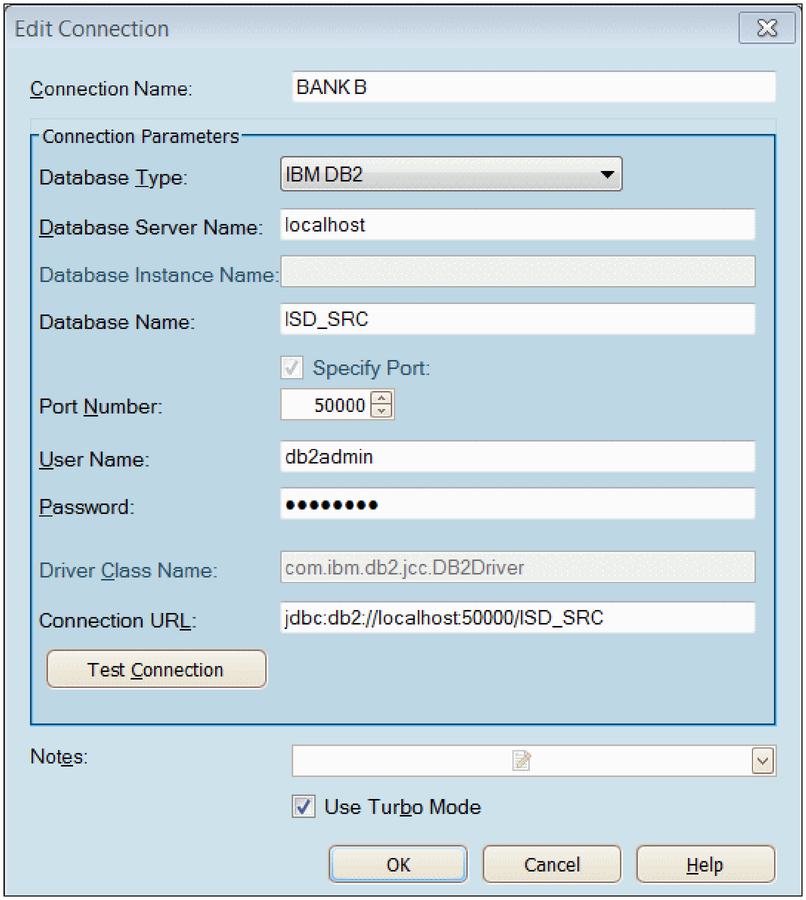
8.2.2 Importing the source data

To import the source data, complete these steps:

1. On the Data Sets page (Figure 8-6 on page 216), click the **Source Data Set** tab. Then in the Import Objects pane (right side), click the **Click here to add a new connection!** link.
2. In the Edit Connection window (Figure 8-7 on page 218), define the parameters for a JDBC connection to the data source:
   a. Complete the following fields:
      - Database Type
      - Database Server Name
      - Database Name
      - Port Number
      - User Name
      - Password
      - Connection URL

   b. Click the **Test Connection** button to ensure that the connection information is correct.

   c. Click **OK**. The Edit Connection window closes, and you see the **Data Sets** tab again.


Edit Connection

Connection Name: BANK B

Connection Parameters

Database Type: IBM DB2

Database Server Name: localhost

Database Instance Name:

Database Name: ISD_SRC

☑ Specify Port:

Port Number: 50000

User Name: db2admin

Password: ••••••••

Driver Class Name: com.ibm.db2.jcc.DB2Driver

Connection URL: jdbc:db2://localhost:50000/ISD_SRC

Test Connection

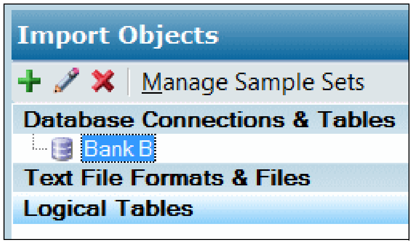
Notes:

☑ Use Turbo Mode

OK | Cancel | Help

Figure 8-7 Creating a data connection

3. Click the data connection to select it. In the Import Objects pane (Figure 8-8), click the green plus sign (**+**) above the data connection name.


Import Objects

Manage Sample Sets

Database Connections & Tables

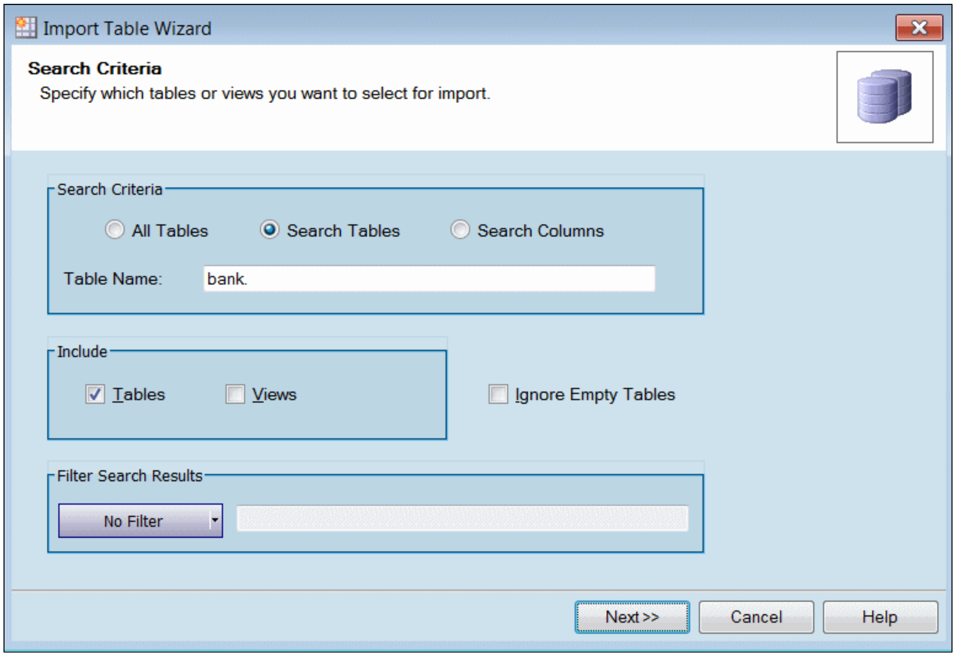
Bank B

Text File Formats & Files

Logical Tables

Figure 8-8 Bank B data connection

4. In the Import Table Wizard window—Search Criteria window (Figure 8-9), for Table Name, type `bank.` (with the period). Then click **Next**.



Import Table Wizard

Search Criteria

Specify which tables or views you want to select for import.

Search Criteria

○ All Tables ● Search Tables ○ Search Columns

Table Name: bank.

Include

☑ Tables □ Views

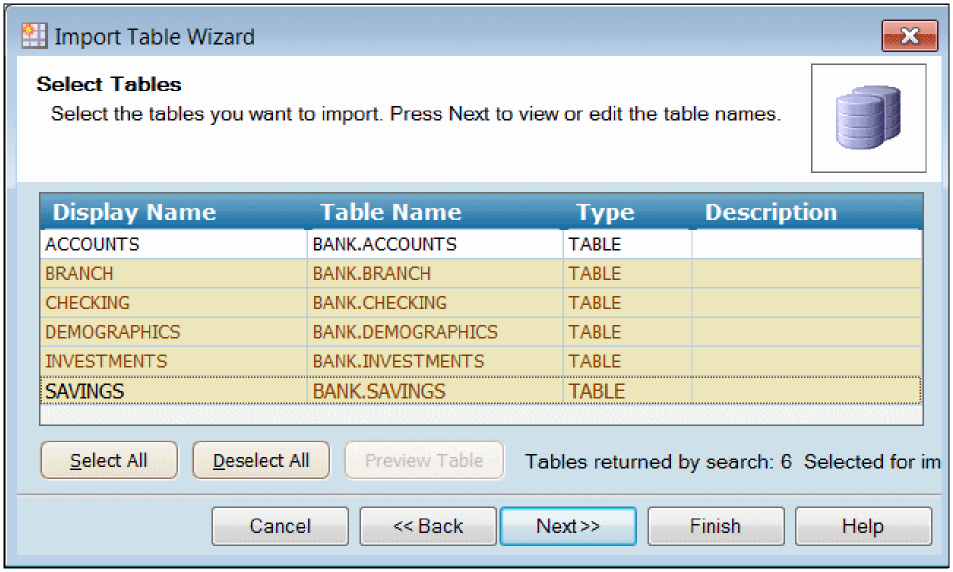
□ Ignore Empty Tables

Filter Search Results

No Filter

Figure 8-9 Import table wizard

5. In the list of tables to select (Figure 8-10) from the bank schema, press and hold down the CTRL key while selecting all of the tables except for ACCOUNTS. Then click **Finish**.



Import Table Wizard

Select Tables

Select the tables you want to import. Press Next to view or edit the table names.

Display Name	Table Name	Type	Description
ACCOUNTS	BANK.ACCOUNTS	TABLE	
BRANCH	BANK.BRANCH	TABLE	
CHECKING	BANK.CHECKING	TABLE	
DEMOGRAPHICS	BANK.DEMOGRAPHICS	TABLE	
INVESTMENTS	BANK.INVESTMENTS	TABLE	
SAVINGS	BANK.SAVINGS	TABLE	

Tables returned by search: 6 Selected for im

Figure 8-10 Selecting tables to import

Figure 8-11 shows the imported table definitions in the Import Objects pane.

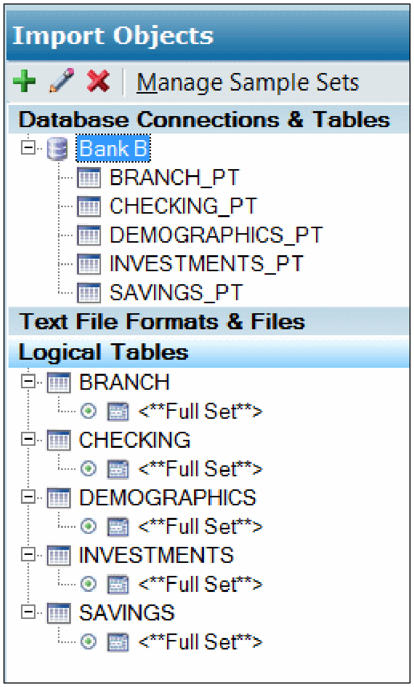


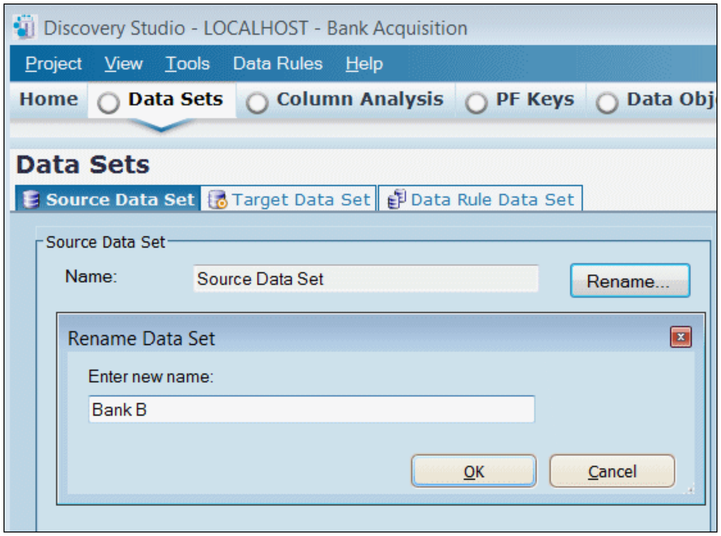
Figure 8-11 List of imported tables

When you import data, InfoSphere Discovery creates a *logical table* for each table and file you import. A logical table functions similar to a database view. You can add or remove columns, add a WHERE clause, or join the logical table to other tables, without modifying the original source table. Essentially, you can create views of the source data (tables and files) without defining them ahead of time and without obtaining support from a DBA.

You can preview data in a logical table by using the **Preview Data** button. This button is helpful, especially when you want to see how a WHERE clause or a join condition influences the data the logical table retrieves. With logical tables, you can make adjustments in real time.

You can also rename a data set:

1. On the Source Data Set tab (Figure 8-12), click the **Rename** button.
2. In the Rename Data Set dialog box (inset in Figure 8-12), for the name, type `Bank B`, and then click **OK**.



Name: Source Data Set

Enter new name: Bank B

Figure 8-12 Renaming a data set

8.2.3 Importing the target data

Target data consists of data from databases and data from files.

Importing target data from databases

To import the target data, complete these steps:

1. Click the **Target Data Set** tab. Then in the Import Objects pane (right side), click the **Click here to add a new connection!** link. Create a connection to the same Bank B database that we analyzed earlier in this chapter.
2. In the Import Table Wizard—Search Criteria window (Figure 8-9), search for `bank.` (including the period), and select the **ACCOUNTS** table. Click **OK** to import it. Then rename the target data set to `Bank A`.
3. Select **Project** → **Save** to save your project.

Importing data from files

For this project, you do not need to analyze data that is stored in structured files. However, the steps are provided as follows as a reference if you need to import data from files by using the *file import wizard* in InfoSphere Discovery:

1. On the **Data Sets** tab, select the data set where you want to import a file. In the Import Objects pane (Figure 8-13), select the **Text File Formats & Files** line.

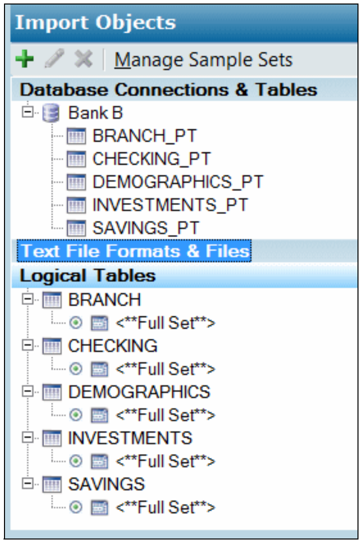
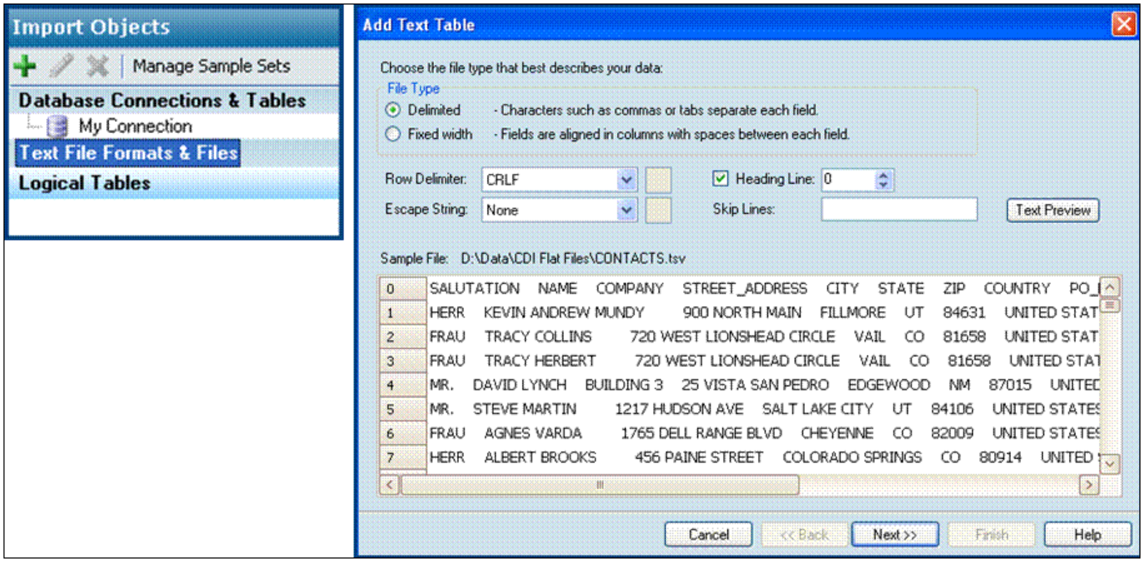


Figure 8-13 Text files and formats

2. Click the green **+** button. Browse to the first file you want to import, and open it.

3. In the Add Text Table wizard (Figure 8-14), select the appropriate options for Row Delimiter and Heading Line, and then click **Next**.



Choose the file type that best describes your data:

File Type

● Delimited - Characters such as commas or tabs separate each field.

○ Fixed width - Fields are aligned in columns with spaces between each field.

Row Delimiter: CRLF

Escape String: None

☑ Heading Line: 0

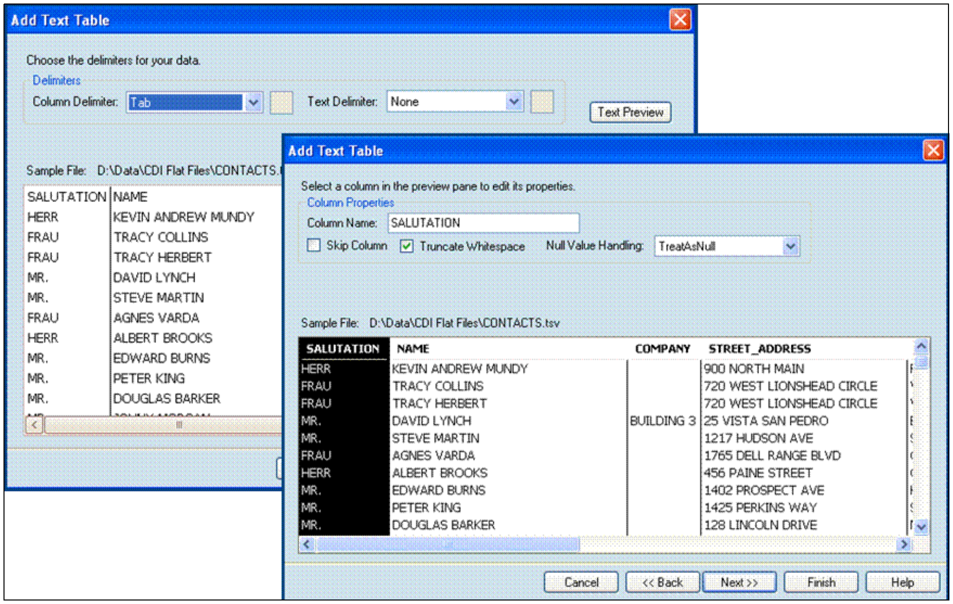
Skip Lines:

Sample File: D:\Data\CDI Flat Files\CONTACTS.tsv

0	SALUTATION	NAME	COMPANY	STREET_ADDRESS	CITY	STATE	ZIP	COUNTRY	PO_
1	HERR	KEVIN ANDREW MUNDY		900 NORTH MAIN	FILLMORE	UT	84631	UNITED STAT	
2	FRAU	TRACY COLLINS		720 WEST LIONSHEAD CIRCLE	VAIL	CO	81658	UNITED STAT	
3	FRAU	TRACY HERBERT		720 WEST LIONSHEAD CIRCLE	VAIL	CO	81658	UNITED STAT	
4	MR.	DAVID LYNCH	BUILDING 3	25 VISTA SAN PEDRO	EDGEWOOD	NM	87015	UNITEC	
5	MR.	STEVE MARTIN		1217 HUDSON AVE	SALT LAKE CITY	UT	84106	UNITED STATES	
6	FRAU	AGNES VARDA		1765 DELL RANGE BLVD	CHEYENNE	CO	82009	UNITED STATES	
7	HERR	ALBERT BROOKS		456 PAINE STREET	COLORADO SPRINGS	CO	80914	UNITED	

Figure 8-14 Choosing the file type

4. Select the Column Delimiter, such as comma, tab, or pipe (upper left window in Figure 8-15), and then you see the data align. Click **Next**.
5. Define the column properties (lower right window in Figure 8-15), and then click **Finish**.



Choose the delimiters for your data.

Delimiters

Column Delimiter: Tab

Text Delimiter: None

Sample File: D:\Data\CDI Flat Files\CONTACTS.

SALUTATION	NAME
HERR	KEVIN ANDREW MUNDY
FRAU	TRACY COLLINS
FRAU	TRACY HERBERT
MR.	DAVID LYNCH
MR.	STEVE MARTIN
FRAU	AGNES VARDA
HERR	ALBERT BROOKS
MR.	EDWARD BURNS
MR.	PETER KING
MR.	DOUGLAS BARKER

Select a column in the preview pane to edit its properties.

Column Properties

Column Name: SALUTATION

☐ Skip Column ☑ Truncate Whitespace Null Value Handling: TreatAsNull

Sample File: D:\Data\CDI Flat Files\CONTACTS.tsv

SALUTATION	NAME	COMPANY	STREET_ADDRESS
HERR	KEVIN ANDREW MUNDY		900 NORTH MAIN
FRAU	TRACY COLLINS		720 WEST LIONSHEAD CIRCLE
FRAU	TRACY HERBERT		720 WEST LIONSHEAD CIRCLE
MR.	DAVID LYNCH	BUILDING 3	25 VISTA SAN PEDRO
MR.	STEVE MARTIN		1217 HUDSON AVE
FRAU	AGNES VARDA		1765 DELL RANGE BLVD
HERR	ALBERT BROOKS		456 PAINE STREET
MR.	EDWARD BURNS		1402 PROSPECT AVE
MR.	PETER KING		1425 PERKINS WAY
MR.	DOUGLAS BARKER		128 LINCOLN DRIVE

Figure 8-15 Defining the column delimiter and column properties

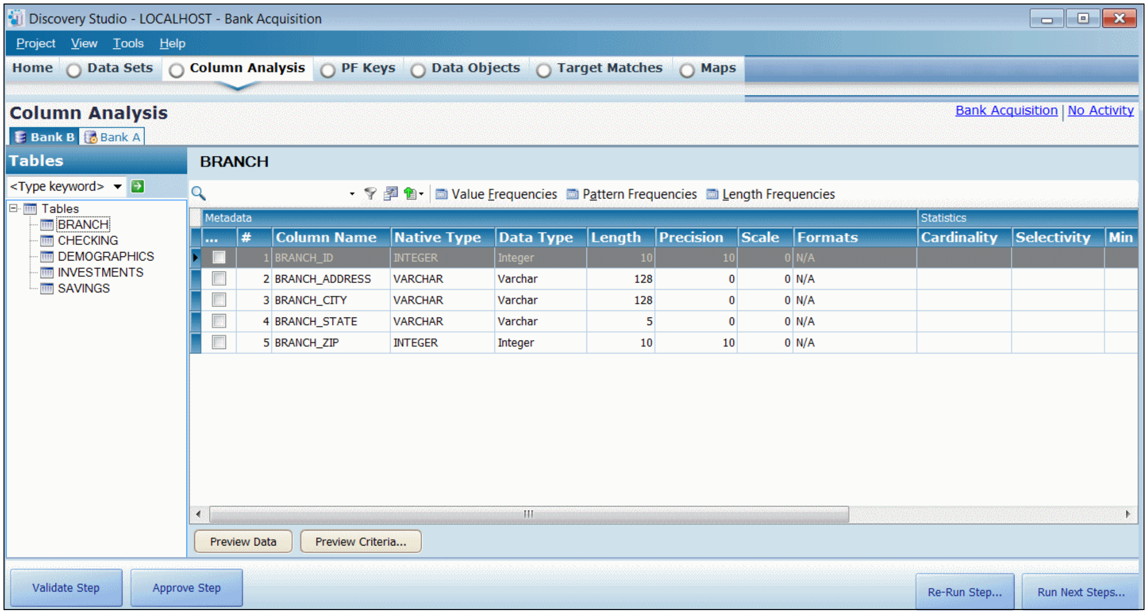
When you import data, InfoSphere Discovery does not import the data immediately. Instead, it reads the metadata to retrieve information about the columns and table or file structures. It does not analyze the data in depth until you complete the next task, *column analysis*. Now, the tables and files you imported are in the Data Sets area. InfoSphere Discovery creates a logical table, which is a database view, for each table and file that you import.

8.3 Performing column analysis

Looking at the workflow bar under the main menu, Column Analysis is the next step. Also known as *profiling*, column analysis provides an initial data inventory of useful metadata and statistical results for each physical asset that you analyze. *Column analysis* is an automated process that is easy to learn and perform. For each column in each table or file that you analyze, the column analysis results include statistics, such as column name, data type, length, minimum, maximum and mode values, cardinality, and selectivity. These results provide a solid data inventory that helps you to understand the data better, while providing a launching point for data relationship discovery.

To perform column analysis, complete these steps:

1. Click the **Column Analysis** tab (Figure 8-16).
2. In the lower right corner of the window, click **Run Next Steps**.



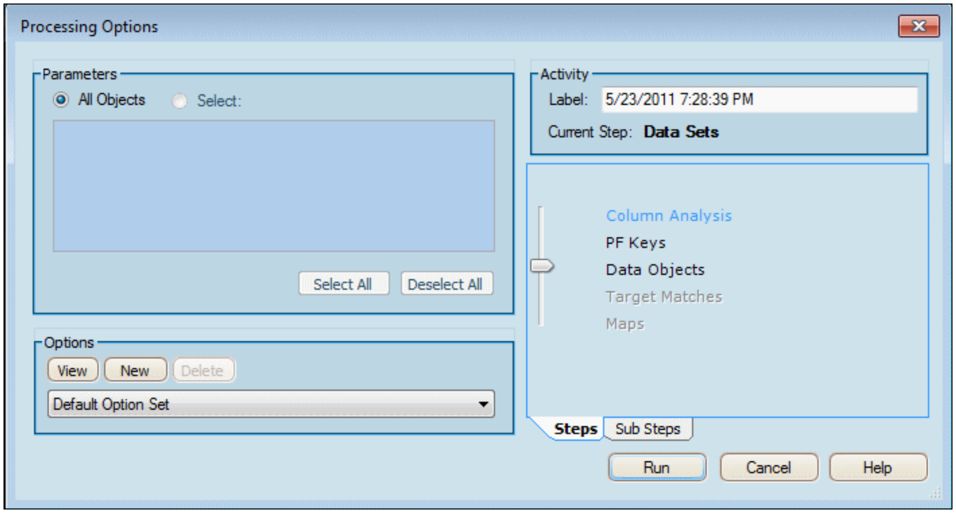
Metadata									Statistics		
...	#	Column Name	Native Type	Data Type	Length	Precision	Scale	Formats	Cardinality	Selectivity	Min
□	1	BRANCH_ID	INTEGER	Integer	10	10	0	N/A			
□	2	BRANCH_ADDRESS	VARCHAR	Varchar	128	0	0	N/A			
□	3	BRANCH_CITY	VARCHAR	Varchar	128	0	0	N/A			
□	4	BRANCH_STATE	VARCHAR	Varchar	5	0	0	N/A			
□	5	BRANCH_ZIP	INTEGER	Integer	10	10	0	N/A			

Figure 8-16 Column Analysis page and the Run Next Steps button

3. In the Processing Options pane (Figure 8-17), determine how many steps to run. Click and drag the slider on the right side to the **Data Objects** step.

   Before running the steps, you can set substeps, such as determining whether to perform Data Type Discovery or Column Classification when you run column analysis. Most steps have substeps that help you refine the discovery process. Sensitive data might exist in the data, but you do not know exactly what or where.

   Click the **Sub Steps** tab.



● All Objects ○ Select:

Default Option Set

Label: 5/23/2011 7:28:39 PM

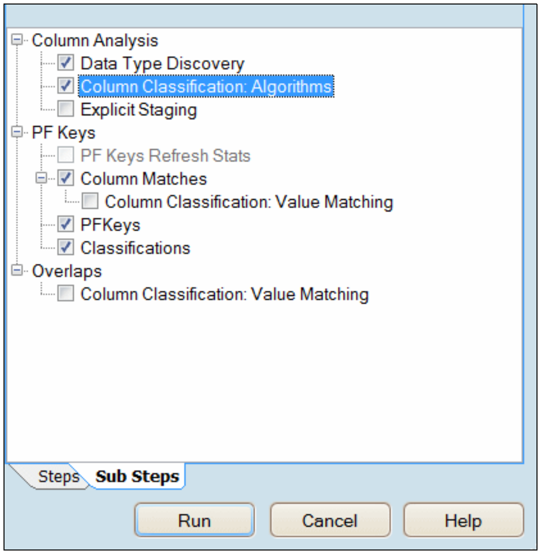
Current Step: **Data Sets**

Figure 8-17 Processing Options window

4. On the Sub Steps page (Figure 8-18), select **Column Classification: Algorithms**, which signals InfoSphere Discovery to find sensitive data using built-in pattern matching algorithms. In addition to this option, you can also select the following options, as we do in this example:

   – **Data Type Discovery**
   – **Column Matches**
   – **PF Keys**
   – **Classifications**

   Then click **Run**.



- Column Analysis
  - ☑ Data Type Discovery
  - ☑ Column Classification: Algorithms
  - □ Explicit Staging
- PF Keys
  - □ PF Keys Refresh Stats
  - ☑ Column Matches
    - □ Column Classification: Value Matching
  - ☑ PFKeys
  - ☑ Classifications
- Overlaps
  - □ Column Classification: Value Matching

Figure 8-18 Sub Steps

Discovery Studio then submits the task for InfoSphere Discovery to execute.

8.3.1 Monitoring tasks with the activity viewer

InfoSphere Discovery provides an *activity viewer* to monitor tasks. You click the hyperlink in the upper right corner of the Discovery Studio window. In the example Figure 8-19, the link indicates the message “Currently 1 Active Task.”

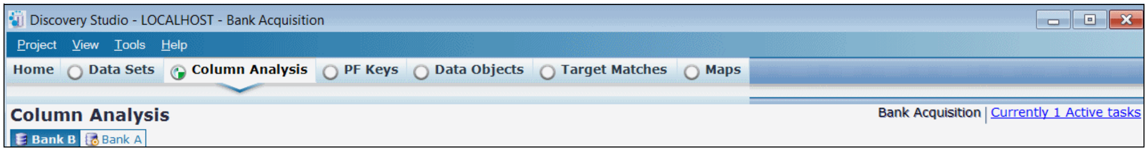
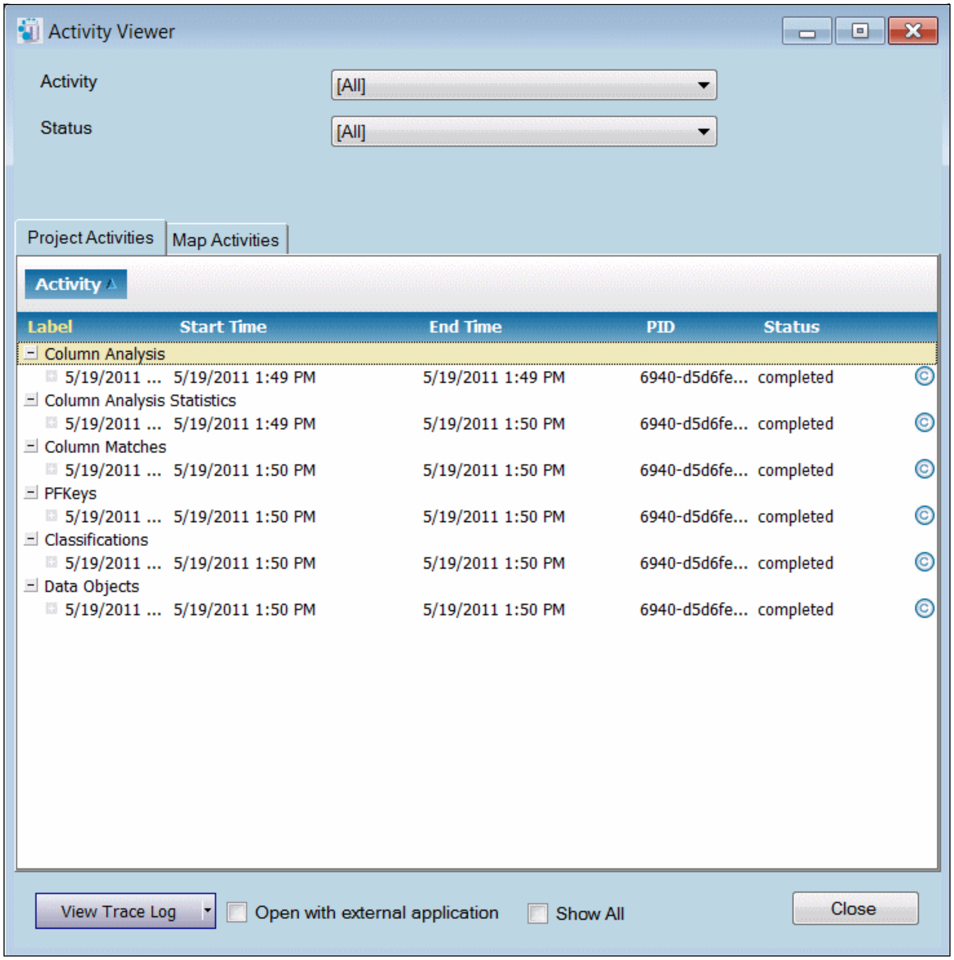


Figure 8-19 Activity Viewer link

When you click the link, the Activity Viewer window opens, showing the status of currently running tasks and previously executed tasks. This information helps with debugging and estimating how long a task will run. The options include viewing the trace log, error log, or debug log. Figure 8-20 shows the activity viewer for the bank scenario project.



Activity [All]

Status [All]

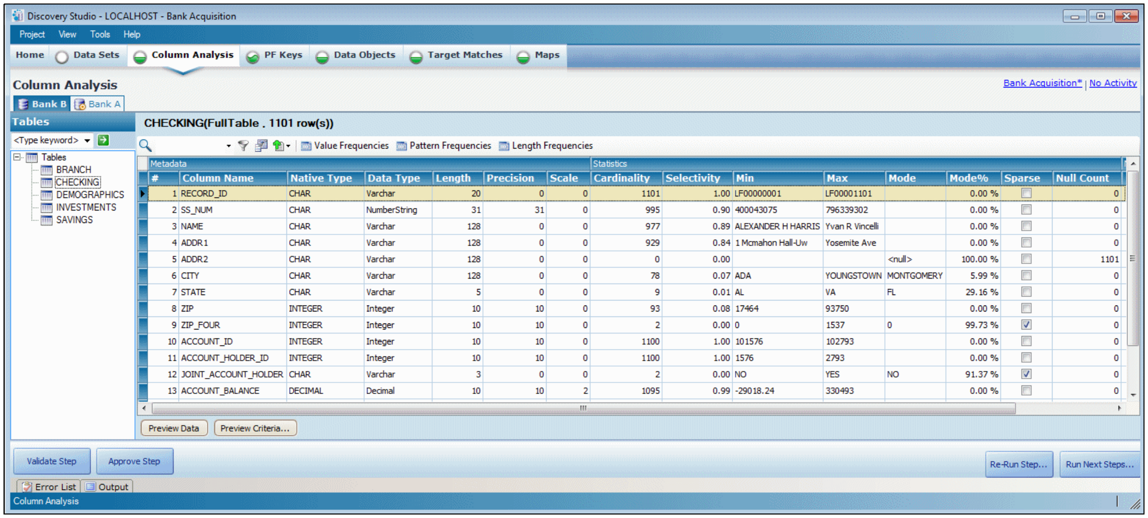
Label	Start Time	End Time	PID	Status
Column Analysis				
5/19/2011 ...	5/19/2011 1:49 PM	5/19/2011 1:49 PM	6940-d5d6fe...	completed
Column Analysis Statistics				
5/19/2011 ...	5/19/2011 1:49 PM	5/19/2011 1:50 PM	6940-d5d6fe...	completed
Column Matches				
5/19/2011 ...	5/19/2011 1:50 PM	5/19/2011 1:50 PM	6940-d5d6fe...	completed
PFKeys				
5/19/2011 ...	5/19/2011 1:50 PM	5/19/2011 1:50 PM	6940-d5d6fe...	completed
Classifications				
5/19/2011 ...	5/19/2011 1:50 PM	5/19/2011 1:50 PM	6940-d5d6fe...	completed
Data Objects				
5/19/2011 ...	5/19/2011 1:50 PM	5/19/2011 1:50 PM	6940-d5d6fe...	completed

□ Open with external application □ Show All

Figure 8-20 Activity Viewer window

8.3.2 Reviewing the column analysis results

After completing the Column Analysis task, you view the results by selecting the **Column Analysis** tab. Figure 8-21 shows an example of the results.



CHECKING(FullTable . 1101 row(s))

Metadata							Statistics							
#	Column Name	Native Type	Data Type	Length	Precision	Scale	Cardinality	Selectivity	Min	Max	Mode	Mode%	Sparse	Null Count
1	RECORD_ID	CHAR	Varchar	20	0	0	1101	1.00	LF00000001	LF00001101		0.00 %	□	0
2	SS_NUM	CHAR	NumberString	31	31	0	995	0.90	400043075	796339302		0.00 %	□	0
3	NAME	CHAR	Varchar	128	0	0	977	0.89	ALEXANDER H HARRIS	Yvan R Vincelli		0.00 %	□	0
4	ADDR1	CHAR	Varchar	128	0	0	929	0.84	1 Mcmahon Hall-Uw	Yosemite Ave		0.00 %	□	0
5	ADDR2	CHAR	Varchar	128	0	0	0	0.00			<null>	100.00 %	□	1101
6	CITY	CHAR	Varchar	128	0	0	78	0.07	ADA	YOUNGSTOWN	MONTGOMERY	5.99 %	□	0
7	STATE	CHAR	Varchar	5	0	0	9	0.01	AL	VA	FL	29.16 %	□	0
8	ZIP	INTEGER	Integer	10	10	0	93	0.08	17464	93750		0.00 %	□	0
9	ZIP_FOUR	INTEGER	Integer	10	10	0	2	0.00	0	1537	0	99.73 %	☑	0
10	ACCOUNT_ID	INTEGER	Integer	10	10	0	1100	1.00	101576	102793		0.00 %	□	0
11	ACCOUNT_HOLDER_ID	INTEGER	Integer	10	10	0	1100	1.00	1576	2793		0.00 %	□	0
12	JOINT_ACCOUNT_HOLDER	CHAR	Varchar	3	0	0	2	0.00	NO	YES	NO	91.37 %	☑	0
13	ACCOUNT_BALANCE	DECIMAL	Decimal	10	10	2	1095	0.99	-29018.24	330493		0.00 %	□	0

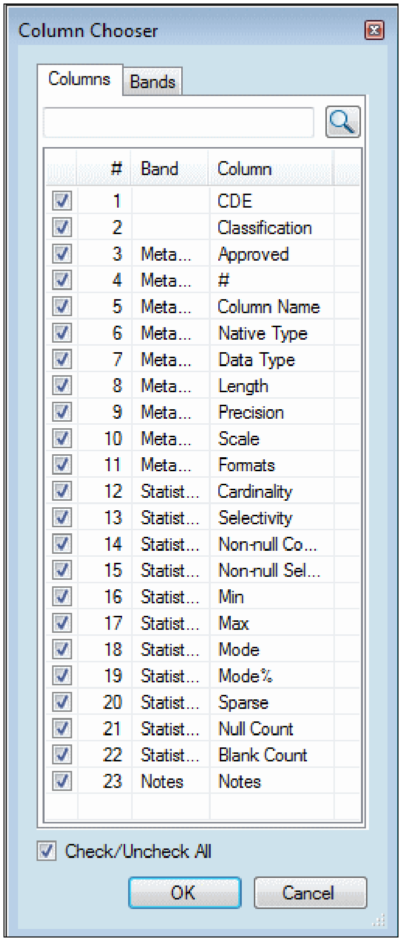
Figure 8-21 Column analysis results

The results are organized by data set and, within each data set, by table (or file). You select a table from the list in the left pane, and the results are displayed in the grid in the right pane. You control the metadata and statistics that are presented by using the *Column Chooser* function.

Column analysis results include two sections: metadata and statistics. With the Column Chooser function, you can view 11 metadata attributes, 11 statistical attributes, and free-text notes.

8.3.3 Metadata and statistical results

Figure 8-22 shows the Column Chooser feature with all available columns selected.



	#	Band	Column
☑	1		CDE
☑	2		Classification
☑	3	Meta...	Approved
☑	4	Meta...	#
☑	5	Meta...	Column Name
☑	6	Meta...	Native Type
☑	7	Meta...	Data Type
☑	8	Meta...	Length
☑	9	Meta...	Precision
☑	10	Meta...	Scale
☑	11	Meta...	Formats
☑	12	Statist...	Cardinality
☑	13	Statist...	Selectivity
☑	14	Statist...	Non-null Co...
☑	15	Statist...	Non-null Sel...
☑	16	Statist...	Min
☑	17	Statist...	Max
☑	18	Statist...	Mode
☑	19	Statist...	Mode%
☑	20	Statist...	Sparse
☑	21	Statist...	Null Count
☑	22	Statist...	Blank Count
☑	23	Notes	Notes

☑ Check/Uncheck All

Figure 8-22 Column Chooser

You can choose to view the following metadata values:

Critical Data Element (CDE)
Whether this column is labeled as a critical data element.

Classification The classifications (if any) assigned to this column either manually or by using automated column classification.

The column number within the table or file.

Column Name The name of the column identified in the logical table.

Native Type The data type stored natively with the table or file. All flat files have a native data type of "varchar," because all columns are text strings.

Data Type The discovered data type of the column. If Data Type Discovery was performed during Column Analysis (enabled by default), this type reflects the data type of the contents of the column, regardless of the native data type. InfoSphere Discovery scans each column to determine if it contains numeric values, dates, or text strings. Then it updates the Data Type value with this information.

Length The defined length of the column.

Precision The precision of the column.

Scale The number of decimals after the decimal point.

Formats Discovered data formats for numeric columns.

You can view the values of several statistical attributes. These values include the number of unique values in a column with the current primary sample set. Null and empty values are not included when calculating cardinality. Cardinality is calculated on an individual column basis and is not the result of comparison to another column. This value is never greater than the total number of rows in the column, and is used in calculating several other statistics.

In addition, you can view the following statistical attributes values:

Selectivity The degree of uniqueness of the values (including nulls) in the column, calculated as:

```
Cardinality / (Row Count – Null Count)
```

Selectivity is calculated on each column individually and is not the result of comparison to another column. This value is never greater than 100%.

Non-null Count The number of rows with non-null values for this column.

Non-null Selectivity The degree of uniqueness of all non-null values in the column, calculated as follows:

```
Cardinality / Non-null Count
```

Min The smallest or lowest value in the column, calculated numerically for numeric columns and alphabetically for other columns.

Max The largest or greatest value in the column, calculated numerically for numeric columns and alphabetically for other columns.

Mode The most common value in the column, not including null values. This value is calculated only if a particular value is displayed in more than 5% of the rows.

Mode % The number of times the mode (most common value) is displayed in this column, as a percentage of all values in the column.

Sparse Indicates whether the column is sparse, based on the Mode %. A sparse column contains mostly the same value except for a few exceptions. Sparse columns are treated differently than non-sparse columns during processing. The large percentage of duplicate values makes it more difficult for InfoSphere Discovery to immediately determine whether a column match is a valid relationship.

Null Count The number of rows where the column value is null.

Blank Count The number of rows in the column that are blank (empty).

Notes A blank field for notes, descriptions, or comments about the column. You click the cell and enter a description of the column. The name of the logged-in user and the current time stamp are automatically added to the note.

8.3.4 Value, pattern, and length frequencies

In addition to the column metadata and statistics, InfoSphere Discovery calculates *value*, *pattern*, and *length frequencies*. You display the frequencies by using the buttons in the *Column Analysis* area, as shown in Figure 8-23.

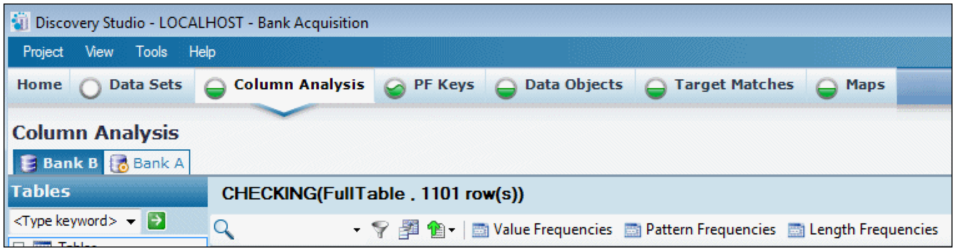
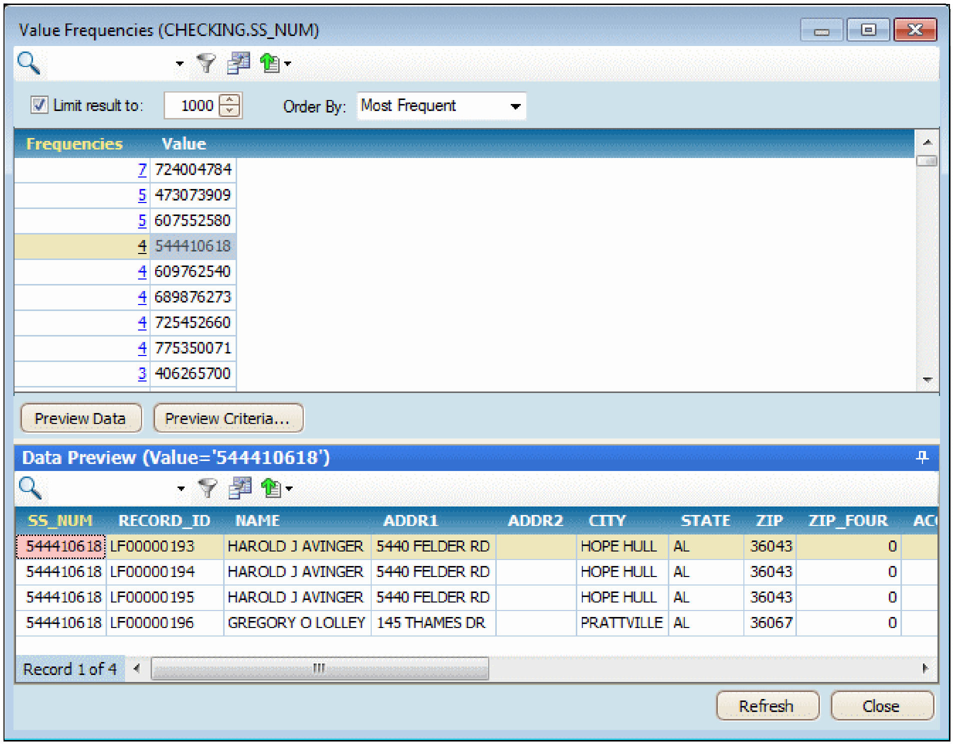


Figure 8-23 Value, Pattern, and Length Frequencies

By selecting **Value Frequencies** (Figure 8-23 on page 233), each unique value in the column and the number of times (frequency) that each value appears in the column are displayed. The frequency value is a hyperlink. Clicking a value selects the rows with that value. Figure 8-24 shows rows where CHECKING.SS_NUM is equal to 544410618.



Value Frequencies (CHECKING.SS_NUM)

☑ Limit result to: 1000 Order By: Most Frequent

Frequencies	Value
7	724004784
5	473073909
5	607552580
4	544410618
4	609762540
4	689876273
4	725452660
4	775350071
3	406265700

Data Preview (Value='544410618')

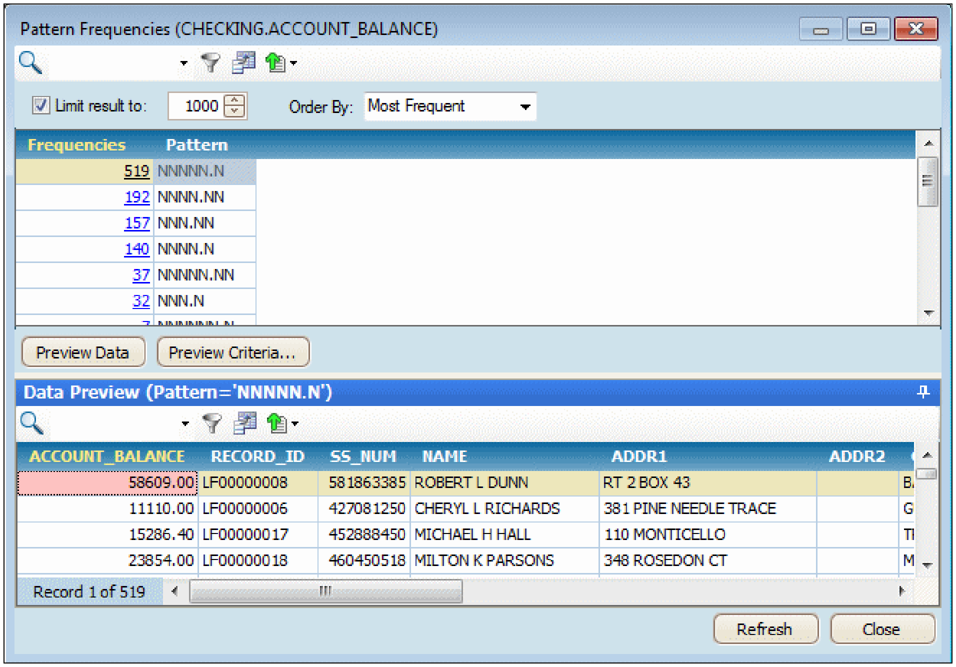
SS_NUM	RECORD_ID	NAME	ADDR1	ADDR2	CITY	STATE	ZIP	ZIP_FOUR	AC
544410618	LF00000193	HAROLD J AVINGER	5440 FELDER RD		HOPE HULL	AL	36043	0	
544410618	LF00000194	HAROLD J AVINGER	5440 FELDER RD		HOPE HULL	AL	36043	0	
544410618	LF00000195	HAROLD J AVINGER	5440 FELDER RD		HOPE HULL	AL	36043	0	
544410618	LF00000196	GREGORY O LOLLEY	145 THAMES DR		PRATTVILLE	AL	36067	0	

Record 1 of 4

Figure 8-24 Value Frequencies

By selecting **Pattern Frequencies** (Figure 8-23 on page 233), each unique pattern for the selected column and the number of times (frequency) that each pattern appears in the column are displayed. The frequency value is a hyperlink. Clicking a link selects the rows with the selected pattern. In patterns, the letter N signifies a numeric value, and the letter A specifies an alphabetic character. Other values are expressed with the character itself. For example, the value `SSN: 555-11-2222` is expressed with a pattern as `AAA: NNN-NN-NNNN`.

Figure 8-25 shows the rows with the pattern `NNNNN.NN` for CHECKING.ACCOUNT_BALANCES.



☑ Limit result to: 1000 Order By: Most Frequent

Frequencies	Pattern
519	NNNNN.N
192	NNNN.NN
157	NNN.NN
140	NNNN.N
37	NNNNN.NN
32	NNN.N

Data Preview (Pattern='NNNNN.N')

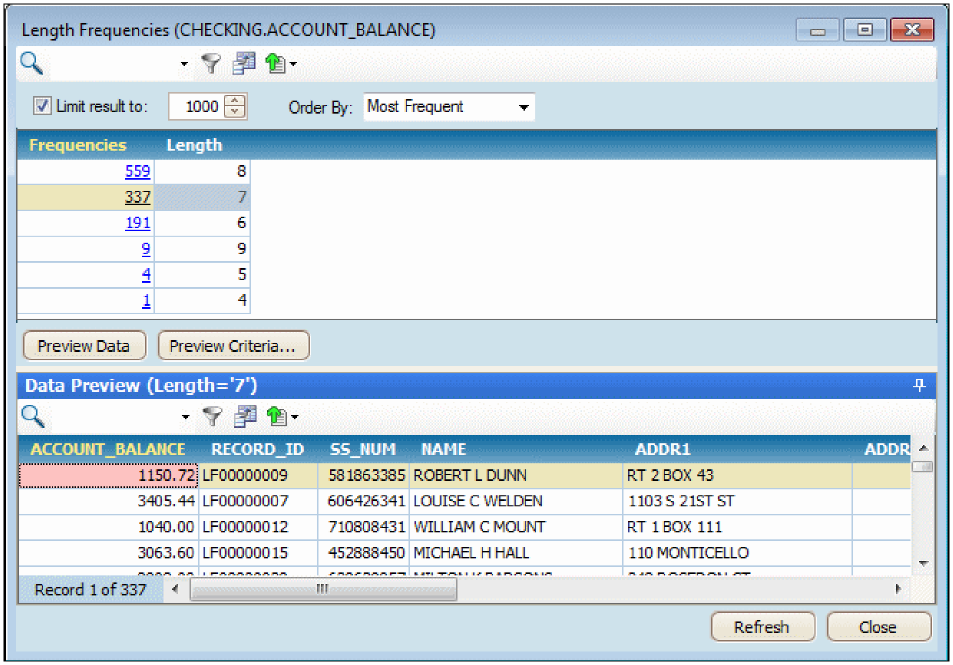
ACCOUNT_BALANCE	RECORD_ID	SS_NUM	NAME	ADDR1	ADDR2	
58609.00	LF00000008	581863385	ROBERT L DUNN	RT 2 BOX 43		B
11110.00	LF00000006	427081250	CHERYL L RICHARDS	381 PINE NEEDLE TRACE		G
15286.40	LF00000017	452888450	MICHAEL H HALL	110 MONTICELLO		T
23854.00	LF00000018	460450518	MILTON K PARSONS	348 ROSEDON CT		M

Record 1 of 519

Figure 8-25 Pattern Frequencies

By selecting **Length Frequencies** (Figure 8-23 on page 233), the unique length of each column value and the number of times (frequency) that each value appears in the column are displayed. The frequency value is a hyperlink. Clicking a link selects the rows with the selected length for this column.

Figure 8-26 shows CHECKING.ACCOUNT_BALANCE values that have a length of seven.



Length Frequencies (CHECKING.ACCOUNT_BALANCE)

☑ Limit result to: 1000 Order By: Most Frequent

Frequencies	Length
559	8
337	7
191	6
9	9
4	5
1	4

Data Preview (Length='7')

ACCOUNT_BALANCE	RECORD_ID	SS_NUM	NAME	ADDR1	ADDR
1150.72	LF00000009	581863385	ROBERT L DUNN	RT 2 BOX 43	
3405.44	LF00000007	606426341	LOUISE C WELDEN	1103 S 21ST ST	
1040.00	LF00000012	710808431	WILLIAM C MOUNT	RT 1 BOX 111	
3063.60	LF00000015	452888450	MICHAEL H HALL	110 MONTICELLO	

Record 1 of 337

Figure 8-26 Length Frequencies

After performing column analysis, you can easily generate a project report with the results. The project report contains all of the column analysis information *except* the value, pattern, or length frequencies, which might be too large to print in a report. To export frequencies, display one, and then export it.

To run the project report, complete these steps:

1. In Discovery Studio, select **Tools** → **Reports** → **Project Report** (Figure 8-27).

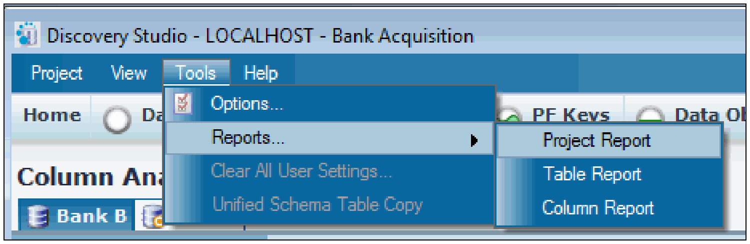
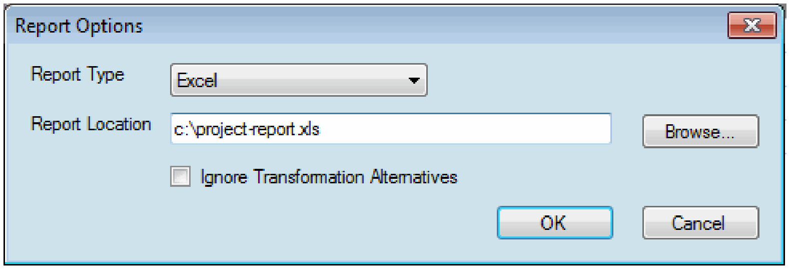


Figure 8-27 Selecting a project report

2. In the Report Options window (Figure 8-28), select the format you want for the reports, such as Excel or HTML, and then select a file destination. Click **OK** to generate the report.



Report Options

Report Type: Excel

Report Location: c:\project-report.xls

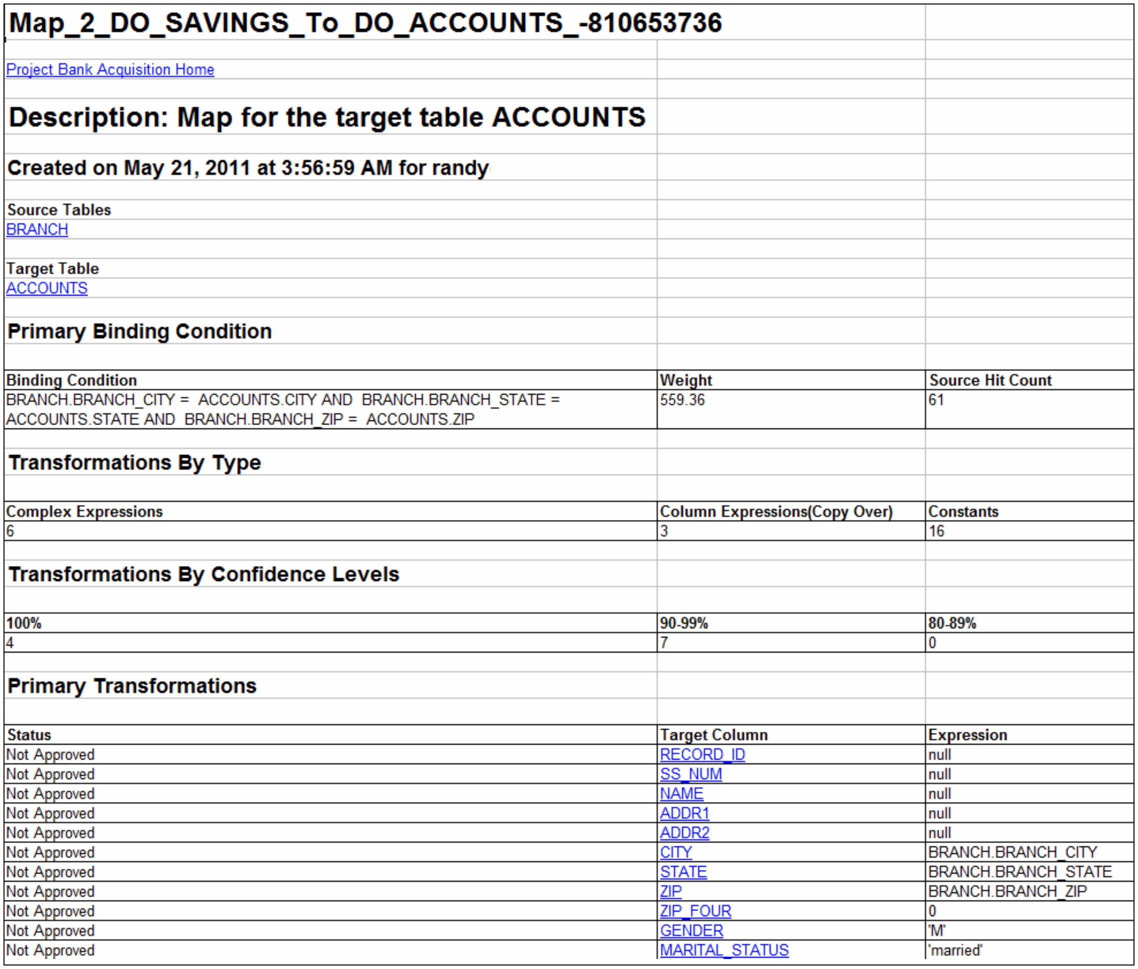
Browse...

□ Ignore Transformation Alternatives

OK Cancel

Figure 8-28 Project report location

The project report provides a way to disseminate column analysis results. Figure 8-29 shows one page of a project report that highlights a transformation map for the ACCOUNTS table.



Map_2_DO_SAVINGS_To_DO_ACCOUNTS_-810653736

Project Bank Acquisition Home

Description: Map for the target table ACCOUNTS

Created on May 21, 2011 at 3:56:59 AM for randy

Source Tables
BRANCH

Target Table
ACCOUNTS

Primary Binding Condition

Binding Condition	Weight	Source Hit Count
BRANCH.BRANCH_CITY = ACCOUNTS.CITY AND BRANCH.BRANCH_STATE = ACCOUNTS.STATE AND BRANCH.BRANCH_ZIP = ACCOUNTS.ZIP	559.36	61

Transformations By Type

Complex Expressions	Column Expressions(Copy Over)	Constants
6	3	16

Transformations By Confidence Levels

100%	90-99%	80-89%
4	7	0

Primary Transformations

Status	Target Column	Expression
Not Approved	RECORD_ID	null
Not Approved	SS_NUM	null
Not Approved	NAME	null
Not Approved	ADDR1	null
Not Approved	ADDR2	null
Not Approved	CITY	BRANCH.BRANCH_CITY
Not Approved	STATE	BRANCH.BRANCH_STATE
Not Approved	ZIP	BRANCH.BRANCH_ZIP
Not Approved	ZIP_FOUR	0
Not Approved	GENDER	'M'
Not Approved	MARITAL_STATUS	'married'

Figure 8-29 Project report example

8.4 Identifying and classifying sensitive data

In this scenario, we selected the **column classification: algorithms** option when we ran the Column Analysis step. Automatic column classification helps you find sensitive data or other critical data. This area is important within information governance. InfoSphere Discovery finds critical data elements (such as sensitive data) embedded in source data, within or across databases. For example, InfoSphere Discovery can find national ID values that are stored in free-text comment fields.

Column classification finds sensitive data with two approaches:

- *Value matching* searches for values that match existing reference data.
- *Pattern matching* searches for data patterns by using algorithms that include regular expressions, data type filters, and thresholds for minimum and maximum length to identify and classify sensitive data elements. With this approach, InfoSphere Discovery finds data values based on defined patterns. A set of algorithms is shipped with InfoSphere Discovery, and custom algorithms can be defined by an organization.

  For example, the built-in algorithm for a credit card number detects columns that contain values that conform to the following regular expression:

  ```
  ^(?:4[0-9]{12}(?:[0-9]{3})?|5[1-5][0-9]{14}|6(?:011|5[0-9][0-9])[0-9
  ]{12}|3[47][0-9]{13}|3(?:0[0-5]|[68][0-9])[0-9]{11}|(?:2131|1800|35\
  \d{3})\\d{11})$
  ```

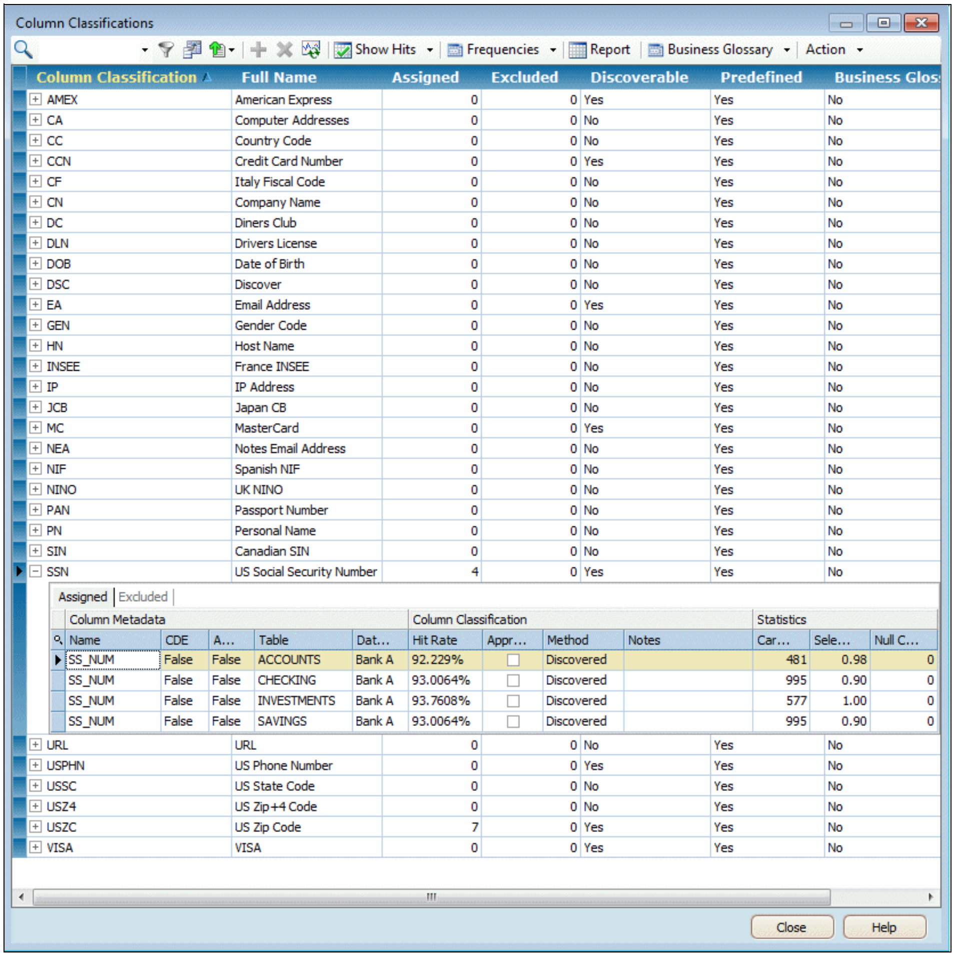
  Additionally, the algorithm validates the Luhn checksum algorithm and ensures that the values are 13–19 digits in length. Algorithms can find anchored values or values embedded in free-form text.

Both column classification approaches work to identify and classify critical data. The best approach depends on the need. For example, the requirement might be to search for a value such as counter party, which might look like any name. Value matching with a set of reference data (actual counter party names) works better than searching for a pattern, which might turn up any column with any names, not specifically counter party names. Alternatively, the requirement might be to search for credit card numbers, meaning any credit card number. Regardless of whether we have reference values, the pattern matching approach works better.

8.4.1 Column classification view

You use the Column Classification View to see if InfoSphere Discovery found any sensitive data. Go to **View** → **Column Classification View** to open the column classification view.

In the Column Classifications window (Figure 8-30), you see that InfoSphere Discovery found four columns with U.S. Social Security Numbers (SSN).



Column Classification	Full Name	Assigned	Excluded	Discoverable	Predefined	Business Glos
AMEX	American Express	0	0	Yes	Yes	No
CA	Computer Addresses	0	0	No	Yes	No
CC	Country Code	0	0	No	Yes	No
CCN	Credit Card Number	0	0	Yes	Yes	No
CF	Italy Fiscal Code	0	0	No	Yes	No
CN	Company Name	0	0	No	Yes	No
DC	Diners Club	0	0	No	Yes	No
DLN	Drivers License	0	0	No	Yes	No
DOB	Date of Birth	0	0	No	Yes	No
DSC	Discover	0	0	No	Yes	No
EA	Email Address	0	0	Yes	Yes	No
GEN	Gender Code	0	0	No	Yes	No
HN	Host Name	0	0	No	Yes	No
INSEE	France INSEE	0	0	No	Yes	No
IP	IP Address	0	0	No	Yes	No
JCB	Japan CB	0	0	No	Yes	No
MC	MasterCard	0	0	Yes	Yes	No
NEA	Notes Email Address	0	0	No	Yes	No
NIF	Spanish NIF	0	0	No	Yes	No
NINO	UK NINO	0	0	No	Yes	No
PAN	Passport Number	0	0	No	Yes	No
PN	Personal Name	0	0	No	Yes	No
SIN	Canadian SIN	0	0	No	Yes	No
SSN	US Social Security Number	4	0	Yes	Yes	No

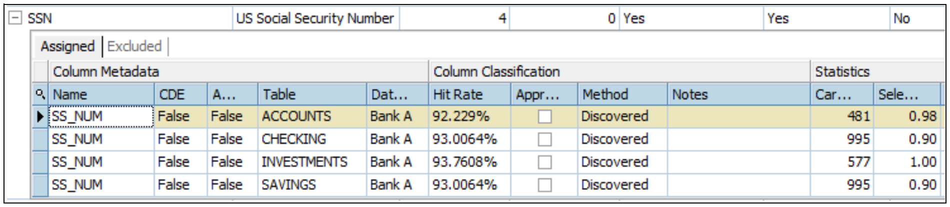
Assigned | Excluded

Column Metadata					Column Classification				Statistics		
Name	CDE	A...	Table	Dat...	Hit Rate	Appr...	Method	Notes	Car...	Sele...	Null C...
SS_NUM	False	False	ACCOUNTS	Bank A	92.229%	□	Discovered		481	0.98	0
SS_NUM	False	False	CHECKING	Bank A	93.0064%	□	Discovered		995	0.90	0
SS_NUM	False	False	INVESTMENTS	Bank A	93.7608%	□	Discovered		577	1.00	0
SS_NUM	False	False	SAVINGS	Bank A	93.0064%	□	Discovered		995	0.90	0

Column Classification	Full Name	Assigned	Excluded	Discoverable	Predefined	Business Glos
URL	URL	0	0	No	Yes	No
USPHN	US Phone Number	0	0	Yes	Yes	No
USSC	US State Code	0	0	No	Yes	No
USZ4	US Zip+4 Code	0	0	No	Yes	No
USZC	US Zip Code	7	0	Yes	Yes	No
VISA	VISA	0	0	Yes	Yes	No

Figure 8-30 Column classification view

Figure 8-31 shows a close-up view of the four columns in the column classification view. Of the values, 92%–93% in each column matched the algorithm for SSN.



⊟ SSN	US Social Security Number	4	0	Yes	Yes	No

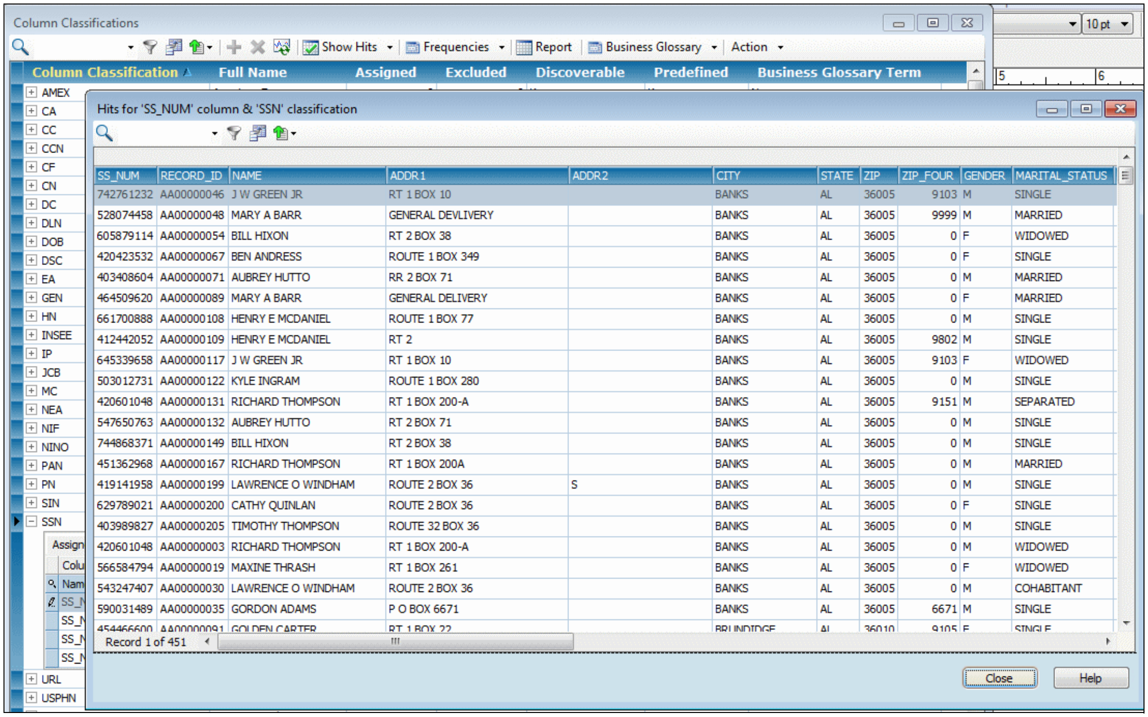
Assigned | Excluded

Column Metadata					Column Classification				Statistics	
Name	CDE	A...	Table	Dat...	Hit Rate	Appr...	Method	Notes	Car...	Sele...
SS_NUM	False	False	ACCOUNTS	Bank A	92.229%	□	Discovered		481	0.98
SS_NUM	False	False	CHECKING	Bank A	93.0064%	□	Discovered		995	0.90
SS_NUM	False	False	INVESTMENTS	Bank A	93.7608%	□	Discovered		577	1.00
SS_NUM	False	False	SAVINGS	Bank A	93.0064%	□	Discovered		995	0.90

Figure 8-31 Column classifications

8.4.2 Displaying hits for classification columns

To view the hits for classification columns, highlight the first row for the ACCOUNTS table, and then click the **Show Hits** button in the menu bar. A list of rows that meet the algorithm for U.S. SSN is shown in the results window (Figure 8-32).



Hits for 'SS_NUM' column & 'SSN' classification

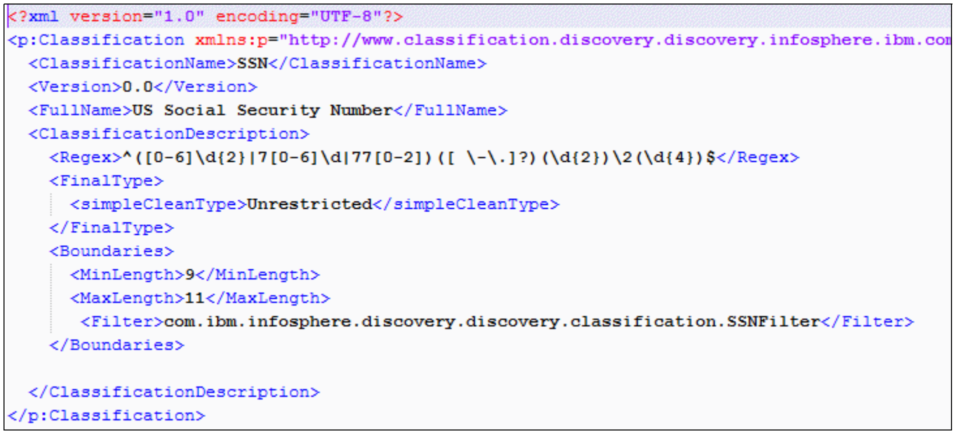
SS_NUM	RECORD_ID	NAME	ADDR1	ADDR2	CITY	STATE	ZIP	ZIP_FOUR	GENDER	MARITAL_STATUS
742761232	AA00000046	J W GREEN JR	RT 1 BOX 10		BANKS	AL	36005	9103	M	SINGLE
528074458	AA00000048	MARY A BARR	GENERAL DEVLIVERY		BANKS	AL	36005	9999	M	MARRIED
605879114	AA00000054	BILL HIXON	RT 2 BOX 38		BANKS	AL	36005	0	F	WIDOWED
420423532	AA00000067	BEN ANDRESS	ROUTE 1 BOX 349		BANKS	AL	36005	0	F	SINGLE
403408604	AA00000071	AUBREY HUTTO	RR 2 BOX 71		BANKS	AL	36005	0	M	MARRIED
464509620	AA00000089	MARY A BARR	GENERAL DELIVERY		BANKS	AL	36005	0	F	MARRIED
661700888	AA00000108	HENRY E MCDANIEL	ROUTE 1 BOX 77		BANKS	AL	36005	0	M	SINGLE
412442052	AA00000109	HENRY E MCDANIEL	RT 2		BANKS	AL	36005	9802	M	SINGLE
645339658	AA00000117	J W GREEN JR	RT 1 BOX 10		BANKS	AL	36005	9103	F	WIDOWED
503012731	AA00000122	KYLE INGRAM	ROUTE 1 BOX 280		BANKS	AL	36005	0	M	SINGLE
420601048	AA00000131	RICHARD THOMPSON	RT 1 BOX 200-A		BANKS	AL	36005	9151	M	SEPARATED
547650763	AA00000132	AUBREY HUTTO	RT 2 BOX 71		BANKS	AL	36005	0	M	SINGLE
744868371	AA00000149	BILL HIXON	RT 2 BOX 38		BANKS	AL	36005	0	M	SINGLE
451362968	AA00000167	RICHARD THOMPSON	RT 1 BOX 200A		BANKS	AL	36005	0	M	MARRIED
419141958	AA00000199	LAWRENCE O WINDHAM	ROUTE 2 BOX 36	S	BANKS	AL	36005	0	M	SINGLE
629789021	AA00000200	CATHY QUINLAN	ROUTE 2 BOX 36		BANKS	AL	36005	0	F	SINGLE
403989827	AA00000205	TIMOTHY THOMPSON	ROUTE 32 BOX 36		BANKS	AL	36005	0	M	SINGLE
420601048	AA00000003	RICHARD THOMPSON	RT 1 BOX 200-A		BANKS	AL	36005	0	M	WIDOWED
566584794	AA00000019	MAXINE THRASH	RT 1 BOX 261		BANKS	AL	36005	0	F	WIDOWED
543247407	AA00000030	LAWRENCE O WINDHAM	ROUTE 2 BOX 36		BANKS	AL	36005	0	M	COHABITANT
590031489	AA00000035	GORDON ADAMS	P O BOX 6671		BANKS	AL	36005	6671	M	SINGLE
454466600	AA00000091	GOLDEN CARTER	RT 1 BOX 22		BRUNDIDGE	AL	36010	9105	F	SINGLE

Record 1 of 451

Figure 8-32 Hits for column classifications

8.4.3 Column classification algorithms

How did InfoSphere Discovery find these rows? InfoSphere Discovery used a column classification algorithm that checked for strings with 9–11 characters that matched a regular expression. The algorithm is defined in an XML specification and looks similar to the example in Figure 8-33.



```
<?xml version="1.0" encoding="UTF-8"?>
<p:Classification xmlns:p="http://www.classification.discovery.discovery.infosphere.ibm.co
  <ClassificationName>SSN</ClassificationName>
  <Version>0.0</Version>
  <FullName>US Social Security Number</FullName>
  <ClassificationDescription>
    <Regex>^([0-6]\d{2}|7[0-6]\d|77[0-2])([ \-\.]?)(\d{2})\2(\d{4})$</Regex>
    <FinalType>
      <simpleCleanType>Unrestricted</simpleCleanType>
    </FinalType>
    <Boundaries>
      <MinLength>9</MinLength>
      <MaxLength>11</MaxLength>
       <Filter>com.ibm.infosphere.discovery.discovery.classification.SSNFilter</Filter>
    </Boundaries>

  </ClassificationDescription>
</p:Classification>
```

Figure 8-33 Column classification algorithm

The regular expression is defined in the <regex> tag:

```
^([0-6]\d{2}|7[0-6]\d|77[0-2])([ \-\.]?)(\d{2})\2(\d{4})$
```

Specifically, the algorithm defines a regular expression that looks for four variants of SSN:

- SSNs with no delimiters between the three groups of numeric digits
- SSNs delimited by a space ( )
- SSNs delimited by a dash (-)
- SSNs delimited by a period (.)

The regular expression indicates that the second section of the SSN must have 2 digits and the third section must have 4 digits. If the first section starts with a digit of 0–6 inclusive, it can be followed by any digit. If the first section starts with 7, the next digit must be 0–6 inclusive, and third digit has no restrictions. If the first section starts with 77, the third digit must be 0–2 inclusive.

The <filter> tag in this algorithm invokes a Java filter called *SSNFilter*, which ensures that no section consists only of zeros.

InfoSphere Discovery ships with several built-in pattern-matching classification algorithms, including the following algorithms:

- US Social Security Number
- Email Address
- Credit Card Number
- US Phone Number
- US ZIP Code

Column analysis reveals rich information that helps organizations manage their information assets. Column analysis is only the beginning. It does not reveal how tables and columns in a database are related to each other, nor does it reveal how data in one system is related to other systems. It is a great starting point for asking more questions about the data, which is the point where data discovery begins.

8.5 Assigning InfoSphere Business Glossary terms to physical assets

Defining and using business terms correctly is challenging for most organizations. After a merger or acquisition, the need for consistent business term usage grows even more. Fortunately, when Bank A buys Bank B, Bank A decides to use InfoSphere Business Glossary to build and manage a glossary of business terms.

Bank A uses InfoSphere Business Glossary to define business terms. After defining the terms in InfoSphere Business Glossary, InfoSphere Discovery imports the terms. The imported business glossary terms are called *column classifications* in InfoSphere Discovery. These classifications are similar to the classifications that are used to discover sensitive data.

By using its pattern-matching and value-matching approaches, InfoSphere Discovery finds database columns that match the imported business terms. This process maps the imported business terms to their related physical assets (tables and files) in the database systems of the bank. For example, Bank A defines the term *Customer Identifier* in InfoSphere Business Glossary, but it does not know every location where customer IDs reside in its database systems. Bank A uses the column classification capability of InfoSphere Discovery to determine the locations where customer IDs reside throughout its databases.

When Bank A finishes the column classification effort, InfoSphere Discovery exports the classifications back to InfoSphere Business Glossary, enriching the metadata results.

8.5.1 Importing, mapping, and exporting term assignments

InfoSphere Discovery imports business glossary terms defined in InfoSphere Business Glossary so that InfoSphere Discovery can map the terms to physical assets. After the mapping exercise, InfoSphere Discovery exports the results, and InfoSphere Business Glossary imports the new terms and assignments.

The process involves the following primary steps:

1. InfoSphere Discovery imports business glossary terms from InfoSphere Business Glossary.
2. InfoSphere Discovery maps the business glossary terms to physical assets.
3. InfoSphere Discovery exports the terms and term assignments back to InfoSphere Business Glossary.

Automatic column classification works in two ways with InfoSphere Discovery:

- Pattern matching with Column Classification Algorithms. This approach uses regular expressions and other parameters to classify columns based on the patterns of data that they contain.
- Value matching uses existing reference data to classify columns by comparing data values to the reference samples.

Both column classification approaches work to map business glossary terms to physical assets. The best approach depends on the business term. The business glossary term *counter party* represents an individual, household, or organization that serves the role of counter party in a contract. The value domain for counter party is *party*, which is a text string of variable length. Few rules govern the domain of possible values. The primary rule is that a valid counter party must exist in the master list of parties for the bank. In this situation, pattern matching is not successful, because it is impossible to distinguish a counter party from any other type of name, based on pattern matching alone.

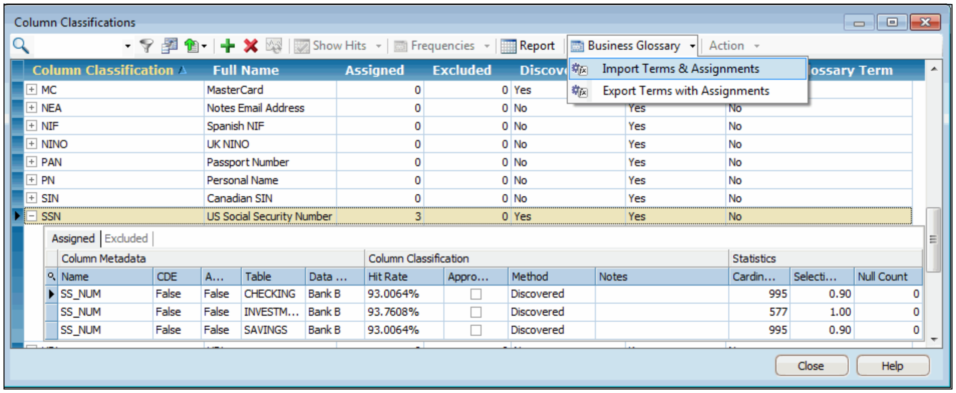
Value matching against a set of reference data (actual counter party names) is the best choice for a term such as counter party. By using an existing list (or lists) of counter parties, InfoSphere Discovery scans the data of the bank to find any columns with values that match the reference list. If a certain number of rows match the reference list, based on a definable threshold, InfoSphere Discovery flags the column as a potential match. A data analyst reviews potential matches and approves the correct ones.

Alternatively, if the bank wants to find credit card numbers stored in their databases, pattern matching might work better. InfoSphere Discovery ships with a built-in algorithm to identify credit card numbers and classify them by type, such as Visa, MasterCard, American Express, or Discover. For patterns that are not built into InfoSphere Discovery, with InfoSphere Discovery, you can define an algorithm based on a regular expression and other parameters, such as minimum and maximum length.

8.5.2 Mapping business glossary terms to physical assets

To map InfoSphere Business Glossary terms to physical assets with InfoSphere Discovery, complete these steps:

1. In the Discovery Studio main menu, select **View** → **Column Classification View**.
2. In the Column Classifications window (Figure 8-34), select **Business Glossary** → **Import Terms & Assignments**.



Column Classification	Full Name	Assigned	Excluded	Discov		ossary Term
MC	MasterCard	0	0	Yes		
NEA	Notes Email Address	0	0	No	Yes	No
NIF	Spanish NIF	0	0	No	Yes	No
NINO	UK NINO	0	0	No	Yes	No
PAN	Passport Number	0	0	No	Yes	No
PN	Personal Name	0	0	No	Yes	No
SIN	Canadian SIN	0	0	No	Yes	No
SSN	US Social Security Number	3	0	Yes	Yes	No

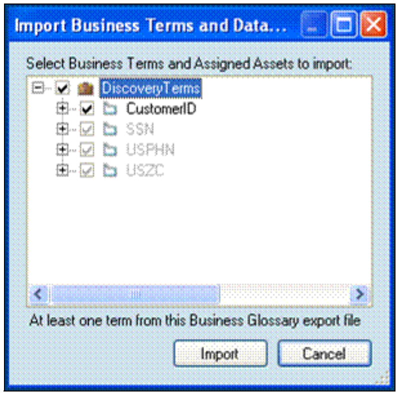
Assigned | Excluded

Column Metadata					Column Classification				Statistics		
Name	CDE	A...	Table	Data ...	Hit Rate	Appro...	Method	Notes	Cardin...	Selecti...	Null Count
SS_NUM	False	False	CHECKING	Bank B	93.0064%	□	Discovered		995	0.90	0
SS_NUM	False	False	INVESTM...	Bank B	93.7608%	□	Discovered		577	1.00	0
SS_NUM	False	False	SAVINGS	Bank B	93.0064%	□	Discovered		995	0.90	0

Figure 8-34 Import business glossary terms and assignments

3. Navigate to the InfoSphere Business Glossary export file, and then open it.

4. In the Import Business Terms and Data window (Figure 8-35), select the business glossary terms to import, and then click **Import**.



Import Business Terms and Data...

Select Business Terms and Assigned Assets to import:

- ☑ DiscoveryTerms
  - ☑ CustomerID
  - ☑ SSN
  - ☑ USPHN
  - ☑ USZC

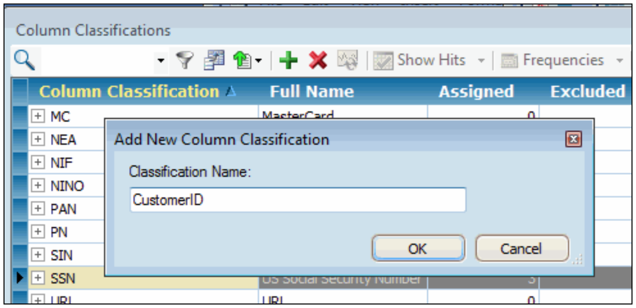
At least one term from this Business Glossary export file

Import | Cancel

Figure 8-35 InfoSphere Business Glossary export file location

You can use the built-in classification algorithms of InfoSphere Discovery or define custom classification algorithms for each of the business glossary terms that you need to map to physical assets. InfoSphere Discovery can classify the assets by using value matching or pattern matching. Some data elements are easier to identify by using pattern matching. Other data elements are easier to identify by using value matching to existing reference data.

5. In the Column Classifications window (Figure 8-36), click the green **+** button to define a new column classification. In the Add new Column Classification dialog box (inset in Figure 8-36), type `CustomerID`, and then click **OK**.



Column Classifications

Column Classification	Full Name	Assigned	Excluded
MC	MasterCard	0	
NEA			
NIF			
NINO			
PAN			
PN			
SIN			
SSN	US Social Security Number	3	
URL	URL	0	

Add New Column Classification

Classification Name: CustomerID

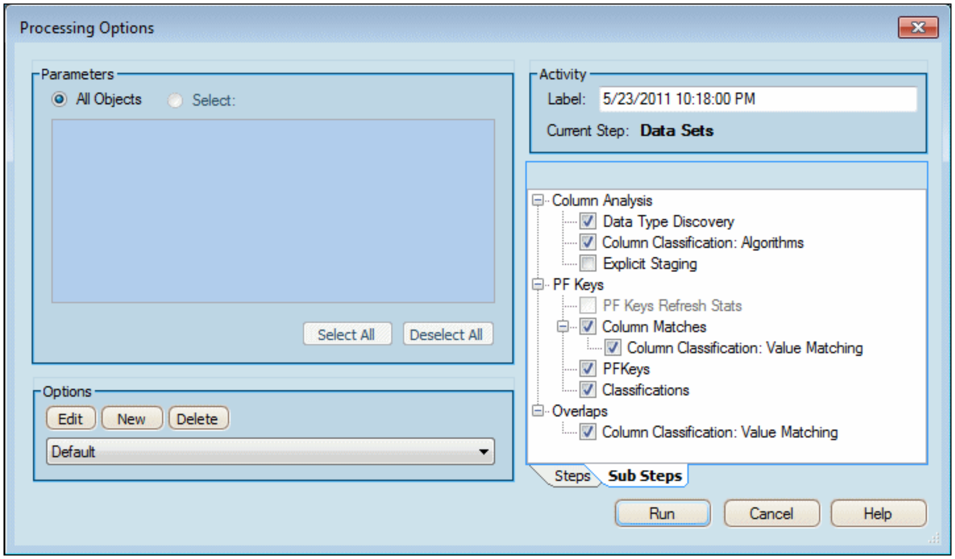
OK | Cancel

Figure 8-36 gAdding column classification

The Column Classification options in InfoSphere Discovery use pattern matching and value matching to identify and classify columns when you run Column Analysis. The **Column Classification: Algorithm** option runs the pattern matching classification task. The **Column Classification: Value**

Matching option runs the value matching classification task. For value matching across data sets, you use the column classification option within the Overlaps step.

6. On the **Data Sets** tab, click the **Run Next Steps** button.
7. On the **Sub Steps** tab (Figure 8-37), select the **Column Classification: Algorithms** and **Column Classification: Value Matching** check boxes. Then click **Run**.



Processing Options

Parameters

● All Objects ○ Select:

Activity

Label: 5/23/2011 10:18:00 PM

Current Step: **Data Sets**

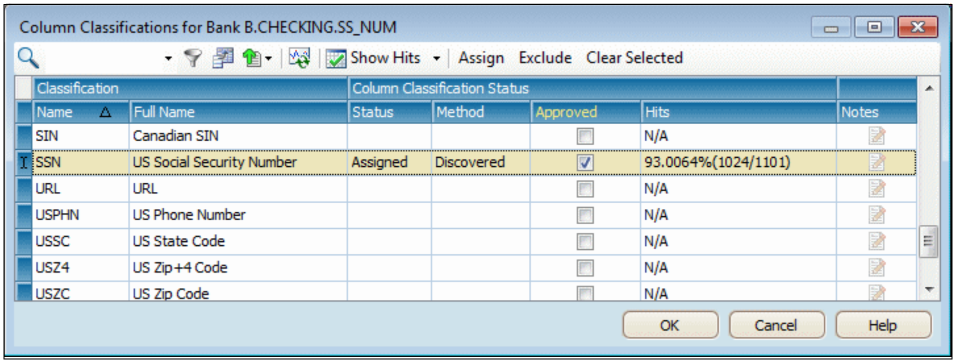
- Column Analysis
  - ☑ Data Type Discovery
  - ☑ Column Classification: Algorithms
  - □ Explicit Staging
- PF Keys
  - □ PF Keys Refresh Stats
  - ☑ Column Matches
    - ☑ Column Classification: Value Matching
  - ☑ PFKeys
  - ☑ Classifications
- Overlaps
  - ☑ Column Classification: Value Matching

Options

Default

Figure 8-37 Substeps for column classification

After the column classification tasks complete, a data analyst or SME reviews the classifications in the **Column analysis** tab and approves the legitimate classifications. Figure 8-38 shows an approved classification for the classification US Social Security Number (SSN).



Classification		Column Classification Status				
Name	Full Name	Status	Method	Approved	Hits	Notes
SIN	Canadian SIN			□	N/A	
SSN	US Social Security Number	Assigned	Discovered	☑	93.0064%(1024/1101)	
URL	URL			□	N/A	
USPHN	US Phone Number			□	N/A	
USSC	US State Code			□	N/A	
USZ4	US Zip+4 Code			□	N/A	
USZC	US Zip Code			□	N/A	

Figure 8-38 Approving column classifications

8. Export the mapping with the InfoSphere Business Glossary export feature:
   a. Open the Column Classification View.
   b. Select **Export Business Glossary Terms and Assignments**.

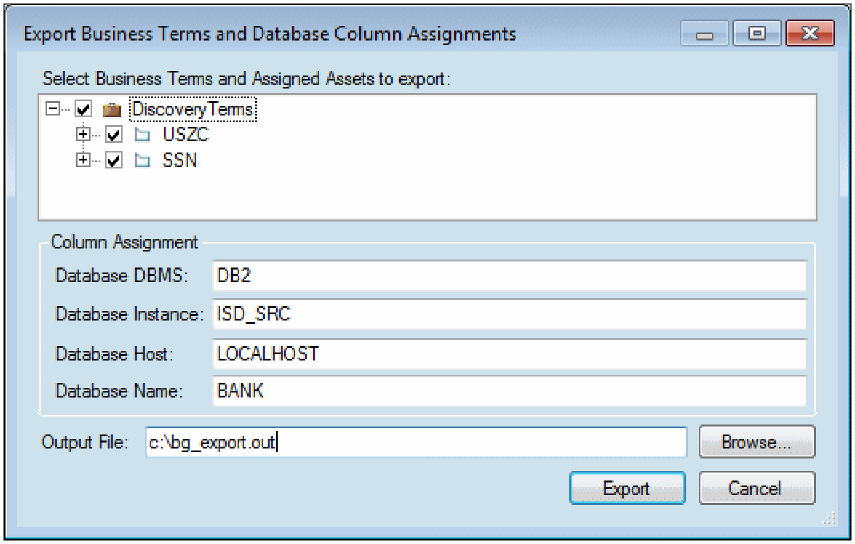
c. In the Export Business Terms and Database Column Assignments window (Figure 8-39), complete the following steps:

   i. Under Select Business Terms and Assigned Assets to export, select the necessary terms and assets to export.

   ii. Under Column Assignment, complete the database information.

   iii. Enter an output file name.

   iv. Click **Export**.



Export Business Terms and Database Column Assignments

Select Business Terms and Assigned Assets to export:

- ☑ DiscoveryTerms
  - ☑ USZC
  - ☑ SSN

Column Assignment

Database DBMS: DB2

Database Instance: ISD_SRC

Database Host: LOCALHOST

Database Name: BANK

Output File: c:\bg_export.out

Figure 8-39 Exporting business terms and column assignments

9. Open InfoSphere Business Glossary, and then import the file that InfoSphere Discovery generated. Verify that you see the imported business terms and assignments.

8.6 Reverse engineering a data model

Bank A accumulated a lot of data over the years. When it acquired Bank B, its data stores grew even more. The data analysts for Bank A believe that they understand their data fairly well. Although many SMEs from some of the older systems left or retired, a few remain.

When Bank A bought Bank B, Bank A decided to keep most of its systems and to onboard the data of Bank B into the systems of Bank A. In a few cases, Bank A decided to keep the systems for Bank B and onboard the data from Bank A into those systems.

The bank acquisition initiated several new initiatives, many of which revolved around customer satisfaction and experience. Some focused on brand recognition and market share. Most initiatives required data movement or data changes at one level or another. The fact that most initiatives involved data was no surprise.

Bank A and Bank B purchased several of their systems from outside vendors, rather than building them. Unfortunately, some vendors never provided data models for their applications. The vendors claimed their applications maintained the integrity of the data as it flowed from table to table. They said that there was no need to understand the inner workings of the tables. Over time, some systems worked better than others. It was not a perfect world, but Bank A and Bank B used coping mechanisms that worked to get the job done. At the end of the day, the banks understood their data was not perfect, but they hoped it was good enough.

Now, with the Bank B acquisition, Bank A faces the challenge of onboarding data into its systems, without having a complete picture of how the applications work at the table level. Many organizations today face the same challenge. InfoSphere Discovery can help by reverse-engineering data models. Through automated data analysis, it interrogates the data to uncover hidden relationships between tables, such as application-enforced primary key pr foreign key pairs, even when referential integrity constraints do not exist.

For example, by inferring that table A is related to table B, and table B is related to table C, and so on, InfoSphere Discovery constructs a graph of inferred relationships, much like a physical data model. This model is derived strictly by direct data analysis, without needing to provide any metadata up front. Because the relationships are inferred by direct data analysis, they might not all be correct. Think of them as candidate keys. An analyst or SME needs to review the results and approve the good relationships, before publishing the model.

After deriving the physical data model, InfoSphere Discovery generates a logical model by applying rules, or heuristics, to the graph of physical relationships. The logical models are called *data objects* in InfoSphere Discovery. Each data object has a root table and potentially one or more child tables that form a hierarchy. Data objects provide a more business-centric view of the data than the physical model, because each object tends to describe a business entity such as account or customer. For example, a data object for an entity, such as customer, might include CUSTOMERS as a root table, with ORDERS, ORDER_LINES, and SALES as children.

Reverse engineering data models provides large productivity gains. With the InfoSphere Discovery approach, you save time and reduce costs and risk mitigation, compared to traditional manual data analysis.

8.6.1 Primary-foreign key candidates

Primary-foreign keys (*PF Keys*) operate by certain rules, which, for performance reasons, are often not enforced directly by the database:

- Primary key columns must have a high uniqueness (also called *selectivity*), because a primary key value is not supposed to be repeated within the primary key column (or columns). InfoSphere Discovery calculates which columns have high selectivity when performing column analysis.
- Foreign key columns typically have low selectivity, because a value might be repeated multiple times, pointing to the primary key. Every foreign key value must be found in the primary key column. Exceptions to this rule are called *orphan records*, because the child exists without the parent. However, the values of a primary key do not need to be found in its foreign key columns. It is acceptable to have a parent without a child.

Consider a typical parent-child relationship, such as customers and orders. Every order (the child record) must point to a customer. Any order that does not point to a customer is an orphan. However, every customer does not need to point to an order. Some customers might not have placed any orders yet.

8.6.2 Discovering primary-foreign key candidates

During column analysis, InfoSphere Discovery derives the domains and value distributions for every column of every table it analyzes. To derive *PF Key* candidates, InfoSphere Discovery performs *column matches*, which calculate the amount of overlapping data values between column pairs. Then it applies a set of heuristics to determine if the relationship is strong enough to meet the requirements of a primary-foreign key pair, with a highly unique primary key, and a foreign key with a low orphan rate.

By default, InfoSphere Discovery searches for primary keys with 80% or greater selectivity on the primary key. It searches for foreign keys with an 80% or greater hit rate from the foreign key to the primary key. The analyst using InfoSphere Discovery raises or lowers these and other thresholds to match the characteristics of the data. With this method, InfoSphere Discovery finds both single-column keys and composite keys.

8.6.3 Displaying the results

InfoSphere Discovery displays the PF Key candidates in a graphical diagram and in a statistical grid. The statistical grid shows the degree to which values in one column match values in another column. The grid provides an easy way to see if there are orphan records or duplicate primary keys.

8.6.4 Data objects

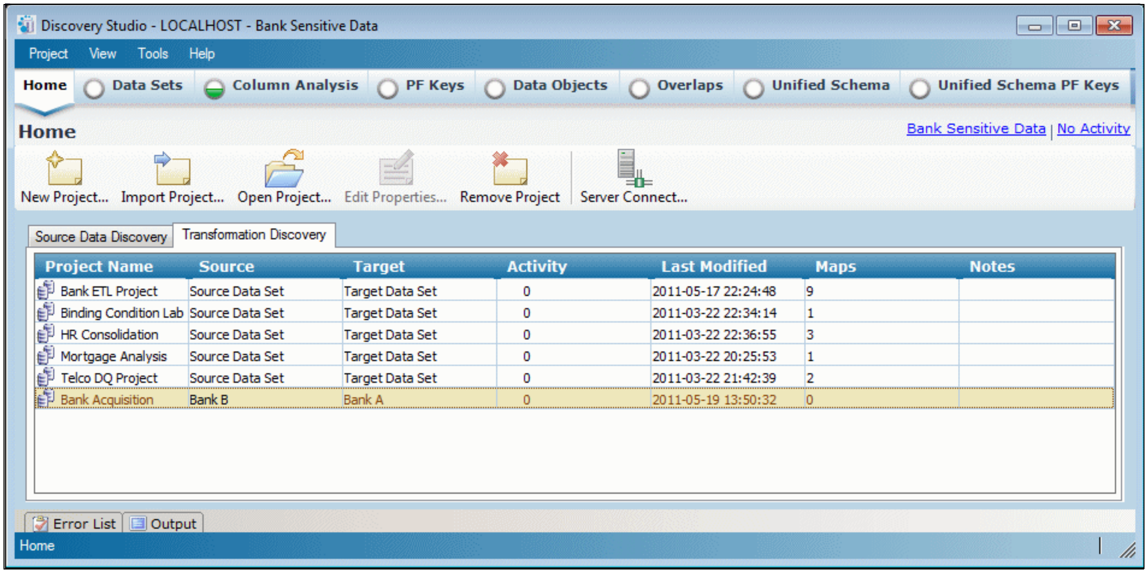
Generating data objects comes after discovering the PF keys. During the Data Objects step, InfoSphere Discovery analyzes the PF Keys results, rather than looking directly at the data. It reviews the PF Key candidates and applies heuristics to determine logical groups of tables. The results indicate how tables are related in a logical hierarchy. Each data object has one root table, and it might have one or more children. Each child might have one or more children as well.

In terms of discovering PF Keys and generating data objects, while the process of proposing candidates is automated, the process of approving them is not. A data analyst or SME must review the candidate keys and candidate data objects to determine the correct ones. InfoSphere Discovery does not guarantee that it will get the right results. It might produce *false positives* or rather results with good statistics, but that are not actual relationships.

8.6.5 Performing transformation discovery

In this scenario, for Bank A to identify PF Keys in its data and then generate data objects from them, complete these steps:

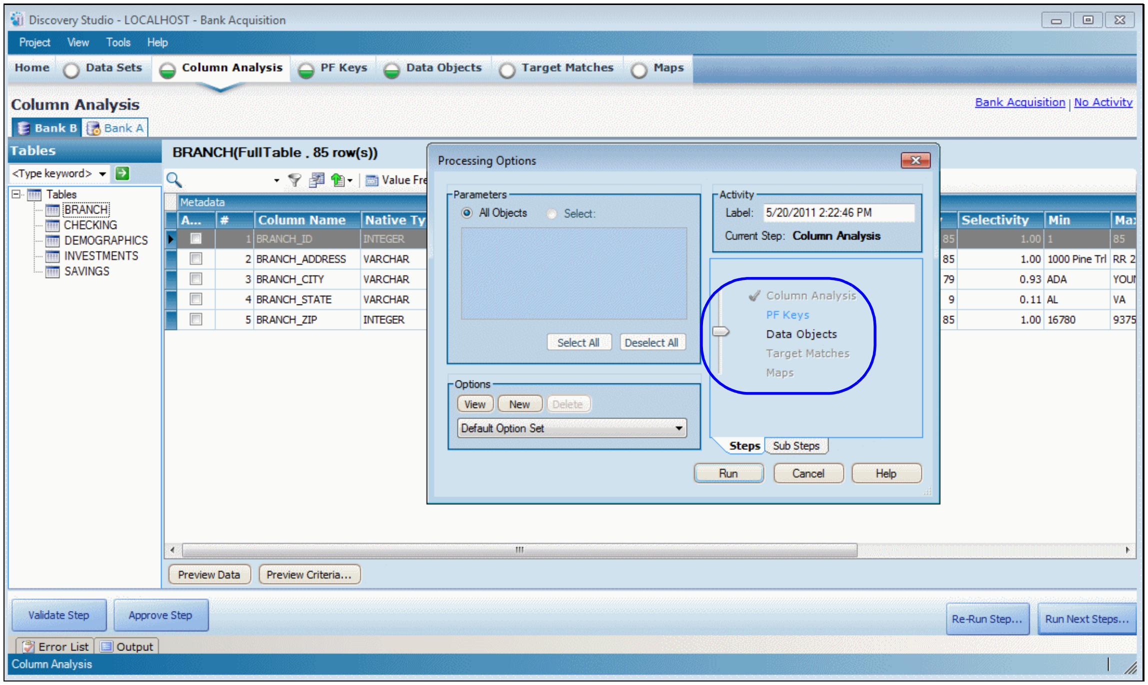
1. Start Discovery Studio.
2. On the **Transformation Discovery** tab (Figure 8-40), highlight the **Bank Acquisition** project, and then click the **Open Project** button.



Project Name	Source	Target	Activity	Last Modified	Maps	Notes
Bank ETL Project	Source Data Set	Target Data Set	0	2011-05-17 22:24:48	9	
Binding Condition Lab	Source Data Set	Target Data Set	0	2011-03-22 22:34:14	1	
HR Consolidation	Source Data Set	Target Data Set	0	2011-03-22 22:36:55	3	
Mortgage Analysis	Source Data Set	Target Data Set	0	2011-03-22 20:25:53	1	
Telco DQ Project	Source Data Set	Target Data Set	0	2011-03-22 21:42:39	2	
Bank Acquisition	Bank B	Bank A	0	2011-05-19 13:50:32	0	

Figure 8-40 Project list

3. On the **Column Analysis** tab, click **Run Next Steps** (Figure 8-41). In the Processing Options window (inset in Figure 8-41), drag the slider down to **Data Objects**. Then click **Run**.



Label: 5/20/2011 2:22:46 PM
Current Step: **Column Analysis**

A...	#	Column Name	Native Ty		Selectivity	Min	Ma:
☐	1	BRANCH_ID	INTEGER	85	1.00	1	85
☐	2	BRANCH_ADDRESS	VARCHAR	85	1.00	1000 Pine Trl	RR 2
☐	3	BRANCH_CITY	VARCHAR	79	0.93	ADA	YOU!
☐	4	BRANCH_STATE	VARCHAR	9	0.11	AL	VA
☐	5	BRANCH_ZIP	INTEGER	85	1.00	16780	9375

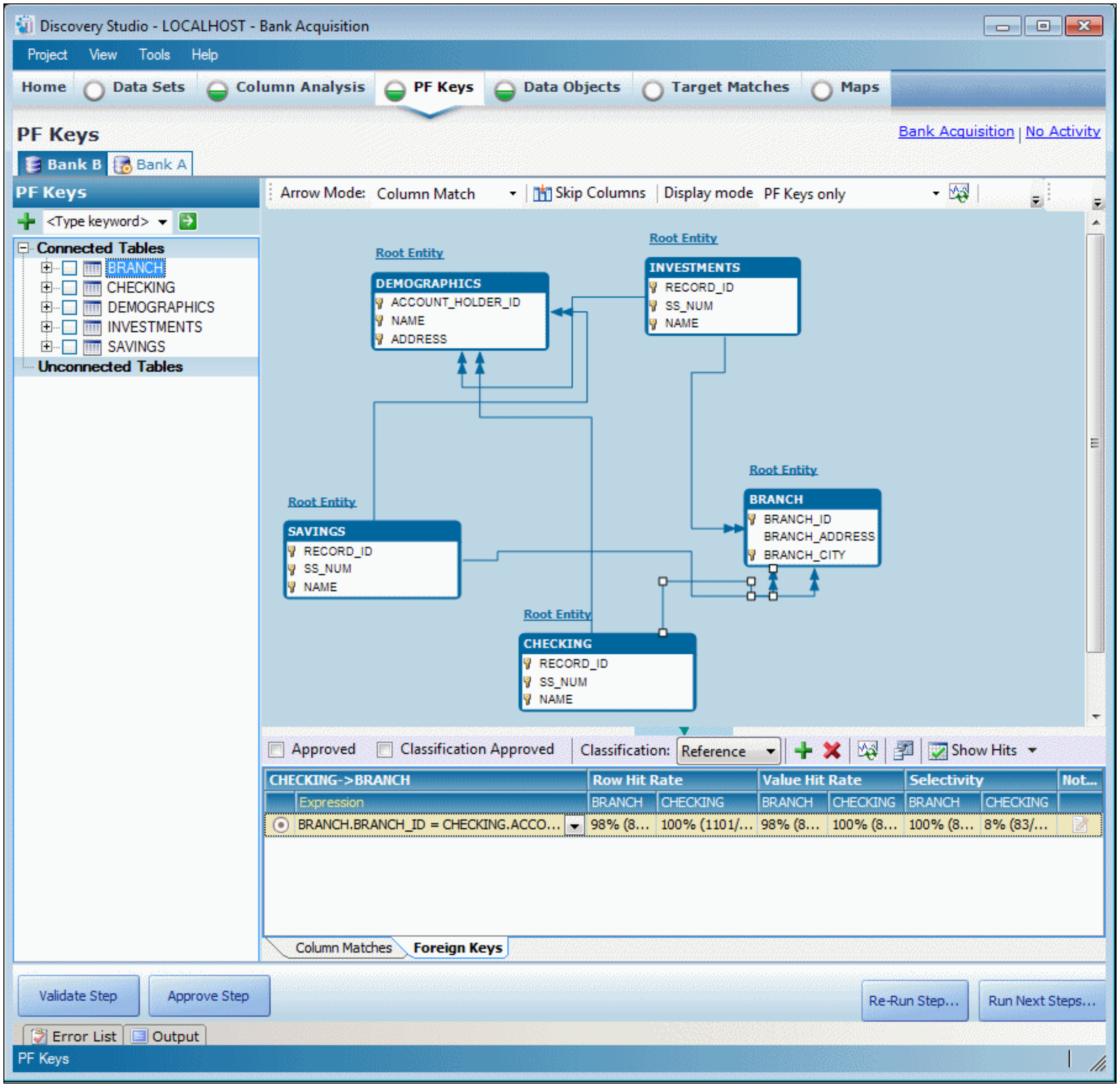
Figure 8-41 Using Column analysis to select the Data Objects processing action

4. Wait until the activity monitor changes from *Currently 1 Active task* to *No activity*. While the project task runs, InfoSphere Discovery locks the project to prevent changes.

5. Go to the **PF Keys** tab, and then highlight the **Bank B** data set. You see a diagram similar to the one shown in Figure 8-42. The diagram shows the PF Key candidates that InfoSphere Discovery discovered.

6. Click the solid line between **Checking** and **Branch** tables. The link represents a candidate PF Key with the following predicate:

```
BRANCH.BRANCH_ID = CHECKING.ACCOUNT_BRANCH
```



Arrow Mode: Column Match

Display mode PF Keys only

□ Approved □ Classification Approved Classification: Reference

CHECKING->BRANCH		Row Hit Rate		Value Hit Rate		Selectivity		Not...
	Expression	BRANCH	CHECKING	BRANCH	CHECKING	BRANCH	CHECKING	
◉	BRANCH.BRANCH_ID = CHECKING.ACCO...	98% (8...	100% (1101/...	98% (8...	100% (8...	100% (8...	8% (83/...	

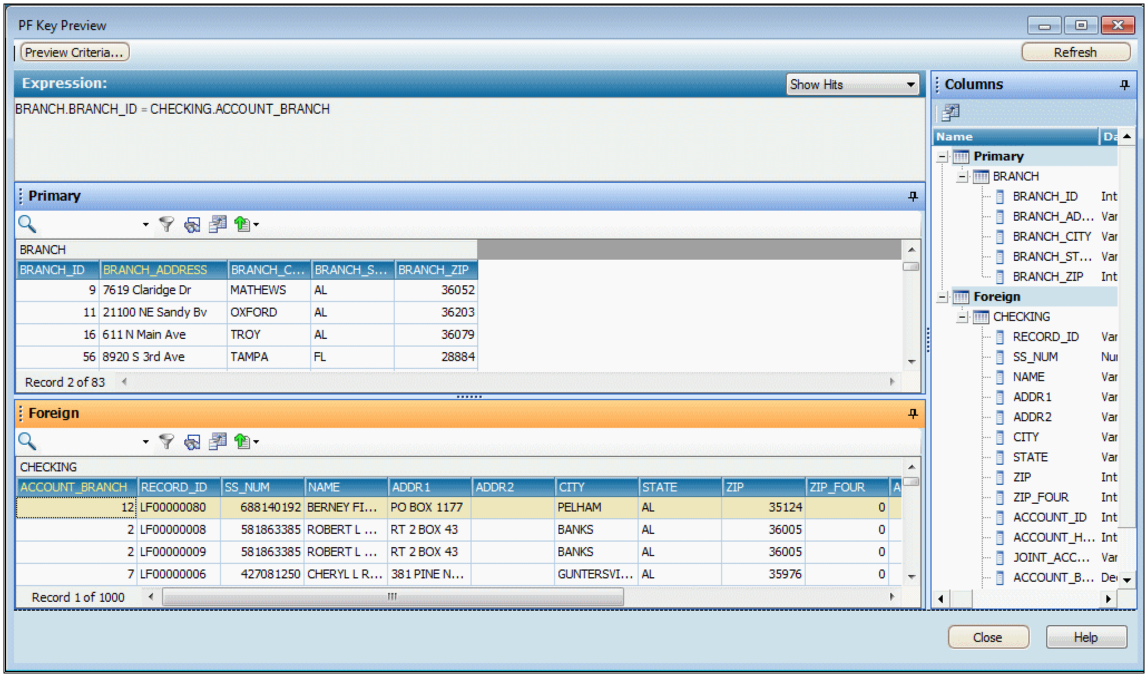
Figure 8-42 Bank B PF Keys

InfoSphere Discovery shows the statistics for this candidate key in the grid below the diagram. The statistics indicate that 100% of the values for the ACCOUNT_BRANCH column in the CHECKING table are found in the BRANCH_ID column of the BRANCH table. This result means that no orphan records are in the CHECKING table, with respect to the BRANCH table. All of the checking accounts belong to a branch.

Similarly, the statistics indicate that 98% of the values in the BRANCH_ID column of the BRANCH table are in the ACCOUNT_BRANCH column of the CHECKING table. That is, there are some branches without checking accounts. This result might seem unlikely at first, until you learn that some branches are dedicated to investment accounts and do not have standard checking accounts.

With InfoSphere Discovery, you can drill down to view the underlying data:

1. Click the **Show Hits** button in the toolbar. The window (Figure 8-43) that opens shows the data for the Primary BRANCH table in the upper panel and the data for the Foreign CHECKING table in the lower panel.
2. To view correlated data, select a primary or foreign row in the grid, and then click the **Focus** button. The button icon includes the image of a pair of binoculars.



BRANCH.BRANCH_ID = CHECKING.ACCOUNT_BRANCH

BRANCH

BRANCH_ID	BRANCH_ADDRESS	BRANCH_C...	BRANCH_S...	BRANCH_ZIP
9	7619 Claridge Dr	MATHEWS	AL	36052
11	21100 NE Sandy Bv	OXFORD	AL	36203
16	611 N Main Ave	TROY	AL	36079
56	8920 S 3rd Ave	TAMPA	FL	28884

Record 2 of 83

CHECKING

ACCOUNT_BRANCH	RECORD_ID	SS_NUM	NAME	ADDR1	ADDR2	CITY	STATE	ZIP	ZIP_FOUR	A
12	LF00000080	688140192	BERNEY FI...	PO BOX 1177		PELHAM	AL	35124	0	
2	LF00000008	581863385	ROBERT L ...	RT 2 BOX 43		BANKS	AL	36005	0	
2	LF00000009	581863385	ROBERT L ...	RT 2 BOX 43		BANKS	AL	36005	0	
7	LF00000006	427081250	CHERYL L R...	381 PINE N...		GUNTERSVI...	AL	35976	0	

Record 1 of 1000

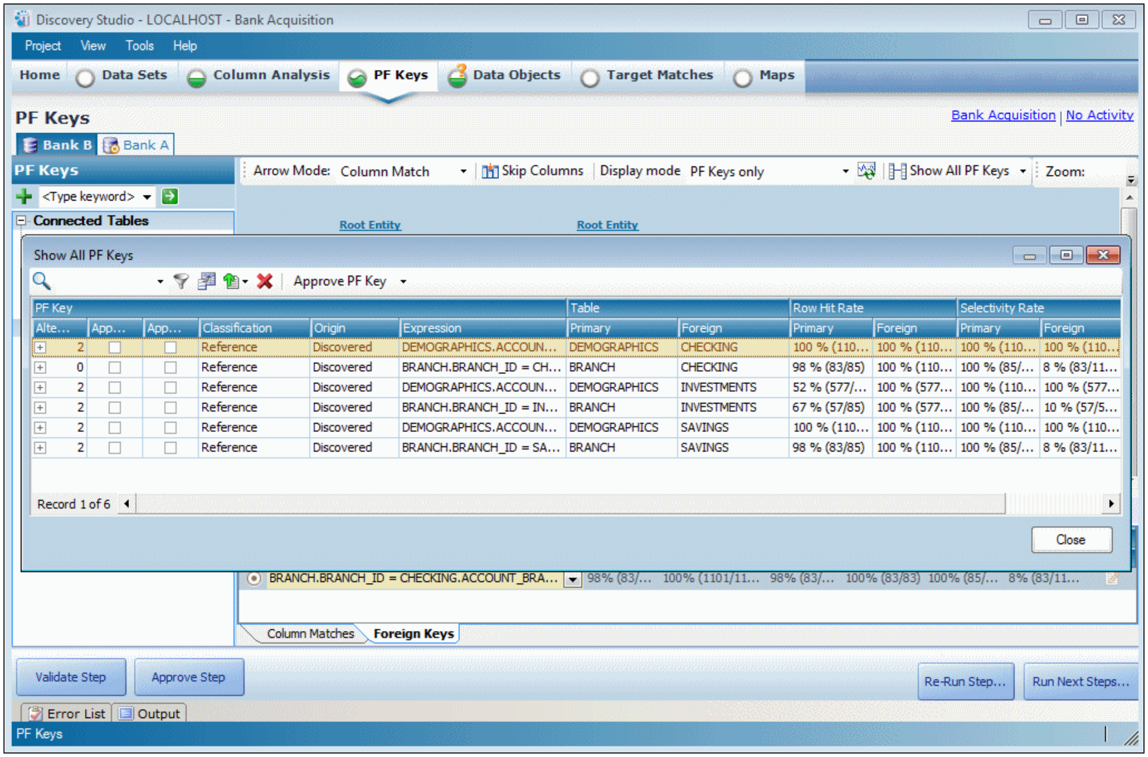
Name	Da
Primary	
BRANCH	
BRANCH_ID	Int
BRANCH_AD...	Var
BRANCH_CITY	Var
BRANCH_ST...	Var
BRANCH_ZIP	Int
Foreign	
CHECKING	
RECORD_ID	Var
SS_NUM	Nui
NAME	Var
ADDR1	Var
ADDR2	Var
CITY	Var
STATE	Var
ZIP	Int
ZIP_FOUR	Int
ACCOUNT_ID	Int
ACCOUNT_H...	Int
JOINT_ACC...	Var
ACCOUNT_B...	De

Figure 8-43 View of the underlying data by using the Show Hits button

3. Select the first row in the primary table. Click the **Focus** button. Now you only see rows in the foreign table that match the BRANCH_ID of the selected row in the primary table. Click **Close** to close the data preview window.

   In addition to the diagram, InfoSphere Discovery presents the PF Key candidates in a tabular report.

4. Click the **Show All PF Keys** button in the toolbar. In the Show All PF Keys window (Figure 8-44), you see a tabular report. Click **Close**.



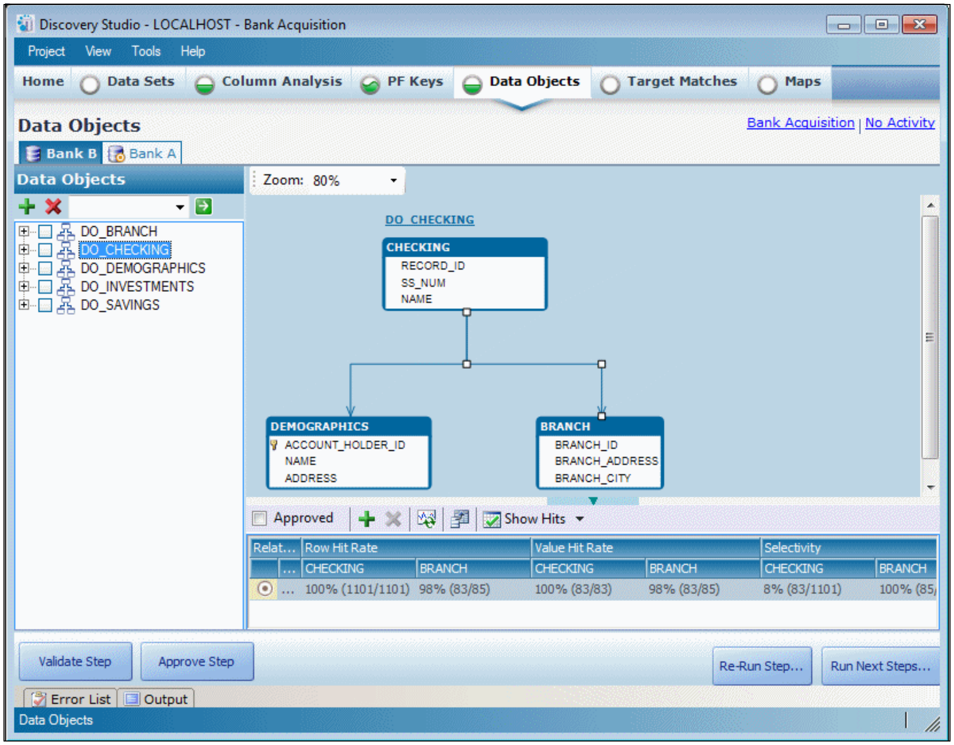
PF Key						Table		Row Hit Rate		Selectivity Rate	
Alte...	App...	App...	Classification	Origin	Expression	Primary	Foreign	Primary	Foreign	Primary	Foreign
2	□	□	Reference	Discovered	DEMOGRAPHICS.ACCOUN...	DEMOGRAPHICS	CHECKING	100 % (110...	100 % (110...	100 % (110...	100 % (110...
0	□	□	Reference	Discovered	BRANCH.BRANCH_ID = CH...	BRANCH	CHECKING	98 % (83/85)	100 % (110...	100 % (85/...	8 % (83/11...
2	□	□	Reference	Discovered	DEMOGRAPHICS.ACCOUN...	DEMOGRAPHICS	INVESTMENTS	52 % (577/...	100 % (577...	100 % (110...	100 % (577...
2	□	□	Reference	Discovered	BRANCH.BRANCH_ID = IN...	BRANCH	INVESTMENTS	67 % (57/85)	100 % (577...	100 % (85/...	10 % (57/5...
2	□	□	Reference	Discovered	DEMOGRAPHICS.ACCOUN...	DEMOGRAPHICS	SAVINGS	100 % (110...	100 % (110...	100 % (110...	100 % (110...
2	□	□	Reference	Discovered	BRANCH.BRANCH_ID = SA...	BRANCH	SAVINGS	98 % (83/85)	100 % (110...	100 % (85/...	8 % (83/11...

Record 1 of 6

BRANCH.BRANCH_ID = CHECKING.ACCOUNT_BRA... 98% (83/... 100% (1101/11... 98% (83/... 100% (83/83) 100% (85/... 8% (83/11...

Figure 8-44 View of the data by using the Show All PF Keys button

5. On the **Data Objects** tab, click the **Bank B** tab, and then select the **CUSTOMERS** table in the left pane (Figure 8-45). The diagram indicates that the CUSTOMERS table is related to the DEMOGRAPHICS and BRANCH tables.



Relat...		Row Hit Rate		Value Hit Rate		Selectivity	
	...	CHECKING	BRANCH	CHECKING	BRANCH	CHECKING	BRANCH
○	...	100% (1101/1101)	98% (83/85)	100% (83/83)	98% (83/85)	8% (83/1101)	100% (85/

Figure 8-45 Data objects for Bank B

PF Keys provides a view similar to a physical data model, and Data Objects provides a view similar to a logical data model.

This information is actionable. For example, if Bank A wants to archive inactive customers from the CUSTOMERS table, the data object that InfoSphere Discovery generated indicates to also archive the data from the DEMOGRAPHICS and BRANCH tables. Archiving data from the CHECKING table alone will not adequately archive all of the related data.

Bank A is satisfied. After reviewing the PF Key candidates and the data objects, the bank and its employees feel more prepared to tackle its many information-centric initiatives.

8.7 Performing value overlap analysis

For several years, before Bank A bought Bank B, both banks stayed busy propagating their data. The offshore test team needed test data for the new application they were developing. The marketing team built a data mart for predictive analytics. The data warehouse team stored everything, at least twice and indefinitely.

How can Bank A and Bank B determine where they have redundant data? The *Overlaps* feature of InfoSphere Discovery identifies column pairs with overlapping data values. Many overlapping values might exist, or just a few might exist. InfoSphere Discovery calculates the degree of overlap and presents the results statistically in a drill-down grid. In set theory, the calculated overlap is a classic intersection.

You represent an overlap with the following notation:

```
Left_Table.Column_m : Right_Table.Column_n
```

Some values in Column_m match some values in Column_n. Left_Table, and Right_Table must be in different data sets for InfoSphere Discovery to identify value overlaps. Overlapping results include the names of the table-column pairs, the percentage of matching values from one column to the other, and the number of rows that match each other.

When Bank A buys Bank B, Bank A wants to know how many customers it shares with Bank B. It runs overlap analysis to compare the data in Bank A with Bank B. If the bank finds a lot of matching information, such as names or addresses, many customers are probably customers of both banks.

8.7.1 Running overlap analysis

From a workflow perspective, you run the Overlaps task any time after you perform Column Analysis. Overlaps follows Data Objects in the workflow bar, but you do not need to run PF Keys or Data Objects before running Overlaps.

To run overlap analysis on the data of the bank, complete these steps:

1. Create a project similar to the Bank Acquisition project (Figure 8-5 on page 215):
   a. For Type, select **Source Data Discovery**.
   b. For the name, type a project name, such as `Bank Overlap Analysis`.
   c. At the bottom of the General group box, clear the **Use Byte Storage For String in Staging** check box.
   d. In the Password group box, clear the **Use Password** check box.
   e. Click **OK**.
2. Rename the existing data set to `Bank B`.

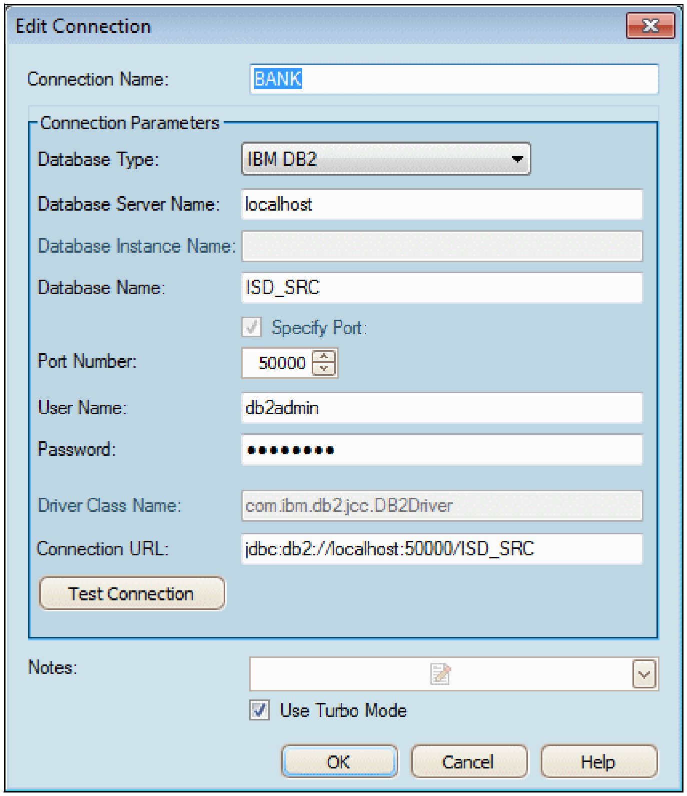
3. In the Edit Connection dialog box (Figure 8-46), create a data connection for Bank B:

   a. Complete the following fields:

      - Database Type
      - Database Server Name
      - Database Name
      - Port Number
      - User Name
      - Password
      - Connection URL

   b. Click the **Test Connection** button to ensure that the connection information is correct.

   c. Click **OK**. The Edit Connection window closes, and you see the **Data Sets** tab again.



Edit Connection

Connection Name: BANK

Connection Parameters

Database Type: IBM DB2

Database Server Name: localhost

Database Instance Name:

Database Name: ISD_SRC

☑ Specify Port:

Port Number: 50000

User Name: db2admin

Password: ●●●●●●●●●

Driver Class Name: com.ibm.db2.jcc.DB2Driver

Connection URL: jdbc:db2://localhost:50000/ISD_SRC

Notes:

☑ Use Turbo Mode

Figure 8-46 Bank data connection

4. Import the following tables from the Bank schema to the Bank B data set, as shown in Figure 8-47.

   – BRANCH
   – CHECKING
   – DEMOGRAPHICS
   – INVESTMENTS
   – SAVINGS

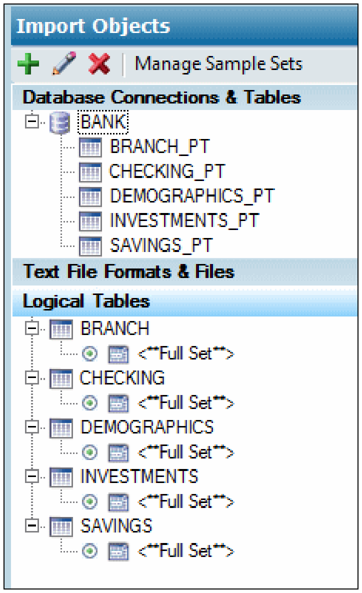


Figure 8-47 Bank Logical Tables

5. Right-click the **Bank B** data set tab, and select **Add a Data Set**. Rename the new data set to `Bank A`.

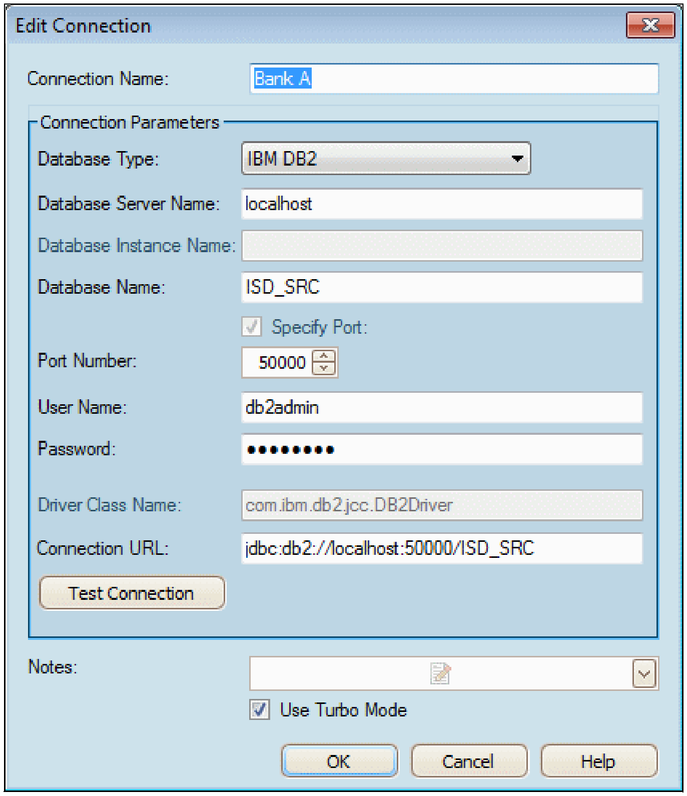
6. Create a data connection for Bank A, as shown in Figure 8-48 on page 263:

   a. Complete the following fields:

      - Database Type
      - Database Server Name
      - Database Name
      - Port Number
      - User Name
      - Password
      - Connection URL

   b. Click the **Test Connection** button to ensure that the connection information is correct.

c. Click **OK**. The Edit Connection window closes, and you see the **Data Sets** tab again.



Edit Connection

Connection Name: Bank A

Connection Parameters

Database Type: IBM DB2

Database Server Name: localhost

Database Instance Name:

Database Name: ISD_SRC

☑ Specify Port:

Port Number: 50000

User Name: db2admin

Password: ••••••••

Driver Class Name: com.ibm.db2.jcc.DB2Driver

Connection URL: jdbc:db2://localhost:50000/ISD_SRC

Notes:

☑ Use Turbo Mode

Figure 8-48 Bank A data connection

7. Import the ACCOUNTS table from the Bank schema into the Bank A data set, as shown in Figure 8-49.

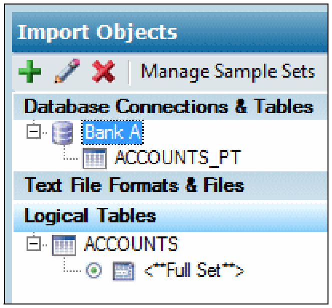


Figure 8-49 ACCOUNTS table imported into the Bank A data set

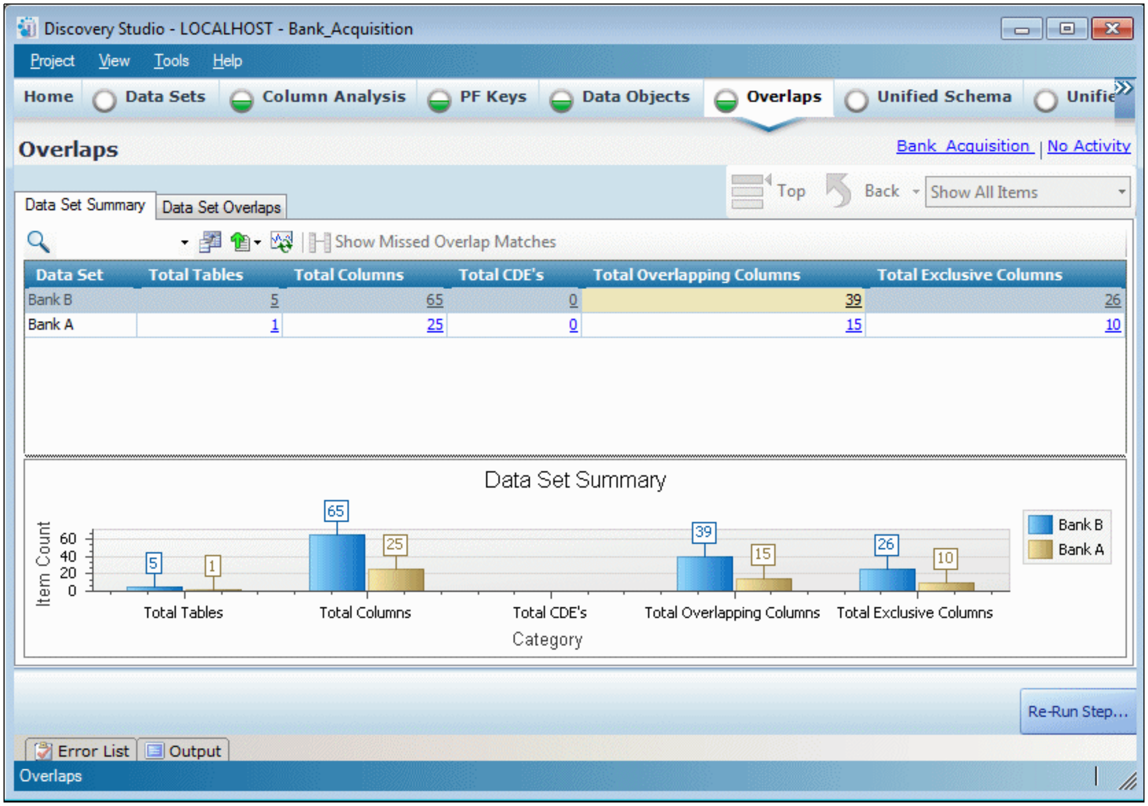
8. From the **Data Sets** tab, click **Run Next Steps**.

9. In the Processing Options window, drag the slider down to the **Overlaps** step, and then click **Run**.

10. After all of the project tasks are completed, go to the **Overlaps** tab (Figure 8-50).

    InfoSphere Discovery presents overlap results in a drill-down tabular grid. In the upper level, you see the number of data sets that are participating in the overlap analysis. For each data set, the grid shows the total number of tables and columns in each data set. Next to that information, you see the total number of overlapping columns and the total number of exclusive columns.

    In addition, you see that the Bank B data set has 39 columns that overlap with the columns in the Bank A data set. Bank B has 26 exclusive columns, which means that those columns do not overlap with any columns in the Bank A data set. The *overlapping columns* and *exclusive columns* values are hyperlinks. You click a link to view the next level of detail.

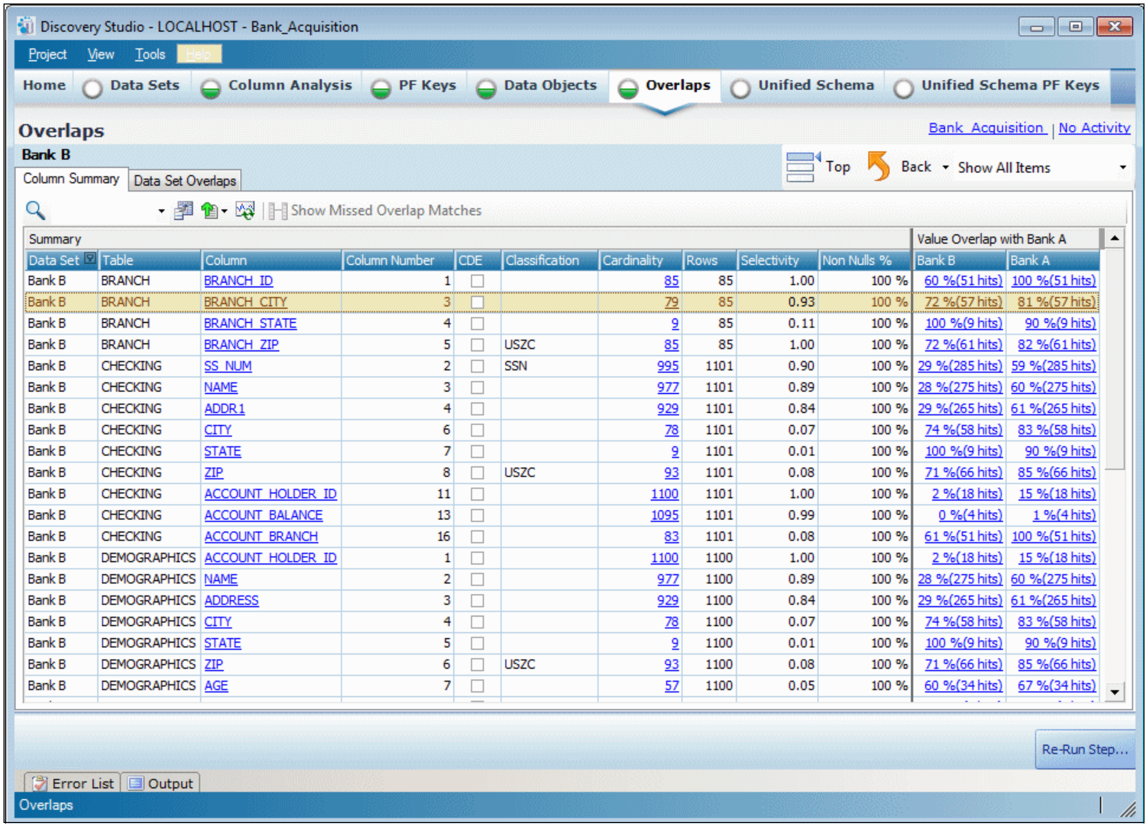


Data Set	Total Tables	Total Columns	Total CDE's	Total Overlapping Columns	Total Exclusive Columns
Bank B	5	65	0	39	26
Bank A	1	25	0	15	10

Figure 8-50 Overlaps home page

8.7.2 Column Summary

Under Total Overlapping Columns for Bank B, click the hyperlinked **39**. You see the Column Summary page (Figure 8-51). The Column Summary page shows a grid with several statistics. All of the tables on the left side of the overlap come from the Bank B data set. The highlighted row indicates that the SS_NUM column in the CHECKING table overlaps with at least one column in the Bank A data set.



Summary										Value Overlap with Bank A	
Data Set	Table	Column	Column Number	CDE	Classification	Cardinality	Rows	Selectivity	Non Nulls %	Bank B	Bank A
Bank B	BRANCH	BRANCH_ID	1	□		85	85	1.00	100 %	60 %(51 hits)	100 %(51 hits)
Bank B	BRANCH	BRANCH_CITY	3	□		79	85	0.93	100 %	72 %(57 hits)	81 %(57 hits)
Bank B	BRANCH	BRANCH_STATE	4	□		9	85	0.11	100 %	100 %(9 hits)	90 %(9 hits)
Bank B	BRANCH	BRANCH_ZIP	5	□	USZC	85	85	1.00	100 %	72 %(61 hits)	82 %(61 hits)
Bank B	CHECKING	SS_NUM	2	□	SSN	995	1101	0.90	100 %	29 %(285 hits)	59 %(285 hits)
Bank B	CHECKING	NAME	3	□		977	1101	0.89	100 %	28 %(275 hits)	60 %(275 hits)
Bank B	CHECKING	ADDR1	4	□		929	1101	0.84	100 %	29 %(265 hits)	61 %(265 hits)
Bank B	CHECKING	CITY	6	□		78	1101	0.07	100 %	74 %(58 hits)	83 %(58 hits)
Bank B	CHECKING	STATE	7	□		9	1101	0.01	100 %	100 %(9 hits)	90 %(9 hits)
Bank B	CHECKING	ZIP	8	□	USZC	93	1101	0.08	100 %	71 %(66 hits)	85 %(66 hits)
Bank B	CHECKING	ACCOUNT_HOLDER_ID	11	□		1100	1101	1.00	100 %	2 %(18 hits)	15 %(18 hits)
Bank B	CHECKING	ACCOUNT_BALANCE	13	□		1095	1101	0.99	100 %	0 %(4 hits)	1 %(4 hits)
Bank B	CHECKING	ACCOUNT_BRANCH	16	□		83	1101	0.08	100 %	61 %(51 hits)	100 %(51 hits)
Bank B	DEMOGRAPHICS	ACCOUNT_HOLDER_ID	1	□		1100	1100	1.00	100 %	2 %(18 hits)	15 %(18 hits)
Bank B	DEMOGRAPHICS	NAME	2	□		977	1100	0.89	100 %	28 %(275 hits)	60 %(275 hits)
Bank B	DEMOGRAPHICS	ADDRESS	3	□		929	1100	0.84	100 %	29 %(265 hits)	61 %(265 hits)
Bank B	DEMOGRAPHICS	CITY	4	□		78	1100	0.07	100 %	74 %(58 hits)	83 %(58 hits)
Bank B	DEMOGRAPHICS	STATE	5	□		9	1100	0.01	100 %	100 %(9 hits)	90 %(9 hits)
Bank B	DEMOGRAPHICS	ZIP	6	□	USZC	93	1100	0.08	100 %	71 %(66 hits)	85 %(66 hits)
Bank B	DEMOGRAPHICS	AGE	7	□		57	1100	0.05	100 %	60 %(34 hits)	67 %(34 hits)

Figure 8-51 Overlaps Column Summary page

The Column Summary page does not indicate the columns on the right side of the overlap. You must drill down one more level to see the tables and columns in the other side.

The Column Summary page has the following rows:

Data Set The name of the data set holding the table in the left side of the overlap.

Table The name of the table in the left side of the overlap.

Column The name of the column in the left side of the overlap.

Column Number The position of the column in the table.

CDE Selected if this column is defined as a *critical data element.*

Cardinality The cardinality of this column.

Rows The number of rows in the table in the left side of the overlap.

Selectivity The selectivity of the column in the left side of the overlap.

Non Nulls % The percentage of rows in the left table that are non-NULL.

A vertical bar separates the left and right sections of the page. The right section shows the overlap percentages between the left column and one or more columns from the other data sets.

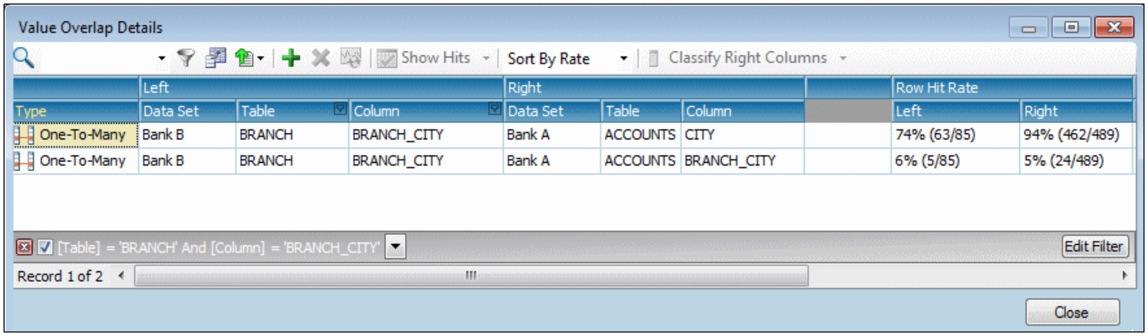
An overlap represents the degree to which values in the left column overlap with values in the right column. It also represents the degree that values in the right column overlap with values in the left column. Therefore, an overlap needs two percentages: left-to-right and right-to-left.

8.7.3 Viewing value overlap details

To see value overlap details, complete these steps:

1. Click the **72% (57 Hits)** link in the row for the BRANCH.BRANCH_CITY column in Bank B.

   This link takes you one level deeper to the Value Overlap Detail window (Figure 8-52). Two columns in Bank A, ACCOUNTS.CITY and ACCOUNTS.BRANCH_CITY, overlap to a degree with values in the BRANCH.BRANCH_CITY column in Bank B. The Row Hit Rate statistics indicate that 74% of the values in BRANCH.BRANCH_CITY overlap with values in ACCOUNTS.CITY. Looking right to left, 94% of the values in ACCOUNTS.CITY overlap with values in BRANCH.BRANCH_CITY.



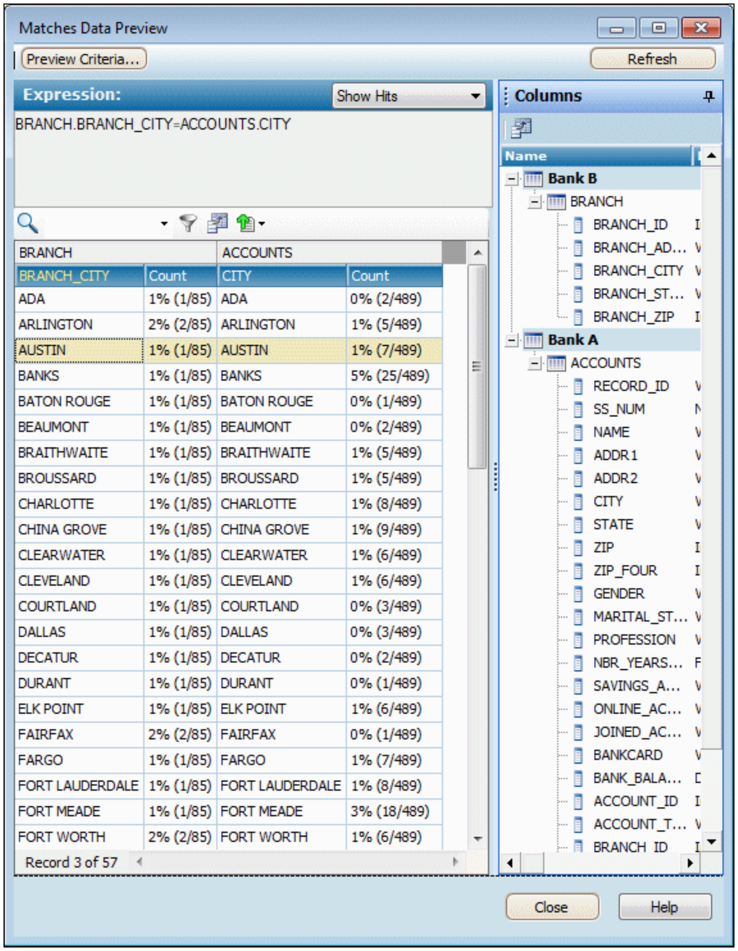
	Left			Right				Row Hit Rate	
Type	Data Set	Table	Column	Data Set	Table	Column		Left	Right
One-To-Many	Bank B	BRANCH	BRANCH_CITY	Bank A	ACCOUNTS	CITY		74% (63/85)	94% (462/489)
One-To-Many	Bank B	BRANCH	BRANCH_CITY	Bank A	ACCOUNTS	BRANCH_CITY		6% (5/85)	5% (24/489)

☒ ☑ [Table] = 'BRANCH' And [Column] = 'BRANCH_CITY'

Figure 8-52 Value Overlap Details

2. Select the first row for ACCOUNTS.CITY.
3. Click the **Show Hits** button in the toolbar.

   The Matches Data Preview window (Figure 8-53) opens showing the matching values for the overlap BRANCH.BRANCH_CITY: ACCOUNTS.CITY. At the simplest level, the data preview shows the list of cities that match across both columns.



Show Hits

BRANCH.BRANCH_CITY=ACCOUNTS.CITY

BRANCH		ACCOUNTS	
BRANCH_CITY	Count	CITY	Count
ADA	1% (1/85)	ADA	0% (2/489)
ARLINGTON	2% (2/85)	ARLINGTON	1% (5/489)
AUSTIN	1% (1/85)	AUSTIN	1% (7/489)
BANKS	1% (1/85)	BANKS	5% (25/489)
BATON ROUGE	1% (1/85)	BATON ROUGE	0% (1/489)
BEAUMONT	1% (1/85)	BEAUMONT	0% (2/489)
BRAITHWAITE	1% (1/85)	BRAITHWAITE	1% (5/489)
BROUSSARD	1% (1/85)	BROUSSARD	1% (5/489)
CHARLOTTE	1% (1/85)	CHARLOTTE	1% (8/489)
CHINA GROVE	1% (1/85)	CHINA GROVE	1% (9/489)
CLEARWATER	1% (1/85)	CLEARWATER	1% (6/489)
CLEVELAND	1% (1/85)	CLEVELAND	1% (6/489)
COURTLAND	1% (1/85)	COURTLAND	0% (3/489)
DALLAS	1% (1/85)	DALLAS	0% (3/489)
DECATUR	1% (1/85)	DECATUR	0% (2/489)
DURANT	1% (1/85)	DURANT	0% (1/489)
ELK POINT	1% (1/85)	ELK POINT	1% (6/489)
FAIRFAX	2% (2/85)	FAIRFAX	0% (1/489)
FARGO	1% (1/85)	FARGO	1% (7/489)
FORT LAUDERDALE	1% (1/85)	FORT LAUDERDALE	1% (8/489)
FORT MEADE	1% (1/85)	FORT MEADE	3% (18/489)
FORT WORTH	2% (2/85)	FORT WORTH	1% (6/489)

Record 3 of 57

Figure 8-53 Match Data Preview window

What can the bank do with these results? A lot. Now Bank A knows where it has redundant data, and it knows how many customers are shared between Bank A and Bank B. As a whole, the results help Bank A understand the trustworthiness of each of its systems.

After performing column analysis, primary and foreign key analysis, data object generation and overlap analysis, the bank is much better prepared for any initiatives that use the data of the bank. Because most of the projects of the bank are centered around information, the implication is that InfoSphere Discovery is a useful asset that provides many benefits.

8.8 Discovering transformation logic

One year after Bank A buys Bank B, the data quality team raises questions about data quality in the data warehouse of Bank A. The data integration team developed routines to load data from Bank B into the existing data warehouse for Bank A. Recently, someone in finance noticed discrepancies in some finance reports.

Did the alleged problems result from clerical errors, bad data integration code, or something else? Currently, the bank does not know the answer. Fortunately, one of the data quality team members knows that the bank has InfoSphere Discovery. After a few phone calls and emails, the bank decides to use InfoSphere Discovery to reverse engineer transformations between the Bank B data and the Bank A data warehouse.

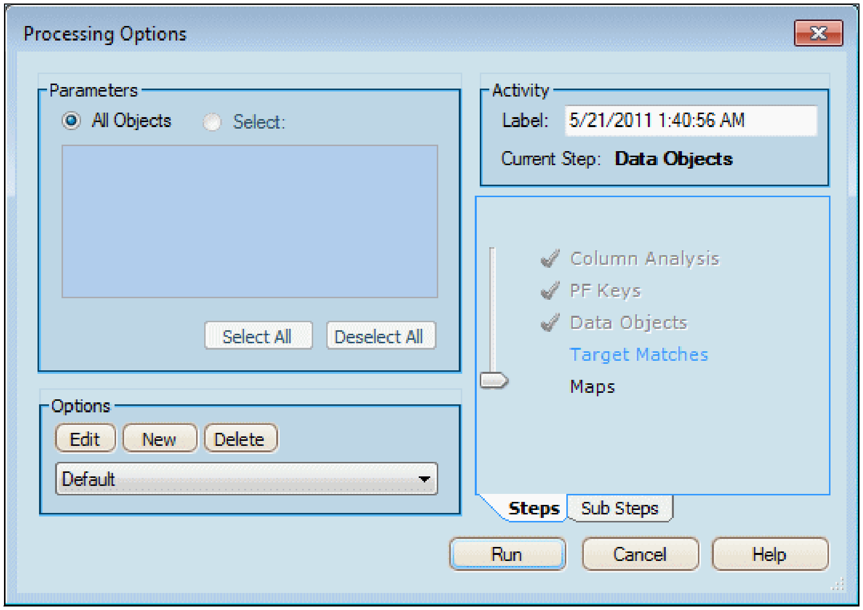
After reverse engineering the transformations, the discovery team helps the bank use the information to validate the flow of data from Bank B to the data warehouse. If the team finds bad transformations, or bad data, the appropriate teams can fix them. Fortunately, Bank A uses IBM InfoSphere QualityStage and InfoSphere DataStage. The discovery team can send the results from InfoSphere Discovery to IBM InfoSphere FastTrack and then to InfoSphere QualityStage or InfoSphere DataStage. When the transformations are in those products, the discovery team is one step closer to fixing the problem.

The transformation analysis feature of InfoSphere Discovery is perfect for this scenario. When you point it to a source and target, it analyzes the data and tells you how to move the data from the source system to the target system. The transformation results for each target column are easy to understand, because they are expressed as Structured Query Language (SQL) transformations.

8.8.1 Performing a transformation discovery

To perform transformation analysis on the data for Bank A, complete these steps to help you discover transformations from the Bank B data to the Bank A data warehouse:

1. Open the transformation analysis project called **Bank Acquisition**. You already ran the Column Analysis, PF Keys, and Data Objects steps.
2. Click **Run Next Steps**.
3. In the Processing Options pane (Figure 8-54), click and drag the slider in the lower right pane to the **Maps** step. Click **Run**.



Processing Options

Parameters: ● All Objects ○ Select:

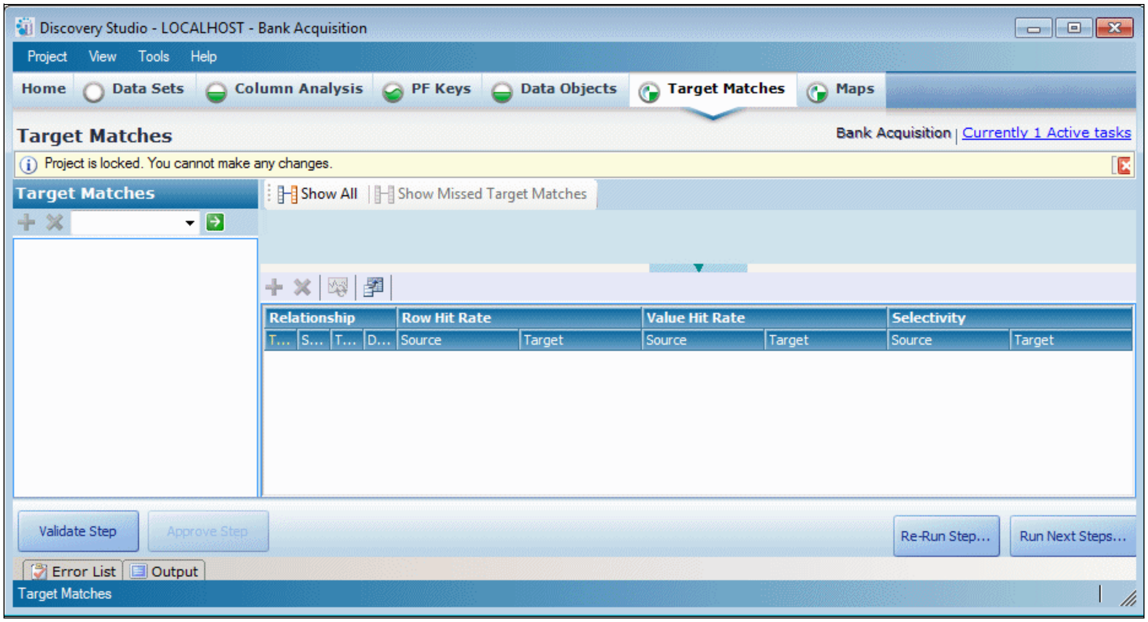
Activity
Label: 5/21/2011 1:40:56 AM
Current Step: **Data Objects**

Options: Default

✓ Column Analysis
✓ PF Keys
✓ Data Objects
Target Matches
Maps

Figure 8-54 Processing Options for Maps

While the project runs, the project remains locked as indicated by the message shown in the example in Figure 8-55. At first, no maps exist.



Project is locked. You cannot make any changes.

Figure 8-55 Target Matches showing that project is locked

8.8.2 Reviewing maps

When the transformation discovery process is complete, several table maps are shown on the **Maps** tab (Figure 8-56 on page 271).

The **Maps** tab has a subworkflow, which includes the following six items:

Summary Indicates the source and target tables and all of the transformations.

Joins If there is more than one source table, you define the join condition in this area.

Bindings The join condition that aligns the source and target rows.

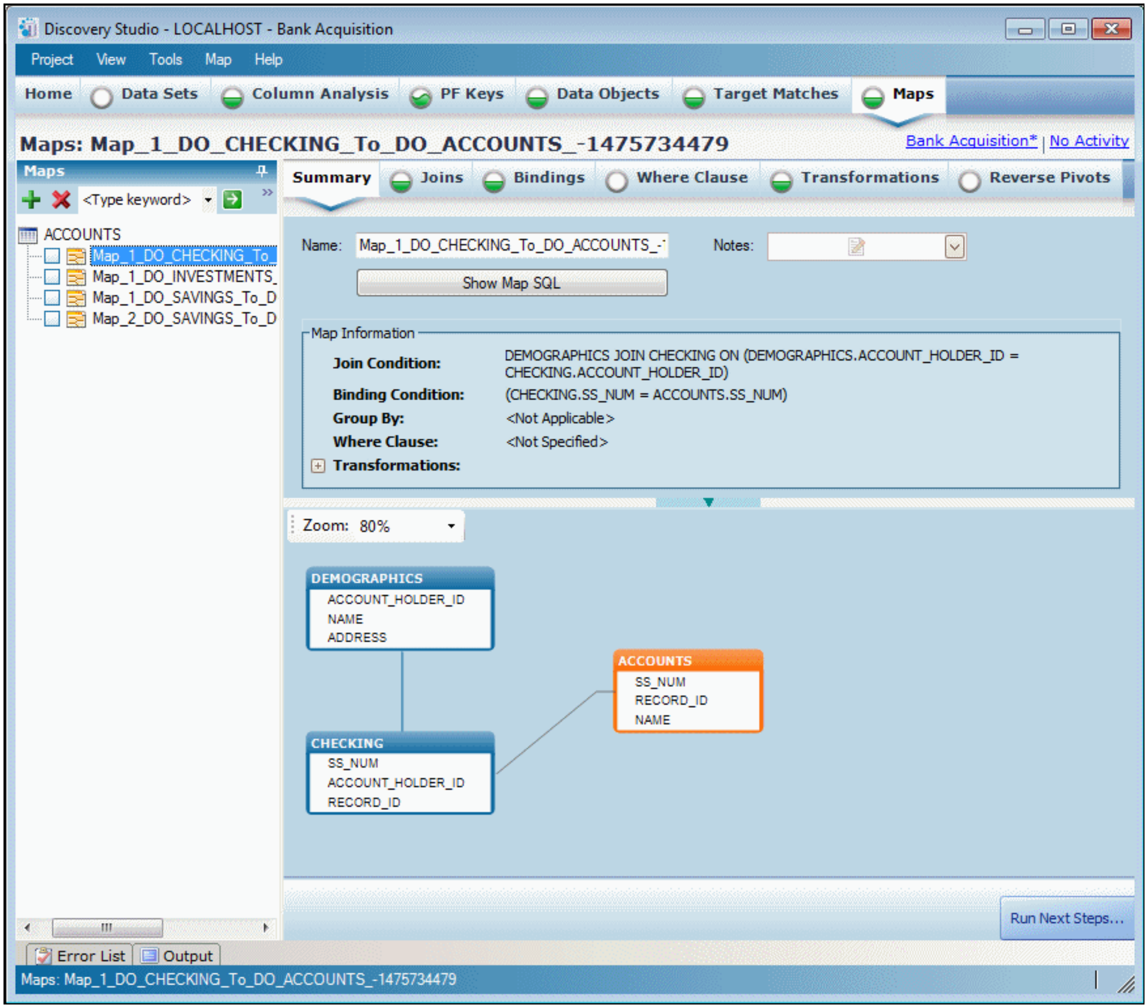
Where Clause An SQL WHERE clause, if needed, for the source tables.

Transformations A target-level view of the transformations.

Reverse Pivots Define pivot groups when your map requires a reverse pivot.

To view a map, complete these steps:

1. From the four maps for this project (left pane in Figure 8-56), select the first map, **Map_1_DO_CHECKING** (with other characters after it). The details of the map are displayed in the right pane.



Maps: Map_1_DO_CHECKING_To_DO_ACCOUNTS_-1475734479

Name: Map_1_DO_CHECKING_To_DO_ACCOUNTS_-1

Notes:

Map Information

Join Condition: DEMOGRAPHICS JOIN CHECKING ON (DEMOGRAPHICS.ACCOUNT_HOLDER_ID = CHECKING.ACCOUNT_HOLDER_ID)

Binding Condition: (CHECKING.SS_NUM = ACCOUNTS.SS_NUM)

Group By: <Not Applicable>

Where Clause: <Not Specified>

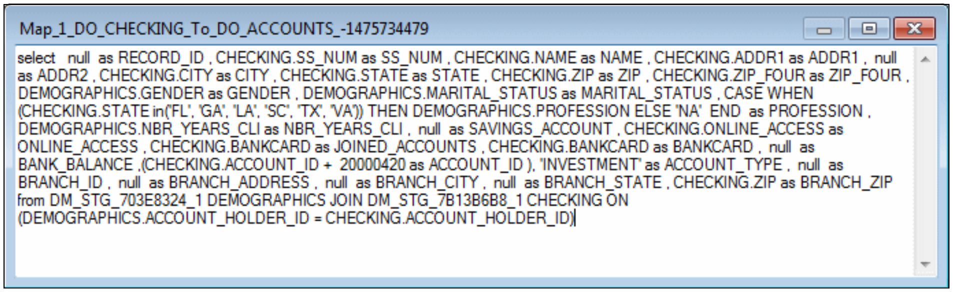
Transformations:

Maps: Map_1_DO_CHECKING_To_DO_ACCOUNTS_-1475734479

Figure 8-56 Maps summary page

The diagram indicates that Map_1_DO_CHECKING uses the DEMOGRAPHICS table and the CHECKING table as sources. The only target table in this project is ACCOUNTS, meaning that all maps will use ACCOUNTS as the target.

2. Click the **Show Map SQL** button. A window opens similar to the one in Figure 8-57. This window shows, in one long SQL statement, the transformations for all of the target columns in the map.



```
select  null  as RECORD_ID , CHECKING.SS_NUM as SS_NUM , CHECKING.NAME as NAME , CHECKING.ADDR1 as ADDR1 ,  null
as ADDR2 , CHECKING.CITY as CITY , CHECKING.STATE as STATE , CHECKING.ZIP as ZIP , CHECKING.ZIP_FOUR as ZIP_FOUR ,
DEMOGRAPHICS.GENDER as GENDER , DEMOGRAPHICS.MARITAL_STATUS as MARITAL_STATUS , CASE WHEN
(CHECKING.STATE in('FL', 'GA', 'LA', 'SC', 'TX', 'VA')) THEN DEMOGRAPHICS.PROFESSION ELSE 'NA'  END  as PROFESSION ,
DEMOGRAPHICS.NBR_YEARS_CLI as NBR_YEARS_CLI ,  null  as SAVINGS_ACCOUNT , CHECKING.ONLINE_ACCESS as
ONLINE_ACCESS , CHECKING.BANKCARD as JOINED_ACCOUNTS , CHECKING.BANKCARD as BANKCARD ,  null  as
BANK_BALANCE ,(CHECKING.ACCOUNT_ID +  20000420 as ACCOUNT_ID ), 'INVESTMENT' as ACCOUNT_TYPE ,  null  as
BRANCH_ID ,  null  as BRANCH_ADDRESS ,  null  as BRANCH_CITY ,  null  as BRANCH_STATE , CHECKING.ZIP as BRANCH_ZIP
from DM_STG_703E8324_1 DEMOGRAPHICS JOIN DM_STG_7B13B6B8_1 CHECKING ON
(DEMOGRAPHICS.ACCOUNT_HOLDER_ID = CHECKING.ACCOUNT_HOLDER_ID)
```

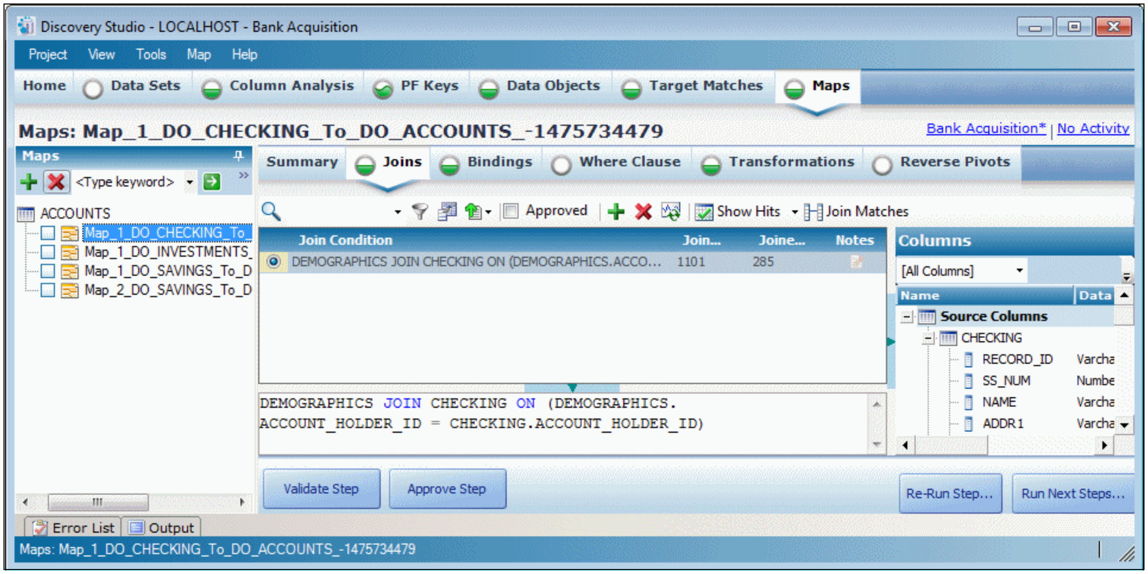
Figure 8-57   Show map SQL

3. Close the Show Map SQL window.

Joins

Click the **Joins** tab (Figure 8-58). The two source tables, DEMOGRAPHICS and CHECKING, have the following join condition:

```
DEMOGRAPHICS JOIN CHECKING ON (DEMOGRAPHICS.ACCOUNT_HOLDER_ID =
CHECKING.ACCOUNT_HOLDER_ID)
```



Join Condition	Join...	Joine...	Notes
● DEMOGRAPHICS JOIN CHECKING ON (DEMOGRAPHICS.ACCO...	1101	285	

```
DEMOGRAPHICS JOIN CHECKING ON (DEMOGRAPHICS.
ACCOUNT_HOLDER_ID = CHECKING.ACCOUNT_HOLDER_ID)
```

Columns

[All Columns]

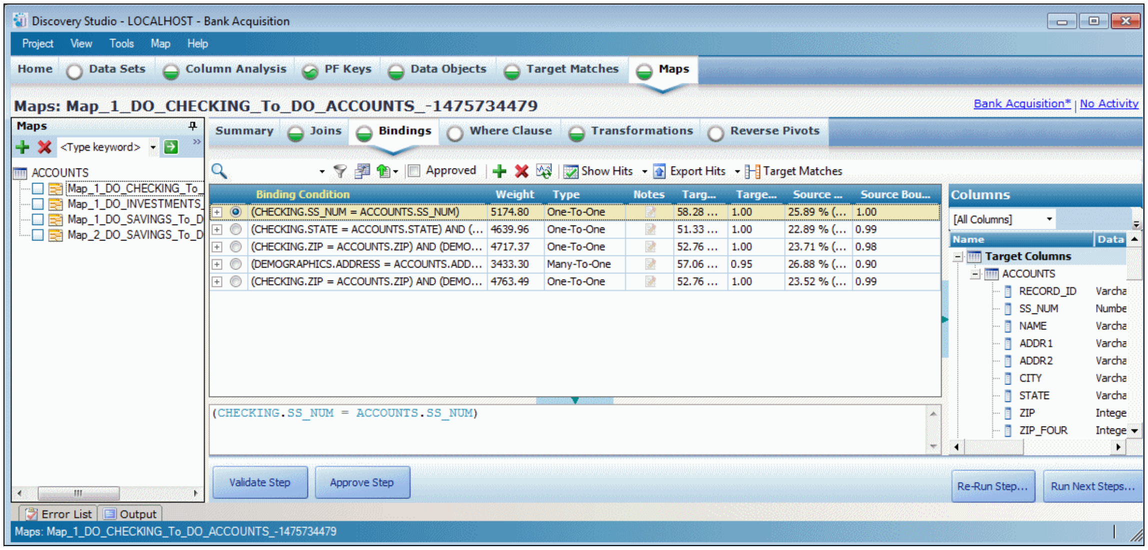
Name	Data
Source Columns	
CHECKING	
RECORD_ID	Varcha
SS_NUM	Numbe
NAME	Varcha
ADDR1	Varcha

Figure 8-58   Maps Joins page

With InfoSphere Discovery, you can join the two source tables, because it knows the PF Key between these tables.

Bindings

Select the **Bindings** tab (Figure 8-59). The binding condition is a join that aligns records in the source and target tables. Even if your source and target tables are in different platforms or database systems, InfoSphere Discovery copies the source and target data to the staging area. By copying the data to the staging area, InfoSphere Discovery can apply the binding condition between the source and target tables, regardless of where they reside.



Maps: Map_1_DO_CHECKING_To_DO_ACCOUNTS_-1475734479

ACCOUNTS
- Map_1_DO_CHECKING_To_
- Map_1_DO_INVESTMENTS_
- Map_1_DO_SAVINGS_To_D
- Map_2_DO_SAVINGS_To_D

Binding Condition	Weight	Type	Notes	Targ...	Targe...	Source ...	Source Bou...
● (CHECKING.SS_NUM = ACCOUNTS.SS_NUM)	5174.80	One-To-One		58.28 ...	1.00	25.89 % (...	1.00
○ (CHECKING.STATE = ACCOUNTS.STATE) AND (...	4639.96	One-To-One		51.33 ...	1.00	22.89 % (...	0.99
○ (CHECKING.ZIP = ACCOUNTS.ZIP) AND (DEMO...	4717.37	One-To-One		52.76 ...	1.00	23.71 % (...	0.98
○ (DEMOGRAPHICS.ADDRESS = ACCOUNTS.ADD...	3433.30	Many-To-One		57.06 ...	0.95	26.88 % (...	0.90
○ (CHECKING.ZIP = ACCOUNTS.ZIP) AND (DEMO...	4763.49	One-To-One		52.76 ...	1.00	23.52 % (...	0.99

```
(CHECKING.SS_NUM = ACCOUNTS.SS_NUM)
```

Columns

[All Columns]

Name	Data
Target Columns	
ACCOUNTS	
RECORD_ID	Varcha
SS_NUM	Numbe
NAME	Varcha
ADDR1	Varcha
ADDR2	Varcha
CITY	Varcha
STATE	Varcha
ZIP	Intege
ZIP_FOUR	Intege

Maps: Map_1_DO_CHECKING_To_DO_ACCOUNTS_-1475734479

Figure 8-59 Maps Bindings page

When InfoSphere Discovery runs the transformation analysis, it automatically copies the source and target data to its staging area. Copying the source and target data to the staging area can be problematic, such as for performance reasons or because of limited database storage capacity. In this case, use the logical table feature in the Data Sets area to define data samples before you run the transformation analysis. On the bank data, you can define a sample based on customer names or regions.

Map_1_DO_CHECKING has the following binding condition:

```
(CHECKING.SS_NUM = ACCOUNTS.SS_NUM)
```

InfoSphere discovered and tested several possible binding conditions, including the following conditions:

- `(CHECKING.STATE = ACCOUNTS.STATE) AND (CHECKING.ZIP = ACCOUNTS.ZIP) AND (DEMOGRAPHICS.ADDRESS = ACCOUNTS.ADDR1) AND (DEMOGRAPHICS.MARITAL_STATUS = ACCOUNTS.MARITAL_STATUS) AND (DEMOGRAPHICS.ZIP = ACCOUNTS.BRANCH_ZIP)`
- `(CHECKING.ZIP = ACCOUNTS.ZIP) AND (DEMOGRAPHICS.ADDRESS = ACCOUNTS.ADDR1)`
- `(DEMOGRAPHICS.ADDRESS = ACCOUNTS.ADDR1)`
- `(CHECKING.ZIP = ACCOUNTS.ZIP) AND (DEMOGRAPHICS.ADDRESS = ACCOUNTS.ADDR1) AND (DEMOGRAPHICS.MARITAL_STATUS = ACCOUNTS.MARITAL_STATUS)`

InfoSphere Discovery tests each condition and determines that the best one is the condition based solely on SS_NUM. It calls the best condition the *primary binding condition*. The others are called *alternative binding conditions*. It uses the primary binding condition when it discovers transformations for this map. To use a different binding condition, you select the condition and rerun the transformation step.

Each potential binding condition comes with several statistics. For example, each binding condition has a type. The type indicates how the source rows are related to the target rows. There are four types:

One-To-One Each row in the source relates to one row in the target. For example, each customer in the source exists one time in the target. This type is usually the simplest type of map to understand.

One-To-Many Each row in the source relates to many rows in the target. This type of map might become a reverse pivot.

Many-To-One Many rows in the source relate to one row in the target. Aggregations are usually many-to-one.

Many-To-Many Many rows in the source map to many rows in the target. This type of binding condition is difficult to turn into a map, because InfoSphere Discovery cannot pinpoint which source rows map to which target rows. This type of binding condition requires more refinement to turn it into one of the other types.

The **Bindings** tab includes the following other helpful statistics:

Target Hits
The number of target rows after applying the binding condition. The percentage reflects the number of target entities found in the source.

Target Bound Selectivity
The selectivity of the combined target columns used in the binding condition.

Source Hits
The number of source rows after applying the binding condition. The percentage reflects the number of source entities found in the target.

Source Bound Selectivity
The selectivity of the combined source columns used in the binding condition.

The binding condition type and statistics help you understand the quality of the binding condition. If you see low hit rates or low selectivity, you might not be matching the source and target rows correctly. Another possibility is that your data samples do not match. You might not be pulling the same slices of data from the source and target systems.

You can see the rows in the source and target that meet the binding condition by viewing the hits and misses. Hits are rows that exist in the source and target, based on passing the binding condition. Misses are rows that are in one system (source or target), but not in the other.

Transformations

Select the **Transformations** tab. The transformation results are in a grid similar to the example shown in Figure 8-60.

Each row of the grid represents a column in the target table, ACCOUNTS. The six columns in the grid describe the transformations:

#
The column number of the target column. For example, 1 represents the first column in the target table.

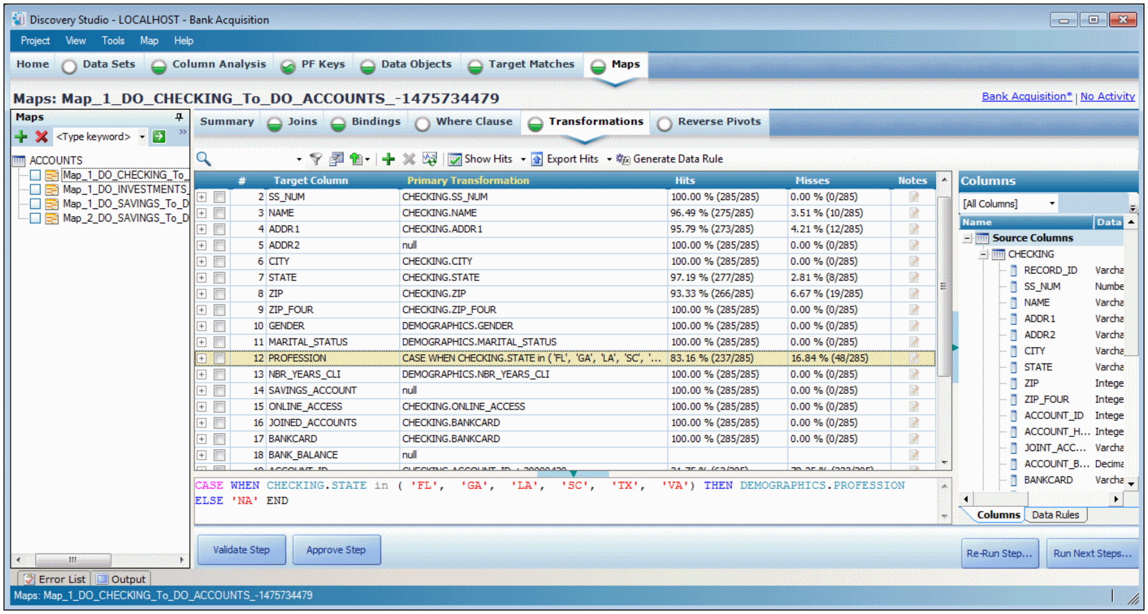
Target Column
The name of the target column.

Primary Transformation
The transformation, described in SQL. Alternative transformations might exist. To see if any alternative transformations exist, expand the tree by clicking the **+** next to a transformation.

Hits
The percentage (and literal number) of target rows that satisfy the transformation.

Misses The percentage (and literal number) of target rows that do not satisfy the transformation.

Notes Optional free-text notes.



#	Target Column	Primary Transformation	Hits	Misses	Notes
2	SS_NUM	CHECKING.SS_NUM	100.00 % (285/285)	0.00 % (0/285)	
3	NAME	CHECKING.NAME	96.49 % (275/285)	3.51 % (10/285)	
4	ADDR1	CHECKING.ADDR1	95.79 % (273/285)	4.21 % (12/285)	
5	ADDR2	null	100.00 % (285/285)	0.00 % (0/285)	
6	CITY	CHECKING.CITY	100.00 % (285/285)	0.00 % (0/285)	
7	STATE	CHECKING.STATE	97.19 % (277/285)	2.81 % (8/285)	
8	ZIP	CHECKING.ZIP	93.33 % (266/285)	6.67 % (19/285)	
9	ZIP_FOUR	CHECKING.ZIP_FOUR	100.00 % (285/285)	0.00 % (0/285)	
10	GENDER	DEMOGRAPHICS.GENDER	100.00 % (285/285)	0.00 % (0/285)	
11	MARITAL_STATUS	DEMOGRAPHICS.MARITAL_STATUS	100.00 % (285/285)	0.00 % (0/285)	
12	PROFESSION	CASE WHEN CHECKING.STATE in ('FL', 'GA', 'LA', 'SC', '...	83.16 % (237/285)	16.84 % (48/285)	
13	NBR_YEARS_CLI	DEMOGRAPHICS.NBR_YEARS_CLI	100.00 % (285/285)	0.00 % (0/285)	
14	SAVINGS_ACCOUNT	null	100.00 % (285/285)	0.00 % (0/285)	
15	ONLINE_ACCESS	CHECKING.ONLINE_ACCESS	100.00 % (285/285)	0.00 % (0/285)	
16	JOINED_ACCOUNTS	CHECKING.BANKCARD	100.00 % (285/285)	0.00 % (0/285)	
17	BANKCARD	CHECKING.BANKCARD	100.00 % (285/285)	0.00 % (0/285)	
18	BANK_BALANCE	null			

```
CASE WHEN CHECKING.STATE in ( 'FL',  'GA',  'LA',  'SC',  'TX',  'VA') THEN DEMOGRAPHICS.PROFESSION
ELSE 'NA' END
```

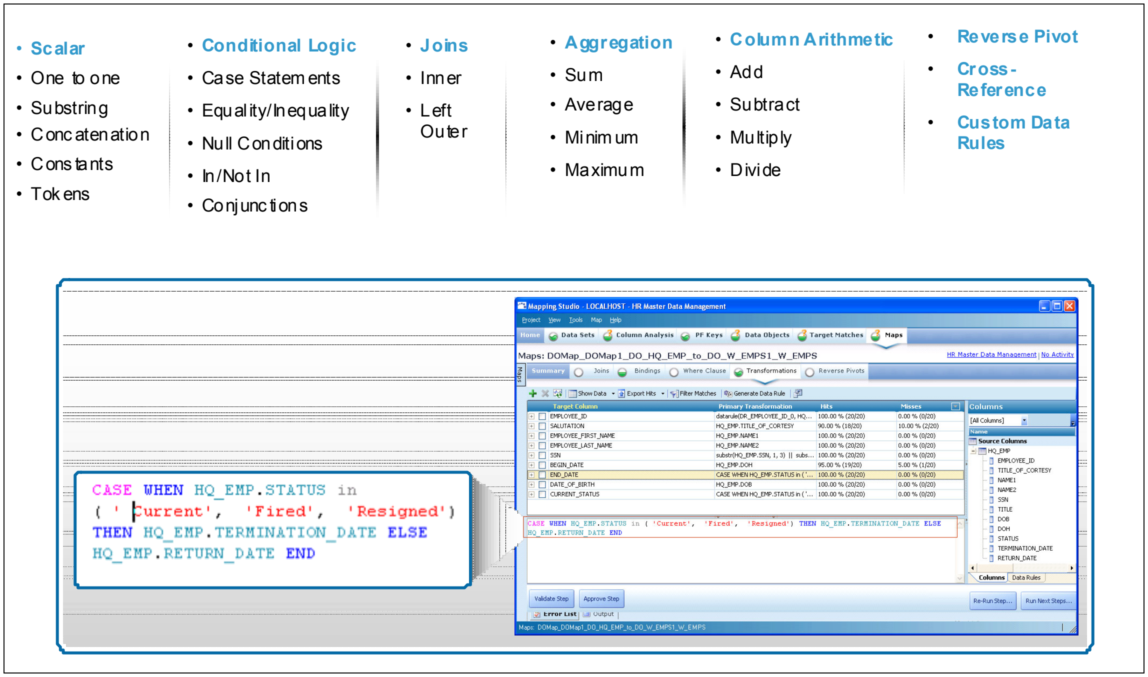
Figure 8-60 Maps Transformations page

InfoSphere Discovery finds many kinds of transformations. Figure 8-61 on page 277 illustrates the kinds of transformations it discovers. The simplest transformations are constants or one-to-one mappings. For example, the second column in the target table is SS_NUM. InfoSphere Discovery can populate it 100% of the time by inserting the value from CHECKING.SS_NUM.

You see a more complex transformation in column 12, Profession. The transformation is a CASE statement that reads as follows:

```
CASE WHEN CHECKING.STATE in ( 'FL',  'GA',  'LA',  'SC',  'TX',  'VA')
THEN DEMOGRAPHICS.PROFESSION ELSE 'NA' END
```

This statement looks at the state of the account holder. If the state is in the list, the rules states that the target must be populated with the profession of the account holder from the DEMOGRAPHICS table. If the state is not in the list, the target is populated with NA. At first, you might not know if the transformation is valid.



- **Scalar**
- One to one
- Substring
- Concatenation
- Constants
- Tokens

- **Conditional Logic**
- Case Statements
- Equality/Inequality
- Null Conditions
- In/Not In
- Conjunctions

- **Joins**
- Inner
- Left Outer

- **Aggregation**
- Sum
- Average
- Minimum
- Maximum

- **Column Arithmetic**
- Add
- Subtract
- Multiply
- Divide

- **Reverse Pivot**
- **Cross-Reference**
- **Custom Data Rules**

```
CASE WHEN HQ_EMP.STATUS in
( 'Current',  'Fired',  'Resigned')
THEN HQ_EMP.TERMINATION_DATE ELSE
HQ_EMP.RETURN_DATE END
```

Mapping Studio - LOCALHOST - HR Master Data Management

Maps: DOMap_DOMap1_DO_HQ_EMP_to_DO_W_EMPS1_W_EMPS

Target Column	Primary Transformation	Hits	Misses
EMPLOYEE_ID	datarule(DR_EMPLOYEE_ID_0, HQ...	100.00 % (20/20)	0.00 % (0/20)
SALUTATION	HQ_EMP.TITLE_OF_CORTESY	90.00 % (18/20)	10.00 % (2/20)
EMPLOYEE_FIRST_NAME	HQ_EMP.NAME1	100.00 % (20/20)	0.00 % (0/20)
EMPLOYEE_LAST_NAME	HQ_EMP.NAME2	100.00 % (20/20)	0.00 % (0/20)
SSN	substr(HQ_EMP.SSN, 1, 3) || subs...	100.00 % (20/20)	0.00 % (0/20)
BEGIN_DATE	HQ_EMP.DOH	95.00 % (19/20)	5.00 % (1/20)
END_DATE	CASE WHEN HQ_EMP.STATUS in ('...	100.00 % (20/20)	0.00 % (0/20)
DATE_OF_BIRTH	HQ_EMP.DOB	100.00 % (20/20)	0.00 % (0/20)
CURRENT_STATUS	CASE WHEN HQ_EMP.STATUS in ('...	100.00 % (20/20)	0.00 % (0/20)

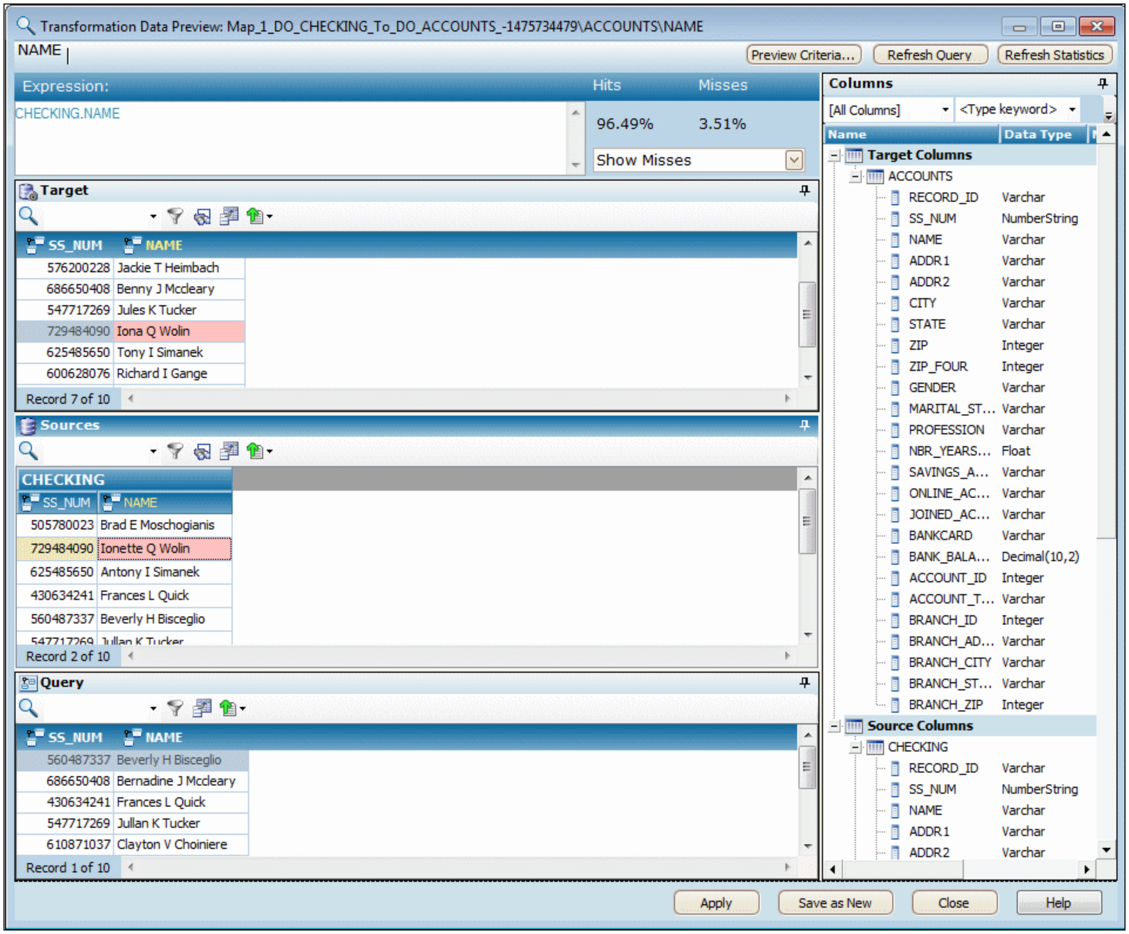
```
CASE WHEN HQ_EMP.STATUS in ( 'Current',  'Fired',  'Resigned') THEN HQ_EMP.TERMINATION_DATE ELSE
HQ_EMP.RETURN_DATE END
```

Figure 8-61 Discovery of complex transformations

Why is a rule for a column, such as Profession, based on a state? You might need to perform research to find the answer as indicated by the following steps. Upon further research, you might discover that some states prohibit using certain demographic data such as *profession*.

1. From the results shown in Figure 8-60 on page 276, select the third row, **NAME**. The transformation is a one-to-one map from CHECKING.NAME. The statistics (Misses) indicate that 3.51% of the target rows do not meet the rule. After you match the source and target rows with the binding condition, SS_NUM, some names do not match.
2. Click **Show Misses** to view the names that do not match.

   You see a list similar to the example in Figure 8-62 on page 278. Closer inspection of the data indicates that the people are the same, but their names are spelled slightly differently. The given name of one person is *Iona* in the target, but *Ionette* in the source. The given name of another person is *Jackie* in the target, but *Jacqueline* in the source system.



Transformation Data Preview: Map_1_DO_CHECKING_To_DO_ACCOUNTS_-1475734479\ACCOUNTS\NAME

NAME

Expression: CHECKING.NAME

Hits	Misses
96.49%	3.51%

Show Misses

Target

SS_NUM	NAME
576200228	Jackie T Heimbach
686650408	Benny J Mcdeary
547717269	Jules K Tucker
729484090	Iona Q Wolin
625485650	Tony I Simanek
600628076	Richard I Gange

Record 7 of 10

Sources

CHECKING

SS_NUM	NAME
505780023	Brad E Moschogianis
729484090	Ionette Q Wolin
625485650	Antony I Simanek
430634241	Frances L Quick
560487337	Beverly H Bisceglio
547717269	Jullan K Tucker

Record 2 of 10

Query

SS_NUM	NAME
560487337	Beverly H Bisceglio
686650408	Bernadine J Mcdeary
430634241	Frances L Quick
547717269	Jullan K Tucker
610871037	Clayton V Choiniere

Record 1 of 10

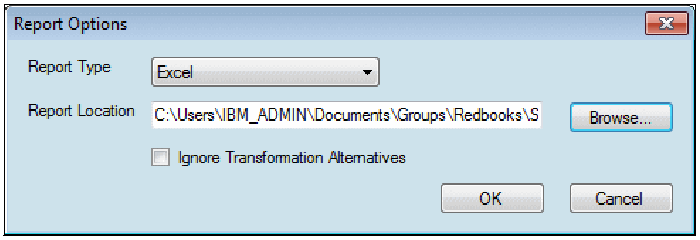
Columns

[All Columns] <Type keyword>

Name	Data Type
Target Columns	
ACCOUNTS	
RECORD_ID	Varchar
SS_NUM	NumberString
NAME	Varchar
ADDR1	Varchar
ADDR2	Varchar
CITY	Varchar
STATE	Varchar
ZIP	Integer
ZIP_FOUR	Integer
GENDER	Varchar
MARITAL_ST...	Varchar
PROFESSION	Varchar
NBR_YEARS...	Float
SAVINGS_A...	Varchar
ONLINE_AC...	Varchar
JOINED_AC...	Varchar
BANKCARD	Varchar
BANK_BALA...	Decimal(10,2)
ACCOUNT_ID	Integer
ACCOUNT_T...	Varchar
BRANCH_ID	Integer
BRANCH_AD...	Varchar
BRANCH_CITY	Varchar
BRANCH_ST...	Varchar
BRANCH_ZIP	Integer
Source Columns	
CHECKING	
RECORD_ID	Varchar
SS_NUM	NumberString
NAME	Varchar
ADDR1	Varchar
ADDR2	Varchar

Figure 8-62 Hits for transformations

3. Select **Tools** → **Reports** → **Project Report** to generate a project report.
4. In the Report Options window (Figure 8-63), for Project Type, select **Excel**, and then, for Report Location, enter a path to the `c:\` drive. Then click **OK**.



Report Options

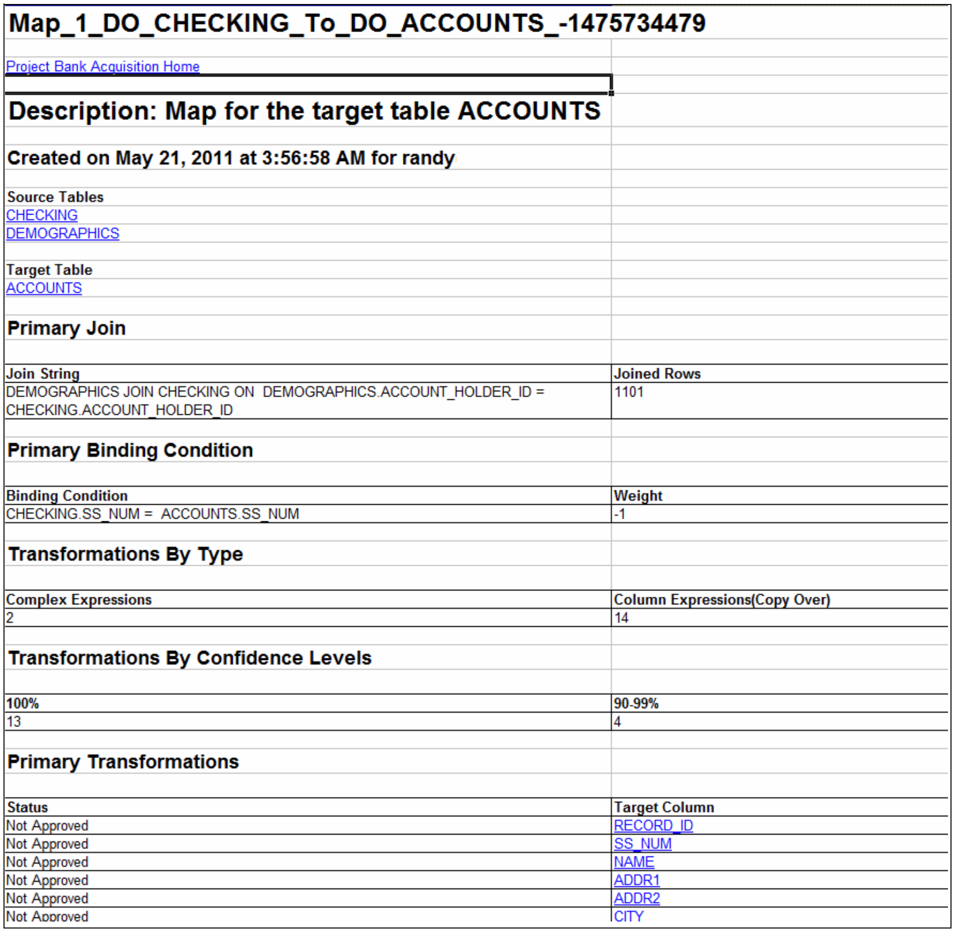
Report Type: Excel

Report Location: C:\Users\IBM_ADMIN\Documents\Groups\Redbooks\S

☐ Ignore Transformation Alternatives

Figure 8-63 Project report location

The report contains all of the column analysis results, PF Keys, data objects, and transformations. Figure 8-64 shows an example of documented transformations from the map, Map_1_DO_CHECKING. You send a copy of the report to your data quality team and indicate where you found data quality problems.



Map_1_DO_CHECKING_To_DO_ACCOUNTS_-1475734479

Project Bank Acquisition Home

Description: Map for the target table ACCOUNTS

Created on May 21, 2011 at 3:56:58 AM for randy

Source Tables
CHECKING
DEMOGRAPHICS

Target Table
ACCOUNTS

Primary Join

Join String	Joined Rows
DEMOGRAPHICS JOIN CHECKING ON DEMOGRAPHICS.ACCOUNT_HOLDER_ID = CHECKING.ACCOUNT_HOLDER_ID	1101

Primary Binding Condition

Binding Condition	Weight
CHECKING.SS_NUM = ACCOUNTS.SS_NUM	-1

Transformations By Type

Complex Expressions	Column Expressions(Copy Over)
2	14

Transformations By Confidence Levels

100%	90-99%
13	4

Primary Transformations

Status	Target Column
Not Approved	RECORD_ID
Not Approved	SS_NUM
Not Approved	NAME
Not Approved	ADDR1
Not Approved	ADDR2
Not Approved	CITY

Figure 8-64 Project report example

You have now completed a successful project. You uncovered bad data in several columns. Bad address data caused some statements to go to the incorrect addresses. Bad name data caused the bank to address customers by the wrong names. You documented the rules by generating the project report, and you sent discrepancies to the appropriate groups for data fixes.

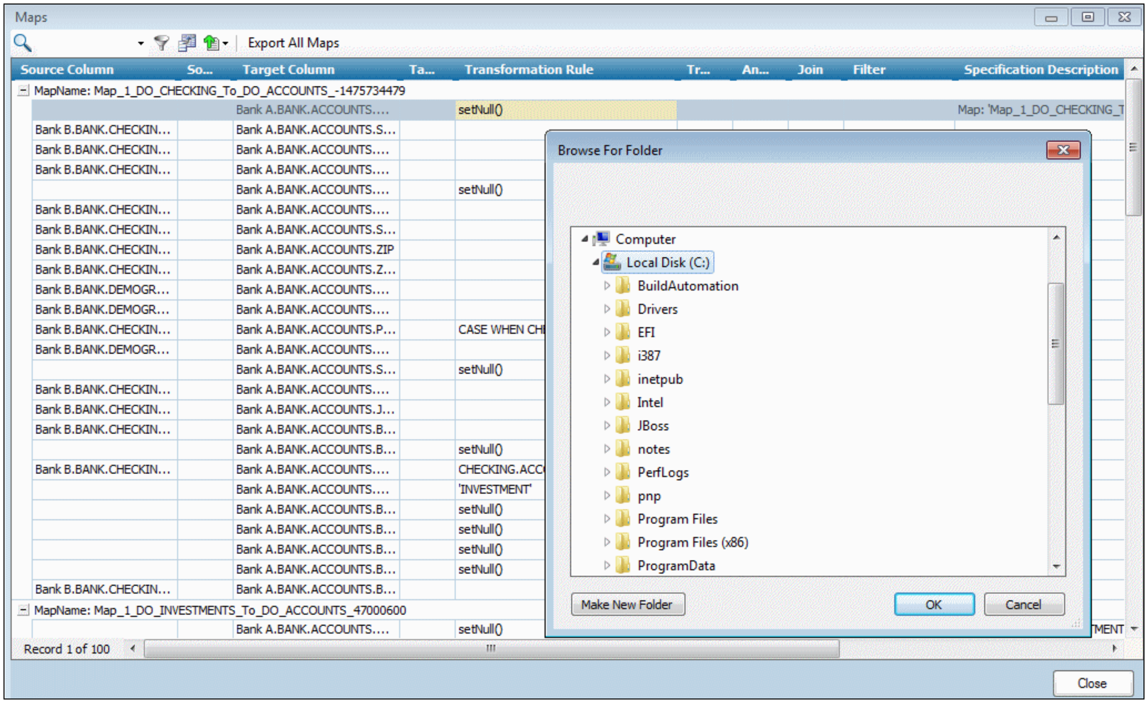
Transformation analysis goes deeper than you learned in this bank project, but what you did in this project was good enough to meet your objectives.

8.8.3 Exporting transformation results to InfoSphere FastTrack

The current data integration code that loads the data warehouse ACCOUNTS table is written in COBOL. The existing code uses several routines that are hard to understand and debug. Fortunately, we used InfoSphere Discovery to find a direct path from the source tables to the target. Now Bank A knows the underlying SQL transformations to populate the ACCOUNTS table in the data warehouse.

At a meeting recently, the data integration team decided to modernize the data integration process and rewrite it with InfoSphere DataStage. Fortunately, the data integration team can use the mapping results to write the new job. You send them an InfoSphere FastTrack export of your maps, so that they can import them into InfoSphere DataStage:

1. From the Discovery Studio menu, select **Project** → **Export** → **FastTrack Mapping CSV**.
2. In the window similar to the one in Figure 8-65, select a file location, and then click **OK**. InfoSphere Discovery exports one file for each map to the location you specified. The files are in comma-separated values (CSV) format.



Source Column	So...	Target Column	Ta...	Transformation Rule	Tr...	An...	Join	Filter	Specification Description
MapName: Map_1_DO_CHECKING_To_DO_ACCOUNTS_-1475734479									
		Bank A.BANK.ACCOUNTS....		setNull()					Map: 'Map_1_DO_CHECKING_T
Bank B.BANK.CHECKIN...		Bank A.BANK.ACCOUNTS.S...							
Bank B.BANK.CHECKIN...		Bank A.BANK.ACCOUNTS....							
Bank B.BANK.CHECKIN...		Bank A.BANK.ACCOUNTS....							
		Bank A.BANK.ACCOUNTS....		setNull()					
Bank B.BANK.CHECKIN...		Bank A.BANK.ACCOUNTS....							
Bank B.BANK.CHECKIN...		Bank A.BANK.ACCOUNTS.S...							
Bank B.BANK.CHECKIN...		Bank A.BANK.ACCOUNTS.ZIP							
Bank B.BANK.CHECKIN...		Bank A.BANK.ACCOUNTS.Z...							
Bank B.BANK.DEMOGR...		Bank A.BANK.ACCOUNTS....							
Bank B.BANK.DEMOGR...		Bank A.BANK.ACCOUNTS....							
Bank B.BANK.CHECKIN...		Bank A.BANK.ACCOUNTS.P...		CASE WHEN CH					
Bank B.BANK.DEMOGR...		Bank A.BANK.ACCOUNTS....							
		Bank A.BANK.ACCOUNTS.S...		setNull()					
Bank B.BANK.CHECKIN...		Bank A.BANK.ACCOUNTS....							
Bank B.BANK.CHECKIN...		Bank A.BANK.ACCOUNTS.J...							
Bank B.BANK.CHECKIN...		Bank A.BANK.ACCOUNTS.B...							
		Bank A.BANK.ACCOUNTS.B...		setNull()					
Bank B.BANK.CHECKIN...		Bank A.BANK.ACCOUNTS....		CHECKING.ACC					
		Bank A.BANK.ACCOUNTS....		'INVESTMENT'					
		Bank A.BANK.ACCOUNTS.B...		setNull()					
		Bank A.BANK.ACCOUNTS.B...		setNull()					
		Bank A.BANK.ACCOUNTS.B...		setNull()					
		Bank A.BANK.ACCOUNTS.B...		setNull()					
Bank B.BANK.CHECKIN...		Bank A.BANK.ACCOUNTS.B...							
MapName: Map_1_DO_INVESTMENTS_To_DO_ACCOUNTS_47000600									
		Bank A.BANK.ACCOUNTS....		setNull()					MENT

Figure 8-65 InfoSphere FastTrack mapping export

3. Send these files to the data integration team.

8.9 Conclusion

In conclusion, this chapter explained how to discover data relationships by using InfoSphere Discovery. Specifically, this chapter explained the steps to perform column analysis, identify and classify sensitive data, assign business terms to physical assets, perform value overlap analysis, and discover transformation logic.

Chapter 9, “Data quality assessment and monitoring” on page 283, focuses on data quality assessment and monitoring using IBM InfoSphere Information Analyzer.

9

Data quality assessment and monitoring

This chapter focuses on assessing and monitoring data quality. An important factor in metadata management is establishing a set of acceptance criteria and corrective measures to standardize data formats, ensure data quality, and completeness of data. These guidelines help in meeting corporate or regulatory requirements for information governance and are a key piece during implementing an information integration solution.

This chapter explains how to use IBM InfoSphere Information Analyzer, one of the IBM InfoSphere Information Server product modules, to assess and maintain data quality. It includes the following sections:

- Introduction to IBM InfoSphere Information Analyzer
- InfoSphere Information Analyzer data rules
- Creating a rule
- Data rule examples
- Data rules and performance consideration
- Rule sets and metrics
- Monitoring data quality
- Using HTTP/CLI API
- Managing rules

- Deploying rules, rule sets, and metrics
- Rule stage for InfoSphere DataStage
- Conclusion

9.1 Introduction to IBM InfoSphere Information Analyzer

InfoSphere Information Analyzer plays a prominent role in the *understand* phase of an information integration project.

9.1.1 InfoSphere Information Analyzer and information governance

Trusted information is in the core of many business initiatives. It is one of the foundations for decision-making processes and initiatives to optimize revenue opportunities. It enables the creation and nurture of collaborative business processes and empower the risk and compliance initiatives of an organization. A primary drive of information governance is the ability to establish the framework of organization, technology, and process that promotes the creation and maintenance of trusted information.

Information quality is a mission that encompasses the entire information life cycle. From source to the ultimate destination, data goes through a long sequence of pipes and jobs that convert it from one form to another and move it from one container to the next. Along this chain of processes are numerous opportunities for things to break, to go astray, and to produce wrong and unreliable data. Information governance is put in place to minimize such occurrences and optimize resources to achieve the best possible results toward building trust in the information.

InfoSphere Information Analyzer supports an evolutionary process of knowledge creation at all levels as shown in Figure 9-1 on page 285. The base for understanding is learning the facts, using profiling to find all that is to be known about the data, one column at a time. Knowledge evolves by extracting patterns and developing data rules to monitor the quality of data. Values are attached to patterns and presented as metrics for management to act upon, to evaluate the cost benefit of decisions and actions, and to measure the progress of their plans execution.

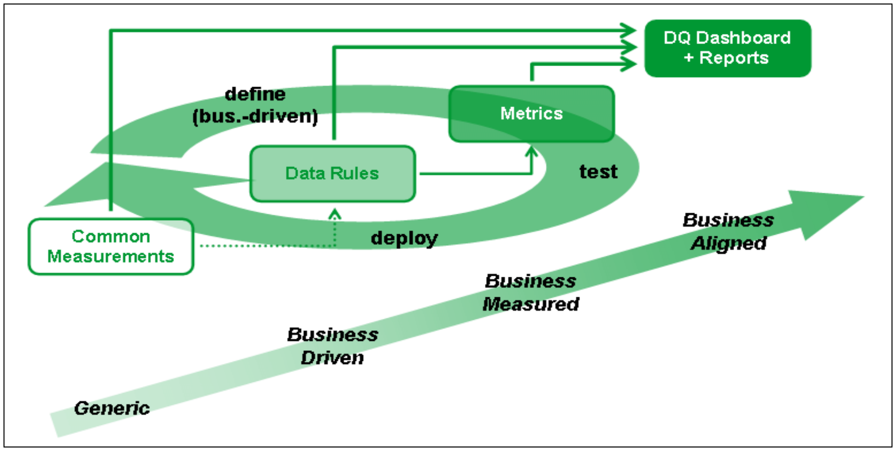


Figure 9-1 Evolution of understanding

InfoSphere Information Analyzer has a role in all phases of data processing from the start, analyzing the data source, all the way to monitoring the quality of data that goes into management reports and decision making models.

InfoSphere Information Analyzer supports the entire life cycle by helping oversee the creation and use of highly trusted information (Figure 9-2 on page 286). Starting at the source (1 in Figure 9-2), data comes in, from internal or external sources, and is profiled and screened for errors. Going through processing and transformation to feed the various data initiatives (2 in Figure 9-2), whether they use Master Data Management, Data Warehouse, Analytics, or other applications, the output of these processes is analyzed and screened for errors.

InfoSphere Information Analyzer looks for erroneous codes and values as they might result by coding errors or unanticipated input. The information gleaned by the analysis and data rules is used to inform the business about the quality and risk associated with the data (3 in Figure 9-2). This information is also used to inform the information governance authorities (4 in Figure 9-2) about the progress the company is making advancing information quality objectives.

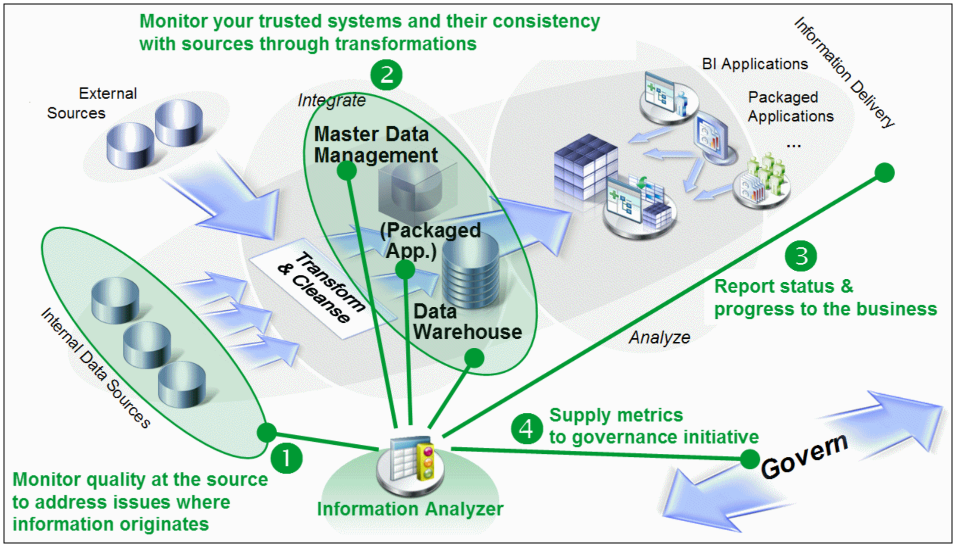

Monitor your trusted systems and their consistency with sources through transformations
2
Integrate
External Sources
BI Applications
Packaged Applications
...
Information Delivery
Master Data Management
(Packaged App.)
Transform & Cleanse
Data Warehouse
Internal Data Sources
3
Report status & progress to the business
Analyze
4
Supply metrics to governance initiative
Govern
1
Monitor quality at the source to address issues where information originates
Information Analyzer

Figure 9-2 Applying InfoSphere Information Analyzer - The solution perspective

9.1.2 InfoSphere Information Analyzer and InfoSphere Information Server

InfoSphere Information Analyzer is an integral product module to the InfoSphere Information Server suite. As such, it benefits from the common services available in the suite including administrative services and security, a parallel execution engine, and shared metadata repository. With this integration, users of InfoSphere Information Analyzer can contribute to the metadata enrichment effort. They can share knowledge acquired in the data analysis process with other individuals downstream such as analysts, developers, and administrators.

InfoSphere Information Analyzer is integrated into the InfoSphere Information Server suite. It includes the following features:

- One application server running a set of suite and product-specific services
- One dynamic repository to capture all metadata in a shared common model
- One operational repository that unifies the collection of runtime events or metadata
- One or more IBM InfoSphere DataStage parallel engines because most processing is carried out by the parallel engine

- A client user interface where all InfoSphere Information Analyzer functionality is driven through a rich client framework
- A set of common services, which include reporting, scheduling, security, logging, and licensing

9.1.3 Metadata data repository

InfoSphere Information Analyzer uses the common metadata repository that is shared by all of InfoSphere Information Server suite components. As noted previously, the repository contains metadata for many objects including actual, physical data sources. Each type is organized in a hierarchy of objects in an increasing level of granularity. This metadata is used by InfoSphere Information Analyzer to identify the data sources it should be applied to.

The metadata for data sources is organized into a hierarchy, starting with hosts at the top level going down to databases, schemas, tables, and columns, as shown in Figure 9-3. Whether you bring in metadata by using the metadata management and import facility for InfoSphere Information Analyzer or by using metadata broker directly into the repository, this hierarchy is the order in which metadata is discovered and imported. It is also the order in which you will see the resources in a tree view when you select data to apply analysis on or select columns to bind to logical variable in the data rules.

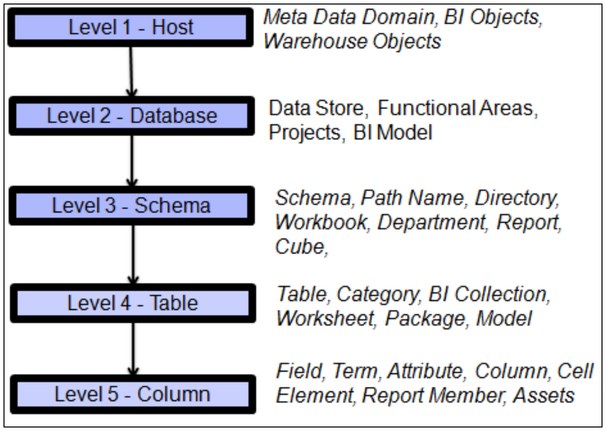
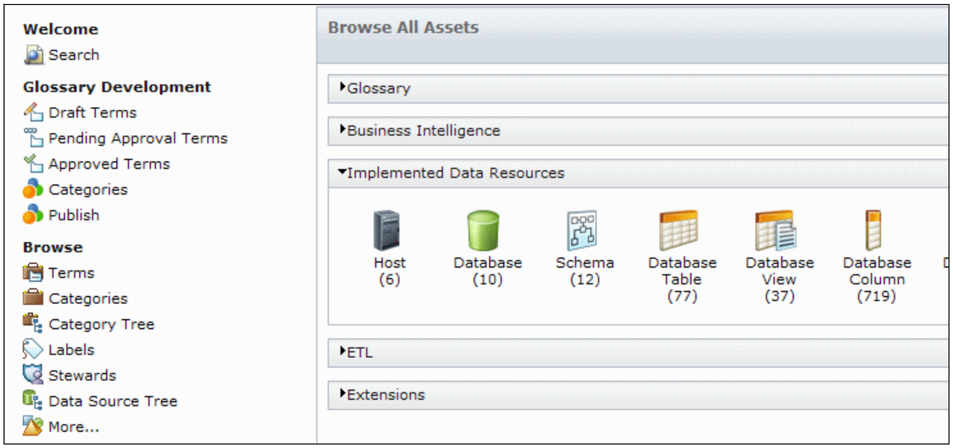

Level 1 - Host
Meta Data Domain, BI Objects, Warehouse Objects
Level 2 - Database
Data Store, Functional Areas, Projects, BI Model
Level 3 - Schema
Schema, Path Name, Directory, Workbook, Department, Report, Cube,
Level 4 - Table
Table, Category, BI Collection, Worksheet, Package, Model
Level 5 - Column
Field, Term, Attribute, Column, Cell Element, Report Member, Assets

Figure 9-3 Metadata hierarchy in a metadata repository

The metadata for these data source objects imported into the repository is visible and accessible by the different InfoSphere Information Server components as needed. This hierarchy of objects is reflected in the repository structure as shown in Figure 9-4.



Welcome

Search

Glossary Development

Draft Terms

Pending Approval Terms

Approved Terms

Categories

Publish

Browse

Terms

Categories

Category Tree

Labels

Stewards

Data Source Tree

More...

Browse All Assets

▸Glossary

▸Business Intelligence

▾Implemented Data Resources

Host	Database	Schema	Database Table	Database View	Database Column
(6)	(10)	(12)	(77)	(37)	(719)

▸ETL

▸Extensions

Figure 9-4 Data assets hierarchy or inventory view from InfoSphere Business Glossary

By using the shared metadata repository and source objects within multiple suite components, you can share information between those components and enhance collaboration between stakeholders. For example, you might reference the shipping address (Address_ShipTo) table in an IBM InfoSphere FastTrack mapping specification. This specification is then used in an IBM InfoSphere DataStage job and in an information analysis job.

A data analyst might review analysis results and create annotations and notes for a column or table in the metadata repository for business and other users to view. A project administrator for InfoSphere Information Analyzer publishes the analysis results, making them available to InfoSphere DataStage users to read and use. A developer can view the notes or annotations created by the data analyst and make design decisions based on the findings of the analyst. These notes might include information about data that contains nulls, special characters, invalid codes, or other data phenomenon. Regardless of the findings, this information highlights what the data contains and helps the developer deal with the potential problem data rather than run into the unexpected.

For example, the data analyst might suggest that the data type of a particular column in the table must be changed or that a table must have a foreign key established. Alternatively, the data analyst might suggest that a table has nonvalid values that job developers must be made aware of. As another example, a developer reviews the notes from the data analyst. The developer sees that the data analyst created a mapping specification in InfoSphere FastTrack that mapped columns and terms in the Bank_Accounts table to columns and terms in two other tables to extend and transform the data. The developer retrieves the job that was generated in InfoSphere FastTrack. Then the developer uses the generated job to extend the existing job that is associated with the Bank_accounts table. The hierarchy of objects remains intact.

9.2 InfoSphere Information Analyzer data rules

A primary thrust of an information governance initiative is data quality. All drivers of information governance rely on the premise that the information we report and use to make decisions is reliable and of high quality, meaning it is correct, complete, precise, and timely. Such drivers include risk avoidance, compliance, and enhanced and improved decision making. Organizations cannot afford the risk of bad data and, therefore, must be on guard constantly to avoid errors and prevent their ill effect.

InfoSphere Information Analyzer supports data analysis, profiling, and data rules. The profiling function performs detailed column analysis, extracting all possible information about data column by column. Analysis results include value distribution, pattern distribution format, length distribution, and more. An analyst evaluating the analysis results can determine if there are data issues, values out of range, wrong code vales, nulls or non-unique identifiers, and more. In addition, the analyst performs primary and foreign key analysis, validating that primary keys exist and are unique and that data integrity is not violated.

Although data profiling does well in analyzing data and communicating everything about the data, businesses often have specific requirements. They want to know that the data meets certain data standards of quality. They want to know that all codes in a column are valid and, if not, that they identify the violating records. They want to know that data integrity is maintained and that all values are within an agreeable range. They want to know that summary reports reflect information from all sites, all stores, and all branches and that no information is missing. Also, when data does not meet these requirements, they want to know, and they want the violators to be identified.

By using data rules, you can specify the conditions that you want the data to meet and test the data for conformance with these conditions. Data rules can take many forms. A data rule can be as simple as validating that a data item is not null, is of a certain type, or is within a given range. Data rules can also be complex, involving matching values to a given patterns and formats or multiple data sources with specified join conditions.

By having defined such conditions, InfoSphere Information Analyzer, using InfoSphere Information Server connectivity services and the parallel execution engine, scans the data in the data source and identifies records that do not meet the conditions. Often, a record might violate more than a single condition. Combining data rules into data rule sets enables checking data simultaneously for multiple conditions. Running a data rule set can reveal an individual rule violation and a multirule violation, revealing patterns and dependencies that otherwise might remain hidden from the common analyst.

Statistics and exception details are usually insufficient for providing management with insight into the cost or impact of potential issues. Management will ask for a summary as in the following examples:

- How do you summarize our data quality situation?
- Are we doing better or worse?
- How much does bad data cost us?

Management wants to see a single number, indicator, or metric. Metrics provide the capability to summarize rule execution results into a single figure and possibly translate it from units of errors to units of cost, labor, or lost sales.

As alluded to previously, the stakes in data quality are high. Data quality rules represent corporate knowledge that is formed by business and IT people who represent corporate experience and understanding of the data, systems, and organizations that use the data. Data rules reflect the particular ways that an organization conducts business, runs operations, and reports on them. Data rules are created after analysis of the impact that data quality might have on the operational and financial performance of an organization. The rules usually originate from business units that have a stake in data quality and are controlled and monitored by these units.

Business rules that are expressed in plain language must be translated into a formal representation, by using the functions and operators provided by InfoSphere Information Analyzer. Rules are validated and tested before they are used. Following these initial steps of creating and approving data rules for use, rules are applied to monitor the quality of data streams. The detection of trends that reflect a systemic problem and deterioration in data quality must trigger an initiative to identify the source of the problem and issue a corrective response.

9.2.1 Roles in data rules and data quality management

Data rules and data quality management has four types of roles: data rule owner, steward, developer, and administrator. These roles can be loosely mapped into InfoSphere Information Analyzer roles as the rule manager role (for both owner and steward), the rule author role (for developer), and the rule administrator role (for administrator).

Important: The roles must be assigned to different individuals to ensure checks and balances of the process. The only exception is that the data rule steward and the data rule developer can be the same person.

Data rule owner

The data rule owner is a business person, such as a functional unit manager. This person is responsible for the data quality and integrity in their particular domain and ensures that the data rules protect the business operations and processes. The data rule owner has the following responsibilities:

- Oversees operations and business processes within their area of operation and as such for the completeness, correctness, relevance, and timeliness of the data that the operation consumes or produces
- Identifies sensitive data and articulates the rules that are required to maintain a specified data quality standard
- Updates the rules, sets threshold values, and initiates actions when rules execution results show that quality measure surpass threshold values

Data rule steward

The data rule steward is a business or IT person who is responsible for managing the rule. Specifically, the data rule steward has the following responsibilities:

- Ensures that data rule remains congruent with its source policies and objectives
- Proposes changes to the rules and documents and explains the reason for and effect of these changes
- Proposes retirement of rules and documents the reasons
- Ensures that all required rule definition and properties have been documented according to established standards, including business processes in which each rule is in effect
- Ensures that rules are uniformly applied across the company

Data rule developer

The data rule developer is a technical person. This person is responsible for translating a data rule expressed in natural language that is submitted by the data rule owner and the steward into a formal rule expression to be used by the system. The rule developer expresses the rule in a way that provides optimal performance. The rule developer tests the rule definition and makes any changes that are necessary to reflect the intent of the rule and reliably provide the desired results.

The rule developer monitors the performance of the rules and makes any adjustments to the definition of the rule or other system aspects to guarantee optimal performance.

Data rules administrator

The data rules administrator is an IT person with administrative rights for data rule management. The data rules administrator maintains the integrity of the rule repository and assigns access permissions to individuals who are authorized to handle data rules.

9.2.2 Properties of the InfoSphere Information Analyzer data rules

InfoSphere Information Analyzer provides a broad range of features to support rule creation, maintenance, and deployment processes. InfoSphere Information Analyzer rules have the following properties:

Logical definition Initial definition of a rule is a functional expression that does not require any knowledge of the data. Logical expression uses logical variables. The system enables syntactical validation of the rules.

Reusable Logical rule definition can be bound to different data sources, creating multiple executable data rules. When changes are made to the logical rule definition, those changes apply to all the executable data rules without requiring any updates to those date rules.

Quickly evaluated Rules can be tested interactively against real or test data to determine whether they meet rule business requirements.

Flexible output A user can tailor rule output, specifying the statistics to be produced and how exception information should look.

Historical Rule execution results can be retained over a long time and analyzed to reveal trends or patterns.

Managed Rule status can be defined according to the phase in the development process (draft, accepted, and so on.)

Categorical Rules are organized into folders that can be used to classify rules by project, subject area, type of test, types of data, frequency of execution, and so on.

Auditable All changes made to rules are recorded, enabling the identification of who modified a rule and when.

9.2.3 Data rules management

A basic premise that the organization needs to adopt is that data rules represent knowledge assets. The rules develop and evolve over time and reflect the specific structure, operations, plans, and strategy of the organization. Rules, rule sets, and metrics are applied on relevant and key business data and reflect the values of an organization.

Usually a rule originates with a business person, such as a data analyst, who is concerned with the quality of information that they use in their daily operations and decision making processes. This person wants assurance that the data they are using is correct and complete. Establish the criteria relevant to key data such as the following examples:

- The customer name must be complete (that is, no blank values), and zero tolerance for error is allowed.
- If the order status is canceled, the ship date must be blank. A 2% tolerance for error is allowed because the cancelation might occur after shipment.

A data analyst might set tolerances around key business indicators (KBI) based on the perceived value or impact to the organization. For example, key attributes might be missing for a customer, a patient, or a product, which is significant. If the attributes are missing descriptive text or a demographic flag, the data analyst might be unable to perform some levels of analysis. However, because this missing information does not stop a business, the tolerance is much higher.

The previous general statements are translated into more explicit requirements on specific data elements. For example, consider that these values represent an aggregation of reports from 30 locations, where each location reports on tens of different data items. With the help of a business analyst, these rules can be translated into the following verbal expression:

- Summary report must include daily reports from all 30 locations.
- No more than 2% of the total items reported by the locations can have a missing or null value.

A rule developer turns these requirements into a formal expression by using variables, operators, and functions that are made available by the system. This formal expression can be submitted to the machine and applied to the data. The expression entails identifying the proper data element to be bound to the rule and testing whether the rule is correct. When the developer and the steward are satisfied with test results, and the rule owner approves, the rule is placed in the folder of executable rules. The rule is then scheduled for execution on a periodical basis as determined by the nature of the data flow and the quality requirements established by the rule owner.

Figure 9-5 shows the rule definition process from conception to realization and execution

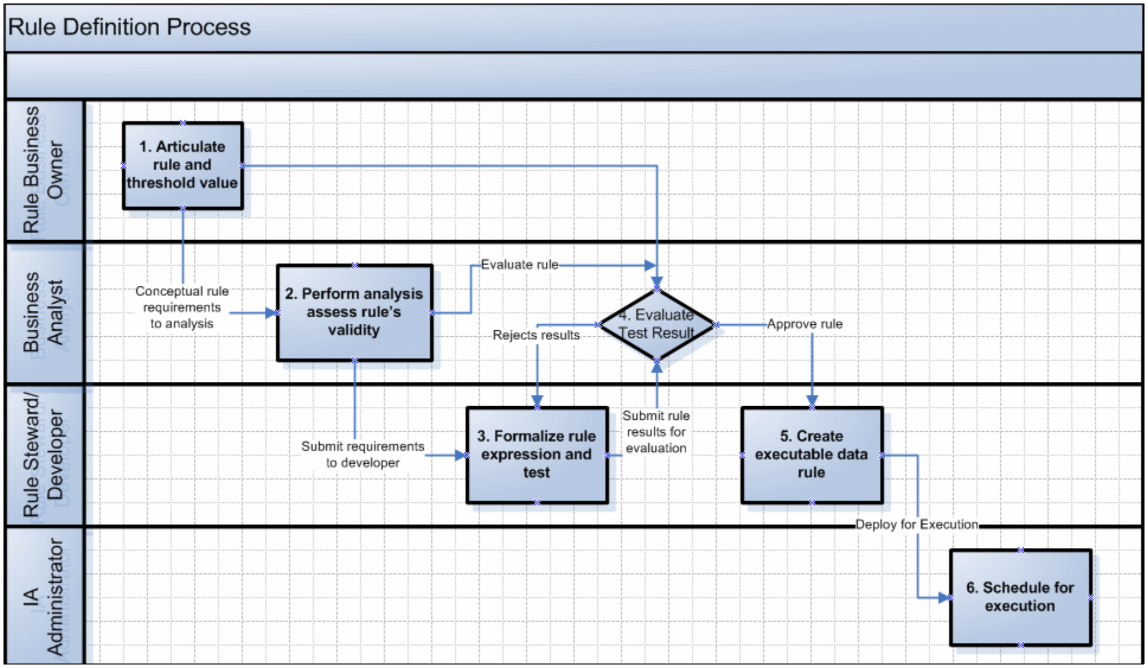

Rule Definition Process
Rule Business Owner
Business Analyst
Rule Steward/ Developer
IA Administrator
1. Articulate rule and threshold value
Conceptual rule requirements to analysis
2. Perform analysis assess rule's validity
Evaluate rule
4. Evaluate Test Result
Rejects results
Approve rule
Submit requirements to developer
3. Formalize rule expression and test
Submit rule results for evaluation
5. Create executable data rule
Deploy for Execution
6. Schedule for execution

Figure 9-5 Rule creation process

Creating a rule

Creating rule entails the following steps:

1. Articulate the rule and threshold values.

   A business initiates a rule by establishing certain conditions that it expects the data to satisfy in order for it to be useful and reliable. It also establishes threshold values for acceptable levels of erroneous data or rule violations. A rule is specified in a business language by using known terminology and by following accepted guidelines for rule expression.

2. Perform analysis and assess the validity of the rule.

   A business analyst or rule steward translates the business requirements into more formal IT requirements. They identify the data source, tables, and columns that will be used to realize the rule, and they create a functional expression of the rule.

3. Create rule definition and test rule.

   A rule developer translates the rule into a formal InfoSphere Information Analyzer rule expression by using the functions and operators provided by

InfoSphere Information Analyzer. The developer establishes links to the actual data set and tests the rule against sample data or test data prepared for that purpose. The rule expression is debugged and modified until the developer is satisfied that the results represent the intent of the rule and that the performance is acceptable.

4. Evaluate test results.

   The rule developer and the business analyst review and evaluate test results to determine that the rule was implemented correctly and will yield the desired results. If they are not satisfied, they go back to the analysis and formulation stage until the results meet the requirements. Otherwise, they will find alternative rule expression to achieve the same objective.

5. Create an executable data rule, and submit it for scheduling.

   The tested rule definition is used to generate an executable data rule. It is bound to the actual data sources and is placed in an executable data rules folder. The administrator is notified of the new rule and on the schedule on which the rule must be run.

6. Schedule rule execution.

   The rule administrator receives information from the rule developer or steward and places it on an appropriate schedule for execution.

Authoring data rules

Natural language is often sloppy and ambiguous. A rule expressed in natural language can often be misinterpreted by a developer, resulting in an inaccurate or wrong functional representation that translates into incorrect execution results.

Data rules are statements of constraints and conditions that the data is expected to meet. A statement that the value of a certain column exists, is numeric, or is smaller or bigger than a given value is evaluated. Records that do not meet this condition are flagged. InfoSphere Information Analyzer does not apply any changes to the data, but evaluates whether the conditions are satisfied.

Rules must be reviewed and refreshed periodically. Changes in data flows, systems, and platforms require that rules are adjusted to the new environment as changes occur. Often changes are subtle and do not trigger impact analysis. The analyst must review the elements on the path to the data that is used by the rule to identify any changes. The analyst must also evaluate their potential impact on the rule result and execution performance.

Data rules can be simple, involving a single operator on a single column, or complex, involving different functions, operators, and data elements from multiple tables. Usually a rule is detailed by a subject matter expert (SME) who is the owner or steward of the domain and then translated into formal expression by a developer. The language used to define a data rules must be carefully crafted to avoid misinterpretation. Data rule definitions must be complete and expressed in a clear and unambiguous language.

9.2.4 Rule definition guidelines for data quality

Guidelines can help you to name and define rules. This section includes the following guidelines:

- Rule naming guidelines
- Rule definition guidelines

Rule naming guidelines

The creation of a rule naming convention is essential to a successful implementation of a rule management system. Rule names must convey what the rule is about. In an environment with hundreds of rules, consider classifying rules so that they can be grouped in a manner that rules can be easily associated with an owner, operation, or schedule and can then be located. Good naming conventions promote the reuse of rules.

Establishing naming conventions is up to the individual organization. Use the following guidelines when developing naming standards:

- Rule definitions, rules, and rule sets cannot have the same name.
- Rules in different projects can have the same name.
- Ensure that names of all published rules are unique.
- Associate the project name with a functional area of the groups to which these rules will apply.
- Use the structure *Prefix-Name-Suffix* for the name of the rule:

  Prefix Used for sorting and classification. Use agreeable and common abbreviations, for example: ERM (Enterprise Risk Management), MKT (Marketing), and ORD (Orders).

  Rule Type (optional) RD=Rule Definition, DR=Data Rule, RS=Rule Set.

  Frequency (optional) Indicates when the rule is executed and can help with the grouping of rules for batch execution, for example: DLY, WK, and MON.

  Number (optional) Can be used to keep track of the rules that you are going to develop. It must match the logical rules number that was used in the design specification. It can also be used to determine the order in which the rule will be executed.

Name of the Rule A combination of terms that identify the subject data element and the purpose of the rule, for example: SSN_Pattern, SSN_Unique, Customer_Number_Exist, and Code_Valid.

Suffix Version, rule type, or sequence.

Putting it all together, two rule definitions are used to monitor weekly the number of gold and platinum customers:

- MKT_01_RD_WK_GoldCustomer_Qualify_V01.1
- MKT_02_RD_WK_PlatinumCustomer_Qualify_V01

Rule definition guidelines

A rule definition is created for a range of users. A developer must have a clear understanding of what the rule is expected to do and translate the definition into a formal, executable expression. For all users, rule definition must be clear and unambiguous. For example, business users might come across a rule over time and need to understand the purpose of the rule.

Use the following guidelines for rule writing:

- Write rules in a declarative non-procedural language.
- Ensure that data rules can be readily understood by any qualified person, with no room for ambiguity or misinterpretation.
- Use “must” and “only” keywords in the expression of rules.
- Avoid extra wording for emphasis, they do not add any additional meaning and can only confuse the reader.
- Express data rules in a positive statement. Indicate the conditions that the data must satisfy, rather than the conditions that must be avoided. This wording simplifies the expression of rules and makes the analysis of results more uniform and consistent, identifying those records that did not meet the rule conditions.

9.3 Creating a rule

The process of creating a data quality rule involves several steps from definition to deployment. This section highlights the following tasks:

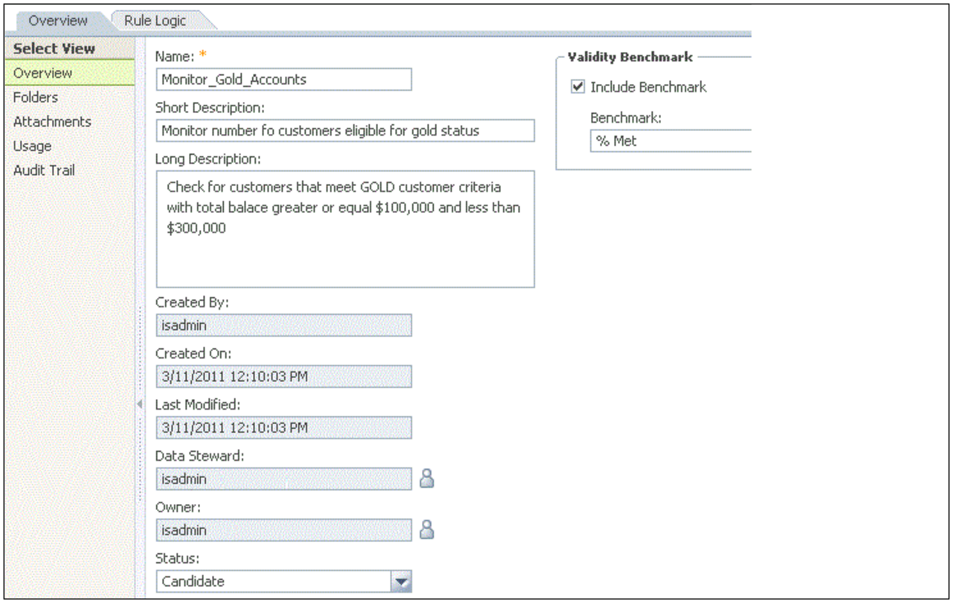
- Creating a rule definition
- Testing a rule
- Generating data rules

9.3.1 Creating a rule definition

Rule definition is the logical formal definition of a rule. A rule definition is a template of logic pattern that you want to explore. You can use a single rule definition to generate multiple data rules by binding the logical variables to different data sources. Every data rule generated from a rule definition requires adjusting the other rule elements, such as join conditions, where applicable, and output columns.

Rule definition specifications

To start a new rule definition, click **Data Quality** on the task menu. A rule definition page (Figure 9-6) opens with the required attributes of rule name and short and long description. It also indicates the creator, data steward, and developer, who are the stakeholders that see through the creation of a rule.



Name: *
Monitor_Gold_Accounts

Short Description:
Monitor number fo customers eligible for gold status

Long Description:
Check for customers that meet GOLD customer criteria with total balace greater or equal $100,000 and less than $300,000

Created By:
isadmin

Created On:
3/11/2011 12:10:03 PM

Last Modified:
3/11/2011 12:10:03 PM

Data Steward:
isadmin

Owner:
isadmin

Status:
Candidate

Validity Benchmark

☑ Include Benchmark

Benchmark:
% Met

Figure 9-6 Rule definition overview pane

Logical rule expression

The logical rule expression is created on the **Rule Logic** tab. A *logical rule expression* is a formal expression that is executable by the engine. It involves the specification of logical variable, checks, and functions that form a constraint on the data that you want to check.

Logical variables

Logical variables are labels, parameters, or place holders for a physical data column to be connected to later. Often general-purpose rules use generic names such as *SourceDate* or *variable1*. Such rules include the “existence” rule, rules that check that a value is not null, or rules that can be applied to a range of data elements. In cases where a rule is applied to certain types of data, such as an identifier or address, it does make sense to name the logical variable to reflect the content that you intend to validate.

Another option is the use business glossary terms, from InfoSphere Business Glossary, as logical variables. Terms in InfoSphere Business Glossary are well-defined and should not cause confusion about which data to include in the rule. Often, glossary terms are already assigned to data elements that realize the concept, which can help to identify the physical data column to be bound to the variable. The assignment of a data asset to a term does not result in automatic binding. However, in cases where it is difficult to identify the correct data element to bind to the variable, this information about data asset assignment might help to make the proper identification.

Usage of InfoSphere Business Glossary terms must be a standard practice whenever possible when implementing data rules for InfoSphere Information Analyzer.

Sometimes, using original business terms is insufficient and must to be qualified. For example, you need to create a rule definition to compare the current account balance with the account balance in a previous month-end. Most likely, account balance is defined once in the business glossary. You need to qualify the “account_balance” business term by adding an appropriate suffix. This way, you have two distinct variables: account_balance_today and account_balance_previous_month_end. Make sure to document all used suffixes.

Functions and checks

InfoSphere Information Analyzer provides the following selection of functions and checks to rule developers:

> Checks to see if the source value is greater than the reference data.

>= Checks to see if the source value is greater than or equal to the reference data.

< Checks to see if the source value is less than the reference data.

<= Checks to see if the source value is less than or equal to the reference data.

Contains Checks the data to see if it contains part of a string. The check returns *true* if the value represented by the reference data is contained by the value represented by the source data. Both the source and reference data must be of the string type.

Exists Checks the data for null values. The check returns *true* if the source data is not null and *false* if it is null.

In_reference_column Checks to determine if the source data value exists in the reference data column. For this check, the source data is checked against every record of the reference column to see if there is at least one occurrence of the source data.

In_reference_list Checks to determine if the source data is in a list of references, for example {'a','b','c'}.

Is_date Checks to determine if the source data represents a valid date. For this type of check, you cannot enter reference data.

Is_numeric Checks to determine if the source data represents a number. For this type of check, you cannot enter reference data.

matches_format Checks to make sure that the data matches the format that you define, as in the following example:

```
IF country='France' then phone matches_format
'99.99.99.99.99'
```

Both source and reference data must be strings.

matches_regex Checks to see if the data matches a regular expression, such as the following example:

```
postal_code matches_regex '^[0-9]{5}$'
```

Both the source and reference data must be strings.

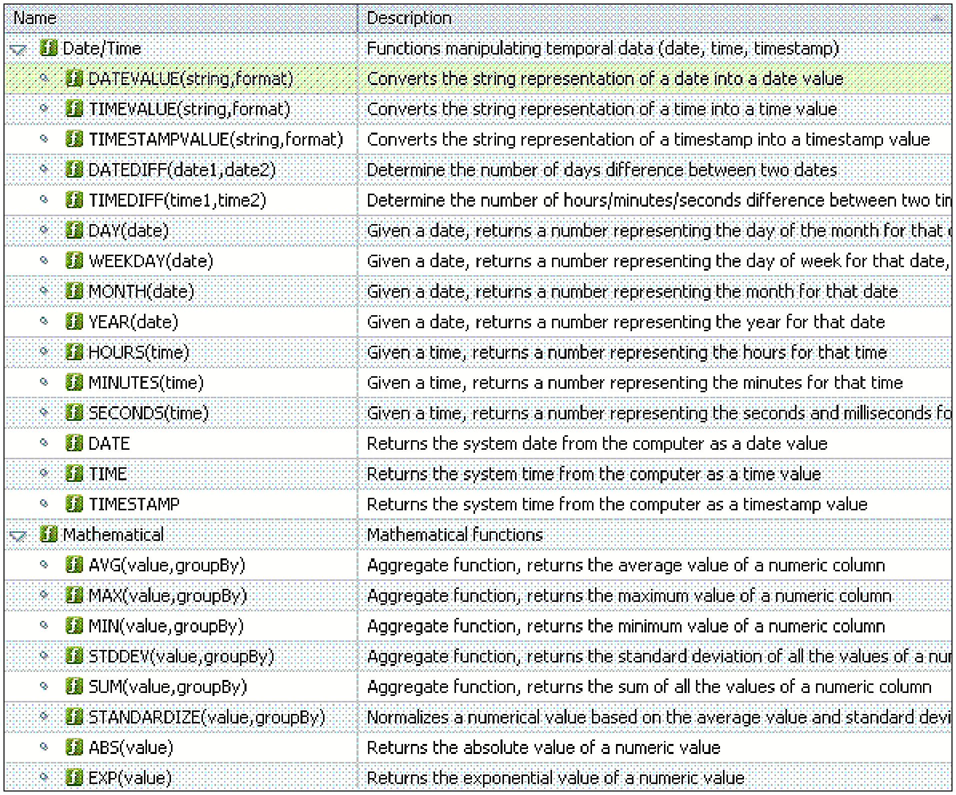
Occurs Checks to evaluate if the source value occurs as many times as specified in the reference data in the source column. The reference data for this check must be numeric. For example, in the firstname column, "John" appears 50 times and the rule logic is written as firstname occurs <100. After you bind the firstname column with the literal John, then records with "John" in the firstname column meet the conditions of the rule logic. The following occurrence check options are available:

- occurs>=
- occurs>
- occurs<=

- occurs<

unique Checks to evaluate if the source data value occurs only one time (a cardinal value) in the source data.

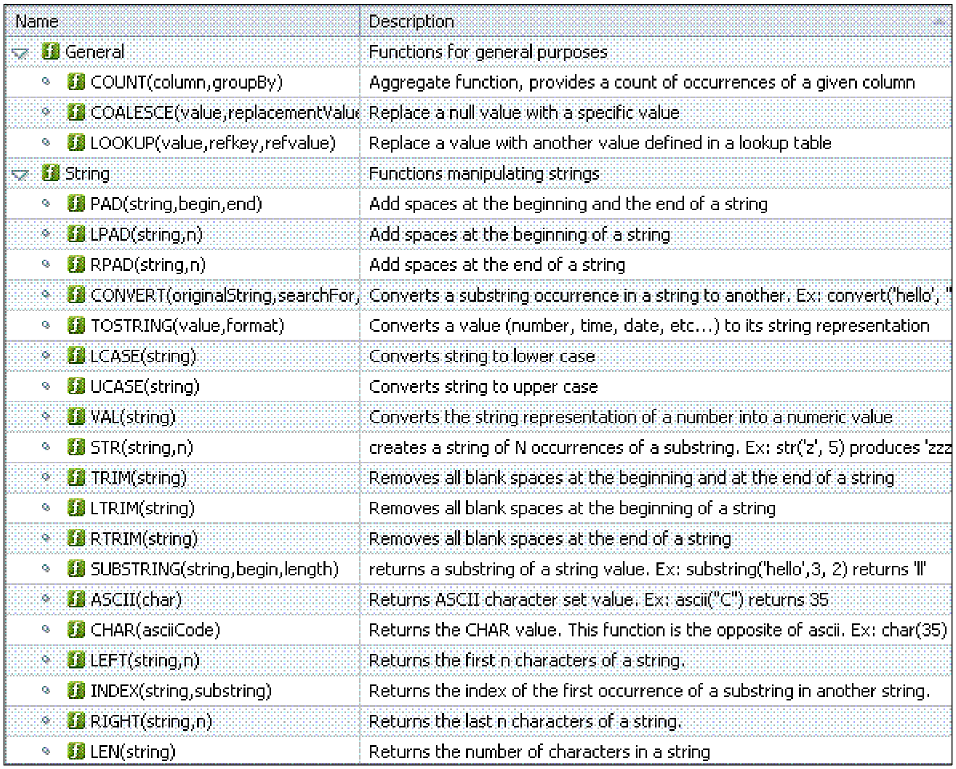
InfoSphere Information Analyzer also offers a rich selection of functions that can be used in rule expressions whether to aggregate data, date, or time related as shown in Figure 9-7.



Name	Description
Date/Time	Functions manipulating temporal data (date, time, timestamp)
DATEVALUE(string,format)	Converts the string representation of a date into a date value
TIMEVALUE(string,format)	Converts the string representation of a time into a time value
TIMESTAMPVALUE(string,format)	Converts the string representation of a timestamp into a timestamp value
DATEDIFF(date1,date2)	Determine the number of days difference between two dates
TIMEDIFF(time1,time2)	Determine the number of hours/minutes/seconds difference between two tin
DAY(date)	Given a date, returns a number representing the day of the month for that (
WEEKDAY(date)	Given a date, returns a number representing the day of week for that date,
MONTH(date)	Given a date, returns a number representing the month for that date
YEAR(date)	Given a date, returns a number representing the year for that date
HOURS(time)	Given a time, returns a number representing the hours for that time
MINUTES(time)	Given a time, returns a number representing the minutes for that time
SECONDS(time)	Given a time, returns a number representing the seconds and milliseconds fo
DATE	Returns the system date from the computer as a date value
TIME	Returns the system time from the computer as a time value
TIMESTAMP	Returns the system time from the computer as a timestamp value
Mathematical	Mathematical functions
AVG(value,groupBy)	Aggregate function, returns the average value of a numeric column
MAX(value,groupBy)	Aggregate function, returns the maximum value of a numeric column
MIN(value,groupBy)	Aggregate function, returns the minimum value of a numeric column
STDDEV(value,groupBy)	Aggregate function, returns the standard deviation of all the values of a nur
SUM(value,groupBy)	Aggregate function, returns the sum of all the values of a numeric column
STANDARDIZE(value,groupBy)	Normalizes a numerical value based on the average value and standard devi
ABS(value)	Returns the absolute value of a numeric value
EXP(value)	Returns the exponential value of a numeric value

Figure 9-7 List of Date/Time and Math functions

A set of functions is also available to manipulate character strings as shown in Figure 9-8.



Name	Description
General	Functions for general purposes
COUNT(column,groupBy)	Aggregate function, provides a count of occurrences of a given column
COALESCE(value,replacementValu	Replace a null value with a specific value
LOOKUP(value,refkey,refvalue)	Replace a value with another value defined in a lookup table
String	Functions manipulating strings
PAD(string,begin,end)	Add spaces at the beginning and the end of a string
LPAD(string,n)	Add spaces at the beginning of a string
RPAD(string,n)	Add spaces at the end of a string
CONVERT(originalString,searchFor,	Converts a substring occurrence in a string to another. Ex: convert('hello', "
TOSTRING(value,format)	Converts a value (number, time, date, etc...) to its string representation
LCASE(string)	Converts string to lower case
UCASE(string)	Converts string to upper case
VAL(string)	Converts the string representation of a number into a numeric value
STR(string,n)	creates a string of N occurrences of a substring. Ex: str('z', 5) produces 'zzz
TRIM(string)	Removes all blank spaces at the beginning and at the end of a string
LTRIM(string)	Removes all blank spaces at the beginning of a string
RTRIM(string)	Removes all blank spaces at the end of a string
SUBSTRING(string,begin,length)	returns a substring of a string value. Ex: substring('hello',3, 2) returns 'll'
ASCII(char)	Returns ASCII character set value. Ex: ascii("C") returns 35
CHAR(asciiCode)	Returns the CHAR value. This function is the opposite of ascii. Ex: char(35)
LEFT(string,n)	Returns the first n characters of a string.
INDEX(string,substring)	Returns the index of the first occurrence of a substring in another string.
RIGHT(string,n)	Returns the last n characters of a string.
LEN(string)	Returns the number of characters in a string

Figure 9-8 List of String and general functions

Using these functions, combined with the creation of compound conditions using NOT, AND, or OR operators, provides a rich canvas to create complex rules.

After a rule is written, you can click the **Validate** button to validate that the rule is syntactically correct.

9.3.2 Testing a rule

Testing a rule means entails setting up a rule definition for test execution. Testing a rule includes the following steps:

1. Bind to the data.

   Variables in the rule definition must be bound to data, either the data on which you want to apply the rule or test data that was prepared for testing this rule and other rules.

To bind the data, highlight a variable in the rule definition, and then select a column from data sources on the **Implemented Data** tab. Select a column, and then click the **Bind** button to establish the bind.

2. Define the joins.

   If the rule involves data from several data sources or tables, the rule engine detects the situation and prompts you to define the join conditions. It presents a table of the source from which you must identify the keys that connect them.

3. Define the output.

   Specify the output that you want to see in the results. Select the type of statistics in the summary report, such as counts and percentage of records that did not meet the rule condition. In addition, select the data columns to appear in the detailed results of records that did not meet the rule conditions. In addition to the columns that participate in the rule evaluation, you might want to select columns that identify the record and that possibly represent factors that might be related to the source of error.

4. Run the test.

   When all the preparations are complete, the rule is ready for running the test. A test job is created and submitted to the parallel engine. (InfoSphere Information Analyzer jobs are specially designed InfoSphere DataStage jobs.)

5. Review the results.

   When the test job is complete, the **Review Results** button is enabled. Summary results provide an execution summary that includes error statistics that you selected, in addition to counts and the percentage of rules that did or did not meet the conditions. Also, the **Results** tab shows the records that did not meet the conditions or meet the conditions as specified. Each record contains the fields that you selected to include in the results output.

9.3.3 Generating data rules

Data rules are executable conditions or instances of the rule logic applied against actual data. They are generated from rule definitions that were tested and approved for use. To generate a rule, you go through similar steps for creating and testing a rule definition. You name and describe the rule, bind the logical variables, define a join condition where needed, and specify the desired output. The data rule is created against the real data stream that you want to validate. It might be different from the data that you used to test the rule definition.

After a data rule is generated, place it into a data rule folder that was created for this purpose. Rule folders group rules in a way that they are easier to manage, representing project, operational area, execution schedule, owner, and so on.

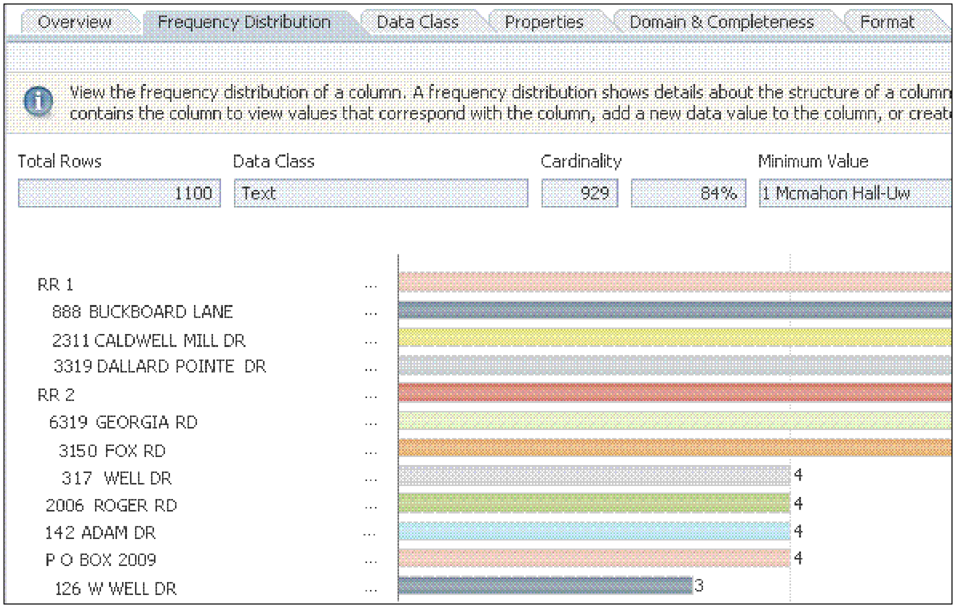
9.4 Data rule examples

Running column analysis is not a prerequisite for data rule execution. Unlike all other types of analysis in InfoSphere Information Analyzer that rely on the results and statistics produced by column analysis jobs, rules do not require it. However, as a good practice, profile the data and become familiar with it. It is not essential to run column analysis every time you execute a rule on that data. Periodically, perform column analysis on the data to verify that no new patterns or values exist to which rules must be adjusted.

9.4.1 Checking for duplicates

In the bank scenario in this book, we are analyzing Bank B data. Because this time is the first time that data is encountered, we start basic profiling by running column analysis. Column analysis highlights data anomalies that we must be aware of going forward.

Running column analysis on Address produced the distribution of address values shown in graphic mode as shown in Figure 9-9.

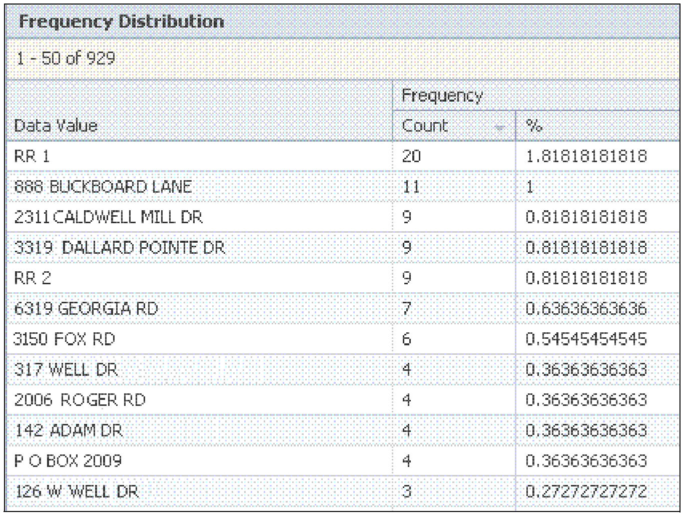


View the frequency distribution of a column. A frequency distribution shows details about the structure of a column
contains the column to view values that correspond with the column, add a new data value to the column, or creat

Total Rows: 1100
Data Class: Text
Cardinality: 929 84%
Minimum Value: 1 Mcmahon Hall-Uw

Figure 9-9 Address values in a graphic frequency distribution

Figure 9-10 shows the same distribution of values in tabular form with the value, count, and percentage of total population.



Frequency Distribution

1 - 50 of 929

	Frequency	
Data Value	Count	%
RR 1	20	1.81818181818
888 BUCKBOARD LANE	11	1
2311 CALDWELL MILL DR	9	0.81818181818
3319 DALLARD POINTE DR	9	0.81818181818
RR 2	9	0.81818181818
6319 GEORGIA RD	7	0.63636363636
3150 FOX RD	6	0.54545454545
317 WELL DR	4	0.36363636363
2006 ROGER RD	4	0.36363636363
142 ADAM DR	4	0.36363636363
P O BOX 2009	4	0.36363636363
126 W WELL DR	3	0.27272727272

Figure 9-10 Address values in a tabular frequency distribution

From this distribution, we can see that some addresses appear multiple times. The reason might be that the IT department for Bank B did not clean their data and remove duplicates. Alternatively the frequency of duplicate addresses might be the result of some other reason. To continue investigation, we must identify the individual records of duplicate addresses. Therefore, we create a data rule that identifies such incidents for records that violate the uniqueness requirements.

First, we create a basic rule definition. A basic rule definition can be used to create any number of data rules by binding the variables in the rule to different data columns. Because we are referring to a rule definition, we prefix its name with “Global” to denote that this rule is a basic rule that can be bound to any data element.

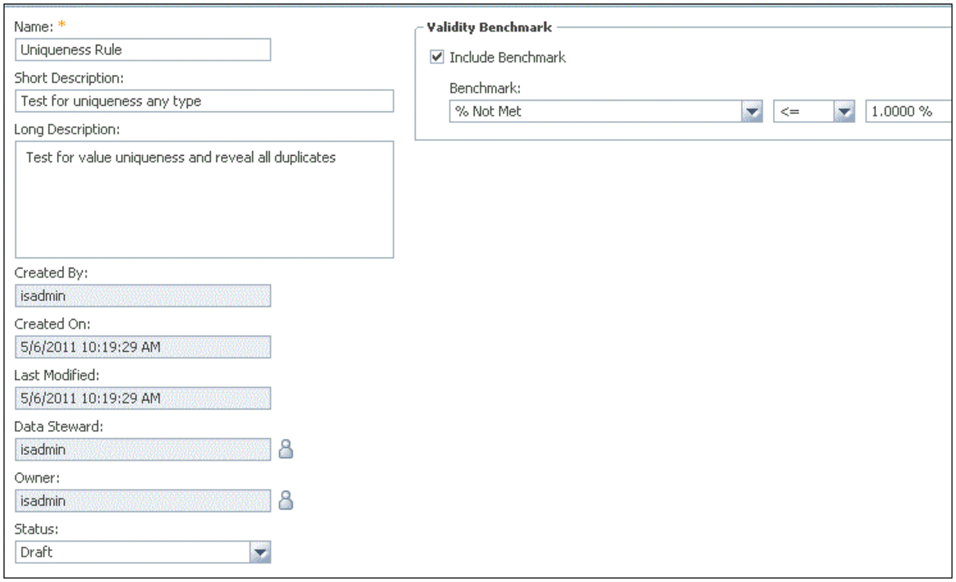
Rule definition is done with local variables that can be specific or general in nature. In general cases, we check for the existence of a value in a column or the uniqueness of value. For a rule that can be potentially applied on every column in the database, the variable can be ‘VARn’ or ‘DATAn’.

In more specific cases, the rule definition has to do with a particular data type (such as Social Security Number (SSN) or Balance), for which we might have specific rules. In this case, we can be specific with the variable names ‘SSN’ or ‘BALANCE’. At this stage, these names are only variable names. The data rule is instantiated only after the variables are bound to a data column.

Creating the rule definition overview or description

A rule definition requires two components. First it requires a name for the rule definition. Second it requires a logical statement (the rule logic), which is a formal expression of the conditions that data is expected to satisfy to meet the test requirements.

Figure 9-11 shows the creation of a simple global rule definition, which is a uniqueness rule that can be used to create multiple data rules in the future.



Name: * Uniqueness Rule

Short Description: Test for uniqueness any type

Long Description: Test for value uniqueness and reveal all duplicates

Created By: isadmin

Created On: 5/6/2011 10:19:29 AM

Last Modified: 5/6/2011 10:19:29 AM

Data Steward: isadmin

Owner: isadmin

Status: Draft

Validity Benchmark

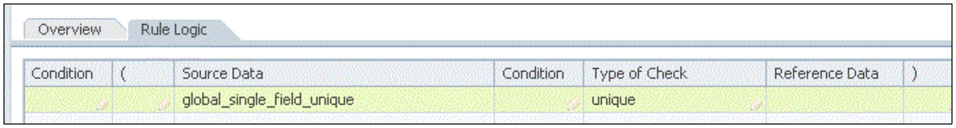
☑ Include Benchmark

Benchmark: % Not Met <= 1.0000 %

Figure 9-11 New rule definition panel

Creating rule logic for the rule definition

Figure 9-12 shows a simple rule that checks whether a field is unique. This rule shows that we only have two items: the local variable of the rules and the type of check that will be done against that local variable.



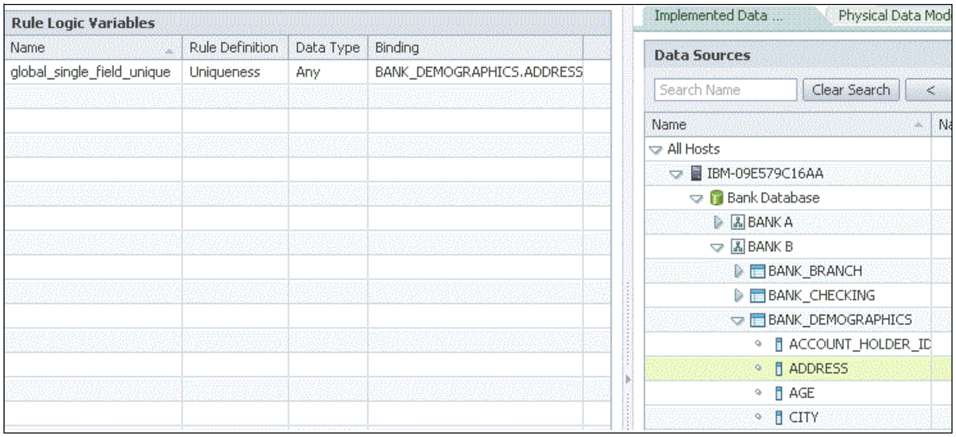
Condition	(	Source Data	Condition	Type of Check	Reference Data	)
		global_single_field_unique		unique		

Figure 9-12 Simple uniqueness rule expression

Similarly simple rules can be set up for other data quality checks such as exist, contains, and equals.

Binding and specifying output

The next step in the process, as mentioned earlier in the chapter, is to bind the variables in the logical rule definition to a physical column in a data source. Figure 9-13 shows that BANK_DEMPOGRAPHICS.ADDRESS is selected from the data sources in the right pane to be bound to the global_single_unique variable.



Rule Logic Variables

Name	Rule Definition	Data Type	Binding
global_single_field_unique	Uniqueness	Any	BANK_DEMOGRAPHICS.ADDRESS

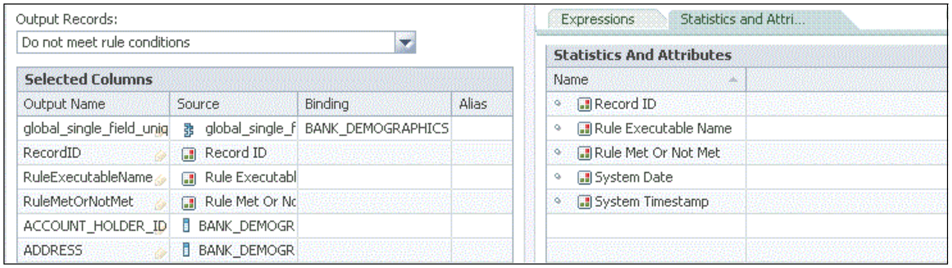
Data Sources

Name

- All Hosts
  - IBM-09E579C16AA
    - Bank Database
      - BANK A
      - BANK B
        - BANK_BRANCH
        - BANK_CHECKING
        - BANK_DEMOGRAPHICS
          - ACCOUNT_HOLDER_ID
          - ADDRESS
          - AGE
          - CITY

Figure 9-13 Binding a database column to a variable

Finally, before execution, we select the summary statistics and the columns that we want to show with the detailed results. Figure 9-14 shows that, on the left side, all of our selections are listed including the columns bound to variables, the statistics, and the additional columns. Selection is done from the pane on the right side on the appropriate tab.



Output Records: Do not meet rule conditions

Selected Columns

Output Name	Source	Binding	Alias
global_single_field_uniq	global_single_f	BANK_DEMOGRAPHICS	
RecordID	Record ID		
RuleExecutableName	Rule Executabl		
RuleMetOrNotMet	Rule Met Or Nc		
ACCOUNT_HOLDER_ID	BANK_DEMOGR		
ADDRESS	BANK_DEMOGR		

Statistics And Attributes

Name
Record ID
Rule Executable Name
Rule Met Or Not Met
System Date
System Timestamp

Figure 9-14 Selecting statistics and additional columns

9.4.2 Generating a data rule

To generate a data rule, follow these steps:

1. Select a rule definition from the list of available rule definitions. Then in the task menu, in the right pane (Figure 9-15), click **Generate Data Rule or Rule Set**.

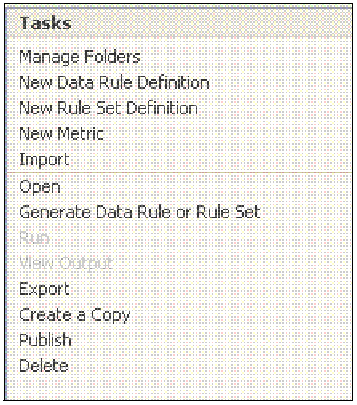


Figure 9-15 Tasks menu for data rules

2. Follow the same steps that you used to create a data definition and run a test. Name the rule (see “Rule naming guidelines” on page 296), create a short and long description, and assign the rule a data (rule) steward and a rule owner. The rule steward is often a different individual from the data steward. If you have not done this process before, complete the Validity Benchmark field in the Definition pane.
3. Bind variables in the data rule definition to columns in the target data sources:
   a. Select the logical variable (`global_single_field_unique` in this example) that you want to bind to an InfoSphere Information Analyzer data source.
   b. On the **Implemented Data** (sources) tab on the right, browse to identify the particular column you want to bind to the logical variable.
   c. Click the **Bind** button at the bottom of the pane to complete the binding.

9.4.3 Use case: Creating a data rule to monitor high value customers

Bank A wants to focus on the high-value customers from both Bank A and Bank B to properly treat them to gain their trust and ensure that they stay with the bank. In addition, customers of Bank A might also have accounts in Bank B that might qualify them for a higher status due to total balances in the larger bank.

Bank A now has two operational units: the original Bank A and the acquired Bank B, whose operations were not integrated.

The objective of this data rule is to calculate the total dollar values of the account balances of a customer and identify gold and platinum customers so that they can be offered the appropriate services. Customers might have both checking and savings accounts that need to be aggregated to balance their total accounts so that their gold or platinum status can be validated. Aggregation must be done across the banks.

Bank A holds only checking accounts. The data is in the BANKA.CHECKING table. This table has several nonvalid accounts that are identified by the account balance `= -999999`.

Bank B holds checking and savings accounts. The data is in the BANKB.CUSTOMERS and BANKB.ACCOUNTS tables. Customers might have checking and savings accounts. Therefore, account balances for both checking and savings accounts must be aggregated to compute the total account balance for a customer. Bank B also keeps track of demographic data about customers in the BANKB.DEMOGRAPHICS table.

Bank A defines two levels of high-value customers:

- Gold customers hold balances in excess of US$100,000.
- Platinum customers hold balances in excess of US$300,000.

Bank A wants to ensure that high value customers are treated appropriately. It wants to offer gold customers new investment opportunities. It also wants to offer platinum customers premium personal customer service when they call in the bank with issues.

The limits of US$100,000 and US$300,000 that qualify customer to be in one group or another might be used in several rules. In addition, the limit can change over time. To make it easier to maintain these rules and other rules, with InfoSphere Information Analyzer, we can create global variables. These variables are managed separately from the rules and can be used in as many rules as needed while managed at a single point.

A global variable can be bound to a data source so that the value of the variable in run time is determined by the value of the data source. For example, you need to specify a rule that includes an exchange rate. Exchange rates change every day and, for some applications, they change a few times a day or every few seconds. Changing the fixed value of a global variable is impractical. Instead, we can bind the global variable to a data source that changes as frequently as needed to reflect market exchange rate as a way to address this problem.

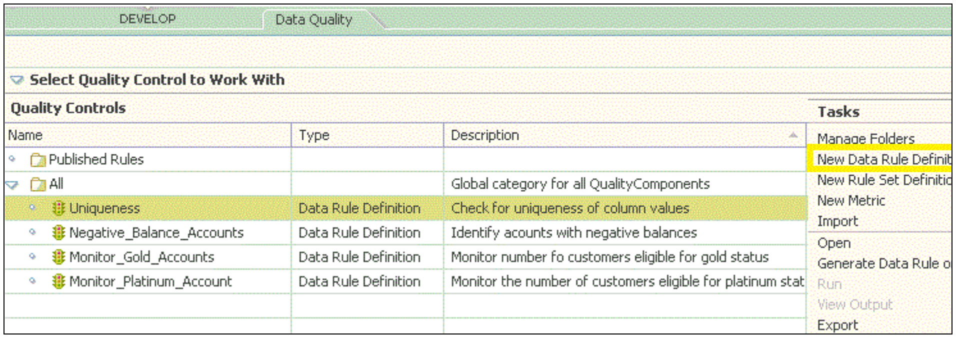
When selecting a data source, make sure that the bindings that you set are of the same data type.

The next step is to create a rule to monitor gold customers.

9.4.4 Creating rules to monitor gold customers

The following example demonstrates the creation of a complex data rule that involves multiple conditions. The rule is intended to count the number of customers that qualify for gold status. It is a none standard rule in the sense that it does not look for records that violate data standards or requirements. However, it demonstrates the versatility of tasks that can be achieved with data rules.

To start the creation of a new data rule definition, from the **DEVELOPMENT** tab, select **Data Quality** → **New Data Rule Definition** (Figure 9-16).



Name	Type	Description
Published Rules		
All		Global category for all QualityComponents
Uniqueness	Data Rule Definition	Check for uniqueness of column values
Negative_Balance_Accounts	Data Rule Definition	Identify acounts with negative balances
Monitor_Gold_Accounts	Data Rule Definition	Monitor number fo customers eligible for gold status
Monitor_Platinum_Account	Data Rule Definition	Monitor the number of customers eligible for platinum stat

Figure 9-16 Creating a data rule definition

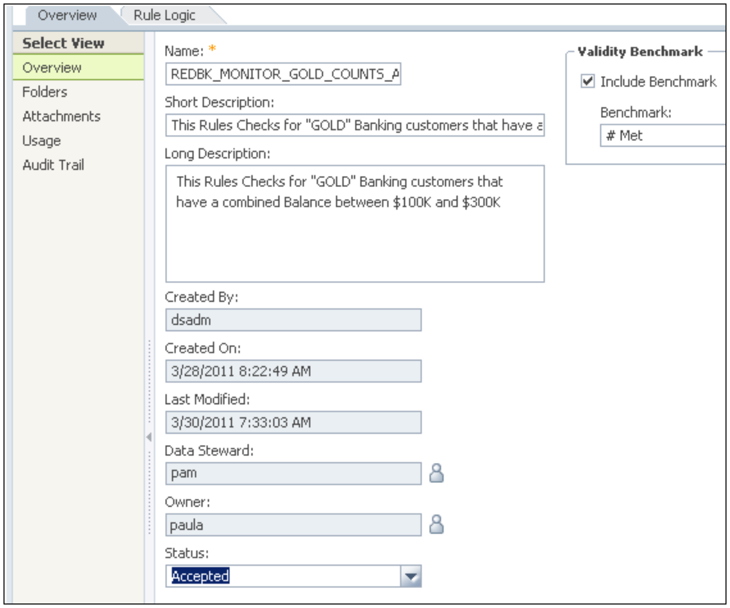
Creating a rule definition

Every rule or rule set must have an "Overview" section. This section contains information and metadata that describe the rule. It contains the name, a description, and other fields that control the functions of the rules (such as threshold values), link to business glossary terms, or monitor the status of the rule. Figure 9-17 on page 311 shows these fields, which are explained as follows:

Name Describes the basic function of the rule definition.

Status Is used to determine if a rule is ready to be promoted (exported) to the next level for testing. While a developer is working on the rule, this field must be set to **Candidate**. When the rule is ready for testing, the status is **Accepted**. After the rule passes acceptance testing and is ready to go into production, the status changes to **Standard**.

Validity Benchmark Determines the aspect of the rules to determine the criteria by which data might be considered good or bad and to turn on the alert. Here we choose **# Met** as the benchmark statistic on which validity will be tested.



Name: * REDBK_MONITOR_GOLD_COUNTS_A

Short Description: This Rules Checks for "GOLD" Banking customers that have a

Long Description: This Rules Checks for "GOLD" Banking customers that have a combined Balance between $100K and $300K

Created By: dsadm

Created On: 3/28/2011 8:22:49 AM

Last Modified: 3/30/2011 7:33:03 AM

Data Steward: pam

Owner: paula

Status: Accepted

Validity Benchmark

☑ Include Benchmark

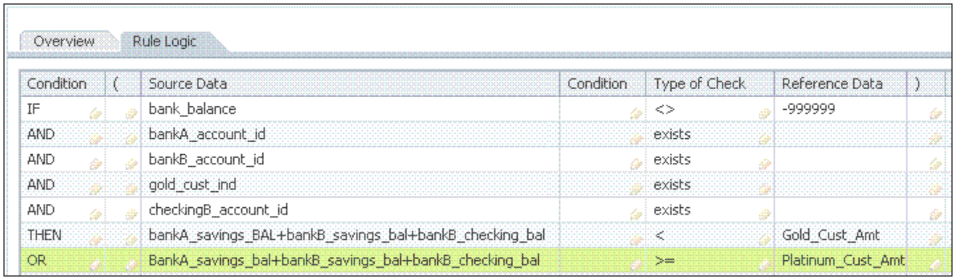
Benchmark: # Met

Figure 9-17 Rule definition pane

Defining the logic of the rule

In the definition of the rules logic, we formally express the intent of the rule. In this case, the intent of the rule is to find all customers that satisfy the gold customer criterion. A gold customer has accounts where the sum of the balances is greater than or equal to US$100,000 and smaller than US$300,000. The requirement is to find these customers and then count and list them. Over time, we will gain a picture of changes and trends in regard to this number and possibly be able to analyze them relative to bank initiatives and promotion campaigns.

Initially we define a filter that specifies the conditions that the customer has accounts in Bank A and Bank B, has a checking account, and is designated as a gold customer. In the rule condition, we look for the total balance that is outside the range of the gold customer criteria. With a negative aspect, the rule marks those customers that are inside the range, namely gold customers. See Figure 9-18 on page 312.

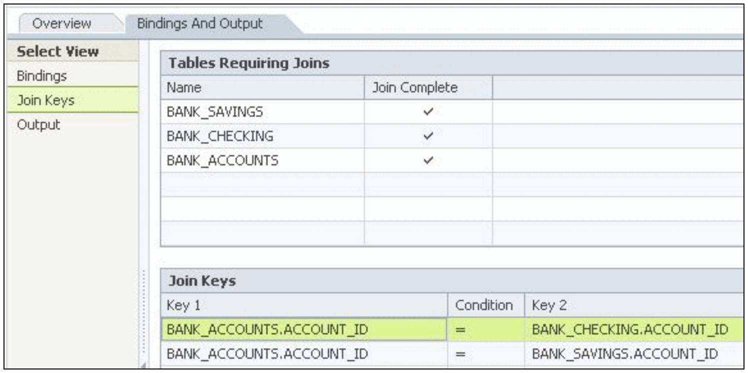


Condition	(	Source Data	Condition	Type of Check	Reference Data	)
IF		bank_balance		<>	-999999	
AND		bankA_account_id		exists		
AND		bankB_account_id		exists		
AND		gold_cust_ind		exists		
AND		checkingB_account_id		exists		
THEN		bankA_savings_BAL+bankB_savings_bal+bankB_checking_bal		<	Gold_Cust_Amt	
OR		BankA_savings_bal+bankB_savings_bal+bankB_checking_bal		>=	Platinum_Cust_Amt	

Figure 9-18 Rule Logic tab for the gold customer rule definition

We can break down this rule into three parts. The first part of the rule, which is the logic before the THEN statement, is called the *filter*. It verifies that the data being validated has a valid bank balance and the account number exists.

The second part of the rule ensures that the ACCOUNT_ID in the BANK_ACCOUNTS table matches the ACCOUNT_ID that is in the BANK_CHECKING and BANK_SAVINGS accounts of the person. When you realize the rule and bind the logical variables to data columns, the system detects that you are involving data from multiple tables. It then prompts you to provide the join conditions as shown in Figure 9-19.



Tables Requiring Joins

Name	Join Complete
BANK_SAVINGS	✓
BANK_CHECKING	✓
BANK_ACCOUNTS	✓

Join Keys

Key 1	Condition	Key 2
BANK_ACCOUNTS.ACCOUNT_ID	=	BANK_CHECKING.ACCOUNT_ID
BANK_ACCOUNTS.ACCOUNT_ID	=	BANK_SAVINGS.ACCOUNT_ID

Figure 9-19 Gold customer rule joins

The third part of the rule is the rule definition logic for a gold customer. The logic is shown as follows:

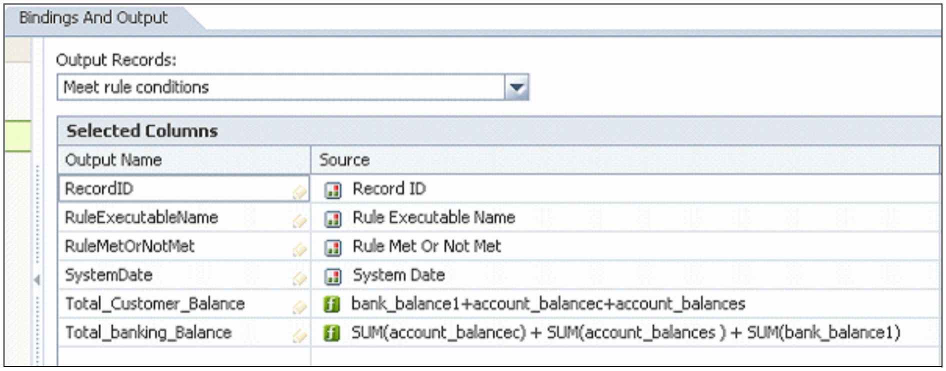
```
Not(A) OR Not(B) = Not(A AND B)
```

Hint: When programming a data definition rule, always try to use the OR condition.

In this example, Not A (customers with a balance less than US$100,000) or Not B (customers with a balance greater than or equal to US$300,000) was used.

Rules outputs for the gold customer

Figure 9-20 shows the selection of statistics and output columns. Keep in mind that columns might also include functions of columns.



Output Records: Meet rule conditions

Selected Columns

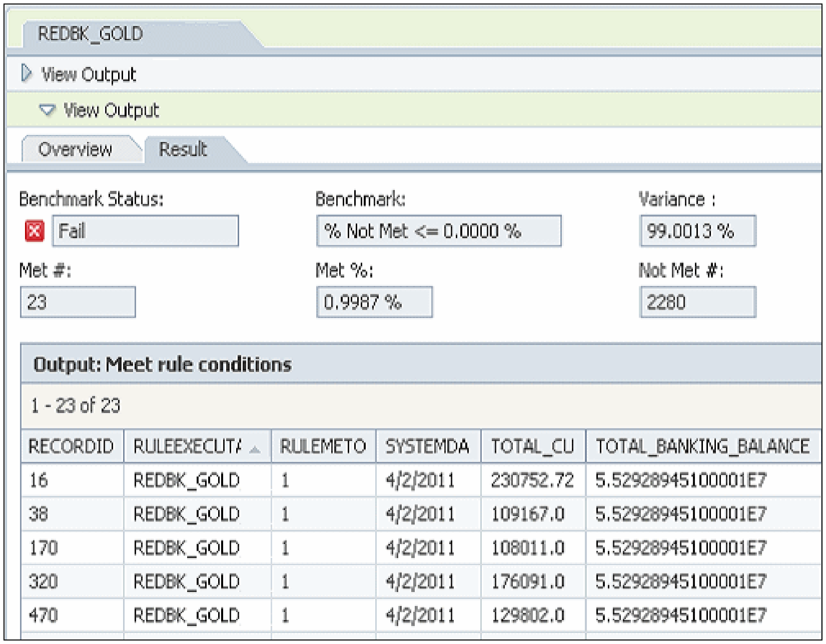
Output Name	Source
RecordID	Record ID
RuleExecutableName	Rule Executable Name
RuleMetOrNotMet	Rule Met Or Not Met
SystemDate	System Date
Total_Customer_Balance	bank_balance1+account_balancec+account_balances
Total_banking_Balance	SUM(account_balancec) + SUM(account_balances) + SUM(bank_balance1)

Figure 9-20 Output column selection for the gold customer

Some internal variables come standard with InfoSphere Information Analyzer that are automatically saved during rules execution. They include items such as RecordID, Rules Executable Name, Rule Met or Not Met, and System Date.

Additionally, logical variables can be included in the selected output in which any of the internal functions can be used. In this example, we used simple addition to sum up the Total_Customer_Balance, plus the use of an aggregate function to calculate the entire sum of all of Total_banking_Balance.

After the rule executes, we can review the results, summary, and details of the records that satisfy the conditions of the rule as shown in Figure 9-21.



Benchmark Status: ☒ Fail

Benchmark: % Not Met <= 0.0000 %

Variance : 99.0013 %

Met #: 23

Met %: 0.9987 %

Not Met #: 2280

Output: Meet rule conditions

1 - 23 of 23

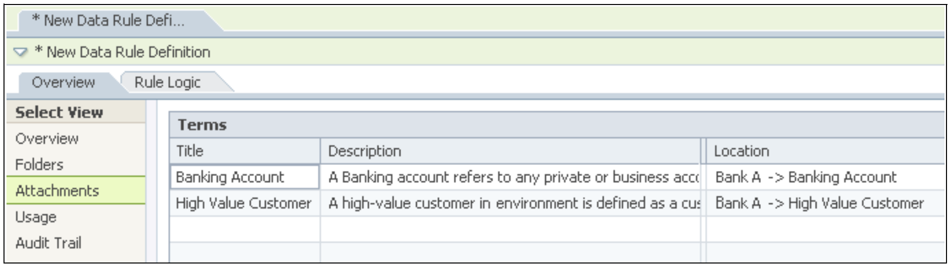
RECORDID	RULEEXECUT/	RULEMETO	SYSTEMDA	TOTAL_CU	TOTAL_BANKING_BALANCE
16	REDBK_GOLD	1	4/2/2011	230752.72	5.52928945100001E7
38	REDBK_GOLD	1	4/2/2011	109167.0	5.52928945100001E7
170	REDBK_GOLD	1	4/2/2011	108011.0	5.52928945100001E7
320	REDBK_GOLD	1	4/2/2011	176091.0	5.52928945100001E7
470	REDBK_GOLD	1	4/2/2011	129802.0	5.52928945100001E7

Figure 9-21 Gold customers outputs

Linking rule definitions to business glossary terms

Attaching a business rule to a data rule broadens the sphere of knowledge that business users can reach by searching or browsing the business glossary. Viewing a business term in the glossary reveals that data rules are associated with the term or the concept represented by the term. It can also reveal that data quality requirements are applied to data associated with the term. Association of business terms to a rule is done from the Attachments view of the Rule Definition pane.

Figure 9-22 shows that the business terms “Banking Account” and “High Value Customer” are associated with this business rule.



Title	Description	Location
Banking Account	A Banking account refers to any private or business acc	Bank A -> Banking Account
High Value Customer	A high-value customer in environment is defined as a cus	Bank A -> High Value Customer

Figure 9-22 Assignments of business terms to rule definitions

9.5 Data rules and performance consideration

As you will find, there is more to data rules than what has been discussed previously. Types of data rules and the way they are constructed have a significant impact on performance and eventually on the results. Certain functions require less resources than others. Applying rules on multiple resources and using joins can complicate processing immensely, drawing heavily on resources. This section highlights some of these issues and, where possible, offers workarounds.

9.5.1 Types of data rules

Given that the checks and function used in a rule affect the amount of resources that need be used to resolve the rule, rules come in two types:

- *Scalar rules* are the simplest rules that can be executed. A scalar rule requires only the field value of a record to do the evaluation. Consider the following example of a scalar rule:

  ```
  If UCASE(country) = ‘US’ then Country_Code = ‘001’
  ```

  This code verifies that records that have US for ‘country’ also have the country_code ‘001’. Although it involves several tests, the rule requires only values in the actual record to be evaluated. Scalar rules execute linearly and can be executed in parallel, taking full advantage of the InfoSphere DataStage parallel engine.

- *Columnar rules* usually require more than a single pass over the column specified in the rule. A simple rule, such as `unique id`, requires two passes over the data first to sort the column and then to determine whether duplicates exist.

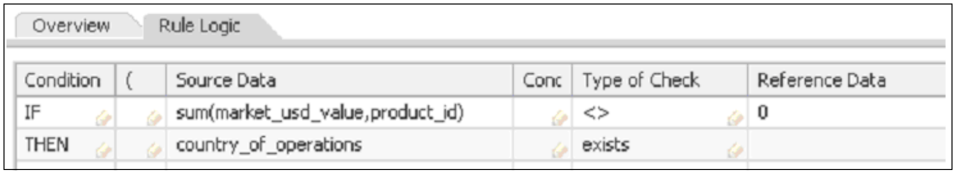
This rule can have a serious impact if executed over a large data set. Occasionally, columnar rules reduce the ability of the job to completely run in parallel. Columnar rules contain aggregation, such as Sum(), Avg(), Min(), Max(), or counting, such as count().

9.5.2 Using join tables in data quality rules

It is common to have a data quality rule, such as the following example, that involves data from multiple tables:

```
Show all Accounts in the Account table with null in the address field
for Accounts with balance value in Holdings table not equal to zero.
```

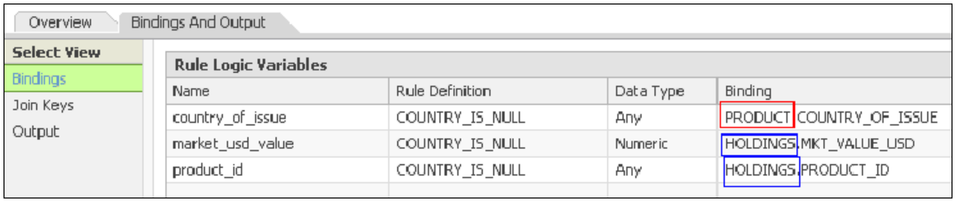
In this case, two tables, Account (product) and Holdings, need to be joined. Data rule definition is built by using user variables. At this point in InfoSphere Information Analyzer, the logical variables in the rule definitions are not bind to the actual data. We use meaningful variable names so that we can ease the translation of the rule and the binding process (Figure 9-23). However, at this stage, they are still only logical variables.



Condition	(	Source Data	Conc	Type of Check	Reference Data
IF		sum(market_usd_value,product_id)		<>	0
THEN		country_of_operations		exists	

Figure 9-23 A rule involving two entities

Generating a data rule from the rule definition requires going through binding actual physical data columns to the variables in the rule definition. As shown in Figure 9-24, the physical columns are in the PRODUCT and HOLDINGS tables.

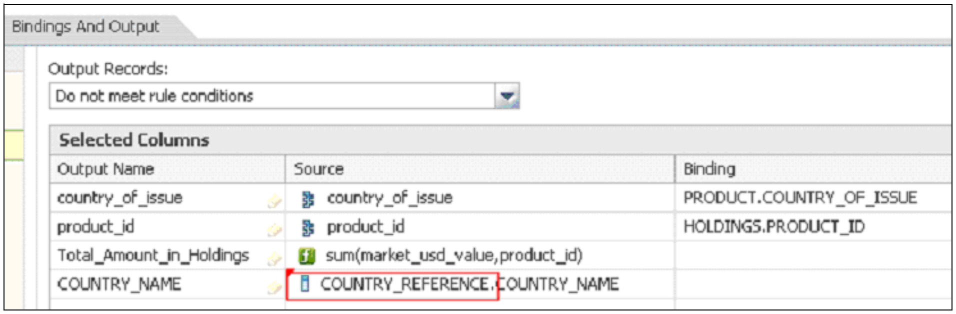


Rule Logic Variables

Name	Rule Definition	Data Type	Binding
country_of_issue	COUNTRY_IS_NULL	Any	PRODUCT.COUNTRY_OF_ISSUE
market_usd_value	COUNTRY_IS_NULL	Numeric	HOLDINGS.MKT_VALUE_USD
product_id	COUNTRY_IS_NULL	Any	HOLDINGS.PRODUCT_ID

Figure 9-24 Binding physical data columns to variables

If variables in the rule are bound to columns in more than one table, it invokes a join step, asking you to identify the join key or keys that enable the association of the tables in question. You must be aware also that specifying fields from other tables in the output will trigger an additional join. Here, we want to display the full country name based on a country code that comes from a country_reference table, as shown in Figure 9-25. This rule involves an additional join to attach the country name to the record information in the detailed results.



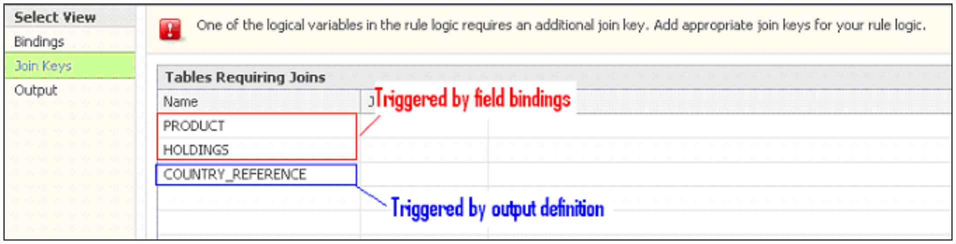
Output Records: Do not meet rule conditions

Selected Columns

Output Name	Source	Binding
country_of_issue	country_of_issue	PRODUCT.COUNTRY_OF_ISSUE
product_id	product_id	HOLDINGS.PRODUCT_ID
Total_Amount_in_Holdings	sum(market_usd_value,product_id)	
COUNTRY_NAME	COUNTRY_REFERENCE.COUNTRY_NAME	

Figure 9-25 Output selection pane

After the system analyzes the rule definition and the output requirements, it determines how many joins are required, as shown in Figure 9-26.



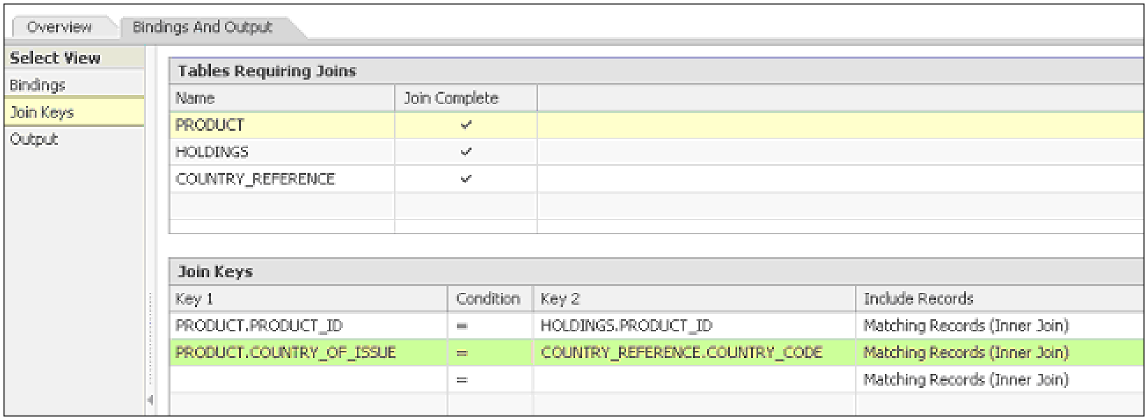
One of the logical variables in the rule logic requires an additional join key. Add appropriate join keys for your rule logic.

Tables Requiring Joins

Name
PRODUCT
HOLDINGS
COUNTRY_REFERENCE

Figure 9-26 Table identified for join

Figure 9-27 shows the join keys for the tables involved.



Tables Requiring Joins

Name	Join Complete
PRODUCT	✓
HOLDINGS	✓
COUNTRY_REFERENCE	✓

Join Keys

Key 1	Condition	Key 2	Include Records
PRODUCT.PRODUCT_ID	=	HOLDINGS.PRODUCT_ID	Matching Records (Inner Join)
PRODUCT.COUNTRY_OF_ISSUE	=	COUNTRY_REFERENCE.COUNTRY_CODE	Matching Records (Inner Join)
	=		Matching Records (Inner Join)

Figure 9-27 Establishing join conditions

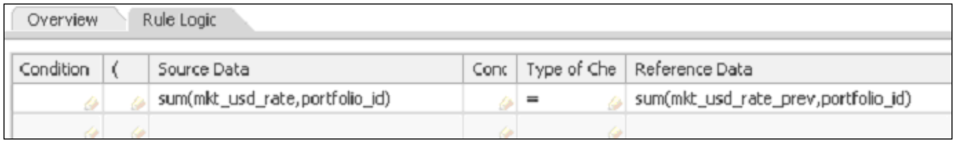
When a data rule is submitted to run, an InfoSphere DataStage job is created and executed. It performs data extraction, the necessary joins, and the evaluation of rule criteria. When doing these tasks, it delegates certain functions to database SQL processing. It does other processes through InfoSphere DataStage operations. In some cases, this processing might raise the issue of cartesian products.

9.5.3 Cartesian products and how to avoid them

A cartesian product is the result of joining two sets of data in a way that all rows in one set are joined with all rows in the other set. Having N records in one table and M records in the other cartesian join generate a new table with NxM records.

Defining rules with joins: Joins are executed first, and then the rule is applied to the result of the join. Always keep in mind this point when defining a data rule for InfoSphere Information Analyzer.

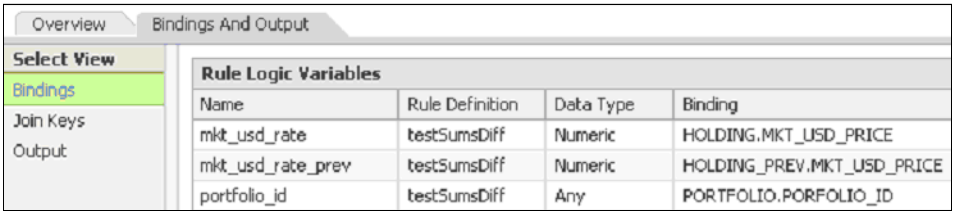
The following example shows multiple rows for each portfolio within the holding table. A new data rule compares the total value of the portfolio within the holding table for today's date and yesterday's date and then reports whether the value is not the same. Figure 9-28 shows how the definition looks. We will look for exceptions to this rule.



Condition	(	Source Data	Conc	Type of Che	Reference Data
		sum(mkt_usd_rate,portfolio_id)		=	sum(mkt_usd_rate_prev,portfolio_id)

Figure 9-28 A rule involving columns from different tables

Figure 9-29 shows bindings of the rule variables.

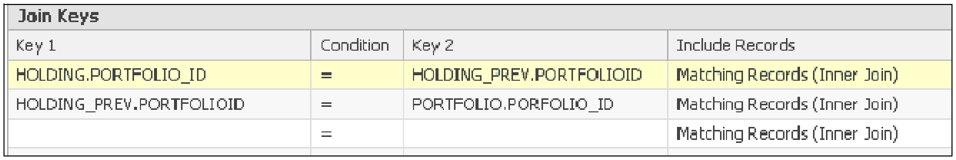


Rule Logic Variables

Name	Rule Definition	Data Type	Binding
mkt_usd_rate	testSumsDiff	Numeric	HOLDING.MKT_USD_PRICE
mkt_usd_rate_prev	testSumsDiff	Numeric	HOLDING_PREV.MKT_USD_PRICE
portfolio_id	testSumsDiff	Any	PORTFOLIO.PORFOLIO_ID

Figure 9-29 Column binding

Figure 9-30 shows the join conditions for the two tables.



Join Keys

Key 1	Condition	Key 2	Include Records
HOLDING.PORTFOLIO_ID	=	HOLDING_PREV.PORTFOLIOID	Matching Records (Inner Join)
HOLDING_PREV.PORTFOLIOID	=	PORTFOLIO.PORFOLIO_ID	Matching Records (Inner Join)
	=		Matching Records (Inner Join)

Figure 9-30 Join conditions for the rule

Looking at the rule definition, we might wrongly assume the following sequence of events during rule processing:

1. Select data from the holding table and from the holding_prev table.
2. Calculate the sum.
3. Join results on portfolio ID.
4. Compare the sums.

However, if we look at the fragments of the generated OSH code, we see the output in Example 9-1 at the beginning sequence of SQL and InfoSphere DataStage operations.

> **OSH:** The OSH Orchestrate scripting language drives the execution of an InfoSphere DataStage job.

Example 9-1 Generated OSH code snippet

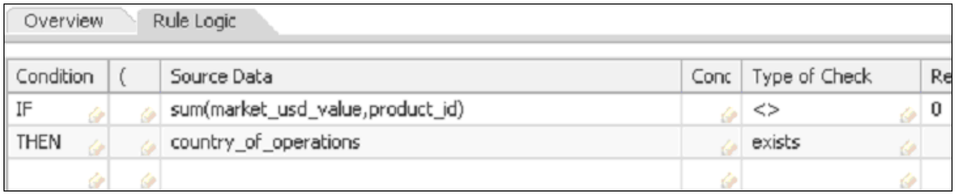
```
Step 1
   select
   "MKT_USD_PRICE" as "mkt_usd_rate",
   "PORTFOLIO_ID" as "key_1553818330"
    from BANK2."HOLDING"
Step 2
   select
   "MKT_USD_PRICE" as "mkt_usd_rate_prev",
   "PORTFOLIOID" as "key_1553818330",
   "PORTFOLIOID" as "key1810270001"
    from BANK2."HOLDING_PREV"
Step 3
   innerjoin
    -key key_1553818330
   0< modify_1.v
   1< modify_2.v
   0> innerjoin_1.v
Step 4
   select
   "PORFOLIO_ID" as "portfolio_id",
   "PORFOLIO_ID" as "key1810270001"
    from BANK2."PORTFOLIO"
Step 5
   innerjoin
    -key key1810270001
   0< modify_4.v
   1< modify_5.v
   0> innerjoin_2.v
```

Steps 3 and 5 are InfoSphere DataStage own join operations. These operations take two input streams (modify_1 and modify-2 or modify_4 and modify_5) to create the inner joins. The way this process works creates a problem. Because multiple records for the portfolio are in the holding and holding_prev tables, trying to join these tables on portfolio_id generates a cartesian product. Therefore, subsequent sums are incorrect.

Several methods are available to avoid this problem. You can perform the joins on the host database, if possible, using the database engine to perform the required joining of facts from the two tables. InfoSphere Information Analyzer provides a filtering clause in the rule to limit the number of records that participate in the join operation.

9.5.4 Applying filtering in data quality rules

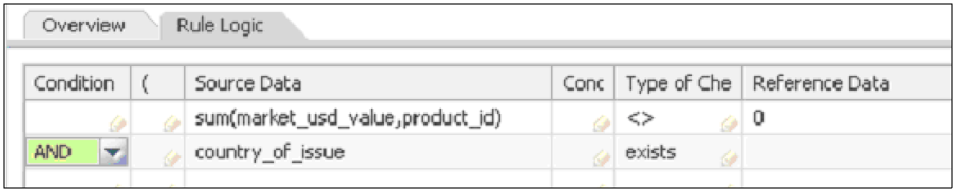
When creating a data rule in InfoSphere Information Analyzer, you can narrow down the scope of records that are evaluated. Creating database views and virtual tables is one way to do it. Another way to filter data is to use an “IF/THEN” construct within a rule definition. Figure 9-31 shows the example used previously that identifies the missing country of operations.



Condition	(	Source Data	Conc	Type of Check	Re
IF		sum(market_usd_value,product_id)		<>	0
THEN		country_of_operations		exists	

Figure 9-31 Rule with an IF/THEN filter

Figure 9-32 shows another, but incorrect, way to code this rule by using the AND operator.



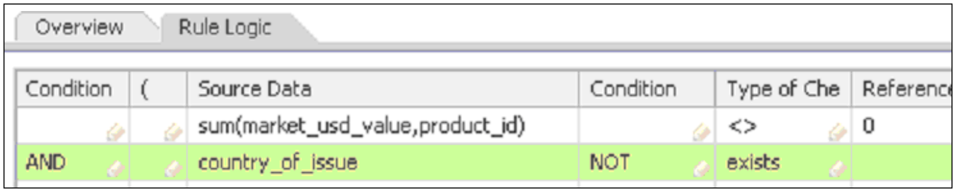
Condition	(	Source Data	Conc	Type of Che	Reference Data
		sum(market_usd_value,product_id)		<>	0
AND		country_of_issue		exists	

Figure 9-32 Rule with an AND operator

Where is the difference? In the first definition, the logic works as follows: Out of all the product table records, select only records where the total market value in the holding table does not equal zero. Then, for any of these records, check if the country of operation is null. As a result, when we define our rule output as “Do not meet rule condition,” we see only records with total market value <> 0 and records with country of operation set to null.

For the second definition, if we keep records in the output that “Do not meet rule condition,” we get records that might violate the first or second condition. This means that you might see records with a valid country of operation but where the total market value equals 0.

Figure 9-33 shows an example where we define the rule with a negative perspective, such as “What is not a valid condition.”



Condition	(	Source Data	Condition	Type of Che	Referenc
		sum(market_usd_value,product_id)		<>	0
AND		country_of_issue	NOT	exists	

Figure 9-33 Negative aspect rule

The rule output changes to show records that “Meet condition.”

Keep in mind that, with either approach, all records are extracted from the source and then joined. Sums are calculated, and then the rule logic is evaluated. Using database views or virtual tables pushes record filtering to the database level.

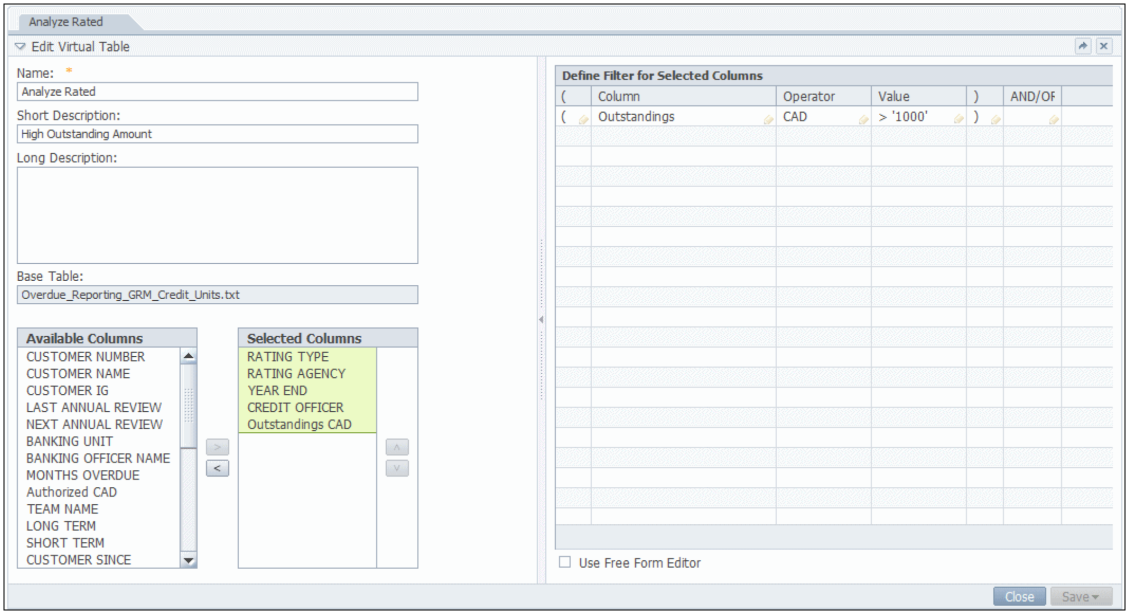
9.5.5 Filtering versus sampling

As you know, sampling is a feature made available to InfoSphere Information Analyzer to perform the analysis only on fraction of the records. However, when sampling is done within a rule, the effect on execution is different. If sampling is done within a rule, all records of the source must be transferred to InfoSphere DataStage before sampling. Sampling reduces the number of records that are evaluated and the size of the output. However, it does not reduce the I/O transfer rate between the data source and InfoSphere DataStage.

However, filtering, as indicated previously, reduces the number of records that are transferred from the source to InfoSphere DataStage. Yet you must remember that filtering and sampling serve different purposes. Although filtering is intended to select a particular subpopulation, sampling is used to select a representative subpopulation.

9.5.6 Virtual tables versus database views

By using a virtual table, InfoSphere Information Analyzer can filter the number of rows and select columns to produce a much smaller and focused table on which analysis and rules can be applied. A virtual table is created by selecting rows using a simple criteria and selecting columns from the columns in the table. No column manipulation is available, as shown in Figure 9-34.



Analyze Rated

Edit Virtual Table

Name: *
Analyze Rated

Short Description:
High Outstanding Amount

Long Description:

Base Table:
Overdue_Reporting_GRM_Credit_Units.txt

Available Columns:
- CUSTOMER NUMBER
- CUSTOMER NAME
- CUSTOMER IG
- LAST ANNUAL REVIEW
- NEXT ANNUAL REVIEW
- BANKING UNIT
- BANKING OFFICER NAME
- MONTHS OVERDUE
- Authorized CAD
- TEAM NAME
- LONG TERM
- SHORT TERM
- CUSTOMER SINCE

Selected Columns:
- RATING TYPE
- RATING AGENCY
- YEAR END
- CREDIT OFFICER
- Outstandings CAD

Define Filter for Selected Columns

(	Column	Operator	Value	)	AND/OF
(	Outstandings	CAD	> '1000'	)	

□ Use Free Form Editor

Figure 9-34 Virtual table definition pane

The database view enables the whole range of SQL capabilities to create views as complex as needed, including complex joins and nested queries. The database view also has the advantage of harnessing the engine of the database and its optimization mechanism to extract the records and columns or their variations as specified. In some cases, the requirements are such that the database view is the only solution.

The metadata of the database view must be imported from the source as any other table in the same database. Although reading data for analysis might require only I/O, the database view might require additional resources from the database server. This approach is something that you need to consult and coordinate with the database administrator to determine the availability of resources or time slots.

A database view might be required for select statements with the following conditions:

- Select Distinct
- Joins
- Counts (*)
- Group By's
- Dynamic SQL Queries or views based upon other values within another table

Consider the following example:

```
Select * from Portfolio_table where effective_date = (select run_date
from batch_run_date_Table)
```

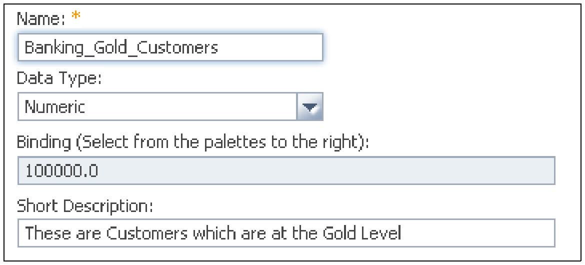
9.5.7 Global variables

Global variables are used to establish values that are used repeatedly. For example, threshold values of US$100,000 and US$300,000 that define the total balances qualifying a customer to be Gold or Platinum customer can be set as global variables. These values can be used in other data rules combined with other conditions. Furthermore, these values can change over time to reflect different eligibility criteria. To avoid the need to update every related rule individually, we can use global variables.

Global variables are available system-wide and are created in advance. They are displayed on the **Global Variable** tab. Global variables can be bound to a concrete concept such as a physical source or a defined constant.

To create global variables, select **Metadata Management** → **Global Logical Variables**.

Figure 9-35 shows a variable definition panel.



Name: *

Banking_Gold_Customers

Data Type:

Numeric

Binding (Select from the palettes to the right):

100000.0

Short Description:

These are Customers which are at the Gold Level

Figure 9-35 Global variable definition

9.6 Rule sets

A *rule* is a condition that you expect data to satisfy. The results of rule execution expose the records that do not meet the condition. But, what if you have other conditions on the same data set and same records? Running these rules separately gives the results for each rule individually and produces lists of records that violate their individual conditions. It is conceivable that some records will appear in multiple results lists. If there is any correlation between these lists, it will be difficult to assess. If there are any dependencies that might cause the occurrence of multiple data errors, they will be impossible to determine.

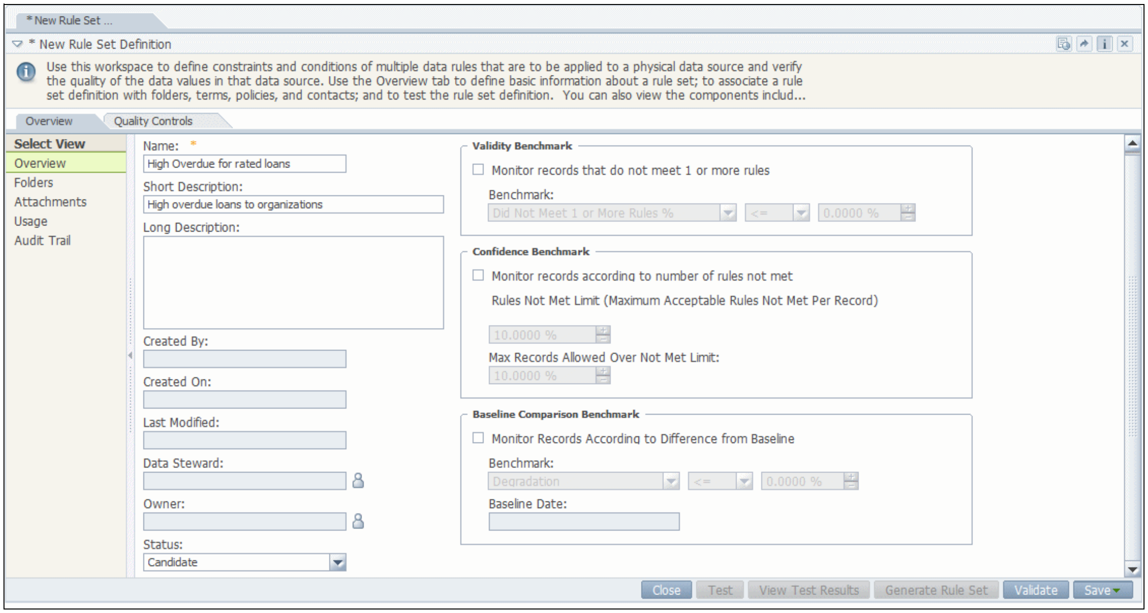
Rule sets are collections of rules that are run together as a single unit to generate statistics of failure or success of rules in a number of levels:

- Rule-level exception, such as the number of records that pass or fail the conditions of a given rule
- Record-level statistics, such as the number of rules that a particular record breaks and the confidence in the record

> **Confidence level:** The confidence level of a record is the percentage of total rules in a rule set violated by the record.

- Rule pattern statistics, such as the number of records that break a certain pattern of rules, such as rules 1, 4, and 5
- Data source-level statistics, such as the average level of a broken rule for each record over the data source

In addition, rule sets provide the means to define certain important statistics. In the Rule Set Definition pane (Figure 9-36), you can establish benchmark values on three important statistics.



*New Rule Set ...

* New Rule Set Definition

Use this workspace to define constraints and conditions of multiple data rules that are to be applied to a physical data source and verify the quality of the data values in that data source. Use the Overview tab to define basic information about a rule set; to associate a rule set definition with folders, terms, policies, and contacts; and to test the rule set definition. You can also view the components includ...

Name: * High Overdue for rated loans

Short Description: High overdue loans to organizations

Long Description:

Created By:

Created On:

Last Modified:

Data Steward:

Owner:

Status: Candidate

Validity Benchmark

□ Monitor records that do not meet 1 or more rules

Benchmark: Did Not Meet 1 or More Rules % <= 0.0000 %

Confidence Benchmark

□ Monitor records according to number of rules not met

Rules Not Met Limit (Maximum Acceptable Rules Not Met Per Record)

10.0000 %

Max Records Allowed Over Not Met Limit: 10.0000 %

Baseline Comparison Benchmark

□ Monitor Records According to Difference from Baseline

Benchmark: Degradation <= 0.0000 %

Baseline Date:

Figure 9-36 Rule Set Definition pane

The Rule Set Definition area contains the following benchmark group boxes:

- Validity Benchmark

  This group box is used to monitor the number or percentage of records that meet or do not meet one or more of the rules in the rule set.

- Confidence Benchmark

  This group box is used to specify a pass or fail threshold according to the number or percentage of rules in a rule set that a record fails. For example, if you have a rule set that contains 10 rules, you might want to consider records that fail half of the rules (50%) as low confidence records. Your data quality standards might be willing to accept a certain number of low confidence records. However, if the number of low confidence records exceeds the set benchmark, the rule set will indicate a failed confidence in the data set.

This group box contains the following threshold settings:

Rules Not Met Limit The maximum percentage of rules that a record does not meet before it is considered low confidence.

Max Records Allowed Over Not Met Limit

The maximum percentage of low confidence records that you are willing to accept before the benchmark fails.

- Baseline Comparison Benchmark

  The baseline comparison quantifies the difference in quality between the current execution of a rule set and a baseline run of a rule set by comparing the distribution of the number of failed rules by record. It evaluates the number of rules that are violated per record against the number of rules that are violated per record in a baseline run.

  It looks at the following two dimensions between the current run and the baseline:

  - It evaluates the distribution of rules violated per record against the distribution number of rules violated per record in the baseline run.
  - It evaluates the total number of records that have one or more violations in the current run versus the baseline run.

> **Important rule set definition:** Rules must be consistent in the perspective in which they are expressed. All rules must be expressed either in a positive perspective (“what we want”) or in a negative perspective (“what we do not want”). If the rules are not consistent in perspective, the exceptions to the rules will reflect a mix of both good and bad records. It is difficult to sort through details to identify the difference. The associated statistics will be meaningless or will provide false impressions and decisions driven by the wrong information.

Consider the following examples of a rule set definition:

- You are evaluating to see that data exists in a particular reference list.
- You want to evaluate that a particular field is in a specific range of values, such as a compound condition involving AND or OR. For example, the field is greater than *x* AND less than *y*.
- You want to evaluate the data to see if it matches a particular format.

Rule sets can also be used to break down complex rule logic into smaller, simpler rules. Executing them as a rule set yields equivalent results. In addition, smaller, simpler rules are more reusable than large complex ones. A collection of simple rules can be used as building blocks to create complex logic in the form of a rule set.

Because rules in a rule set are run simultaneously, rather than running them separately, the execution performance of rule sets is much better than running rules on their own.

9.7 Metrics

Metrics provide the capability to consolidate measurements from various data analysis steps into a single, meaningful measurement for data quality management purposes. Metrics can be used to reduce hundreds of detailed analytical results into a few meaningful measurements that effectively convey the overall data quality condition.

Metrics are user-defined objects that do not analyze data. Instead, they provide mathematical calculation capabilities that can be performed on statistical results from data rules, data rule sets, and metrics.

Metrics can be applied to single or multiple rules or rule sets. When naming a metric, it is best to include the type of measure that you are doing in the name. Include a measure interval if relevant, such as `AcctTable.Name_Exists_CostFactor_EOM`, which indicates the end of month. Likewise, an interval of `AcctTable.Name_Exists_CostFactor_EOD` refers to the end of day. When the interval is applied to a single rule or a single rule set, you can add clarity by including the name of that rule or rule set.

Now that we have established some standards for naming conventions, we go back to InfoSphere Information Analyzer. At a basic level, a metric can express a cost or weighting factor on a data rule. For example, the cost of correcting a missing date of birth might be US$1.50 per exception, which can be expressed as a metric. In this case, note the following details about the metric:

- The metric has the following condition:

  ```
  Date of Birth Rule Not Met # * 1.5
  ```

- The metric has the following possible result:

  ```
  If Not Met # = 50, then Metric = 75
  ```

At a more compound level, the cost for a missing date of birth might be the same US$1.50 per exception, where a bad customer type is only US$.75, but a missing or bad tax ID costs US$25.00. The metric has the following condition:

```
(Date of Birth Rule Not Met # * 1.5) +
(Customer Type Rule Not Met # * .75) +
(TaxID Rule Not Met # * 2.5)
```

Metrics might also be used as super rules that have access to data rules, rule sets, and metric statistical outputs. These rules can include tests for end-of-day, end-of-month, or end-of-year variances. Alternatively, they might reflect the evaluation of totals between two tables such as a source-to-target process or a source that generates results to both an accepted and a rejected table. The totals must match.

Metrics function

When a large number of data rules is being used, the results from the data rules must be consolidated into meaningful metrics by appropriate business categories. A metric is an equation that uses data rules, rule sets, or other metric results (that is, statistics) as numeric variables in the equation.

The following types of statistics are available for use as variables in metric creation:

- Data rule statistics
  - Number of records tested
  - Number of records that met the data rule conditions
  - Number of records that did not meet the data rule conditions
  - Percentage of records that met the data rule conditions
  - Percentage of records that did not meet the data rule conditions
  - Number of records in the variance from the data rule benchmark
  - Percentage of records in the variance from the data rule benchmark
- Rule set statistics
  - Number of records that met all rules
  - Number of records that failed one or more rules
  - Average number of rule failures per record
  - Standard deviation of the number of rule failures per record
  - Percentage of records that met all rules
  - Percentage of records that failed one or more rules
  - Average percentage of rule failures per record
  - Standard deviation of the percentage of rule failures per record
- Metric statistic, which includes a metric value

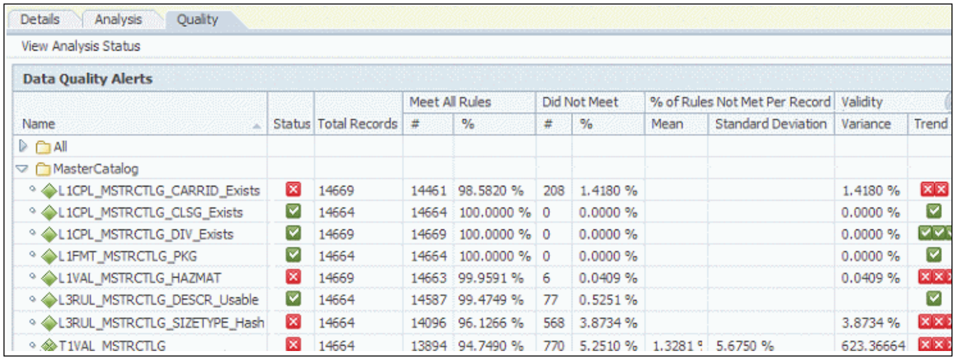
A key system feature in the creation of metrics is the capability for you to use weights, costs, and literals in the design of the metric equation. With this capability, you can develop metrics with the following qualities:

- Reflect the relative importance of various statistics (that is, applying weights).
- Reflect the business costs of data quality issues (such as applying costs).
- Use literals to produce universally used quality-control program measurements such as errors per million parts.

9.8 Monitoring data quality

Data rules, rule sets, and metrics are all executable objects that can be run as needed or on schedule. Each object generates a set of results, statistics, and detailed results recorded in the InfoSphere Analyzer Database, which is a special database workspace designated for storing analysis and data rules execution results. Because these objects run repeatedly, they create a time series of events that you can track, annotate, report, and trend over time.

Monitoring can be done in several different ways. The quality alert report in InfoSphere Information Analyzer presents results over time, highlighting up or down trends (Figure 9-37). Red icons with an X indicate down or worsening trends, and green icons with a check mark indicate improving trends.



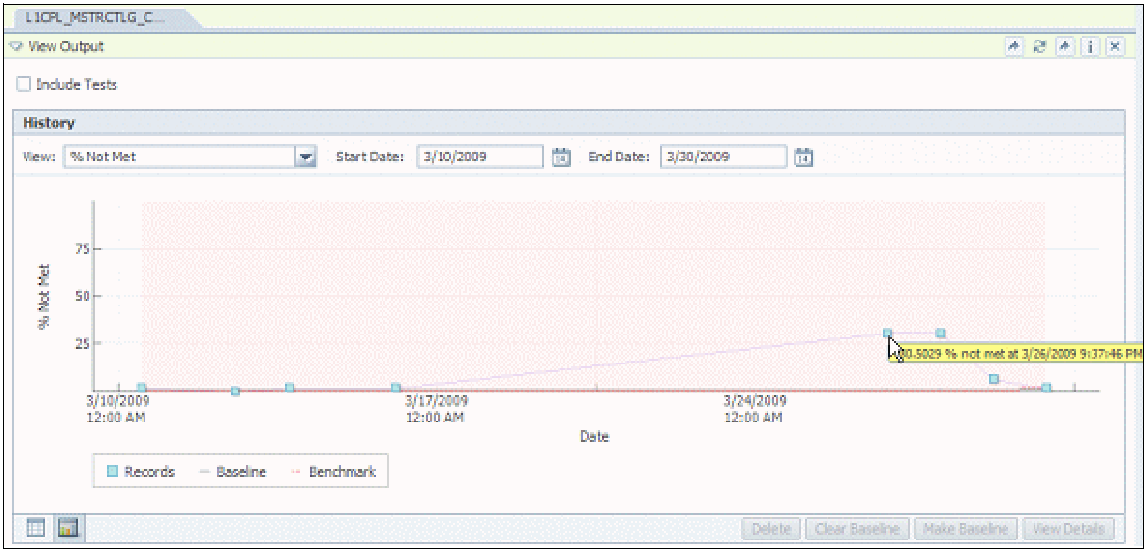
View Analysis Status

Data Quality Alerts

			Meet All Rules		Did Not Meet		% of Rules Not Met Per Record		Validity	
Name	Status	Total Records	#	%	#	%	Mean	Standard Deviation	Variance	Trend
All										
MasterCatalog										
L1CPL_MSTRCTLG_CARRID_Exists	☒	14669	14461	98.5820 %	208	1.4180 %			1.4180 %	☒☒
L1CPL_MSTRCTLG_CLSG_Exists	☑	14664	14664	100.0000 %	0	0.0000 %			0.0000 %	☑
L1CPL_MSTRCTLG_DIV_Exists	☑	14669	14669	100.0000 %	0	0.0000 %			0.0000 %	☑☑☑
L1FMT_MSTRCTLG_PKG	☑	14664	14664	100.0000 %	0	0.0000 %			0.0000 %	☑
L1VAL_MSTRCTLG_HAZMAT	☒	14669	14663	99.9591 %	6	0.0409 %			0.0409 %	☒☒☒
L3RUL_MSTRCTLG_DESCR_Usable	☑	14664	14587	99.4749 %	77	0.5251 %				☑
L3RUL_MSTRCTLG_SIZETYPE_Hash	☒	14664	14096	96.1266 %	568	3.8734 %			3.8734 %	☒☒☒
T1VAL_MSTRCTLG	☒	14664	13894	94.7490 %	770	5.2510 %	1.3281	5.6750 %	623.36664	☒☒☒

Figure 9-37 Data quality alert report

For individual rules, you can view a longer history of results in a tabular or graphical mode, which shows given statistics over time (Figure 9-38).



□ Include Tests

History

View: % Not Met

Start Date: 3/10/2009

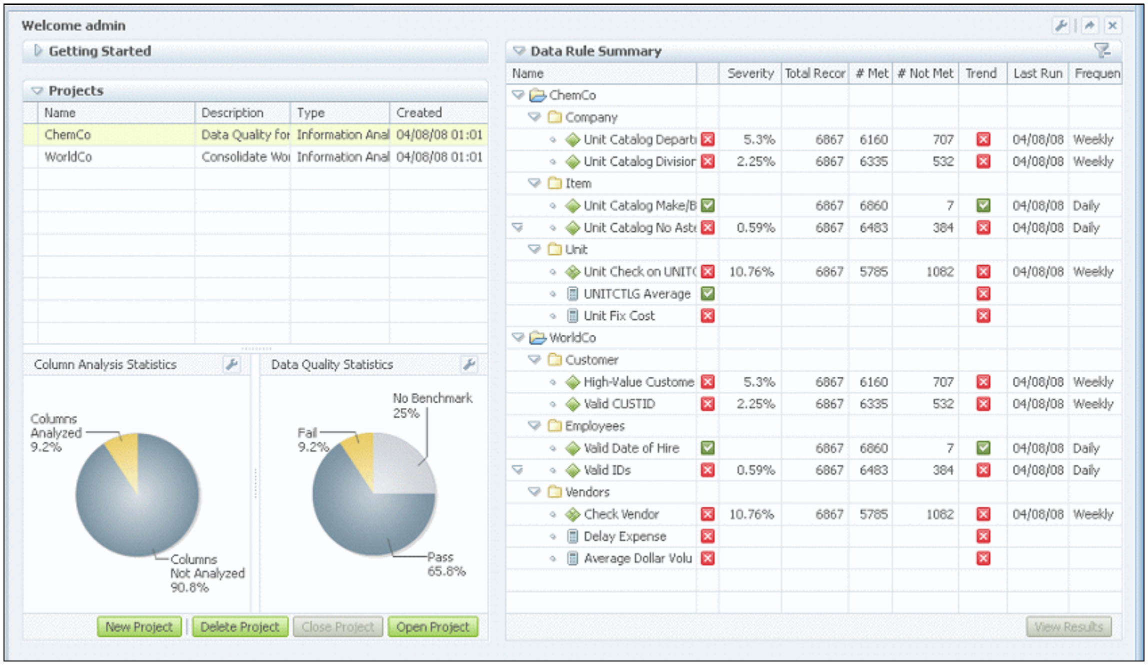
End Date: 3/30/2009

30.5029 % not met at 3/26/2009 9:37:46 PM

Figure 9-38 Time series of quality rule run results

We can see the individual runs and the percentage of errors produced over a period of time. Over longer periods of time, we might be able to identify cycles in the trend if errors spike during particular days in the month or particular months of the year.

Users with access to the InfoSphere Information Analyzer client can view a summary of all that is happening in the InfoSphere Information Analyzer dashboard. At a glance, a user can see which projects are out there, the status of analysis, and a summary of results of data quality rules runs as shown in Figure 9-39 on page 331. The dashboard can be customized to show any information generated by InfoSphere Information Analyzer.



Name	Description	Type	Created
ChemCo	Data Quality for	Information Anal	04/08/08 01:01
WorldCo	Consolidate Wo	Information Anal	04/08/08 01:01

Name		Severity	Total Recor	# Met	# Not Met	Trend	Last Run	Frequen
ChemCo								
Company								
Unit Catalog Depart	☒	5.3%	6867	6160	707	☒	04/08/08	Weekly
Unit Catalog Division	☒	2.25%	6867	6335	532	☒	04/08/08	Weekly
Item								
Unit Catalog Make/B	☑		6867	6860	7	☑	04/08/08	Daily
Unit Catalog No Ast	☒	0.59%	6867	6483	384	☒	04/08/08	Daily
Unit								
Unit Check on UNIT(	☒	10.76%	6867	5785	1082	☒	04/08/08	Weekly
UNITCTLG Average	☑					☒		
Unit Fix Cost	☒					☒		
WorldCo								
Customer								
High-Value Custome	☒	5.3%	6867	6160	707	☒	04/08/08	Weekly
Valid CUSTID	☒	2.25%	6867	6335	532	☒	04/08/08	Weekly
Employees								
Valid Date of Hire	☑		6867	6860	7	☑	04/08/08	Daily
Valid IDs	☒	0.59%	6867	6483	384	☒	04/08/08	Daily
Vendors								
Check Vendor	☒	10.76%	6867	5785	1082	☒	04/08/08	Weekly
Delay Expense	☒					☒		
Average Dollar Volu	☒					☒		

Figure 9-39 InfoSphere Information Analyzer dashboard

You can view all information from one dashboard. If you want to view full results of an analysis or data-rule runs, you can perform the following tasks:

- Export or extract results for additional analysis.
- Report results, and then deliver those results to others, on an as-needed or scheduled basis, and through various alternative formats.

You can develop a report by using standard report templates. A wide-range of report templates is available. Some are for the data quality analysis functions, and some are associated with data rules, rule sets, and metrics.

In addition, you can extract data from the InfoSphere Information Analyzer repository into external tools to generate reports or dynamic dashboards.

9.9 Using HTTP/CLI API

You can use a command-line interface (CLI) or HTTP to request the execution of various InfoSphere Information Analyzer commands or to extract data from the InfoSphere Information Analyzer results repository IADB (also known as the *database workspace*). This way, you can create custom reports and extract data to populate data marts that feed dashboards.

The following types of functions are covered by the HTTP/CLI API:

- Creation and modification of InfoSphere Information Analyzer projects
- Registration of sources in the project
- Creation of virtual columns and virtual tables
- Creation, modification, and deletion of rules and rule sets
- Creation and modification of global variables
- Run column analysis (base profile)
- Retrieval of column analysis results (base profile)
- Retrieval of frequency distributions
- Execution of rules and rulesets
- Retrieval of rules and rulesets execution history
- Retrieval of rules and rulesets output tables

With this functionality, you can integrate InfoSphere Information Analyzer in third-party environments without the use of the rich client. You can create custom reports by extracting and combining results of several requests in one report and use XSLT to format as needed.

9.10 Managing rules

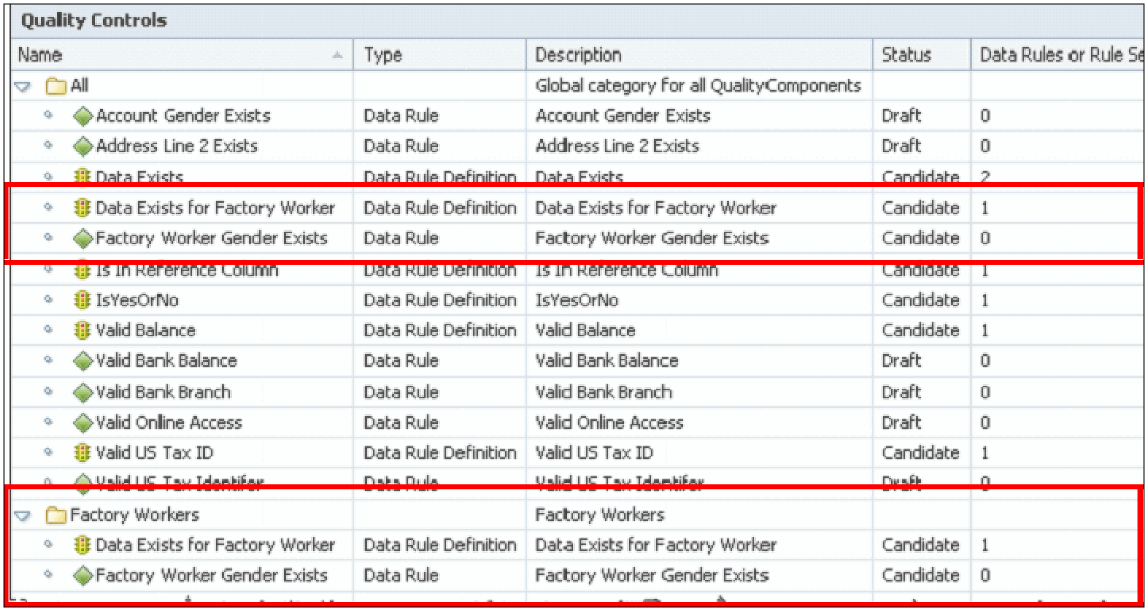
InfoSphere Information Analyzer operates within the confines of a project. Every analysis task or every rule definition testing and execution task is done within the context of an InfoSphere Information Analyzer project. The project consists of a set of data assets, tables, columns, files, and fields selected from the available metadata previously imported into the repository. It includes a group of authorized users with their designated roles within InfoSphere Information Analyzer. Roles for users are defined in the InfoSphere Information Server administrative console with specific roles for analysis and separate roles for data rules.

Any number of projects can exist simultaneously. They can have the same or different data sources and the same or different users and roles. An authorized user in a project can perform various tasks, running analysis tasks, data quality-related tasks, or both.

Rules created in one project are visible only to users who are authorized in that project. Rules can be published to make them visible and available for use in all other projects. Project staff can manage its rules within a multilevel structure of folders that it can create to suit its needs, distribution of subjects, responsibilities, or deployment schedules, as shown in Figure 9-40 on page 333.

A user with the role of rule manager can decide if a rule is ready to be promoted from *draft* and *candidate* to *approved* and *standard*. It is up to the manager of the organization, through policies and process, to decide the meaning of the status and how to use it. Promotion from *draft* to *candidate* might be that the rule has moved from the stage of formulation to evaluation where possibly more tests are required to determine that it serves the intended purpose.

Some rules can be basic and simple enough that they can be promoted directly to *approved* or *standard*, but others might require more testing and evaluation. After changing the status to *approved* or *standard*, the rule is locked from further editing. A rule cannot be changed intentionally or erroneously unless its status is changed back to *draft* or *candidate* by the rule manager. The status of rules is displayed in the Status column in the rule overview pane (Figure 9-40). The change to the rule status is applied in the rule edit pane.



Quality Controls

Name	Type	Description	Status	Data Rules or Rule Se
All		Global category for all QualityComponents		
Account Gender Exists	Data Rule	Account Gender Exists	Draft	0
Address Line 2 Exists	Data Rule	Address Line 2 Exists	Draft	0
Data Exists	Data Rule Definition	Data Exists	Candidate	2
Data Exists for Factory Worker	Data Rule Definition	Data Exists for Factory Worker	Candidate	1
Factory Worker Gender Exists	Data Rule	Factory Worker Gender Exists	Candidate	0
Is In Reference Column	Data Rule Definition	Is In Reference Column	Candidate	1
IsYesOrNo	Data Rule Definition	IsYesOrNo	Candidate	1
Valid Balance	Data Rule Definition	Valid Balance	Candidate	1
Valid Bank Balance	Data Rule	Valid Bank Balance	Draft	0
Valid Bank Branch	Data Rule	Valid Bank Branch	Draft	0
Valid Online Access	Data Rule	Valid Online Access	Draft	0
Valid US Tax ID	Data Rule Definition	Valid US Tax ID	Candidate	1
Valid US Tax Identifier	Data Rule	Valid US Tax Identifier	Draft	0
Factory Workers		Factory Workers		
Data Exists for Factory Worker	Data Rule Definition	Data Exists for Factory Worker	Candidate	1
Factory Worker Gender Exists	Data Rule	Factory Worker Gender Exists	Candidate	0

Figure 9-40 Quality Controls pane

When reviewing results, you can choose several monitoring approaches:

- You can work entirely within the user interface, reviewing and annotating information.
- You can choose to export results for additional analysis.
- You can choose to report results and deliver those results to others, on an as needed or scheduled basis, and through various alternative formats.

9.11 Deploying rules, rule sets, and metrics

Typically data rules, rule sets, and metrics in a monitoring environment are targeted to production data. The data is new data that comes through the pipes and is processed to populate various databases and data warehouses and to feed reports and decision making models. This environment requires more explicit control over tasks such as definition and change.

After a rule, rule sets, or metrics were defined, tested, and approved in the development environment, they must be moved and deployed in the production environment. This task is performed by the rule administrator, who exports the rules from the development environment and imports them into the production environment.

The rule administrator uses the Export and Import tasks on the Data quality pane to perform the function. The Export task is done in the context of a project. Choosing the Export task, the administrator selects the items to export, which include the following items:

- Project folders
- Binding of variables
- Global variables
- Output configuration
- Results history
- Folder support
- Metrics and benchmarks
- Key and cross-domain analysis
- Public rules

After they are selected, the administrator proceeds with the export process to produce an export file.

The import occurs in the production environment. By using the Import task in that environment, all elements in the export file are imported and placed in their proper places.

All import and export actions produce an audit event that can be traced by administrators.

As a preferred practice, the approach for rule definition, testing, and deployment must be clearly established. In such an environment, naming standards become important in moving from the initial design and testing to the deployed rules, rule sets, or metrics. Rules are no longer under the control of a single user, but people in other roles must now schedule, execute, and monitor them. The ability to correctly identify the rule, rule set, or metric is paramount.

9.12 Rule stage for InfoSphere DataStage

So far, you have seen how you can develop, deploy, and execute data rules against persisted data. You can use InfoSphere Information Analyzer data rules to monitor the quality of your data, checking for business conditions against a specified data source.

With InfoSphere Information Analyzer 8.7, data rules can become more pervasive in terms of the development environment and deployment options. Developers now can integrate data validation rules when they develop data integration and data cleansing jobs. They have access within the InfoSphere DataStage and InfoSphere QualityStage Designer to use existing data rules that are developed in InfoSphere Information Analyzer and can embed them into their jobs.

For example, Figure 9-41 shows a data rule in an InfoSphere DataStage job. You can see immediately how many rows have been analyzed and how many of the rows are valid and invalid. The developer can then choose to process the invalid records differently than the valid records for reporting or automated remediation.

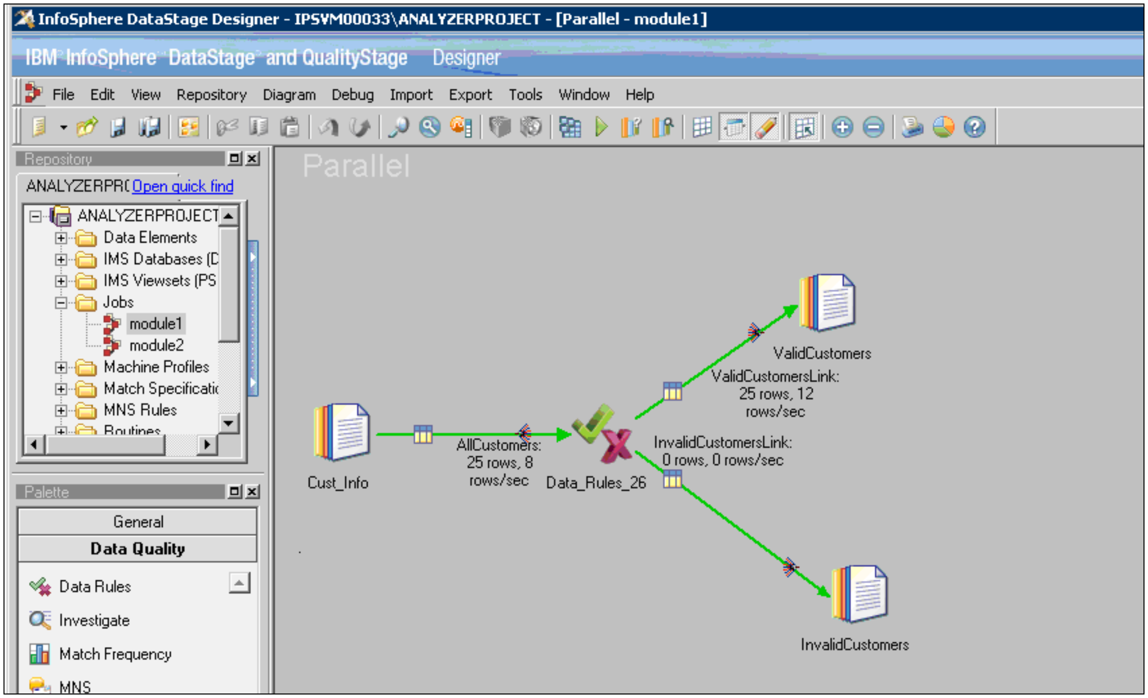
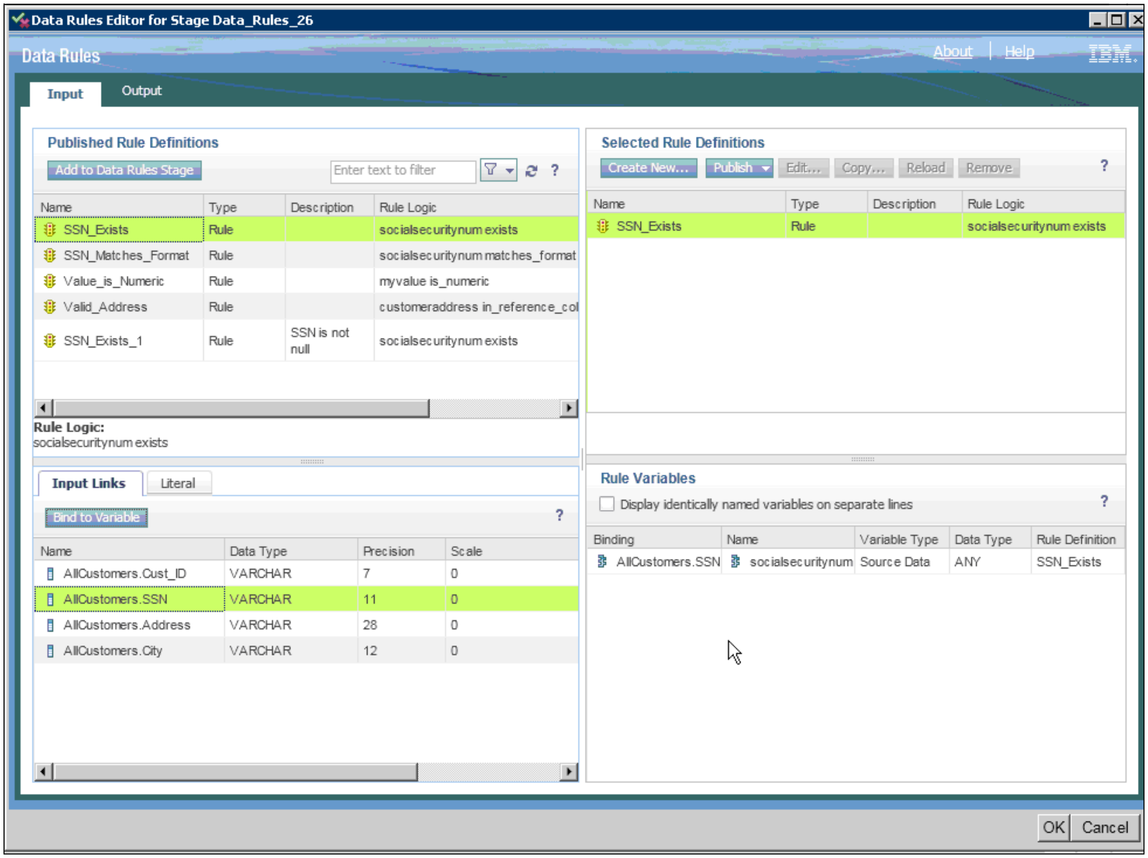


Figure 9-41 InfoSphere DataStage job with Rule Stage

Seamlessly integrated into the InfoSphere DataStage and InfoSphere QualityStage Designer is the user interface (Figure 9-42). With this interface, you can view the definition of the rule and modify it according to your needs. For example, you can determine which rule variables to map to which input links.



Name	Type	Description	Rule Logic
SSN_Exists	Rule		socialsecuritynum exists
SSN_Matches_Format	Rule		socialsecuritynum matches_format
Value_is_Numeric	Rule		myvalue is_numeric
Valid_Address	Rule		customeraddress in_reference_col
SSN_Exists_1	Rule	SSN is not null	socialsecuritynum exists

Rule Logic:
socialsecuritynum exists

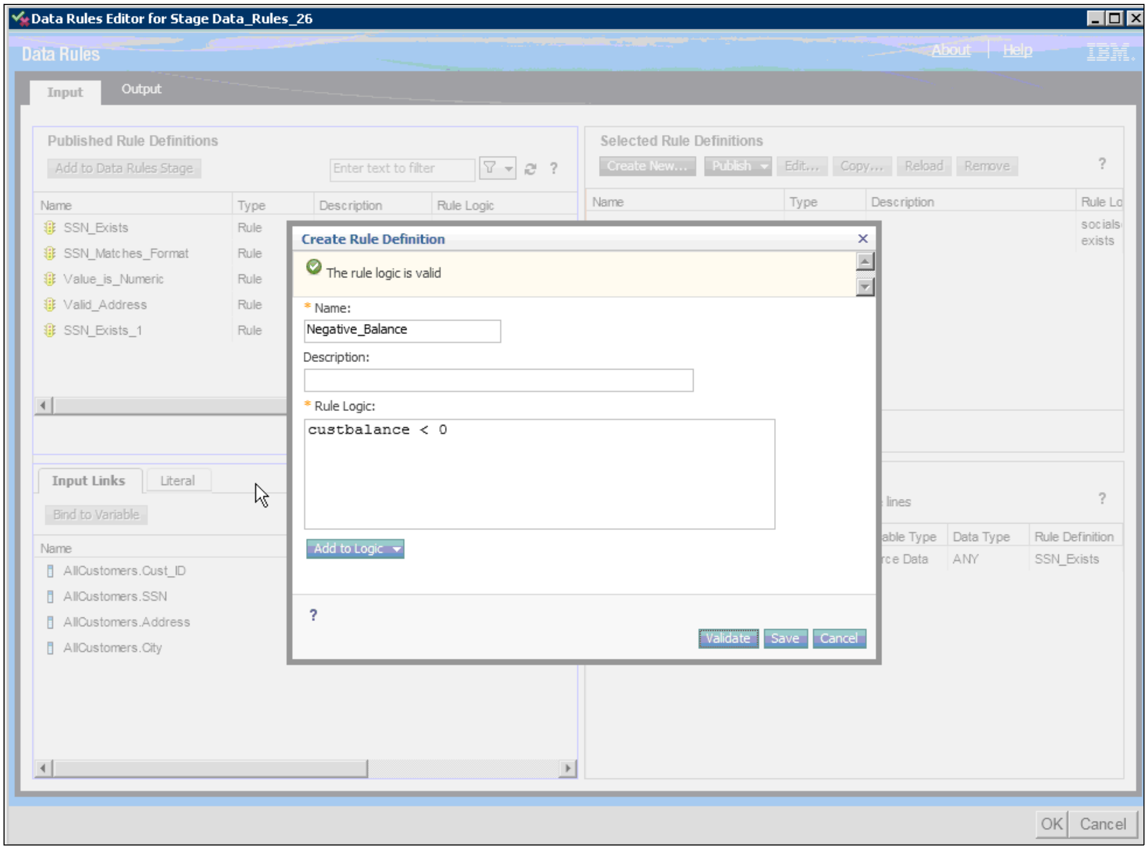
Name	Data Type	Precision	Scale
AllCustomers.Cust_ID	VARCHAR	7	0
AllCustomers.SSN	VARCHAR	11	0
AllCustomers.Address	VARCHAR	28	0
AllCustomers.City	VARCHAR	12	0

Name	Type	Description	Rule Logic
SSN_Exists	Rule		socialsecuritynum exists

Binding	Name	Variable Type	Data Type	Rule Definition
AllCustomers.SSN	socialsecuritynum	Source Data	ANY	SSN_Exists

Figure 9-42 Data Rule editor for Rule Stage

You can also define a new rule from scratch within the InfoSphere DataStage and InfoSphere QualityStage Designer as shown in Figure 9-43.



Name	Type	Description	Rule Logic
SSN_Exists	Rule		
SSN_Matches_Format	Rule		
Value_is_Numeric	Rule		
Valid_Address	Rule		
SSN_Exists_1	Rule		

Name
AllCustomers.Cust_ID
AllCustomers.SSN
AllCustomers.Address
AllCustomers.City

The rule logic is valid

* Name: Negative_Balance

Description:

* Rule Logic:

```
custbalance < 0
```

Figure 9-43 Creating a data rule definition

The same rule definition can now be applied to data stored in a repository or to data being processed in an InfoSphere DataStage or InfoSphere QualityStage job or as a service. You benefit from higher flexibility on how and where to deploy data quality monitoring. Publishing a data rule to the shared repository from InfoSphere DataStage, or from InfoSphere Information Analyzer, makes the rule available for usage across jobs or modules. For example, you might execute the rule first in the data warehouse. If bad data causes disruptions of the business, the same rule can be embedded in the job that loads the warehouse and preempt the bad data that enters the data warehouse.

9.13 Conclusion

In conclusion, this chapter highlighted the different aspects InfoSphere Information Analyzer data rules and their uses to assess and monitor data quality.

Chapter 10, “Building up the metadata repository” on page 339, addresses loading metadata of various asset types into the metadata repository so that you have a single location of metadata. This chapter also introduces InfoSphere Metadata Workbench so that you can view and explore data, monitor data usage, generate lineage reports, and do much more.

10

Building up the metadata repository

The InfoSphere Information Server metadata repository provides stakeholders a single portal of information to monitor the use of information, its quality, and its adherence to the business requirements. The metadata repository enables the sharing wand exchange of information and the documenting and loading of external data assets or processes.

The metadata repository can be considered a warehouse for metadata because it includes metadata that is native to InfoSphere Information Server. It also includes other assets that can be imported from across the enterprise.

Building up the metadata repository is the process of accumulating and storing metadata in this centrally shared metadata repository. It is the process of storing data and business assets so that they can be used for better understanding of the data. It facilitates auditing of data usage and manipulation of the data. It helps to ensure correct usage and validate data reliability.

This chapter includes the following sections:

- Introduction to InfoSphere Metadata Workbench
- Data storage systems
- Data models
- Business intelligence reports

- Information asset enrichment
- Conclusion

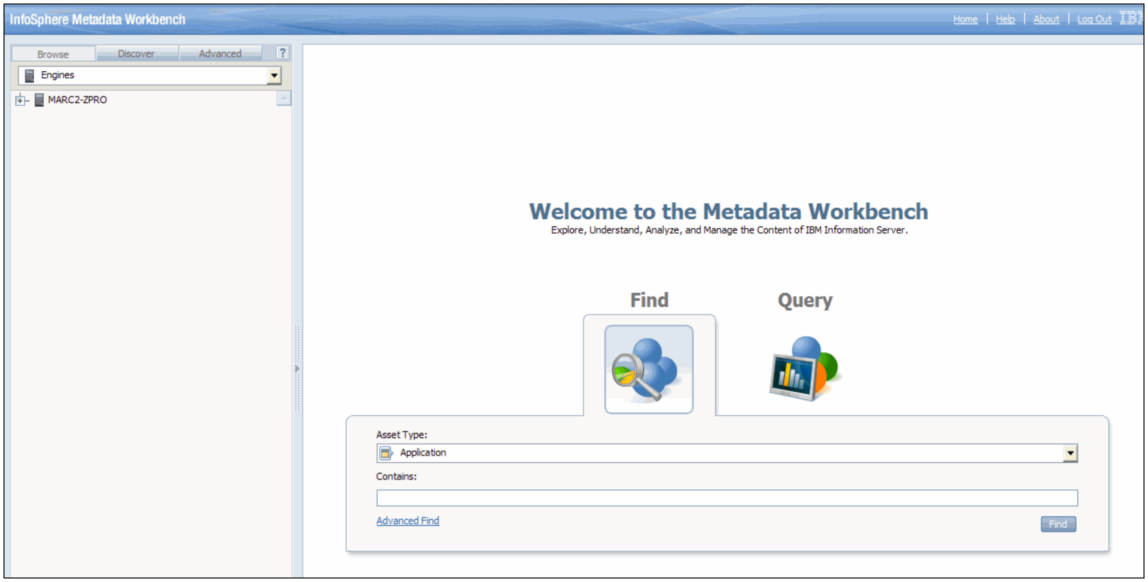
10.1 Introduction to InfoSphere Metadata Workbench

IBM InfoSphere Metadata Workbench is used to enrich the contents of the metadata repository, specifically, the association with business terms, business tables, data stewards, and authoring descriptors. InfoSphere Metadata Workbench gives stakeholders a unified view of information. This way, they can better understand the implied meaning, specification, and published quality assessment. They can also analyze the data flow.

InfoSphere Metadata Workbench is installed as part of the InfoSphere Information Server, and it can be accessed through a web browser. You use the following default URL to access it:

`http://`*`ServerName`*`:9080/workbench`

Figure 10-1 shows the administrative user interface for InfoSphere Metadata Workbench.



Welcome to the Metadata Workbench

Explore, Understand, Analyze, and Manage the Content of IBM Information Server.

Asset Type: Application

Contains:

Figure 10-1 InfoSphere Metadata Workbench

InfoSphere Metadata Workbench provides an administrative user interface to manage and deliver data lineage and to create and apply custom properties. This interface also helps to manage and document external processes and systems in the form of extended data sources and extension mapping documents.

10.2 Data storage systems

Loading data storage systems as a data warehouse or data marts into the metadata repository supports the data lineage requirements of the business. In addition, by loading these systems, you can take advantage of the application of business terms and the enhancement of the descriptors for data assets. Furthermore, the metadata repository sustains and supports more efficient development processes by allowing developers full information sharing.

Figure 10-2 shows the data flow of an information integration system. For more information and instructions for loading the database assets, see 7.5, “Staging database” on page 192.

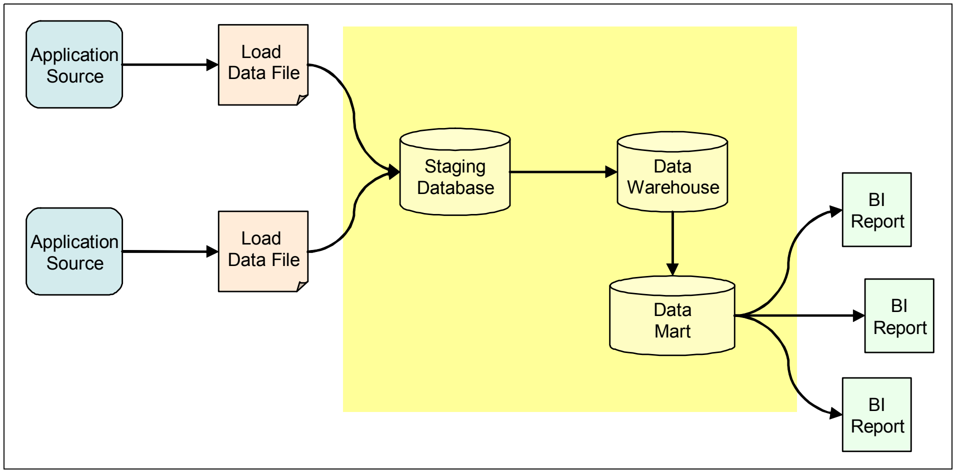


Figure 10-2 Loading staging data in an information integration solution

10.3 Data models

Data models help to describe how to structure, consolidate, and represent data effectively. They also help you to implement a database system to support the data storage or development efforts. Representing data models in the InfoSphere Information Server benefits the stakeholders in that they can trace the database system back to the original design and understand the implied intent and specification.

You can load data models into the InfoSphere Information Server by using InfoSphere Metadata Asset Manager. Users of the Information Server benefit by including such metadata in the Repository. For example, they can search and browse such information from a central asset catalog, view the relationships between models and implemented databases, and use such information for development. Users can also assign terms, labels, stewards to such assets.

In the bank acquisition scenario in this book, we import an InfoSphere Data Architect data model that has been saved in a logical data model (LDM) file format. We developed this LDM file format data model by using the IBM InfoSphere Glossary Pack for Financial Services. We also generated the supporting data warehouse and reporting mart database storage systems.

10.3.1 Loading the data models

To load the data models, complete these steps:

1. Log on to InfoSphere Metadata Asset Manager.
2. Click **New Import Area**.

3. In the New Import Area panel (Figure 10-3), define a new area for the import of a data model. You can then re-import and manage the imported metadata.

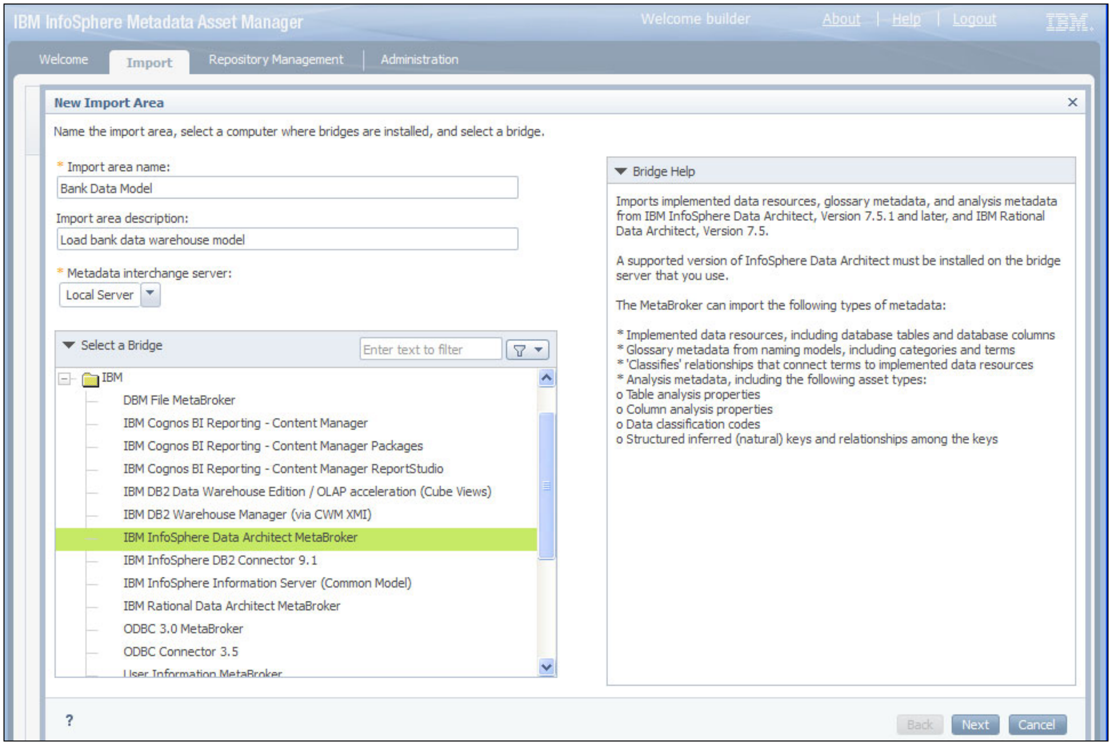
   a. Enter a name for the import area to uniquely identify the import process for future re-import or administration.

   b. Optional: Enter a description for the import area, describing the data model to be imported or the process.

   c. Select a previously defined metadata interchange server. The metadata interchange server defines the connectivity to the InfoSphere Information Server where the metadata of the data model will be imported.

   d. Browse to select a bridge to provide the connection parameters to the load information from the source system. In our example, we select the **IBM InfoSphere Data Architect Metabroker** bridge.

   e. Click **Next**.



Name the import area, select a computer where bridges are installed, and select a bridge.

* Import area name: Bank Data Model

Import area description: Load bank data warehouse model

* Metadata interchange server: Local Server

Select a Bridge

- IBM
  - DBM File MetaBroker
  - IBM Cognos BI Reporting - Content Manager
  - IBM Cognos BI Reporting - Content Manager Packages
  - IBM Cognos BI Reporting - Content Manager ReportStudio
  - IBM DB2 Data Warehouse Edition / OLAP acceleration (Cube Views)
  - IBM DB2 Warehouse Manager (via CWM XMI)
  - IBM InfoSphere Data Architect MetaBroker
  - IBM InfoSphere DB2 Connector 9.1
  - IBM InfoSphere Information Server (Common Model)
  - IBM Rational Data Architect MetaBroker
  - ODBC 3.0 MetaBroker
  - ODBC Connector 3.5
  - User Information MetaBroker

Bridge Help

Imports implemented data resources, glossary metadata, and analysis metadata from IBM InfoSphere Data Architect, Version 7.5.1 and later, and IBM Rational Data Architect, Version 7.5.

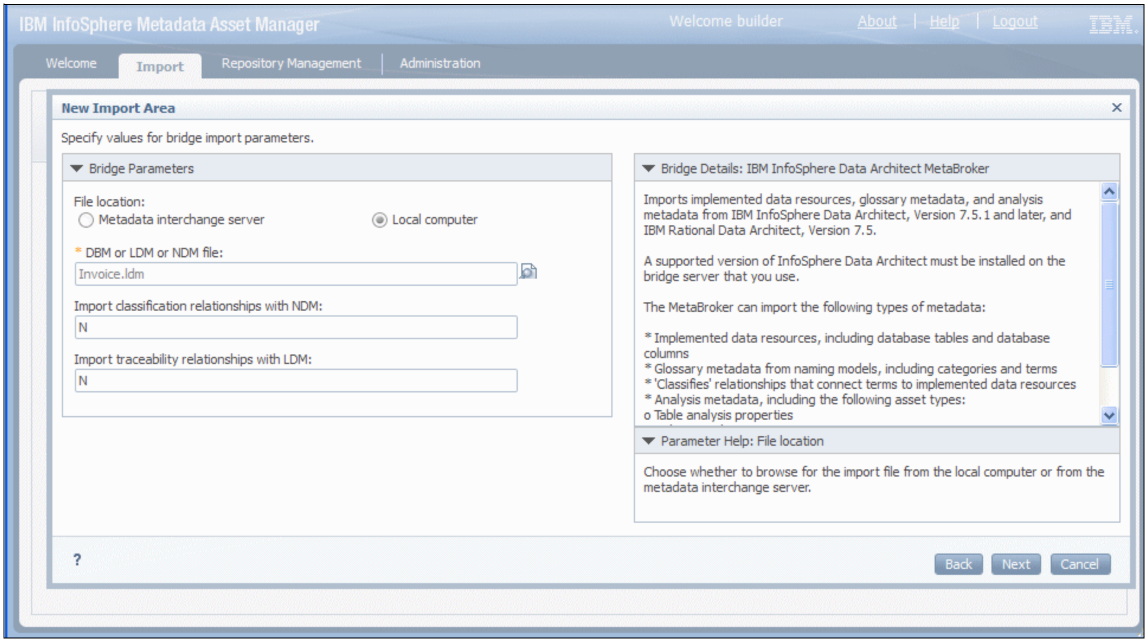
A supported version of InfoSphere Data Architect must be installed on the bridge server that you use.

The MetaBroker can import the following types of metadata:

* Implemented data resources, including database tables and database columns
* Glossary metadata from naming models, including categories and terms
* 'Classifies' relationships that connect terms to implemented data resources
* Analysis metadata, including the following asset types:
o Table analysis properties
o Column analysis properties
o Data classification codes
o Structured inferred (natural) keys and relationships among the keys

Figure 10-3 New import area in the InfoSphere Metadata Asset Manager

4. In the Bridge Parameters group box, specify the values for the bridge import parameters (Figure 10-4), which identify the data model to be loaded.
   a. Browse to select an InfoSphere Data Architect data model. Select an LDM file that represents a logical model or a database management (DBM) file that represents a physical model. You can find the file on InfoSphere Information Server or on local file systems.
   b. Select **Import traceability relationships with LDM**, which captures the relationships between an LDM file and a DBM file.
   c. Click **Next**.



Specify values for bridge import parameters.

Bridge Parameters

File location:
○ Metadata interchange server ● Local computer

* DBM or LDM or NDM file: Invoice.ldm

Import classification relationships with NDM: N

Import traceability relationships with LDM: N

Bridge Details: IBM InfoSphere Data Architect MetaBroker

Imports implemented data resources, glossary metadata, and analysis metadata from IBM InfoSphere Data Architect, Version 7.5.1 and later, and IBM Rational Data Architect, Version 7.5.

A supported version of InfoSphere Data Architect must be installed on the bridge server that you use.

The MetaBroker can import the following types of metadata:

* Implemented data resources, including database tables and database columns
* Glossary metadata from naming models, including categories and terms
* 'Classifies' relationships that connect terms to implemented data resources
* Analysis metadata, including the following asset types:
o Table analysis properties

Parameter Help: File location

Choose whether to browse for the import file from the local computer or from the metadata interchange server.

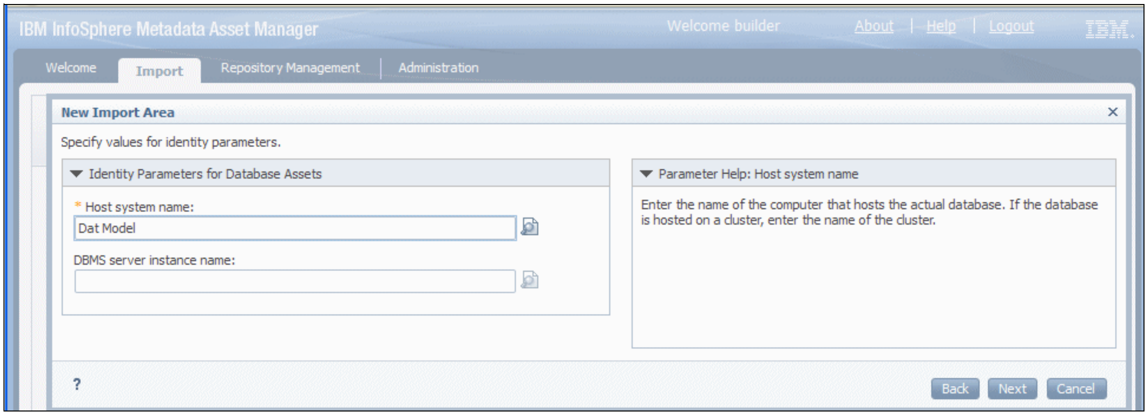
Figure 10-4 Bridge parameters in the import area

5. In the Identity Parameters for Database Assets group box, specify the values for the identity parameters (Figure 10-5). The identity parameters include the host system, which helps a user identify and classify information in the InfoSphere Information Server.

   a. Browse to select an existing host system, or enter the name of the host system. The host system must reflect the server on which the implemented database instance has been installed.

   b. Optional: Enter a name for the database instance. The instance name must reflect the database or node where the physical model has been deployed.

   c. Click **Next**.



Specify values for identity parameters.

* Host system name: Dat Model

DBMS server instance name:

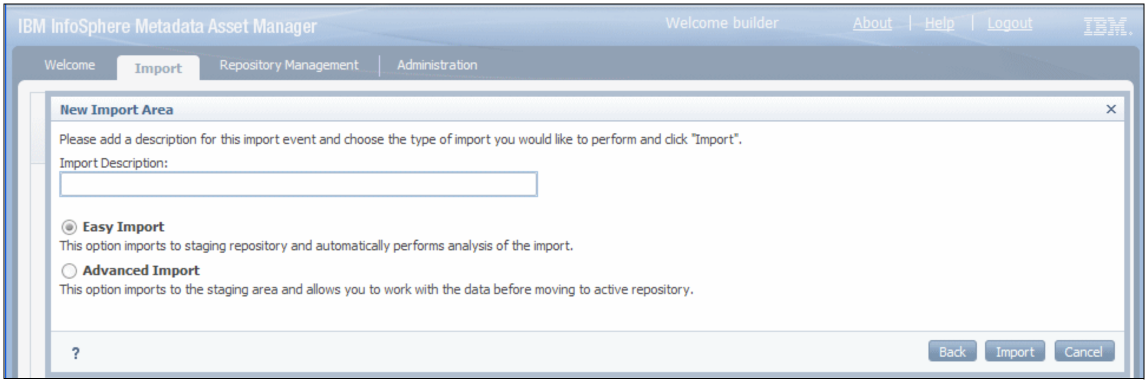
Enter the name of the computer that hosts the actual database. If the database is hosted on a cluster, enter the name of the cluster.

Figure 10-5 Identity parameters in the import area

6. Complete the import event so that you can preview the data model asset before you publish it to the InfoSphere Information Server (Figure 10-6):

   a. Select **Easy Import** to publish the model automatically to the metadata repository. Alternatively, select **Advanced Import** to review the model before its publication to the metadata server.

   b. Click **Import** to complete the process and publish the model to the metadata repository.



New Import Area

Please add a description for this import event and choose the type of import you would like to perform and click "Import".

Import Description:

● **Easy Import**
This option imports to staging repository and automatically performs analysis of the import.

○ **Advanced Import**
This option imports to the staging area and allows you to work with the data before moving to active repository.

Figure 10-6 Complete the import process of the data model

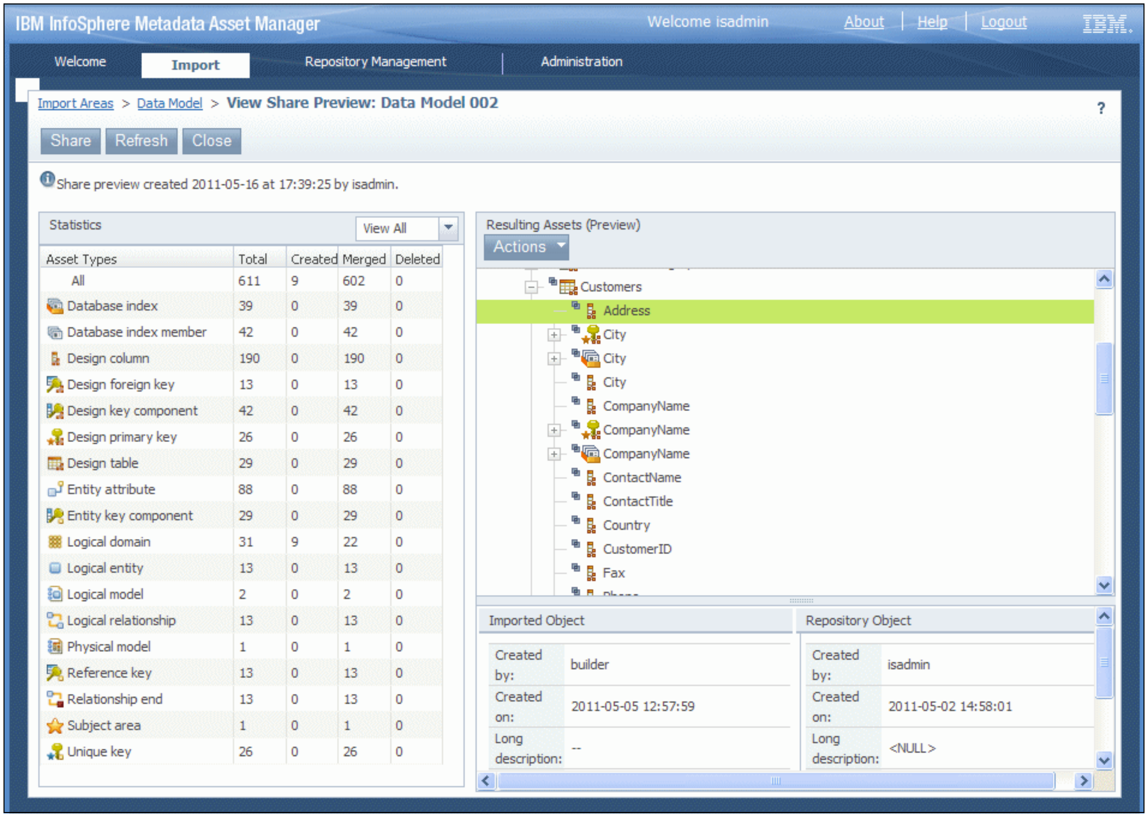
7. Publish the data model asset to the InfoSphere Information Server so that you can preview the asset to be imported. You can also compare it to existing assets that will be merged or otherwise updated.

   a. Select and open the import area that contains the previously imported data model.

   b. Click the **Staged Import** tab.

   c. Click **Preview** to analyze the data model assets to be shared and their effects on the existing assets in the InfoSphere Information Server (Figure 10-7 on page 347). You can preview the type and number of assets to be created, deleted, or merged upon sharing. Browse to select a particular data model asset object, so that you can view the pending changes after it is shared.

   d. Click **Share** to publish the database asset to the InfoSphere Information Server, making the loaded database asset available to all components.



Share preview created 2011-05-16 at 17:39:25 by isadmin.

Asset Types	Total	Created	Merged	Deleted
All	611	9	602	0
Database index	39	0	39	0
Database index member	42	0	42	0
Design column	190	0	190	0
Design foreign key	13	0	13	0
Design key component	42	0	42	0
Design primary key	26	0	26	0
Design table	29	0	29	0
Entity attribute	88	0	88	0
Entity key component	29	0	29	0
Logical domain	31	9	22	0
Logical entity	13	0	13	0
Logical model	2	0	2	0
Logical relationship	13	0	13	0
Physical model	1	0	1	0
Reference key	13	0	13	0
Relationship end	13	0	13	0
Subject area	1	0	1	0
Unique key	26	0	26	0

	Imported Object	Repository Object
Created by:	builder	isadmin
Created on:	2011-05-05 12:57:59	2011-05-02 14:58:01
Long description:	--	<NULL>

Figure 10-7 Sharing data model assets with InfoSphere Information Server

10.3.2 Results

Data models represent the design of the database systems that are defined in the information integration project. Stakeholders want to understand the relationships between the physical database systems and their logical counterparts.

The following information assets are available in the InfoSphere Information Server after loading a data model:

Logical model ( ) The organizational structure or domain concepts of information, independent of any data management system.

Logical entity ( ) The entity design structures of the model.

Logical attribute ( ) The attribute or column definitions in the defined entity structures of the model.

Physical model () The data design and data relationships.

Design table () The design of the database table, its indexes, its keys, and its relationships.

Design column () The design of the database column, its type, its length, and its usage.

You can search, browse, and view the details of the LDM assets, the implemented physical models, and database systems from the components of InfoSphere Information Server, including InfoSphere Metadata Asset Manager.

10.4 Business intelligence reports

Business intelligence (BI) reporting refers to the process of publishing, distributing, and reviewing data results and information. Critical to your success is the ability to understand the meaning and authenticity of these reports. The reports are routinely generated against data sources, such as data marts or data warehouses.

Consider the following questions:

- How can we better understand the current quality and status of our reports?
- When was the content feeding the report last updated?
- Which InfoSphere DataStage and InfoSphere QualityStage jobs or other extract, transform, and load data integration processes were sequenced during an update?

BI reports are report templates that are created in the reporting tool. The BI reports include report fields, several of which are non-data fields, such as page numbers and section headers. Other report fields are data fields that retrieve or calculate data from a data source.

BI reports contain source information from the underlying BI model that aggregates and reads information from a data source. In the bank example, we import a Cognos BI report and framework manager model, as shown in Figure 10-8.

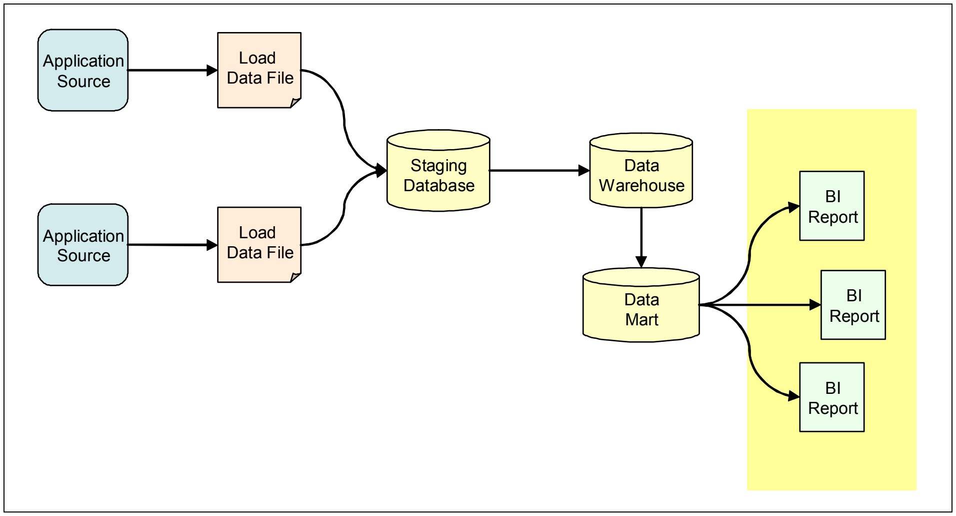


Figure 10-8 BI reports in the data flow of an information integration project

10.4.1 Loading BI reports

To load BI reports, complete these steps:

1. Log on to InfoSphere Metadata Asset Manager.
2. Click **New Import Area**.

3. In the New Import Area panel (Figure 10-9), define a new area for the import of a BI report. You can then re-import and manage the imported metadata.

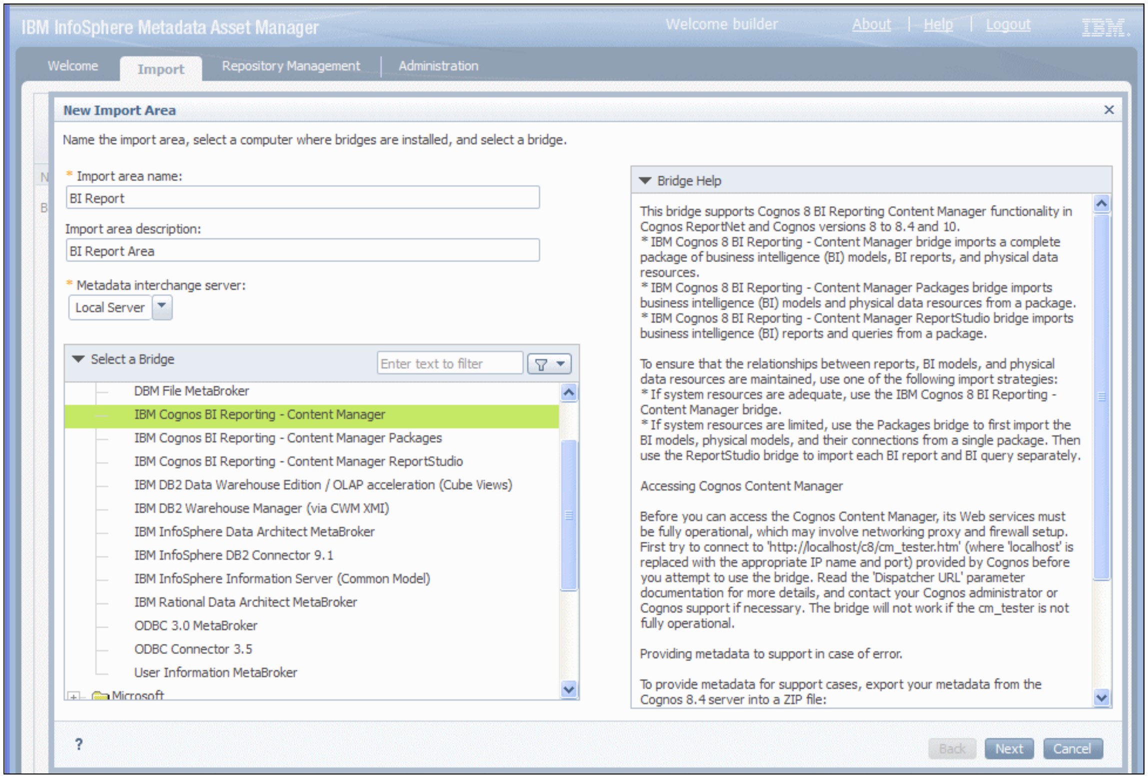
   a. Enter a name for the import area to uniquely identify the import process for future re-import or administration.

   b. Optional: Enter a description for the import area to identify the report to be imported or the process.

   c. Select a previously defined metadata interchange server. The metadata interchange server defines the connectivity to the InfoSphere Information Server where the metadata of the report will be imported.

   d. Browse to select a bridge, which provides the connection parameters to the load information from the source system. In this example, we select the **IBM Cognos BI Reporting - Content Manager** bridge. This bridge connects to the Cognos BI server to load the report metadata.

   e. Click **Next**.



Name the import area, select a computer where bridges are installed, and select a bridge.

* Import area name: BI Report

Import area description: BI Report Area

* Metadata interchange server: Local Server

- DBM File MetaBroker
- IBM Cognos BI Reporting - Content Manager
- IBM Cognos BI Reporting - Content Manager Packages
- IBM Cognos BI Reporting - Content Manager ReportStudio
- IBM DB2 Data Warehouse Edition / OLAP acceleration (Cube Views)
- IBM DB2 Warehouse Manager (via CWM XMI)
- IBM InfoSphere Data Architect MetaBroker
- IBM InfoSphere DB2 Connector 9.1
- IBM InfoSphere Information Server (Common Model)
- IBM Rational Data Architect MetaBroker
- ODBC 3.0 MetaBroker
- ODBC Connector 3.5
- User Information MetaBroker
- Microsoft

This bridge supports Cognos 8 BI Reporting Content Manager functionality in Cognos ReportNet and Cognos versions 8 to 8.4 and 10.
* IBM Cognos 8 BI Reporting - Content Manager bridge imports a complete package of business intelligence (BI) models, BI reports, and physical data resources.
* IBM Cognos 8 BI Reporting - Content Manager Packages bridge imports business intelligence (BI) models and physical data resources from a package.
* IBM Cognos 8 BI Reporting - Content Manager ReportStudio bridge imports business intelligence (BI) reports and queries from a package.

To ensure that the relationships between reports, BI models, and physical data resources are maintained, use one of the following import strategies:
* If system resources are adequate, use the IBM Cognos 8 BI Reporting - Content Manager bridge.
* If system resources are limited, use the Packages bridge to first import the BI models, physical models, and their connections from a single package. Then use the ReportStudio bridge to import each BI report and BI query separately.

Accessing Cognos Content Manager

Before you can access the Cognos Content Manager, its Web services must be fully operational, which may involve networking proxy and firewall setup. First try to connect to 'http://localhost/c8/cm_tester.htm' (where 'localhost' is replaced with the appropriate IP name and port) provided by Cognos before you attempt to use the bridge. Read the 'Dispatcher URL' parameter documentation for more details, and contact your Cognos administrator or Cognos support if necessary. The bridge will not work if the cm_tester is not fully operational.

Providing metadata to support in case of error.

To provide metadata for support cases, export your metadata from the Cognos 8.4 server into a ZIP file:

Figure 10-9 Loading the Cognos BI report

4. Specify the values for the bridge import parameters (Figure 10-10 on page 352). The parameters identify the report to be loaded.

   a. Browse to select the version of the Cognos BI server that contains the report to be imported.

   b. Enter the URL for the Cognos BI server dispatcher, such as the following typical URL:

      ```
      http://Server:9300/p2pd/server/dispatch
      ```

      The dispatcher acts as a gateway to send requests to the Cognos BI server repository. Ensure that the machine can access the Cognos BI server dispatcher.

   c. If Cognos BI authentication has been enabled, complete these steps:

      i. Enter the Cognos BI namespace.
      ii. Enter a user name.
      iii. Enter a user password.

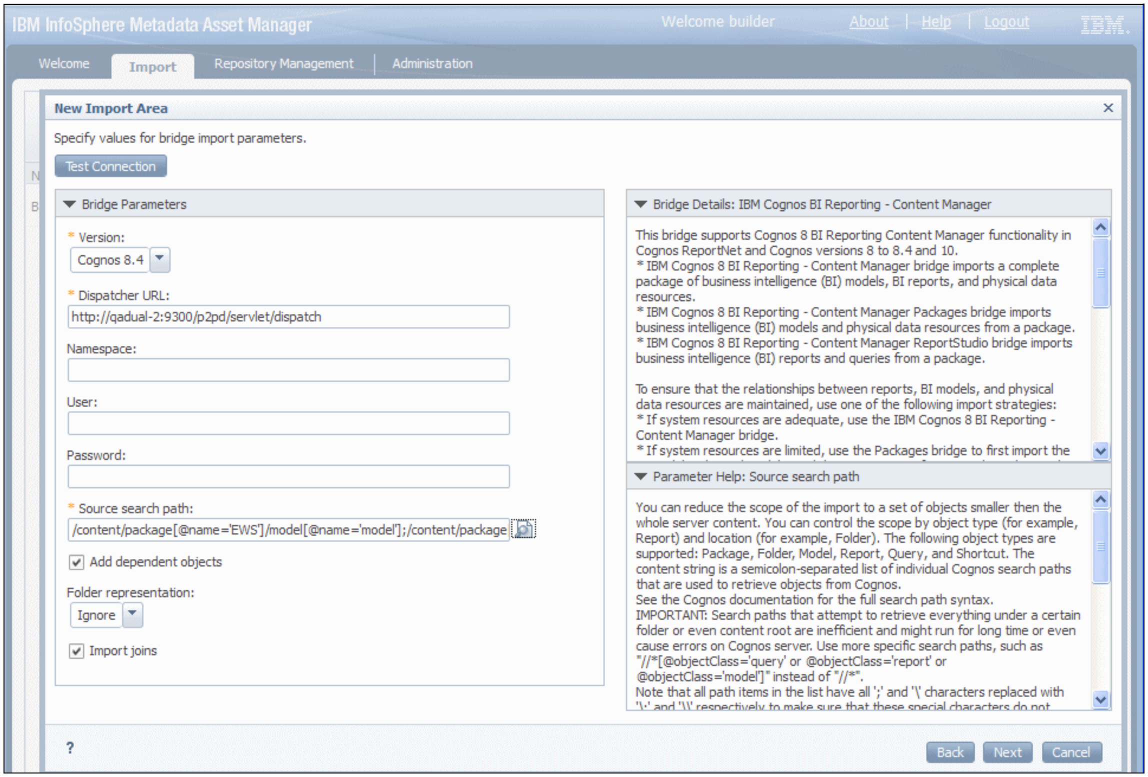
   d. For Source search path, select the report objects to import. Click the **Search** icon (magnifying glass) to browse to select the published packages or folders of the Cognos BI server that contain the reports and framework manager model. Select a single package for import. The import process loads all associated reports and models from the package.

   e. Click **Add Dependent Objects** to capture the referenced report objects for import.

   f. For Folder representation, select **Ignore**, which automatically captures the Cognos BI server and the package or folder location of the report.

   g. Click **Import joins** to capture the Cognos BI framework model entity relationships.

   h. Click **Next**.



New Import Area

Specify values for bridge import parameters.

Bridge Parameters

* Version: Cognos 8.4

* Dispatcher URL: http://qadual-2:9300/p2pd/servlet/dispatch

Namespace:

User:

Password:

* Source search path: /content/package[@name='EWS']/model[@name='model'];/content/package

☑ Add dependent objects

Folder representation: Ignore

☑ Import joins

Bridge Details: IBM Cognos BI Reporting - Content Manager

This bridge supports Cognos 8 BI Reporting Content Manager functionality in Cognos ReportNet and Cognos versions 8 to 8.4 and 10.
* IBM Cognos 8 BI Reporting - Content Manager bridge imports a complete package of business intelligence (BI) models, BI reports, and physical data resources.
* IBM Cognos 8 BI Reporting - Content Manager Packages bridge imports business intelligence (BI) models and physical data resources from a package.
* IBM Cognos 8 BI Reporting - Content Manager ReportStudio bridge imports business intelligence (BI) reports and queries from a package.

To ensure that the relationships between reports, BI models, and physical data resources are maintained, use one of the following import strategies:
* If system resources are adequate, use the IBM Cognos 8 BI Reporting - Content Manager bridge.
* If system resources are limited, use the Packages bridge to first import the

Parameter Help: Source search path

You can reduce the scope of the import to a set of objects smaller then the whole server content. You can control the scope by object type (for example, Report) and location (for example, Folder). The following object types are supported: Package, Folder, Model, Report, Query, and Shortcut. The content string is a semicolon-separated list of individual Cognos search paths that are used to retrieve objects from Cognos.
See the Cognos documentation for the full search path syntax.
IMPORTANT: Search paths that attempt to retrieve everything under a certain folder or even content root are inefficient and might run for long time or even cause errors on Cognos server. Use more specific search paths, such as "//*[@objectClass='query' or @objectClass='report' or @objectClass='model']" instead of "//*".
Note that all path items in the list have all ';' and '\' characters replaced with '\;' and '\\' respectively to make sure that these special characters do not

Figure 10-10 Specifying the Cognos BI report bridge parameters in the import area

5. Specify the values for the identity parameters (Figure 10-11 on page 353). The identity parameters include the host system, which helps a user identify and classify information in the InfoSphere Information Server.

   a. Browse to select an existing host system, or enter the name of the host system. The host system must reflect the server, on which the database system that is accessed by the BI model has been deployed.

   b. Browse to select an existing database system, or enter the name of the database system. The database system name must reflect the name of the database system from where Cognos BI Framework Manager is reading information.

   c. Optional: Select **Use database name to override bridge value**. Alternatively, select **Use database name if bridge value is empty** to determine how the database name is applied to the imported information assets.

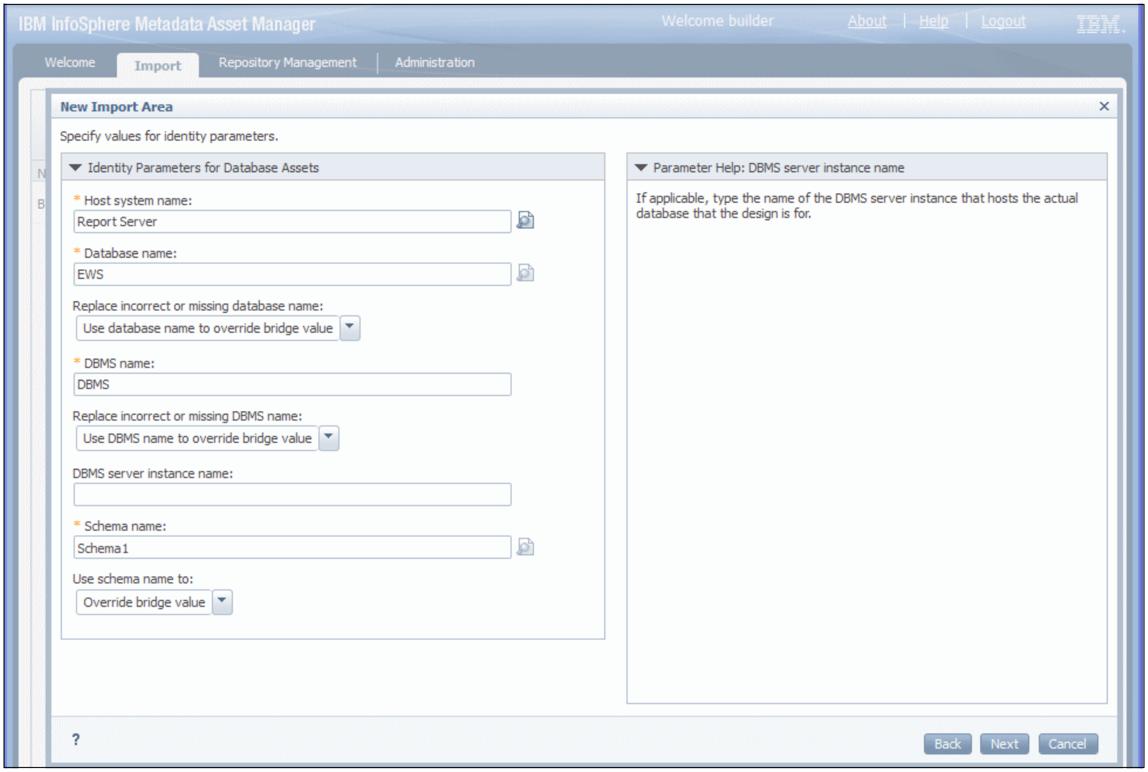
   d. Enter the name of the database management system (DBMS) that houses the database system, such as DB2.

e. Optional: Select **Use DBMS name to override bridge value**. Alternatively, select **Use DBMS name if bridge value is empty** to determine how the DBMS name is applied to the imported information assets.

f. Enter the name of the database schema containing the database tables that are referenced by the Cognos BI Framework Manager model.

g. Optional: For Use schema name, select **Override bridge value** Alternatively, select **Bridge value is empty** to determine how the schema name is applied to the imported information assets.

h. Click **Next**.



New Import Area

Specify values for identity parameters.

Identity Parameters for Database Assets

* Host system name: Report Server

* Database name: EWS

Replace incorrect or missing database name: Use database name to override bridge value

* DBMS name: DBMS

Replace incorrect or missing DBMS name: Use DBMS name to override bridge value

DBMS server instance name:

* Schema name: Schema1

Use schema name to: Override bridge value

Parameter Help: DBMS server instance name

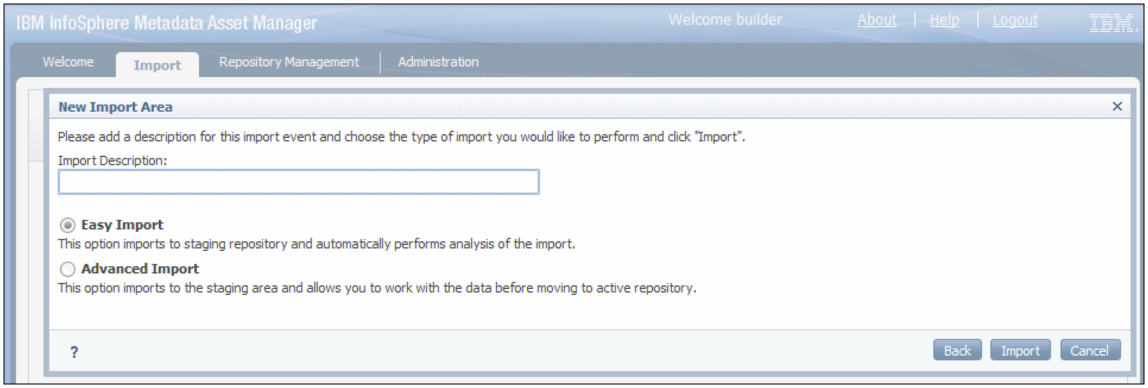
If applicable, type the name of the DBMS server instance that hosts the actual database that the design is for.

Figure 10-11 Define identity parameters in the import area

6. Complete the import event so that you can preview the report asset before you publish it to the InfoSphere Information Server (Figure 10-12):

   a. Select **Easy Import** to automatically publish the report to the metadata repository. Alternatively, select **Advanced Import** to review the report before it is published to the metadata server.

   b. Click **Import** to complete the process and publish the report to the metadata repository.



New Import Area

Please add a description for this import event and choose the type of import you would like to perform and click "Import".

Import Description:

● Easy Import
This option imports to staging repository and automatically performs analysis of the import.

○ Advanced Import
This option imports to the staging area and allows you to work with the data before moving to active repository.

Figure 10-12 Completing the import process of the data model

7. Publish the report metadata to the InfoSphere Information Server so that you can preview the asset to be imported and compare it to the existing assets that will be merged or otherwise updated:

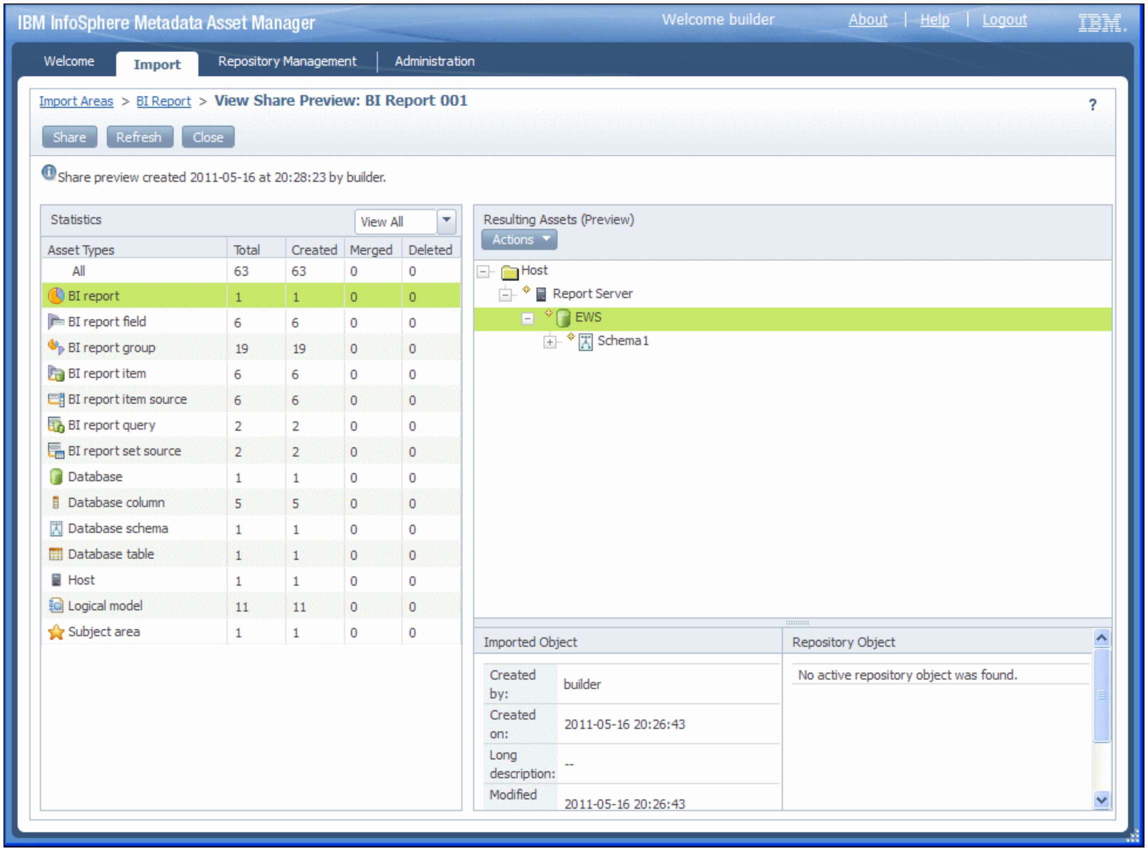
   a. Select and open the import area that contains the previously imported report.

   b. Select the **Staged Import** tab.

c. Click **Preview** to analyze the report assets to be shared and their effects on the existing assets in the InfoSphere Information Server (Figure 10-13).

You can preview the type and number of assets to be created, deleted, or merged after sharing. Browse to select a particular report asset object to view the pending changes after sharing.

d. Click **Share** to publish the database asset to the InfoSphere Information Server, which makes the loaded report available to all components.



Share preview created 2011-05-16 at 20:28:23 by builder.

Asset Types	Total	Created	Merged	Deleted
All	63	63	0	0
BI report	1	1	0	0
BI report field	6	6	0	0
BI report group	19	19	0	0
BI report item	6	6	0	0
BI report item source	6	6	0	0
BI report query	2	2	0	0
BI report set source	2	2	0	0
Database	1	1	0	0
Database column	5	5	0	0
Database schema	1	1	0	0
Database table	1	1	0	0
Host	1	1	0	0
Logical model	11	11	0	0
Subject area	1	1	0	0

Created by:	builder
Created on:	2011-05-16 20:26:43
Long description:	--
Modified	2011-05-16 20:26:43

No active repository object was found.

Figure 10-13 Sharing report assets with InfoSphere Information Server

10.4.2 Results

BI reports represent the consumers of information and the distribution media for business reports for internal and external consumption. Stakeholders want to understand the report and the associated business requirements. Most importantly, they want the ability to trace the report data to the originating source systems, further assessing the data quality of those source systems.

The following information assets are available in the InfoSphere Information Server upon loading a report:

BI report server () The computer system on which the report system resides. The server is created during the import of the BI report.

BI report ( ) The name of the BI report that contains the content for distribution. BI reports include fields.

BI report field ( ) The names of the fields or columns that are included in the report.

BI model () The report data model, against which the report has been developed. BI report models include collections.

BI collection ( ) A structured grouping or entity for containing information. BI collections include members.

BI collection member ()
The columns or attributes of a collection, referencing specific instances of data.

You can search, browse, and view the details of the BI report or report model assets from the components of InfoSphere Information Server, including InfoSphere Metadata Workbench.

10.5 Information asset enrichment

The proper understanding and use of information are critical to manage enterprise metadata and implement information integration solutions.

Developers must be able to browse and view the data structures with which they need to interface. However, in addition to providing them with the technical requirements to fulfill their tasks, they must also have a business understanding of the information and the overall business requirements. This knowledge is critical to achieving efficiencies and the quality of the development effort.

Furthermore, analysts and auditors require insight into the key systems and processes. This way, they can evaluate the application of data quality standards and rules and the adherence to the business requirements or regulatory standards.

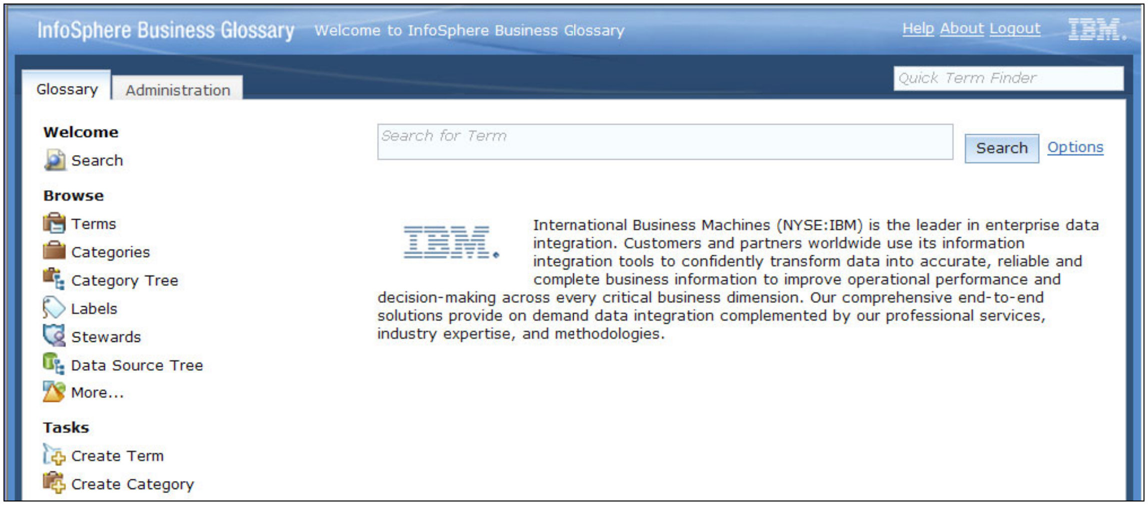
Although the metadata repository allows the load and representation of the data systems, reports, and processes, InfoSphere Information Server provides the framework to enrich the understanding of the information.

10.5.1 Business glossary terms

Achieving strategic business objectives requires a common and shared understanding of the business definitions and vocabulary. The incorrect interpretation or use of data increases the risk of errors and compromises its value. A requirement exists to extend the definition of technical assets with clearly defined and accepted business definitions.

Business metadata documents the business meaning and intent of information, its defined use, and structure. The business meaning defines the vocabulary, which is independent of technology and the source documentation.

Business metadata is created in the IBM InfoSphere Business Glossary (Figure 10-14), so that the business can establish relationships between business understanding and information assets. Stakeholders can search and benefit from the business glossary from all of the components of the InfoSphere Information Server.



International Business Machines (NYSE:IBM) is the leader in enterprise data integration. Customers and partners worldwide use its information integration tools to confidently transform data into accurate, reliable and complete business information to improve operational performance and decision-making across every critical business dimension. Our comprehensive end-to-end solutions provide on demand data integration complemented by our professional services, industry expertise, and methodologies.

Figure 10-14 InfoSphere Business Glossary

Business glossary terms express the precise definition of a singular concept or expression. Terms can further infer their meaning through defined synonyms or related terms.

You can assign business glossary terms to the assets in the metadata repository from the following components of the InfoSphere Information Server:

- IBM InfoSphere Business Glossary
- IBM InfoSphere Business Glossary Representational State Transfer (REST) application programming interface (API)

- IBM InfoSphere Metadata Workbench
- IBM InfoSphere FastTrack
- IBM InfoSphere Information Analyzer

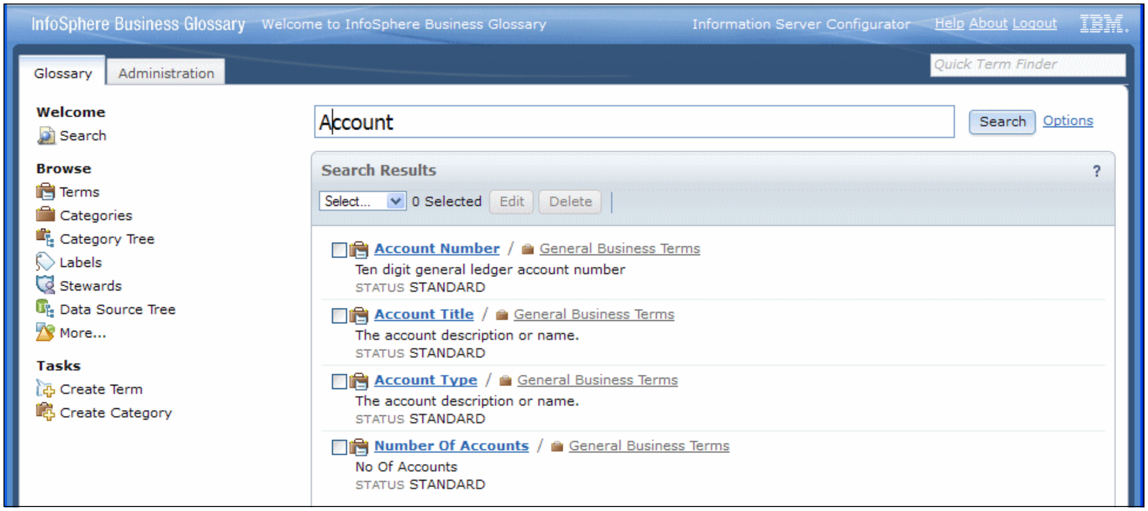
In the bank scenario in this book, we assign a business glossary term to a database table by using the InfoSphere Business Glossary. This action requires a business glossary assigner role.

Assigning business glossary terms

You can assign business glossary terms from and to and asset.

To assign a glossary term to an asset, complete these steps:

1. Log on to the InfoSphere Business Glossary.
2. Search for a glossary term to which to assign a database table.
3. On the Search page (Figure 10-15), enter the name, or partial name, of a term that you want to locate in the glossary. Click **Search** to view a list of matched results.



Account

Account Number / General Business Terms
Ten digit general ledger account number
STATUS STANDARD

Account Title / General Business Terms
The account description or name.
STATUS STANDARD

Account Type / General Business Terms
The account description or name.
STATUS STANDARD

Number Of Accounts / General Business Terms
No Of Accounts
STATUS STANDARD

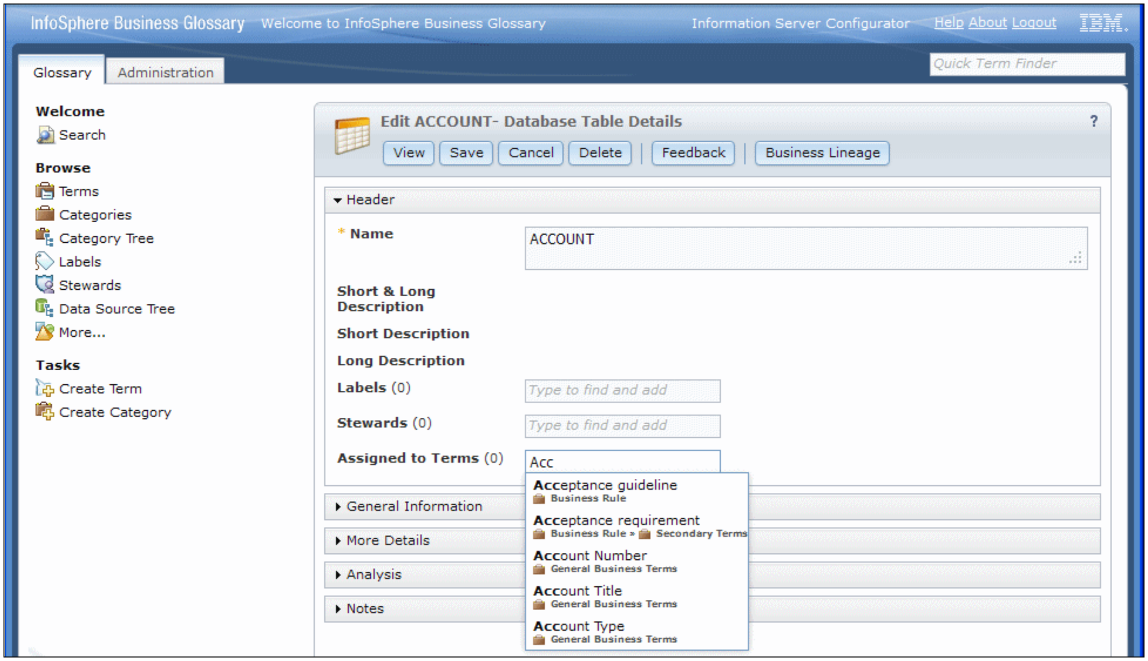
Figure 10-15 Glossary term search results

4. From the list of results, select the appropriate term to view the term details.
5. On the Term Details page, click **Edit** to add an assigned asset.
6. In the Assigned Asset section, enter the name of the database table to be assigned. Repeat steps 4-6 for all asset assignments for this term.
7. Click **Save** to save the term assignments and close the edit window.

To assign a glossary term from an asset, follow these steps:

1. Log on to the InfoSphere Business Glossary.
2. On the Search page, complete these steps:
   a. Click **Options**.
   b. Select **Search** to search for a database table.
   c. Enter the name, or partial name, of the table.
   d. Click **Search** to view a list of matched results.

   Alternatively, select the Data Source Tree or browse for additional information asset types in the business glossary to locate a database table.
3. From the list of results, select the requested table, and then click to view the table details.
4. On the Database Table Details page, click **Edit** to add an assigned asset.
5. On the Edit Account page (Figure 10-16), complete these steps:
   a. In the Assigned to Labels field, enter the name of the label to be assigned. Repeat steps 4 and 5 for all term assignments for this table.
   b. Click **Save** to save the term assignment and close the edit window.



Edit ACCOUNT- Database Table Details

Header

* Name: ACCOUNT

Short & Long Description

Short Description

Long Description

Labels (0): Type to find and add

Stewards (0): Type to find and add

Assigned to Terms (0): Acc

Acceptance guideline
Business Rule

Acceptance requirement
Business Rule » Secondary Terms

Account Number
General Business Terms

Account Title
General Business Terms

Account Type
General Business Terms

Figure 10-16 Term assignment of a database table

You can preview or search on the glossary term to database table relationship, similar to any term to asset relationship, in any component of InfoSphere Information Server.

10.5.2 Business glossary labels

Similar to business glossary terms, labels capture and represent business metadata. This metadata categorizes and imparts meaning, defined categorization, or the typing of assets in the metadata repository.

`Geography` or `Deployment Type` are examples of business labels that provide stakeholders additional understanding and classification. These business labels specifically provide the locale and intended use of the labeled database system as needed in an information integration project. Business requirements and governance regulations mandate capturing and applying business metadata for increased data understanding and validation.

You can assign business glossary labels to the assets in the metadata repository from the following components of InfoSphere Information Server:

- IBM InfoSphere Business Glossary
- IBM InfoSphere Business Glossary REST
- IBM InfoSphere Metadata Workbench

In the bank scenario, we assign a business glossary label to a database table by using InfoSphere Business Glossary. This action requires a business glossary assigner role.

Assigning business glossary labels

To assign a business glossary label to a database table, complete these steps:

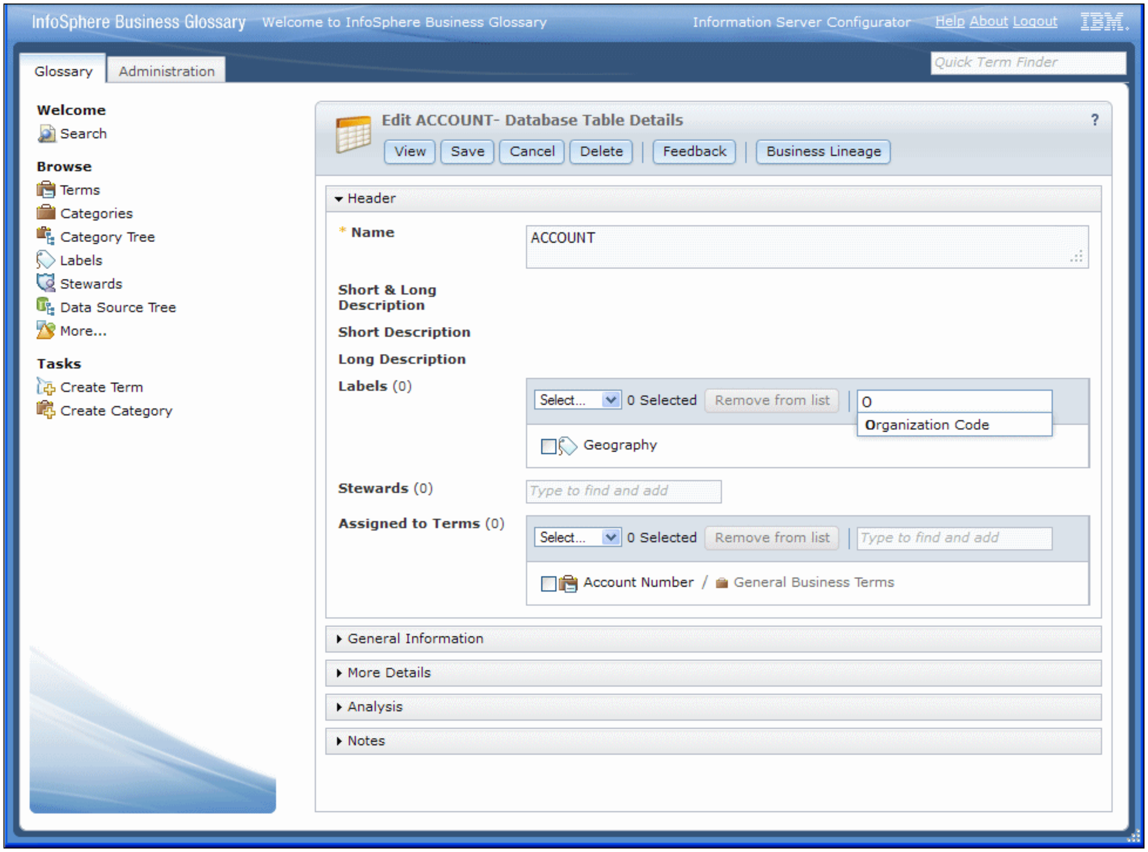
1. Log on to the InfoSphere Business Glossary.
2. On the Search page, complete these steps:
   a. Click **Options**.
   b. Select **Search** to search for a database table.
   c. Enter the name, or partial name, of the table.
   d. Click **Search** to view a list of matched results.

   Alternatively, select the Data Source Tree or browse for additional information asset types in the business glossary to locate a database table.
3. From the list of results, select the requested table, and click to view the table details.
4. On the Database Table Details page, click **Edit** to add a label assignment.

5. On the Edit Account page (Figure 10-17), complete these steps:

   a. In the Assigned to Term field, enter the name of the term to be assigned. Repeat the process for all label assignments for this table.

   b. Click **Save** to save the label assignment and close the Edit window.



Edit ACCOUNT- Database Table Details

* Name: ACCOUNT

Short & Long Description

Short Description

Long Description

Labels (0): 0 Selected; O; Organization Code; □ Geography

Stewards (0)

Assigned to Terms (0): 0 Selected; □ Account Number / General Business Terms

Figure 10-17 Label assignment of a database table

You can view or search a glossary label that is associated with a database table, similar to any detail of the database table, in InfoSphere Metadata Workbench.

10.5.3 Data stewardship

Data stewards are responsible for managing information, including enforcing the data policies and standards, assigning business definitions, and monitoring data quality. Data stewards work closely with the architects, administrators, and developers to define, establish, and monitor quality processes and to mitigate the risks that are associated with inadequate data quality.

Data stewards do not necessarily own the information assets. They are more aligned to the business rather than the technology. Data stewards, however, remain accountable for information and for ensuring the compliance with business requirements and standard practices.

Metadata management requires the creation and empowerment of a data governance strategy, including the stewardship for all assets that are in the metadata repository.

You can assign the data stewards to the assets in the metadata repository from the following components of the InfoSphere Information Server:

- IBM InfoSphere Business Glossary
- IBM InfoSphere Business Glossary REST
- IBM InfoSphere Metadata Workbench
- IBM InfoSphere Information Analyzer
- IBM InfoSphere FastTrack

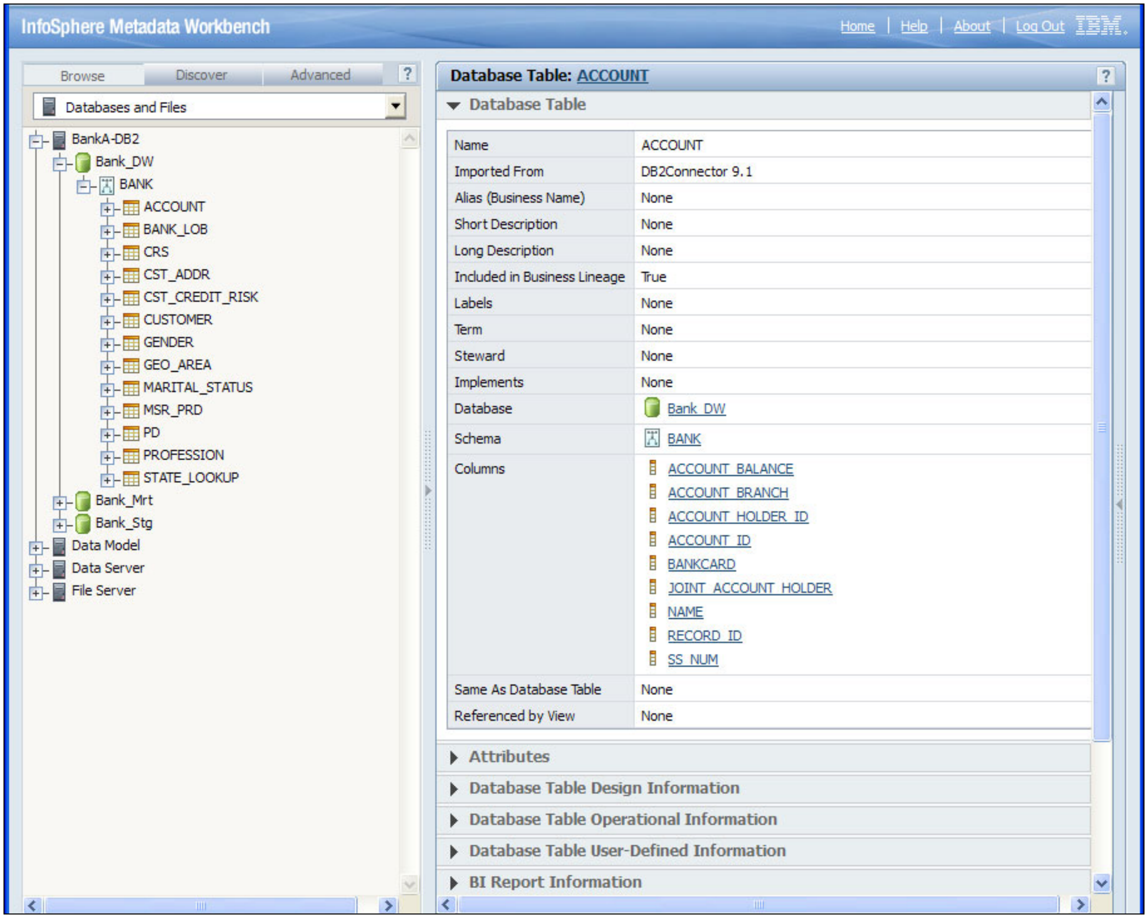
In the example in this book, we assign a data steward to a database table by using InfoSphere Metadata Workbench.

Assigning a data steward

To assign a data steward to a database table, complete these steps:

1. Log on to the InfoSphere Metadata Workbench.
2. From the home page (Figure 10-18), in the left navigation pane, click **Browse**. In the right pane, click **Databases and Tables**. Here you can browse the host systems and navigate their included database systems, by selecting the database table for which a data steward is to be added.

   Alternatively, from the home page, select the asset type **Database Table** and enter the name or partial name of the table to find a specific database table. Click **Find** to view a list of matched results and to select the requested table.
3. On the Database Table Details page, in the right pane, click **Assign to Steward** to add a data steward assignment.



Database Table: ACCOUNT

Database Table

Name	ACCOUNT
Imported From	DB2Connector 9.1
Alias (Business Name)	None
Short Description	None
Long Description	None
Included in Business Lineage	True
Labels	None
Term	None
Steward	None
Implements	None
Database	Bank_DW
Schema	BANK
Columns	ACCOUNT_BALANCE ACCOUNT_BRANCH ACCOUNT_HOLDER_ID ACCOUNT_ID BANKCARD JOINT_ACCOUNT_HOLDER NAME RECORD_ID SS_NUM
Same As Database Table	None
Referenced by View	None

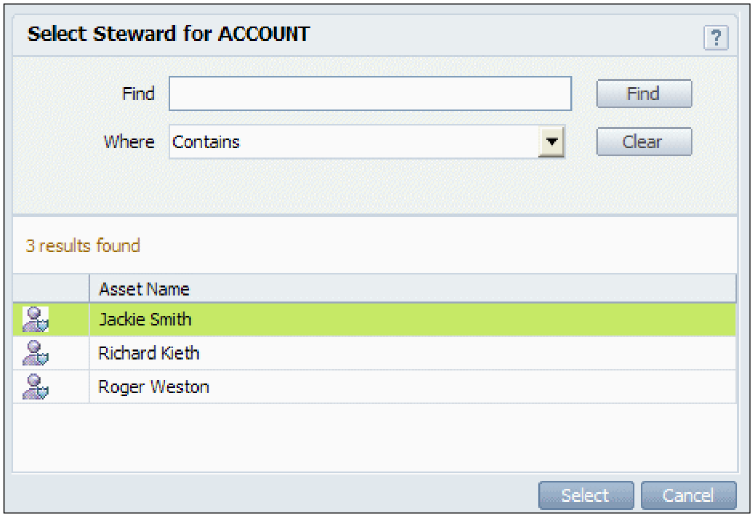
Attributes
Database Table Design Information
Database Table Operational Information
Database Table User-Defined Information
BI Report Information

Figure 10-18 Details of Database Table in InfoSphere Metadata Workbench

4. In the Select Steward for ACCOUNT window (Figure 10-19), select a data steward:

   a. In the Find field, enter the name, or partial name, of the data steward to be assigned to the database table. Click **Find** to view a list of results.

   b. Select the requested data steward, and click **Select** to assign the data steward to the database table.



Where: Contains

3 results found

Asset Name
Jackie Smith
Richard Kieth
Roger Weston

Figure 10-19 Data steward selection window

With InfoSphere Metadata Workbench, you can assign a data steward, business term, or business label to multiple assets in a single action, when searching on or displaying a list of assets.

You can preview or search on the data steward that is associated with a database table, similar to any detail of the database table, in any component of InfoSphere Information Server.

10.5.4 Asset descriptor and alias

Information assets, such as the information assets that are loaded into the metadata repository, require concise and relevant descriptors. These descriptors impart their intended understanding, structure, classification, and specification to the stakeholders. Although the assets include a name and context, they do not facilitate the advanced searching and classifications that are required.

Descriptors facilitate the correct use of information and validate the expected results. They can also provide additional data requirement, specification, or ownership details. The alias name references the synonym or business name of an asset. Often the common name is referenced by stakeholders when they search for information.

Unlike business glossary terms or business glossary labels, which are specific to business metadata and requirements, descriptors are often specific to the technical ownership and use of the assets. Examples include the data owner, developer, or modeler.

Asset descriptors include short and long descriptions, in addition to the ability to author and manage custom descriptors for assets from within InfoSphere Metadata Workbench. With custom descriptors, you can extend and capture additional technical details, such as the data owner, organizational code, or security flag.

You can author certain descriptors, such as the short description, long description, and alias name, from the following components of InfoSphere Information Server:

- IBM InfoSphere Business Glossary
- IBM InfoSphere Metadata Workbench
- IBM InfoSphere Information Analyzer
- IBM InfoSphere FastTrack

In the bank scenario, we create a custom asset descriptor and edit a database table by authoring its descriptors and alias name by using InfoSphere Metadata Workbench. This action requires an InfoSphere Metadata Workbench administrator role.

Creating an asset descriptor

To create a custom asset descriptor, complete these steps:

1. Log on to the InfoSphere Metadata Workbench.
2. Select the **Advanced** tab from the left navigation pane. Then, select the link **Create Custom Attribute**.

3. In the Untitled Custom Attribute Definition window (Figure 10-20 on page 367), complete these steps:

   a. Enter the name of the custom property. The name represents the descriptor that is capturing the technical definition, classification, or structure to be applied to the asset.

   b. Optional: Enter a description for the custom property to be created.

   c. Select the information asset type to which the custom property will be applied. In this example, we select **Database Table, View, Data File Structure**. The list of asset types includes all technical assets that are supported in the InfoSphere Information Server and that are loaded into the metadata repository.

   d. Select the type of custom property to be created. You can create custom properties of the following types:

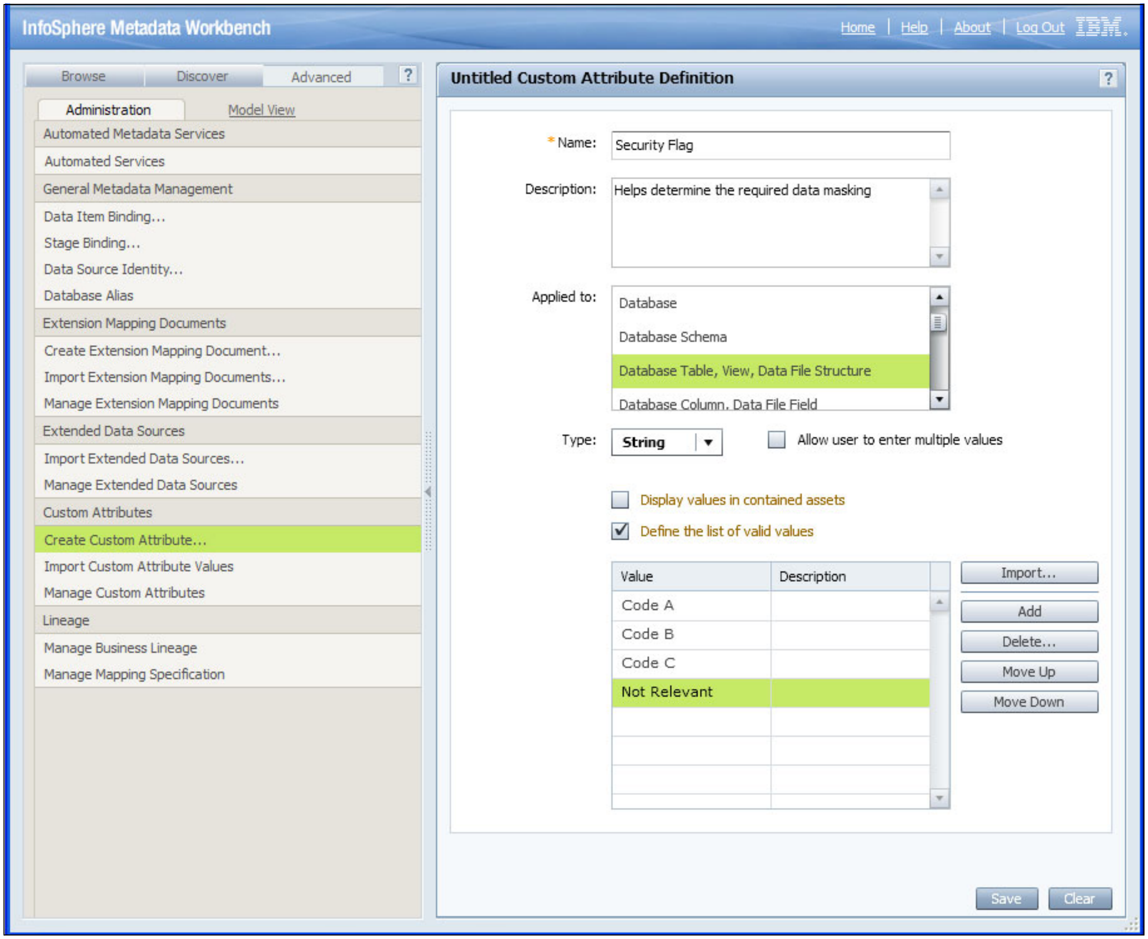
      - String to capture textual descriptions, specifications, or instructions
      - Number to capture numeric information, such as a version number
      - Date to capture time and date information, such as the run time or creation date

   e. Optional: Select **Allow user to enter multiple values** so that a user can author and maintain multiple descriptor values for the asset.

   f. Optional: Select **Display values in contained assets** so that a user can view the descriptor when browsing contained assets. For example, display the descriptor for a database table when viewing the columns of the database. For example, share a linked technical specification that is defined as a custom property on the table.

   g. Optional: Define a set of enumerated values that a user can select and apply to the asset. Select **Define the list of valid values**, and then click **Add** to define the list of values. Alternatively, select **Import** to load a list of values from a text file. For example, you can define a set of organizational codes or security flags.

   h. Click **Save** to create the custom property to represent the asset descriptor.


InfoSphere Metadata Workbench
Home | Help | About | Log Out
IBM.
Browse
Discover
Advanced
?
Administration
Model View
Automated Metadata Services
Automated Services
General Metadata Management
Data Item Binding…
Stage Binding…
Data Source Identity…
Database Alias
Extension Mapping Documents
Create Extension Mapping Document…
Import Extension Mapping Documents…
Manage Extension Mapping Documents
Extended Data Sources
Import Extended Data Sources…
Manage Extended Data Sources
Custom Attributes
Create Custom Attribute…
Import Custom Attribute Values
Manage Custom Attributes
Lineage
Manage Business Lineage
Manage Mapping Specification
Untitled Custom Attribute Definition
?
Import…
Add
Delete…
Move Up
Move Down
Save
Clear

*Name: Security Flag

Description: Helps determine the required data masking

Applied to:
Database
Database Schema
Database Table, View, Data File Structure
Database Column, Data File Field

Type: String

□ Allow user to enter multiple values

□ Display values in contained assets

☑ Define the list of valid values

Value	Description
Code A	
Code B	
Code C	
Not Relevant	

Figure 10-20 Creating a custom attribute in InfoSphere Metadata Workbench

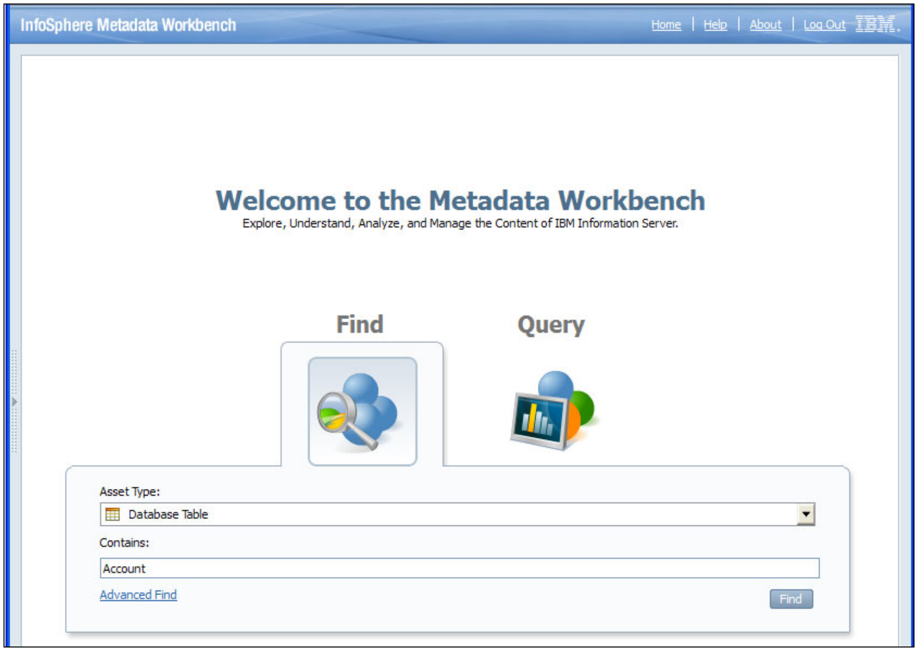
4. Click the **Manage Custom Properties** link to view a listing of all defined properties. You can edit, delete, or export the property definitions.

Creating author descriptors

To create author descriptors, complete these steps:

1. Log on to the InfoSphere Metadata Workbench.
2. On the home page (Figure 10-21), search for a specific database table to author its descriptor properties:

   a. For Asset Type, select **Database Table**.
   b. For Contains, enter the name or partial name of the table.
   c. Click **Find** to view a list of results.



Welcome to the Metadata Workbench

Explore, Understand, Analyze, and Manage the Content of IBM Information Server.

Asset Type: Database Table

Contains: Account

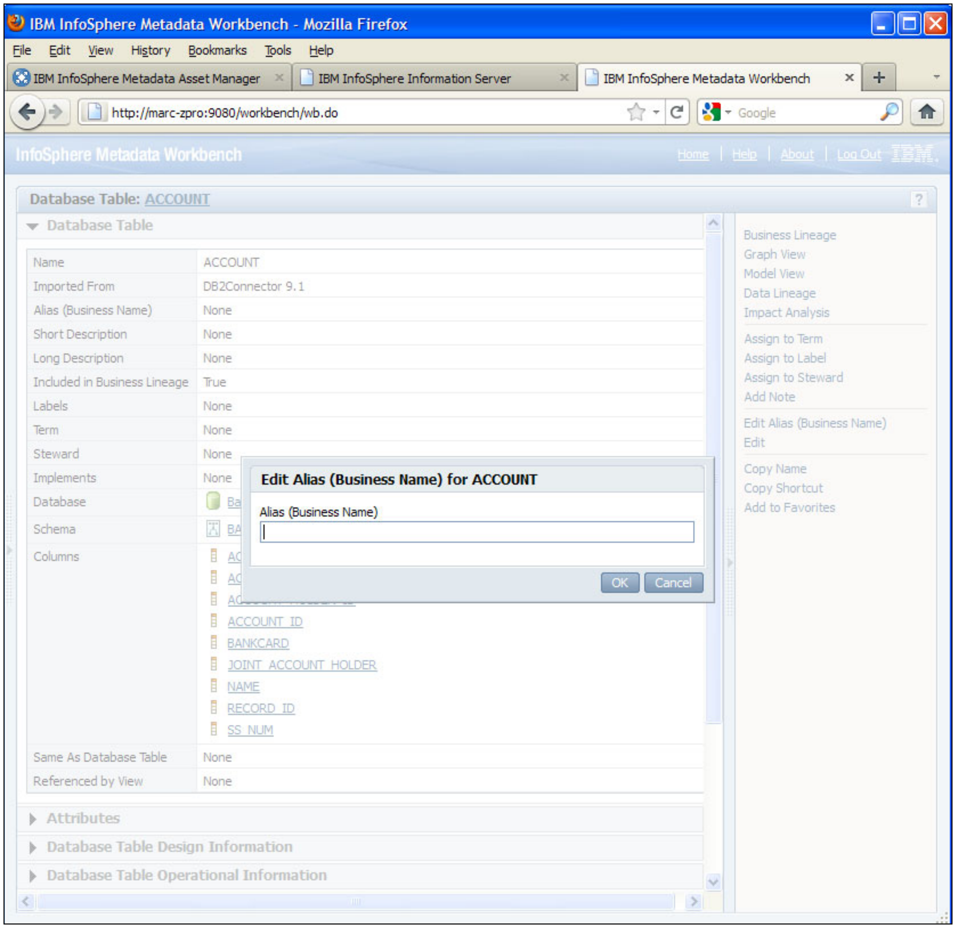
Figure 10-21 Searching for a database table in InfoSphere Metadata Workbench

3. From the list of results, select the requested database table to view the Database Table Details page.
4. In the right action pane, click **Edit** to edit the descriptors of the database table asset.
5. In the Edit window for the asset opens, complete these steps:

   a. Author or edit the short description or long description of the asset.
   b. Author or edit the custom properties that are associated with the asset.

6. In the right action pane, click **Edit Alias (Business Name)** to author the database table alias name.

7. In the Edit Alias window (Figure 10-22), author a new alias name, or edit an existing name. Then click **OK** to save the authoring changes.


IBM InfoSphere Metadata Workbench - Mozilla Firefox
File Edit View History Bookmarks Tools Help
IBM InfoSphere Metadata Asset Manager
IBM InfoSphere Information Server
IBM InfoSphere Metadata Workbench
http://marc-zpro:9080/workbench/wb.do
Google
InfoSphere Metadata Workbench
Home | Help | About | Log Out
IBM.
Business Lineage
Graph View
Model View
Data Lineage
Impact Analysis
Assign to Term
Assign to Label
Assign to Steward
Add Note
Edit Alias (Business Name)
Edit
Copy Name
Copy Shortcut
Add to Favorites
Attributes
Database Table Design Information
Database Table Operational Information
OK
Cancel

Database Table: ACCOUNT

Database Table

Name	ACCOUNT
Imported From	DB2Connector 9.1
Alias (Business Name)	None
Short Description	None
Long Description	None
Included in Business Lineage	True
Labels	None
Term	None
Steward	None
Implements	None
Database	Ba
Schema	BA
Columns	AC AC AC ACCOUNT_ID BANKCARD JOINT_ACCOUNT_HOLDER NAME RECORD_ID SS_NUM
Same As Database Table	None
Referenced by View	None

Edit Alias (Business Name) for ACCOUNT

Alias (Business Name)

Figure 10-22 Authoring an alias name for a database table

Creating import descriptors

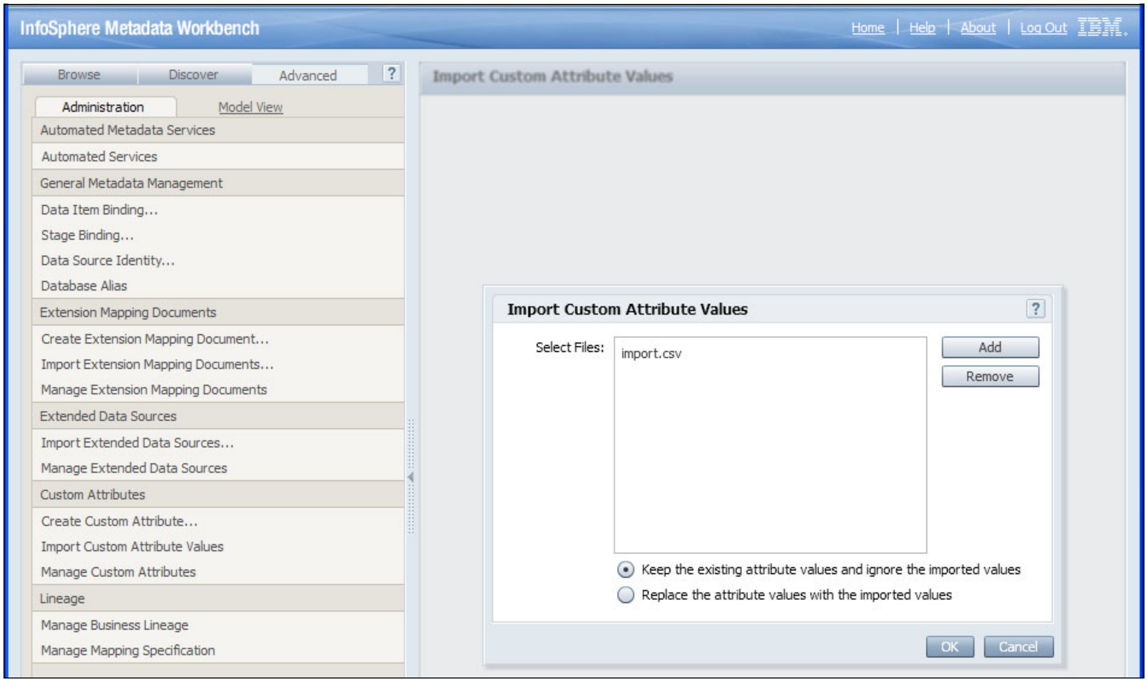
You can import the values of the custom properties that have been defined for information assets in InfoSphere Metadata Workbench, from externally generated input files. With this capability, you can take advantage of easy tagging and the fulfilment of the requirement to document additional details of the asset.

You can extract the input file from InfoSphere Metadata Workbench. The file format resembles a spreadsheet that identifies the asset and lists the custom property values.

To create import descriptors, complete these steps:

1. Log on to the InfoSphere Metadata Workbench.
2. In the left navigation pane, on the **Advanced** tab, select the **Import Custom Attribute Values** link.
3. In the Import Custom Attribute Values panel (Figure 10-23 on page 371), complete these steps:
   a. Click **Add** to browse to select one or more input files.
   b. Optional: Select one of the following actions:
      - Select **Keep the existing attribute values and ignore the imported values** to keep the values that are associated with the assets that are identified in the input file. This selection does not overwrite the existing asset descriptors.
      - Select **Replace the attribute values with the imported values** to remove existing values associated with the assets identified in the input file. This option also replaces those property values with the property values that are defined in the input file.

   c. Click **OK** to complete the process.


InfoSphere Metadata Workbench
Home | Help | About | Log Out
IBM.
Browse
Discover
Advanced
?
Administration
Model View
Automated Metadata Services
Automated Services
General Metadata Management
Data Item Binding...
Stage Binding...
Data Source Identity...
Database Alias
Extension Mapping Documents
Create Extension Mapping Document...
Import Extension Mapping Documents...
Manage Extension Mapping Documents
Extended Data Sources
Import Extended Data Sources...
Manage Extended Data Sources
Custom Attributes
Create Custom Attribute...
Import Custom Attribute Values
Manage Custom Attributes
Lineage
Manage Business Lineage
Manage Mapping Specification
Import Custom Attribute Values
Add
Remove
OK
Cancel

Import Custom Attribute Values

Select Files: import.csv

● Keep the existing attribute values and ignore the imported values

○ Replace the attribute values with the imported values

Figure 10-23 Importing custom property values in InfoSphere Metadata Workbench

Creating export descriptors

You can export the values of custom properties that are defined for information assets in InfoSphere Metadata Workbench for bulk editing or replacing. With this capability, you can take advantage of the maintenance and authoring features. You can extract the file from InfoSphere Metadata Workbench for a list of assets.

To create an export descriptor, complete these steps:

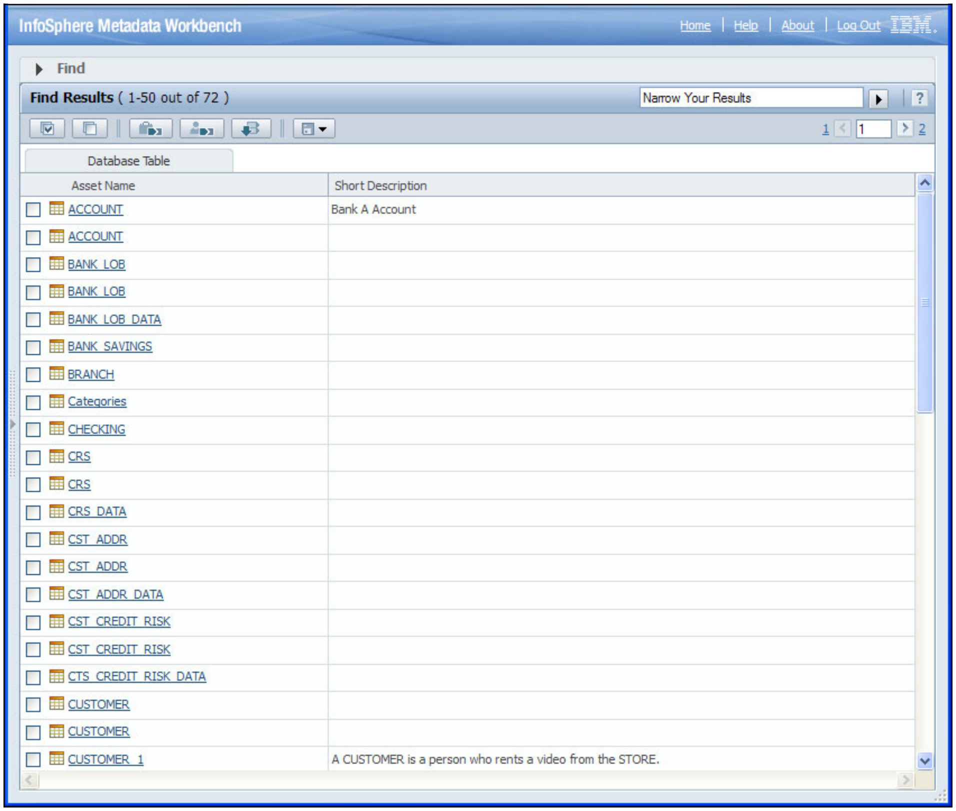
1. Log on to the InfoSphere Metadata Workbench.
2. On the home page, search for a specific asset type, such as database tables, by selecting the asset type **Database Table**. Optionally, enter a partial name of a table to filter the resulting list. Then click **Find** to view the results.

3. In the list of database tables (Figure 10-21), perform general authoring or management tasks:

   a. Select the individual database table assets to include in the export of the custom properties. Alternatively, click **Select All** from the toolbar to select all of the database table assets that are listed.

   b. Select **Export selected assets and their custom attribute values** from the toolbar.

   c. In the File Save window, enter the location to which to save the output file.

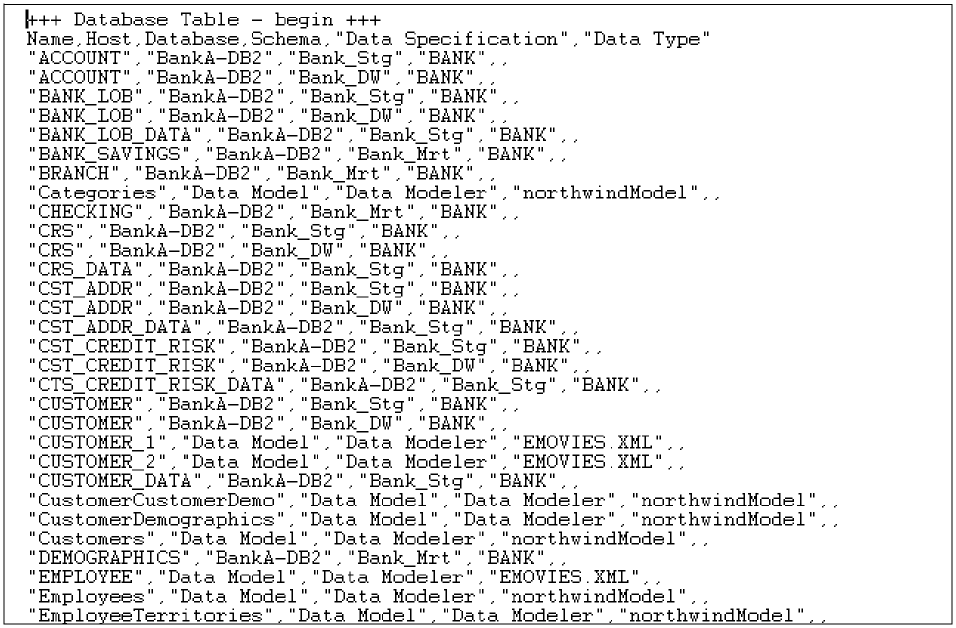

InfoSphere Metadata Workbench
Home | Help | About | Log Out
IBM.
Find
Find Results (1-50 out of 72)
Narrow Your Results
?
1 1 2
Database Table

	Asset Name	Short Description
□	ACCOUNT	Bank A Account
□	ACCOUNT	
□	BANK_LOB	
□	BANK_LOB	
□	BANK_LOB_DATA	
□	BANK_SAVINGS	
□	BRANCH	
□	Categories	
□	CHECKING	
□	CRS	
□	CRS	
□	CRS_DATA	
□	CST_ADDR	
□	CST_ADDR	
□	CST_ADDR_DATA	
□	CST_CREDIT_RISK	
□	CST_CREDIT_RISK	
□	CTS_CREDIT_RISK_DATA	
□	CUSTOMER	
□	CUSTOMER	
□	CUSTOMER_1	A CUSTOMER is a person who rents a video from the STORE.

Figure 10-24 List of assets in InfoSphere Metadata Workbench

4. Edit the custom property values that are associated with the listed assets of the output file.

The export file is in the comma-separated value (CSV) format, delineating the names and identity for each selected asset. It includes a complete list (see Figure 10-25) of all custom attribute definitions that are defined for the asset type and the values, including null or empty values, for each asset.



```
+++ Database Table - begin +++
Name,Host,Database,Schema,"Data Specification","Data Type"
"ACCOUNT","BankA-DB2","Bank_Stg","BANK",,
"ACCOUNT","BankA-DB2","Bank_DW","BANK",,
"BANK_LOB","BankA-DB2","Bank_Stg","BANK",,
"BANK_LOB","BankA-DB2","Bank_DW","BANK",,
"BANK_LOB_DATA","BankA-DB2","Bank_Stg","BANK",,
"BANK_SAVINGS","BankA-DB2","Bank_Mrt","BANK",,
"BRANCH","BankA-DB2","Bank_Mrt","BANK",,
"Categories","Data Model","Data Modeler","northwindModel",,
"CHECKING","BankA-DB2","Bank_Mrt","BANK",,
"CRS","BankA-DB2","Bank_Stg","BANK",,
"CRS","BankA-DB2","Bank_DW","BANK",,
"CRS_DATA","BankA-DB2","Bank_Stg","BANK",,
"CST_ADDR","BankA-DB2","Bank_Stg","BANK",,
"CST_ADDR","BankA-DB2","Bank_DW","BANK",,
"CST_ADDR_DATA","BankA-DB2","Bank_Stg","BANK",,
"CST_CREDIT_RISK","BankA-DB2","Bank_Stg","BANK",,
"CST_CREDIT_RISK","BankA-DB2","Bank_DW","BANK",,
"CTS_CREDIT_RISK_DATA","BankA-DB2","Bank_Stg","BANK",,
"CUSTOMER","BankA-DB2","Bank_Stg","BANK",,
"CUSTOMER","BankA-DB2","Bank_DW","BANK",,
"CUSTOMER_1","Data Model","Data Modeler","EMOVIES.XML",,
"CUSTOMER_2","Data Model","Data Modeler","EMOVIES.XML",,
"CUSTOMER_DATA","BankA-DB2","Bank_Stg","BANK",,
"CustomerCustomerDemo","Data Model","Data Modeler","northwindModel",,
"CustomerDemographics","Data Model","Data Modeler","northwindModel",,
"Customers","Data Model","Data Modeler","northwindModel",,
"DEMOGRAPHICS","BankA-DB2","Bank_Mrt","BANK",,
"EMPLOYEE","Data Model","Data Modeler","EMOVIES.XML",,
"Employees","Data Model","Data Modeler","northwindModel",,
"EmployeeTerritories","Data Model","Data Modeler","northwindModel",,
```

Figure 10-25 Import file format for custom attribute values in InfoSphere Metadata Workbench

10.6 Conclusion

In conclusion, this chapter explained how to use InfoSphere Metadata Workbench to load data storage systems, data models, and BI reports into the metadata repository. It also explained how you can enrich your information assets through asset descriptors, author descriptors, and import and export descriptors.

Chapter 11, “Data transformation” on page 375, explains how to use InfoSphere FastTrack and InfoSphere DataStage to extract, transform, and load data to the metadata repository and data mart for developing BI reports and analysis.

Then Chapter 12, “Enterprise reports and lineage generation” on page 393, explains how to view enterprise data lineage reports and other reports by using the information collected and stored in the metadata repository with InfoSphere Metadata Workbench.

11

Data transformation

IBM InfoSphere FastTrack creates source-target mapping specifications for data integration jobs. IBM InfoSphere DataStage and IBM InfoSphere QualityStage provide the essential integration functionality to integrate data across multiple source and target applications, collecting, transforming, and delivering high volumes of data. InfoSphere QualityStage also provides data cleansing functionality from standardization, to deduplication, to establishing master data.

InfoSphere FastTrack, InfoSphere DataStage, and InfoSphere QualityStage are all IBM InfoSphere Information Server product modules. This chapter focuses on these modules from a mapping and job generation perspective. This chapter also explains how to construct the jobs to enable the automatic generation of lineage and impact analysis reports by IBM InfoSphere Metadata Workbench, contributing to successful metadata management.

This chapter includes the following sections:

- Introduction to InfoSphere FastTrack
- Basic mapping
- Advanced mapping
- Mapping lifecycle management (job generation)
- Metadata sharing (extension mappings)
- InfoSphere DataStage job design
- Shared metadata
- Operational metadata
- Conclusion

11.1 Introduction to InfoSphere FastTrack

InfoSphere FastTrack is one of the InfoSphere Information Server product modules. It creates source-target (or more technically target to source) mapping specifications for data integration jobs.

InfoSphere FastTrack provides a central and secure interface to manage all data integration mappings. If provides the following advantages and functions:

- Is useful for any integration tooling (Java, InfoSphere DataStage, and so on)
- Provides a single identical copy of all metadata
- Is auditable for tracking mapping history
- Provides centralized management review, statistics, custom status codes, and so on
- Supports the reuse of rule specifications across projects (and for future endeavors)
- Provides search and discover functions for locating the “best” column mappings
- Uses business glossary term relationships to help identify appropriate mappings
- Offers increased visibility of mappings through standardized reports
- Establishes data lineage for any source-to-target mappings
- Generates “template” data integration jobs for InfoSphere DataStage and InfoSphere QualityStage

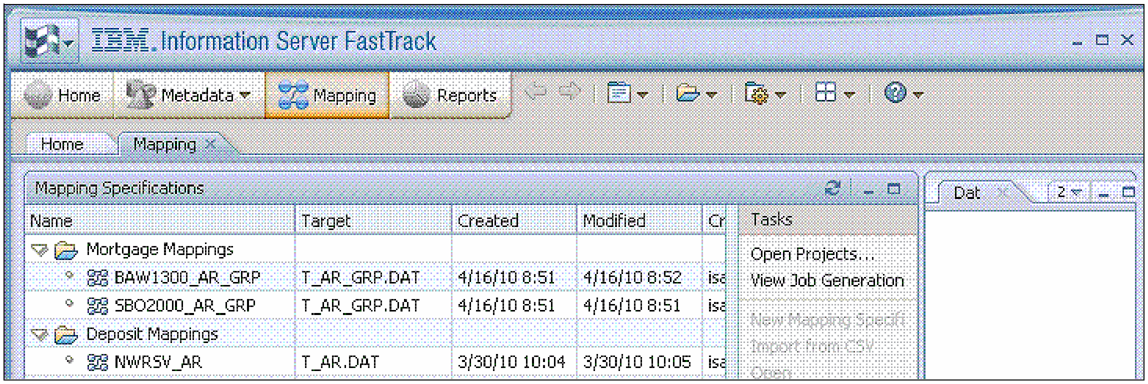
InfoSphere FastTrack mapping is displayed in InfoSphere Metadata Workbench for lineage immediately. Without export, without creation as an extension mapping, and without job generation, you can display an InfoSphere FastTrack mapping directly.

InfoSphere FastTrack is a client-server-based product module. Its user interface is an Eclipse-based application so that users can define mapping specifications that describe source to target data movement. It is supported by a services layer that is implemented as WebSphere Application Server services. These services are installed on the WebSphere Application Server instance of InfoSphere Information Server.

11.1.1 Functionality and user interface

InfoSphere FastTrack mapping specifications are organized into projects. *Projects* are centralized locations where you manage business specifications. The status of each project can be traced through its various stages: Project Review, In Progress, and Deployed. InfoSphere FastTrack projects can contain multiple mapping specifications. There is no limit to the number of projects that can be created. Because InfoSphere FastTrack is an InfoSphere Information Server product module, it allows for seamless sharing of the common security user model of InfoSphere Information Server.

For the bank scenario in this book, we created two projects: Mortgage Mappings and Deposit Mappings (Figure 11-1).



Name	Target	Created	Modified	Cr
Mortgage Mappings				
BAW1300_AR_GRP	T_AR_GRP.DAT	4/16/10 8:51	4/16/10 8:52	isa
SBO2000_AR_GRP	T_AR_GRP.DAT	4/16/10 8:51	4/16/10 8:51	isa
Deposit Mappings				
NWRSV_AR	T_AR.DAT	3/30/10 10:04	3/30/10 10:05	isa

Tasks

Open Projects...
View Job Generation
New Mapping Specifi
Import from CSV
Open

Figure 11-1 InfoSphere FastTrack projects

The InfoSphere FastTrack is a single managed infrastructure to track projects from the requirements phase to project deployment. With InfoSphere FastTrack, you can create the following items:

- Required transformation rules
- Source-to-target column mappings
- The definition of business terms with links to physical data structures

With InfoSphere FastTrack, you can also generate InfoSphere DataStage jobs and historical documentation for project tracking purposes.

11.1.2 Administration

The administration function of InfoSphere FastTrack can be summarized into administering the projects, user roles, and cross-tool permissions of InfoSphere FastTrack.

An *InfoSphere FastTrack project* is a logical unit that contains a set of mapping specifications secured separately from other projects (such as InfoSphere DataStage projects). User access is granted on the level of an InfoSphere FastTrack project. The InfoSphere FastTrack administrator is responsible for creating and removing projects as needed during the development cycle.

InfoSphere FastTrack administrators can create InfoSphere FastTrack projects and assign *users* to those projects. InfoSphere FastTrack users can use the core InfoSphere FastTrack functionality, import metadata, create mapping specifications, and generate reports.

InfoSphere FastTrack administrators provide *cross-tool permissions* for users. For example, InfoSphere FastTrack users must also be InfoSphere DataStage users with appropriate permission to generate InfoSphere DataStage jobs. In addition, InfoSphere FastTrack users must be glossary users and glossary authors in order to use and update InfoSphere Business Glossary.

The InfoSphere FastTrack administrator must ensure that the InfoSphere FastTrack map developers have all of the appropriate suite roles so that they can complete their work.

11.2 Basic mapping

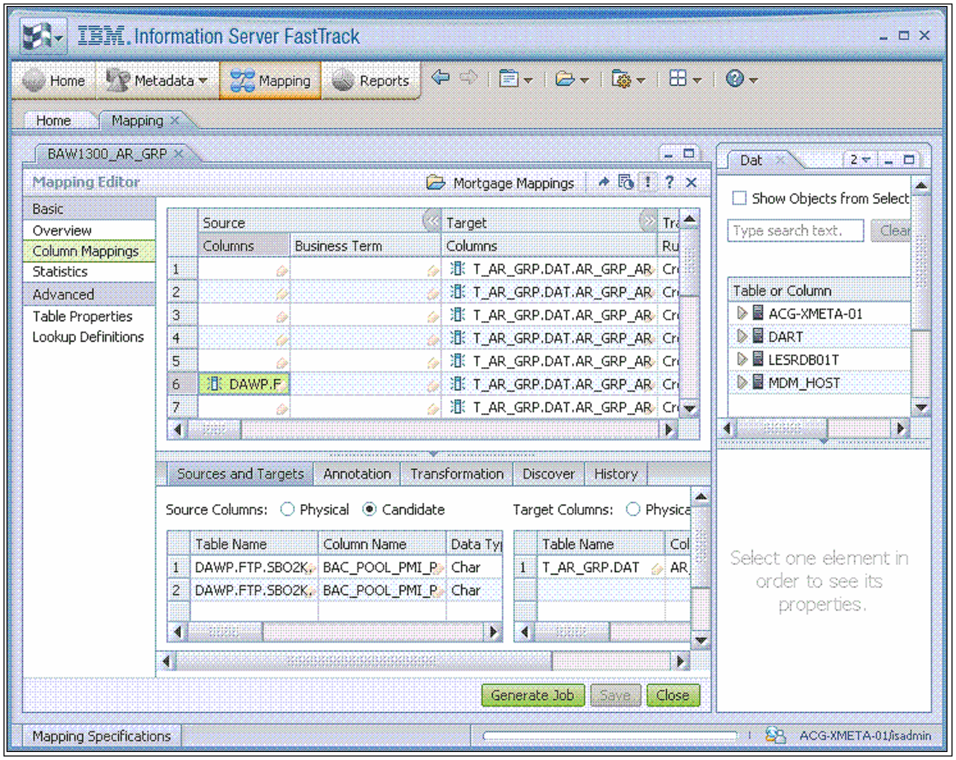
In InfoSphere FastTrack, you can create mapping specifications by using the following methods:

- The drag-and-drop method
- Auto-map wizards
- Importing previously created mapping specifications that were defined in Microsoft Excel or comma-separated value (CSV) files

The first two methods use the InfoSphere FastTrack user interface. With the import method, almost any format of mapping spreadsheet can be used as long as each row in the spreadsheet includes a source column and target columns. Optionally each row can contain mapping rules or transformation pseudo code that describes how the data is transformed before it is written to the target column. These mapping documents can be imported in their entirety. Alternatively, a portion of the document can be imported by specifying start and end rows and column delimiters. Such information as column headings can be filtered out during the import process.

In all cases, the mapping process can be monitored for status (Mapped, Reviewed, and Validated). Statistics are also gathered during map creation such as the number of columns and tables mapped, the map creation date, the last modified date, and any description information.

More than one map can be in the process of being created at any given time, which allows dependencies between processes to be easily synchronized. Also the ability to cut, copy, and paste mapping elements between specifications makes it easy to share specification logic between maps. Figure 11-2 shows two columns being mapped into one column on line six.



	Source		Target	Tr
	Columns	Business Term	Columns	Ru
1			T_AR_GRP.DAT.AR_GRP_AR	Cr
2			T_AR_GRP.DAT.AR_GRP_AR	Cr
3			T_AR_GRP.DAT.AR_GRP_AR	Cr
4			T_AR_GRP.DAT.AR_GRP_AR	Cr
5			T_AR_GRP.DAT.AR_GRP_AR	Cr
6	DAWP.F		T_AR_GRP.DAT.AR_GRP_AR	Cr
7			T_AR_GRP.DAT.AR_GRP_AR	Cr

Source Columns: ○ Physical ● Candidate

	Table Name	Column Name	Data Ty
1	DAWP.FTP.SBO2K.	BAC_POOL_PMI_P	Char
2	DAWP.FTP.SBO2K.	BAC_POOL_PMI_P	Char

Target Columns: ○ Physica

	Table Name	Col
1	T_AR_GRP.DAT	AR

□ Show Objects from Select

Table or Column
ACG-XMETA-01
DART
LESRDB01T
MDM_HOST

Figure 11-2 Two columns mapped into one column

Table mapping supports one to one (1:1) column mapping, one to many (1:many) column mapping, many to one (many:1) column mapping, and many to many (many:many) column mapping.

The column metadata used in these mappings can be imported from the physical database or from a physical data model. Data model imports are done by using the common InfoSphere Information Server import/export manager. (This manager is included with all InfoSphere Information Server modules as part of the core infrastructure.)

InfoSphere FastTrack can also use metadata that was imported by using other InfoSphere Information Server product modules such as InfoSphere Information Analyzer or InfoSphere DataStage connector stages. Consider that you have followed best practices and profiled your source data to help you understand its structure and the types of transformations to apply. In this case, you can use the same metadata that was previously imported by InfoSphere Information Analyzer.

If no shared metadata is in the repository that can be used to create InfoSphere FastTrack mapping documents, maps can be created with columns defined in the map itself. These columns are called *candidate columns*. They can be keyed in manually, or they can be created from existing InfoSphere Business Glossary terms.

While the map is being defined, you can view the underlying data defined by the source or target metadata (if available) as sample data content with security login. The fields of each mapping element are customizable by using display options. These options include sort order, columns displayed, and viewable properties, such as physical column characteristics. All options can be configured as visible or not visible depending on the preference of the user who is creating the map.

11.3 Advanced mapping

Up to this point, we have focused on simple source to target mappings. InfoSphere FastTrack can also help create more complicated mapping specifications. These more complicated maps can populate multiple target tables from a single source table. They can also incorporate switch statement constructs that define values to direct data flow into multiple target tables.

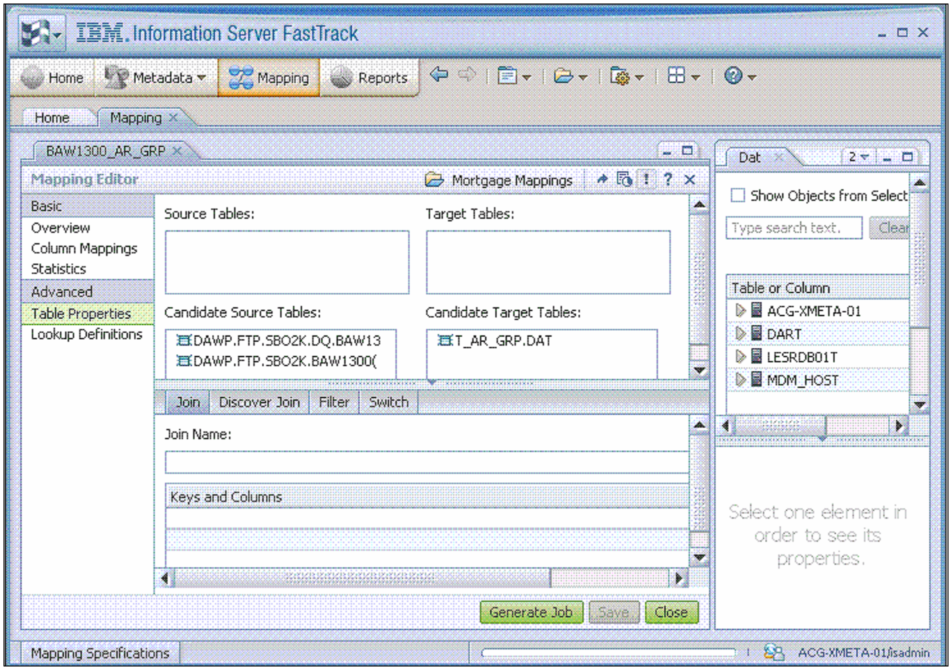
Lookup constructs can also be incorporated. Lookups, for example, are elements of a map that allow for the replacement of code values with textual descriptions of what a code represents. These lookups can be designed for reuse and can then be imported from a previously defined CSV file.

Discovery algorithms and auto-map wizards are also available to facilitate the creation of mapping documents. Also mapping facilities are available to find exact, partial, and lexical matches on column names, which are useful when mapping large sources and targets. Column match details include linkages found in InfoSphere Business Glossary between dissimilar names. For example, Tax ID and SS Num are synonyms in InfoSphere Business Glossary. The relationship can be used to automatically map these elements. The auto-mapping wizards can also identify column names, such as CUST = CUSTOMER, reducing the time required to map hundreds of columns.

Business linkages can be used to create, link, and share standard business terms with physical columns. Business terms can be assigned manually or by using drag-and-drop functionality from shared metadata. Existing InfoSphere Business Glossary terms can be associated with source or target map columns. New InfoSphere Business Glossary terms can also be created by using the InfoSphere FastTrack user interface. The only limitation of this feature is that an InfoSphere Business Glossary category must exist for the new term to be created in it.

InfoSphere Business Glossary terms can be imported from a spreadsheet into a mapping specification. InfoSphere FastTrack can publish relationships between terms and physical columns to the InfoSphere Business Glossary. In InfoSphere Business Glossary, this process is also referred to as *defining related IT assets*.

The creation of business terms and publishing results requires appropriate user permissions to implement. Figure 11-3 shows that we have started the mapping specification with candidate tables. These tables do not yet exist in the repository. There are two source tables being mapped into a single target table. In this example, we do not yet have any joins or lookups defined. If it is necessary, these definitions can be implemented here.



Candidate Source Tables:
DAWP.FTP.SBO2K.DQ.BAW13
DAWP.FTP.SBO2K.BAW1300(

Candidate Target Tables:
T_AR_GRP.DAT

Select one element in order to see its properties.

Figure 11-3 Table mapping

11.4 Mapping lifecycle management (job generation)

InfoSphere FastTrack, as the name (fast track) implies, helps business users to quickly and easily define elements of a data flow. It also helps them to create mapping documents for InfoSphere DataStage jobs by using templates that incorporate major functional and design elements. This process uses the common security user model of InfoSphere Information Server and requires users to have job authoring privileges for InfoSphere DataStage to generate the jobs.

Before jobs can be generated, the mapping specification is run through a validation process. This process identifies inconsistencies and issues that might preclude the successful transformation of InfoSphere FastTrack map into an InfoSphere DataStage job template. Jobs can be generated even with validation errors. Errors are annotated in the generated jobs as comments. The InfoSphere DataStage developer must rectify them before the job is completed and functionally correct.

To generate a job, you must select an InfoSphere DataStage project folder where the generated job is to be written. You must also assign a unique name. If a unique name is not specified, the job generation process prompts you for permission to overwrite the existing job.

Each mapping specification can generate one job. Alternatively, multiple specifications can be combined sequentially or in parallel to generate a single job. When validation is enabled, you can receive messages that validate a filter or function expression. Warning or error messages are generated based on the information that you specify in the mapping specification.

Combining specifications can be done in the following ways:

- Mapping specification A to generate job A
- Mapping specification A, plus mapping specification B, processed sequentially in a single job C
- Mapping specification A, plus mapping specification B, processing in parallel streams in a single job C

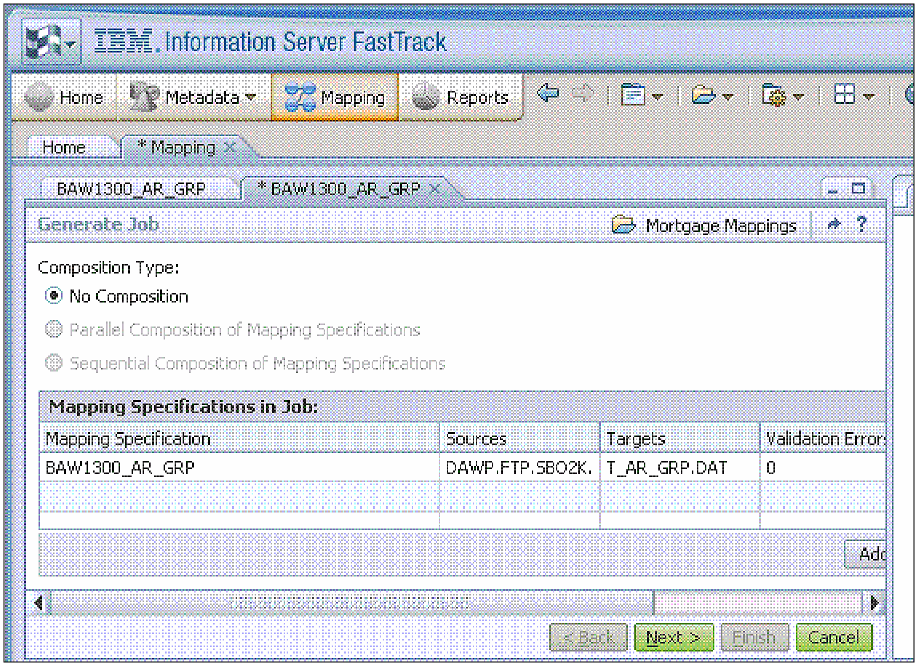
The completion of job generation results in an InfoSphere DataStage job that is ready for a developer to review and complete, after which the job can be deployed. Jobs can include shared containers, lookup stages, transformations, annotations, and more. Simple data integration jobs can be ready for immediate deployment with little or no additional work. More complex jobs can require substantial work to prepare them for deployment.

Lineage is tracked from specification to the deployed job. Generated jobs are automatically documented with specification requirements giving the developer most of the requirements needed to complete the jobs data flow.

When a mapping specification is complete, an InfoSphere DataStage job can be generated.

To generate a job, complete these steps:

1. On the Mapping page, select the mapping specification from which to generate the job. As shown in Figure 11-4, we select mapping specification **BAW1300_AR_GRP** from which to generate a job. Then click **Next**.



Generate Job

Composition Type:

● No Composition

○ Parallel Composition of Mapping Specifications

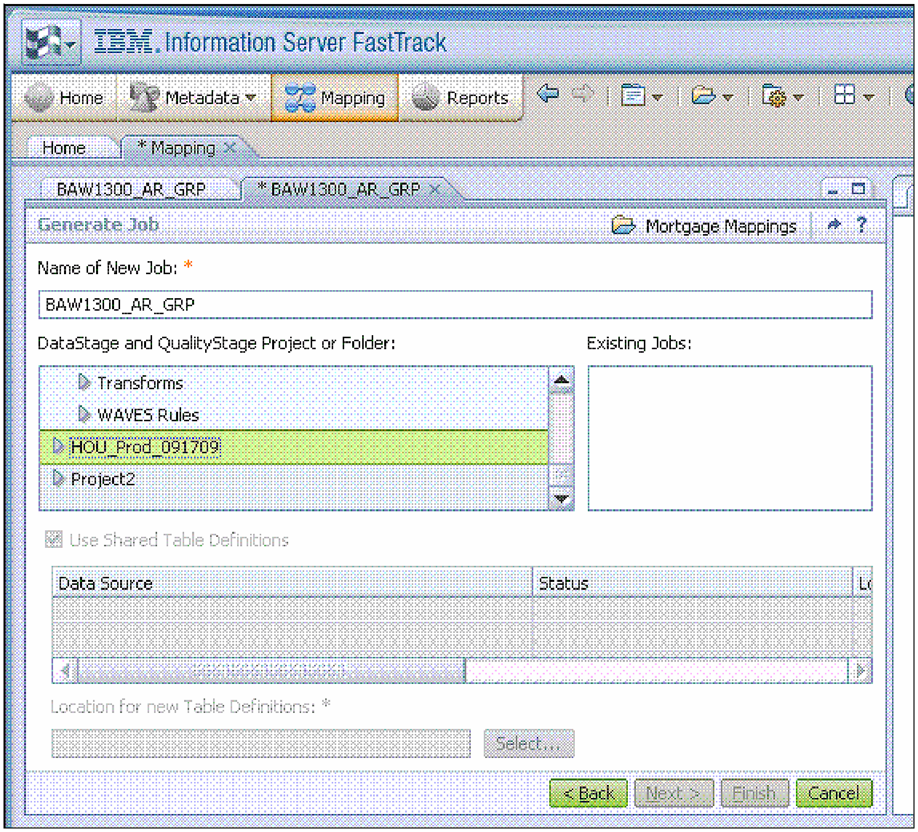
○ Sequential Composition of Mapping Specifications

Mapping Specifications in Job:

Mapping Specification	Sources	Targets	Validation Error
BAW1300_AR_GRP	DAWP.FTP.SBO2K.	T_AR_GRP.DAT	0

Figure 11-4 Generating a job

2. Select an InfoSphere DataStage project in which to have the generated job saved. As shown in Figure 11-5, we select the InfoSphere DataStage project **HOU_Prod_091709**, in which to save the generated job. Then **Next**.



Name of New Job: *

BAW1300_AR_GRP

DataStage and QualityStage Project or Folder:

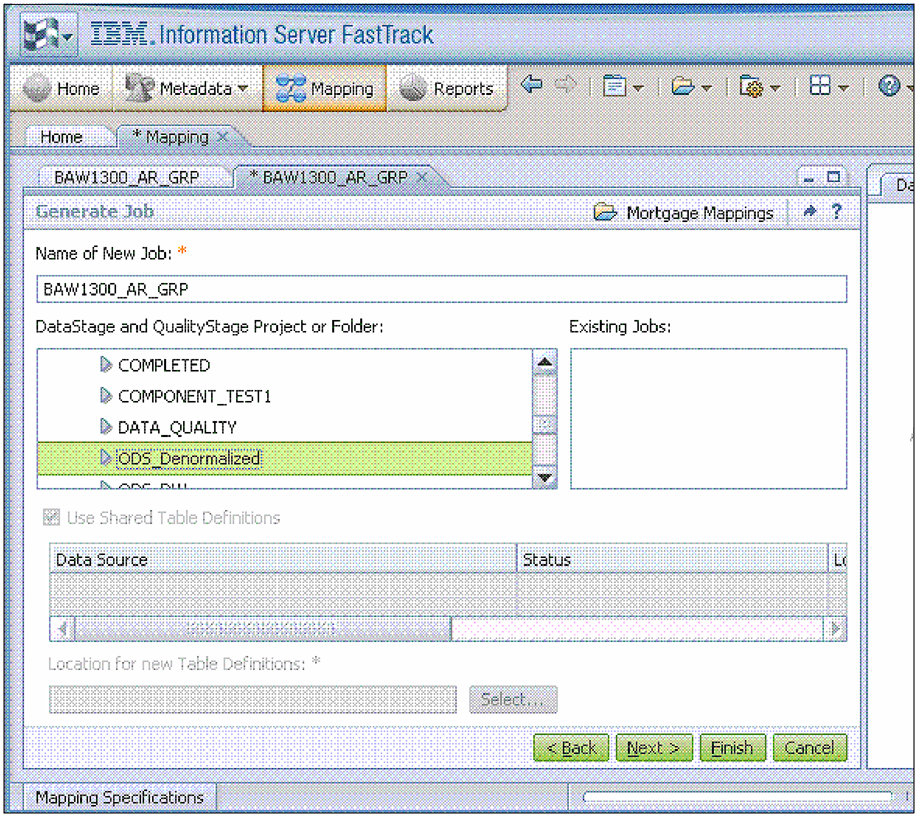
Existing Jobs:

☑ Use Shared Table Definitions

Location for new Table Definitions: *

Figure 11-5 Selecting the project in which a generated job is saved

3. Select a folder inside the project that you selected in the previous. As shown in Figure 11-6, we select the **ODS_Denormalized** folder. Then click **Finish**.



Name of New Job: *

BAW1300_AR_GRP

DataStage and QualityStage Project or Folder:

- COMPLETED
- COMPONENT_TEST1
- DATA_QUALITY
- ODS_Denormalized
- ODS_DW

Existing Jobs:

☒ Use Shared Table Definitions

Data Source	Status	Lo

Location for new Table Definitions: *

Figure 11-6 Selecting the folder within the project in which a generated job is saved

The job template is generated based on the information that you provided.

11.5 Metadata sharing (extension mappings)

InfoSphere FastTrack can help create *Lineage Extension Mappings*, which are data lineage objects. You use them to supplement the built-in data lineage features of InfoSphere Metadata Workbench with lineage elements that are not defined as part of an InfoSphere DataStage project. These elements are existing data integration processes that were developed with a third-generation development tool such as COBOL, C, or C++. You also use Lineage Extension Mappings to describe data flows that are implemented in database stored procedures or other data flows that are not implemented in InfoSphere DataStage.

An Extension Mapping is created by creating a basic mapping document that maps an existing data source column to an existing target column. Each column mapping can contain transformation logic that describes the processes that occur during the flow. The sum of these individual column mappings builds a picture of the functional logic of the data flow.

The sources and targets of the extension mapping are drawn from the existing shared metadata in the repository, eliminating the need to manually supply them in the extension mapping object. This approach can greatly reduce the number of erroneous column definitions that are introduced from manually generated extension mappings, resulting in a higher degree of accuracy. Because these mappings are functionally complete when they are exported, they seamlessly integrate into the existing data lineage that is produced by InfoSphere Metadata Workbench.

11.6 InfoSphere DataStage job design

Design metadata encompasses the details that define the logical steps of a job to move data from source to target. Everything related to a job at its lowest level resolves to program code. InfoSphere DataStage jobs are represented graphically in InfoSphere DataStage Designer (Figure 11-7).

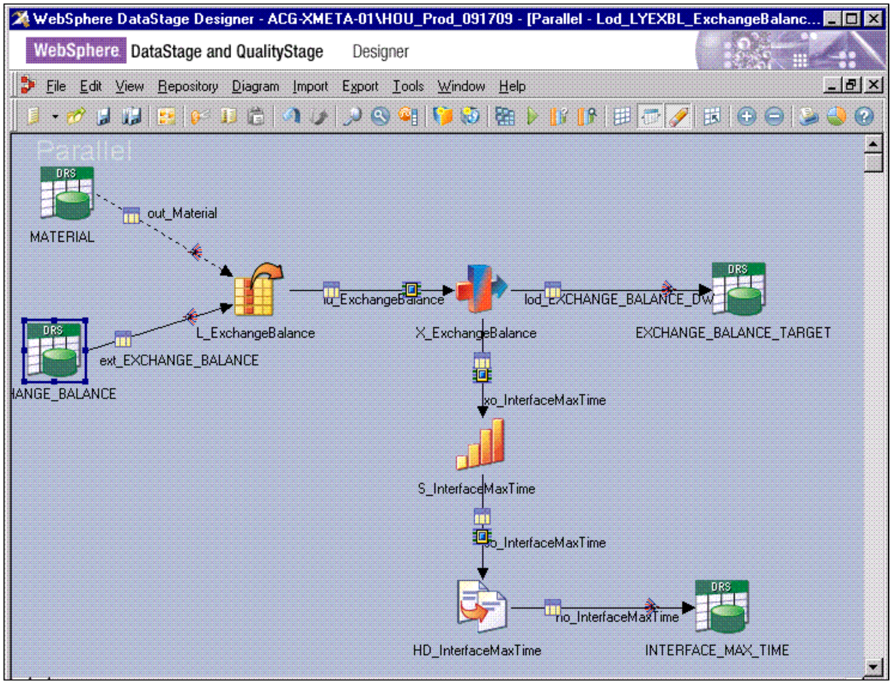
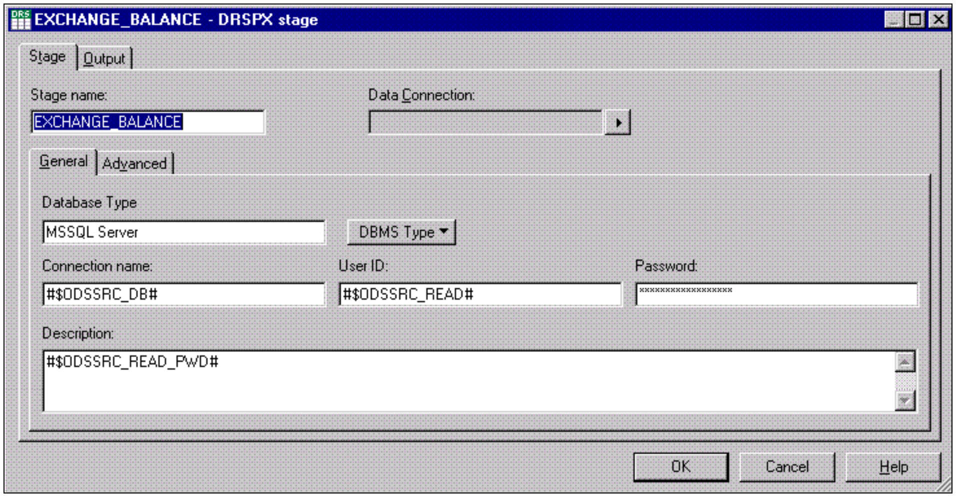


Figure 11-7 InfoSphere DataStage and InfoSphere QualityStage Designer

Each icon that you see in Figure 11-7 on page 386 is called a *stage*. A stage contains various properties, variables, program code, column mappings, and so on. The types of information that each stage contains depends on the stage type.

For example, a database stage, such as the one on the far left side of Figure 11-7 on page 386, contains database connection information and the generated SQL to read a specific table from the database. Figure 11-8 and Figure 11-9 on page 388 show a sample of the types of metadata that a database stage uses to connect and retrieve information from the database. The parameters with the number sign (#) are used in the property fields.



Stage name: EXCHANGE_BALANCE

Data Connection:

Database Type: MSSQL Server

Connection name: #$ODSSRC_DB#

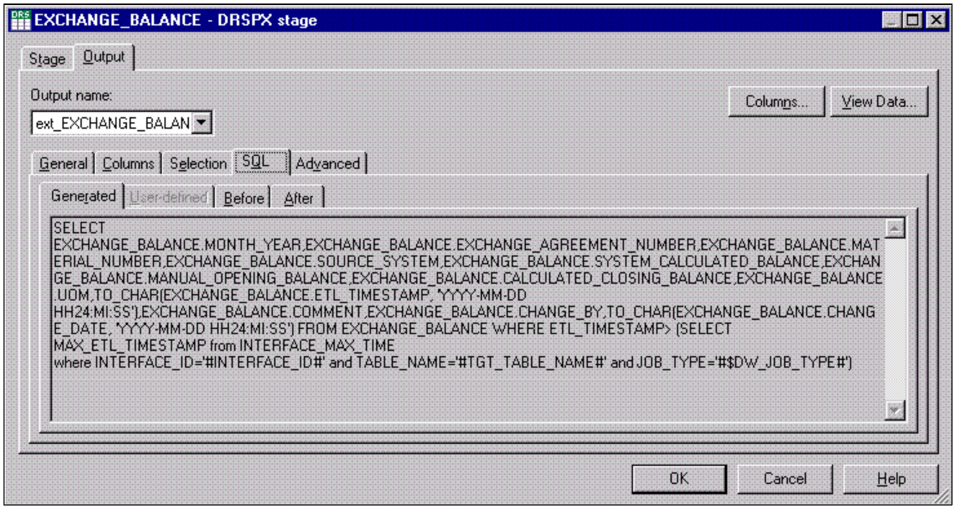
User ID: #$ODSSRC_READ#

Password: ******************

Description: #$ODSSRC_READ_PWD#

Figure 11-8 Sample InfoSphere DataStage general properties

Figure 11-9 shows a sample generated SQL by InfoSphere DataStage.



Output name: ext_EXCHANGE_BALAN

```
SELECT
EXCHANGE_BALANCE.MONTH_YEAR,EXCHANGE_BALANCE.EXCHANGE_AGREEMENT_NUMBER,EXCHANGE_BALANCE.MAT
ERIAL_NUMBER,EXCHANGE_BALANCE.SOURCE_SYSTEM,EXCHANGE_BALANCE.SYSTEM_CALCULATED_BALANCE,EXCHAN
GE_BALANCE.MANUAL_OPENING_BALANCE,EXCHANGE_BALANCE.CALCULATED_CLOSING_BALANCE,EXCHANGE_BALANCE
.UOM,TO_CHAR(EXCHANGE_BALANCE.ETL_TIMESTAMP, 'YYYY-MM-DD
HH24:MI:SS'),EXCHANGE_BALANCE.COMMENT,EXCHANGE_BALANCE.CHANGE_BY,TO_CHAR(EXCHANGE_BALANCE.CHANG
E_DATE, 'YYYY-MM-DD HH24:MI:SS') FROM EXCHANGE_BALANCE WHERE ETL_TIMESTAMP> (SELECT
MAX_ETL_TIMESTAMP from INTERFACE_MAX_TIME
where INTERFACE_ID='#INTERFACE_ID#' and TABLE_NAME='#TGT_TABLE_NAME#' and JOB_TYPE='#$DW_JOB_TYPE#')
```

Figure 11-9 Sample InfoSphere DataStage generated SQL

Job design details impact how seamlessly your data lineage and impact analysis perform.

11.6.1 Job design details

Jobs that follow good metadata management practices facilitate the most robust data lineage information. Jobs can be constructed in a way that requires little or no manual manipulation of the metadata. With the right design, the automated services of InfoSphere Metadata Workbench can trace through the entire data lineage to produce data lineage and impact analysis reports.

In some cases, InfoSphere DataStage jobs were developed before these design considerations were known. Where jobs do not follow best practices, use the guidance provided in this section so that InfoSphere Metadata Workbench can automatically link these jobs together from a data lineage and impact analysis perspective. The job elements that require attention are data connectors, job parameters, SQL statement, and file names.

Database connectors

InfoSphere DataStage has numerous database connector types that can connect to various data sources. A data flow typically consists of multiple jobs. The same connector type must be used consistently throughout the data flow.

For example, if a developer chooses to use the DB2 connector to read and or write data in a job, any preceding and subsequent jobs in the flow must also use the DB2 connector. A project might have been started by using the Open Database Connectivity (ODBC) connector to access the data. At some point in the development process, a design decision was made to use the DB2 connector for performance reasons. The point at which the ODBC connector is substituted for the DB2 connector keeps the automated services from resolving the data lineage automatically.

Where a portion of the jobs uses a mix of connector types, you can use the connector migrator to automate these changes. This tool is supplied with InfoSphere Information Server.

Job parameters

Parameters or parameter sets must use consistent names for design time lineage to function. At run time, all values will be resolved. However, for design time lineage and impact analysis to function, parameter names and default parameter values must match between jobs, which is easily accomplished today by using parameter sets. For jobs that were developed before parameter sets were available, you might want to recode the jobs.

SQL

The automated services of InfoSphere Metadata Workbench have difficulty parsing vendor-specific SQL extensions. When custom SQL is supplied by the job developer, use care to refrain from the use of these extensions where possible. For performance reasons, in all cases, this approach might not be possible.

If the use of vendor-specific extensions is required, manual linkage of data lineage is required.

File names

For job flows that use sequential files, the imported file name and file element name must match the full path of the file as defined within the connector. Without this connection, the shared file definition cannot be linked to the jobs data flow. Because they are normally defined by job parameters, the previous rules might typically apply.

11.7 Shared metadata

Shared metadata objects describe the sources and targets of a data flow and become the fundamental buildings blocks of metadata reporting in regard to the flow of data. Without this shared metadata, there are no connective elements to link job flows.

Some shared metadata is produced as a by-product of activities such as importing metadata for analysis work in InfoSphere Information Analyzer. Other types, such as physical data models, schemes, or flat or complex file metadata, must be imported by the appropriate mechanism. Other disparate metadata that is crucial to your job designs and ultimately your metadata reporting must be configured to the point where it can be automatically identified and related by the automated services of InfoSphere Metadata Workbench. Other metadata that describes the structure of sequential or complex flat file must be created as shared metadata from InfoSphere DataStage jobs.

11.7.1 Shared metadata that must be created

InfoSphere DataStage table definitions can be turned into shared tables and files in the InfoSphere Information Server metadata repository by using the Shared Metadata Creation Wizard. With this wizard, the user to take existing InfoSphere DataStage table definitions and push them into the metadata repository so that they can be used as start or end points for data lineage or impact analysis.

This pushing step is not necessary when importing metadata by using the import/export manager or common connectors. However this step is necessary when sequential files or complex flat files must be represented as shared metadata.

The only guaranteed accurate representation of this metadata is inside InfoSphere DataStage jobs. Because InfoSphere DataStage allows for the manipulation of this metadata on the links inside of jobs, they never need to be saved as InfoSphere DataStage table definitions. InfoSphere DataStage developers commonly change these file definitions where they are used and not in the common InfoSphere DataStage table definition. With this practice, InfoSphere DataStage table definitions can become out of sync with the columns defined on the links in an InfoSphere DataStage job.

To ensure that the shared metadata that is persisted in the repository matches the actual columns, create new table definitions from the links in an InfoSphere DataStage job. These new table definitions can then be used to create the equivalent shared metadata in the InfoSphere Information Server metadata repository. This approach also allows for the correct locators to be generated in the InfoSphere DataStage job, making it easier for InfoSphere Metadata Workbench to generate its lineage and impact analysis reports.

11.8 Operational metadata

Operational metadata describes the events and processes that occur and the objects that are affected when you run an InfoSphere DataStage or InfoSphere QualityStage job. After a job is run, the statistics of the job run (operational metadata) can be imported into the metadata repository. This metadata can then be used by InfoSphere Metadata Workbench to build operational metadata reports. It can also assist in the generation of data lineage and impact analysis reports.

The following types of information are captured as operational metadata:

- The number of rows that were read, written, or referenced
- The tables or files that were read from, written to, or referenced
- The stages and links that were used
- The project in which the job is contained
- Any parameters that are used by the job
- The invocation ID of the job
- Any notes about running the job

11.8.1 Creating operational metadata

Operational metadata is not generated by default. The service that creates it must be enabled by the InfoSphere DataStage administrator in the InfoSphere DataStage Administrator application.

The operational metadata for each job run is stored in a series of XML files. These XML files contain all the information previously mentioned that the jobs have since the last time operational metadata was imported into the metadata repository. This storage of information is accomplished through a series of scripts and is straight forward when the metadata repository used to view this metadata is the same that was used to execute the InfoSphere DataStage jobs. Job run reports are available by using InfoSphere Metadata Workbench.

11.9 Conclusion

In conclusion, this chapter explained InfoSphere FastTrack mapping and InfoSphere DataStage job generation. It addressed considerations to ensure automated generation of data lineage and impact analysis reports for metadata management.

Chapter 12, “Enterprise reports and lineage generation” on page 393, explains how to generate lineage and impact analysis reports by using InfoSphere Metadata Workbench for metadata management.

12

Enterprise reports and lineage generation

Metadata management means understanding where your reporting and analysis data come from and whether these data are accurate and reliable. Thus, when deriving business intelligence (BI) reports, you must validate the quality and correctness of the data to ensure that the report results are accurate and reliable.

IBM InfoSphere Metadata Workbench, one of the IBM InfoSphere Information Server product modules, provides metadata-based reports to ensure that quality data is also the correct data from the correct system. This chapter introduces the tasks and processes that are required to deliver the business and regulatory requirements for reporting and analysis.

This chapter includes the following sections:

- Lineage administration
- Support for InfoSphere DataStage and InfoSphere QualityStage jobs
- Support for external processes
- Support for InfoSphere FastTrack mapping specifications
- Configuring business lineage
- Search and display
- Querying and reporting
- Conclusion

12.1 Lineage administration

With IBM InfoSphere Metadata Workbench, you can view, understand, and analyze the content stored within InfoSphere Information Server repository. Whether it is InfoSphere DataStage jobs, external mapping documents, quality specifications, database systems, or glossary terms and labels, InfoSphere Metadata Workbench delivers a singular portal of information to satisfy the business requirements and promote governance objectives.

The key to understanding information is integration. How is information integrated between InfoSphere DataStage jobs, mappings, and database systems?

The analysis services of InfoSphere Metadata Workbench provide the advanced algorithms and data mining techniques in determining existing relationships that have yet to be formalized. These services provide the added value of InfoSphere Metadata Workbench and its reporting, querying, and analysis capabilities, which are delivered as in data lineage report.

InfoSphere Metadata Workbench acts as a flexible lens into the InfoSphere Information Server repository, providing meaningful views of asset information and its relevant associations. For example, when viewing an IBM InfoSphere DataStage and InfoSphere QualityStage job, you can immediately understand the data source the job is reading from or writing to the last execution time of the job. Further, by selecting the data source, you can appreciate the meaning, owner, data rule, policy, and usage of the asset.

In addition to providing details about information and their relationships, InfoSphere Metadata Workbench provides a wide-angle analysis view. It displays the data lineage between the data sources through their corresponding jobs, processes, and dependent BI reports.

This section highlights the following reports, which are available through InfoSphere Metadata Workbench:

- Business lineage
- Data lineage
- Impact analysis reports

12.1.1 Business lineage

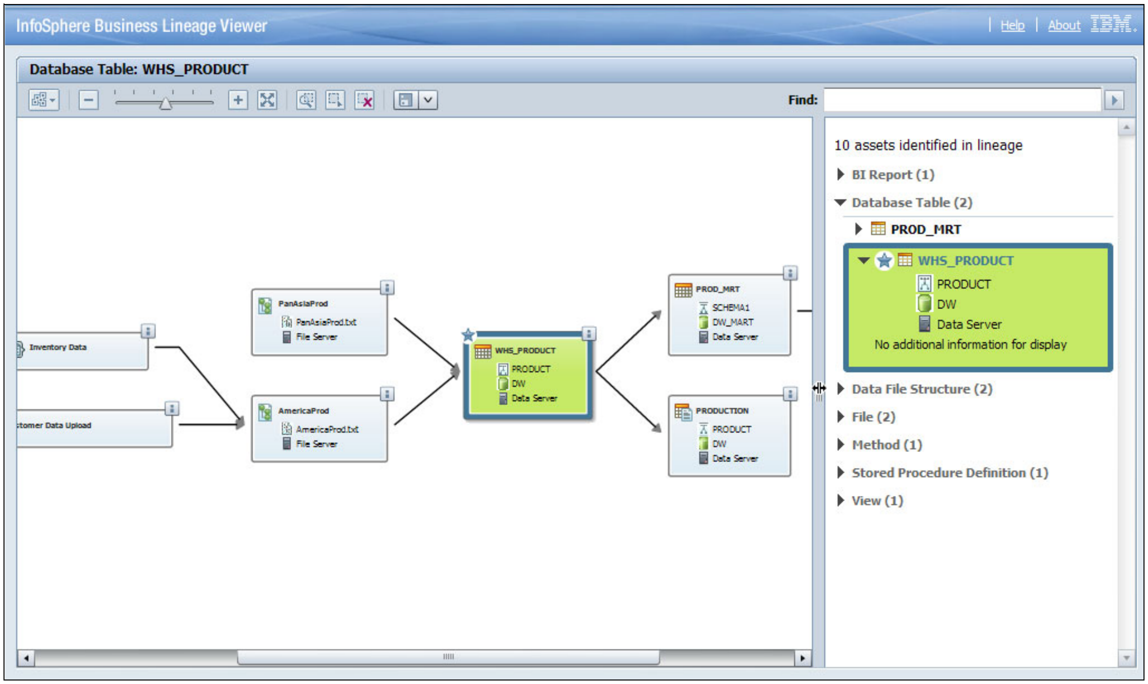
Business lineage analysis reports provide a visual overview of how information flows within the information integration solution, from the consuming BI reports to the systems of records displaying only the relevant data sources.

Business lineage incorporates glossary terms and stewards, affording users an understanding of the meaning of the information, in addition to its defined business or alias name.

A governance czar who inspects weekly customer bank deposits report might want to validate the enterprise definition and application to identify high value customers and trace the origin of information for such a report. The governance czar can satisfy their understanding with such a business lineage report.

Business lineage can be launched from IBM InfoSphere Business Glossary or InfoSphere Metadata Workbench.

Figure 12-1 shows the Business Lineage Viewer.



10 assets identified in lineage

- BI Report (1)
- Database Table (2)
  - PROD_MRT
  - WHS_PRODUCT
    - PRODUCT
    - DW
    - Data Server
    - No additional information for display
- Data File Structure (2)
- File (2)
- Method (1)
- Stored Procedure Definition (1)
- View (1)

Figure 12-1 View of business lineage

12.1.2 Data lineage

Data lineage analysis reports help to visualize how information flows within the integrated solution, from the consuming BI reports through data sources, jobs, or processes to the systems of record, displaying technical data flow.

Data lineage incorporates InfoSphere DataStage and InfoSphere QualityStage jobs, IBM InfoSphere FastTrack mapping specifications, and mapping documents authored within InfoSphere Metadata Workbench. Data lineage analysis reports further allow traceability for individual data columns, displaying transformation logic and runtime statistics.

A data analyst who views the data results within a data storage system might want to understand the originating systems of record or consuming BI reports and systems. Such a report satisfies the stated objectives for auditing and governance of data transformation and change.

Data lineage can be launched from InfoSphere Metadata Workbench, as shown in Figure 12-2.

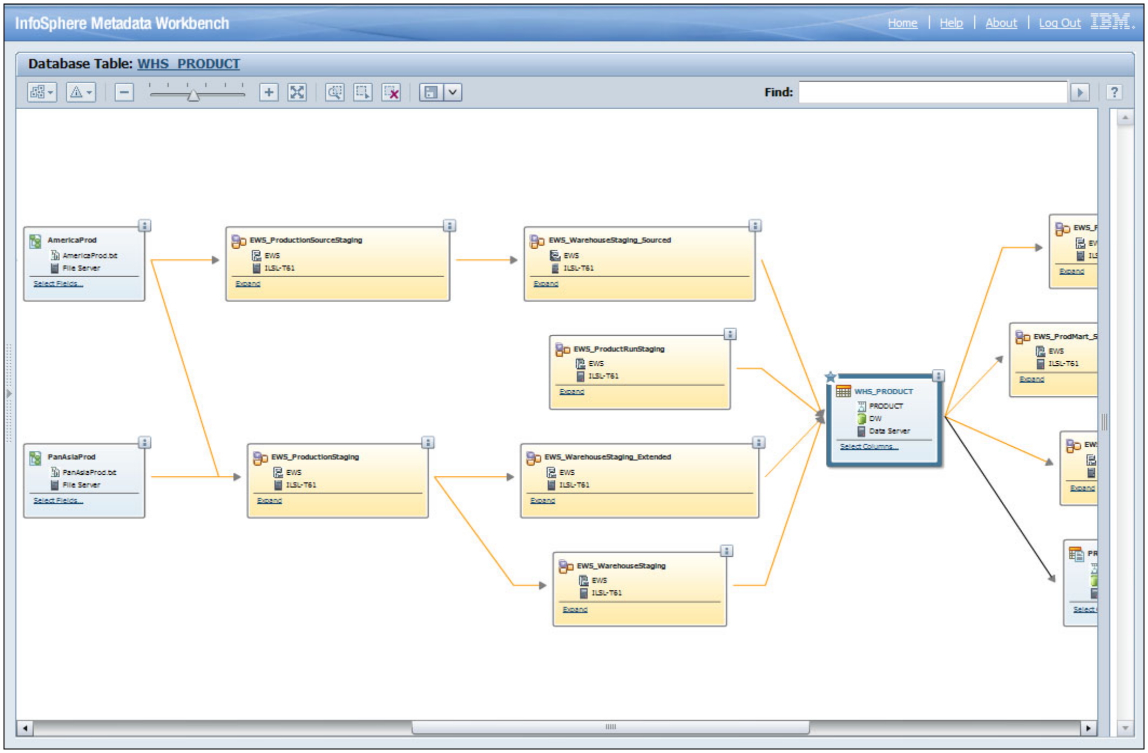


Figure 12-2 View of data lineage

12.1.3 Impact analysis

Impact analysis reports detail the dependencies of information or the impact of a change within a data source or InfoSphere DataStage and InfoSphere QualityStage job. Impact analysis incorporates InfoSphere DataStage and InfoSphere QualityStage jobs, InfoSphere Information Services Director operations, data sources, and InfoSphere FastTrack mapping specifications. It details the usage and dependencies between such assets.

A developer who is required to alter the structure of a database table or update an InfoSphere DataStage and InfoSphere QualityStage job to accommodate changes invokes this report to understand the impact of this change. Downstream InfoSphere DataStage and InfoSphere QualityStage jobs, data sources, or BI reports might require updating to ensure continued data flow as depicted within the integrated solution. A developer can further satisfy their requirement with the ability to print a report of the impacted assets and their assigned data stewards.

Impact analysis can be launched from InfoSphere Metadata Workbench, as shown in Figure 12-3.

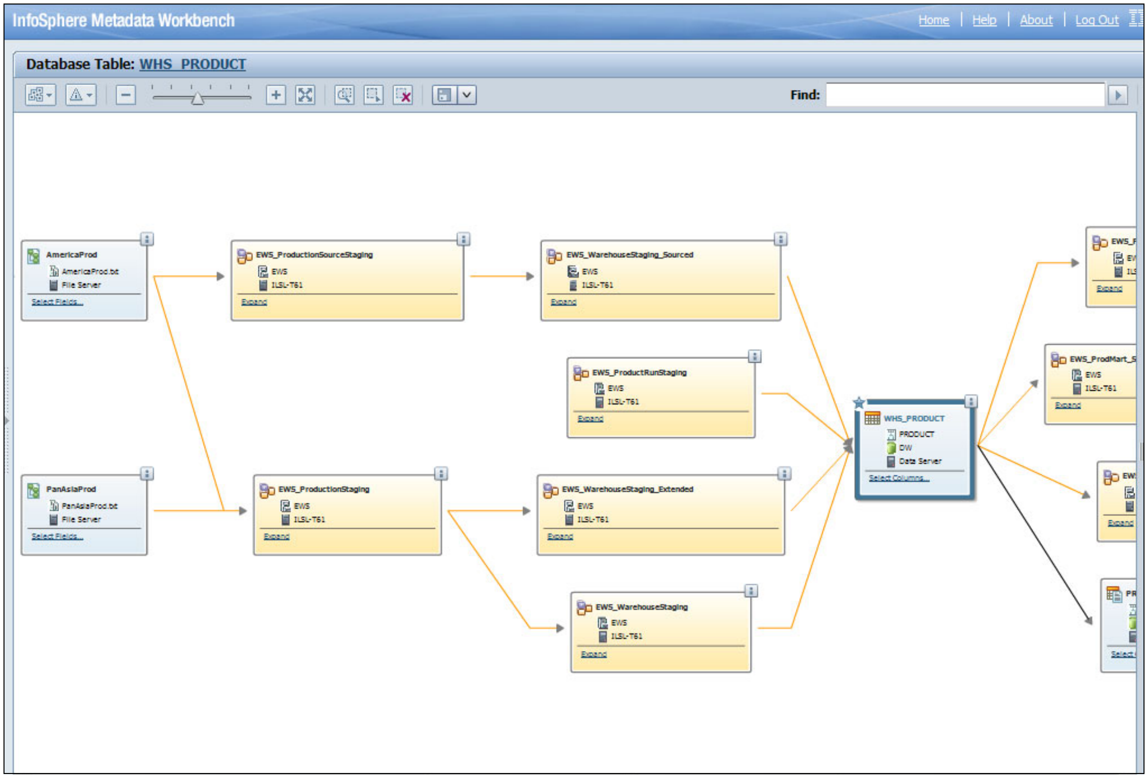


Figure 12-3 View of impact analysis

12.2 Support for InfoSphere DataStage and InfoSphere QualityStage jobs

The analysis services for IBM InfoSphere Metadata Workbench can analyze and infer relationships and links between InfoSphere DataStage and InfoSphere QualityStage jobs and the data sources that those jobs are reading from or writing to.

When parameters are used, the relationships can only be known from the job at run time. You can use that runtime information and blend with the design information to provide the most robust and accurate representation.

The analysis services form the backbone for data lineage, business lineage, and impact analysis reports.

12.2.1 Design lineage

Design metadata is the information that is included within an InfoSphere DataStage and InfoSphere QualityStage job when it is developed. Such jobs can include environment parameters or job parameters in addition to the specific property values for each declared stage.

As a developer creates a job, this person builds the intended data flow from a particular source to a given target within that job. This flow includes any data lookup, filtering, or logging, as shown in Figure 12-4 on page 399.

InfoSphere Metadata Workbench analysis service investigates the design metadata to build relationships and links between the given InfoSphere DataStage and InfoSphere QualityStage job components and stages. Other information that the analysis service might use for the investigation include the source data (that is, the imported databases and data files) and job assets found within the InfoSphere metadata repository.

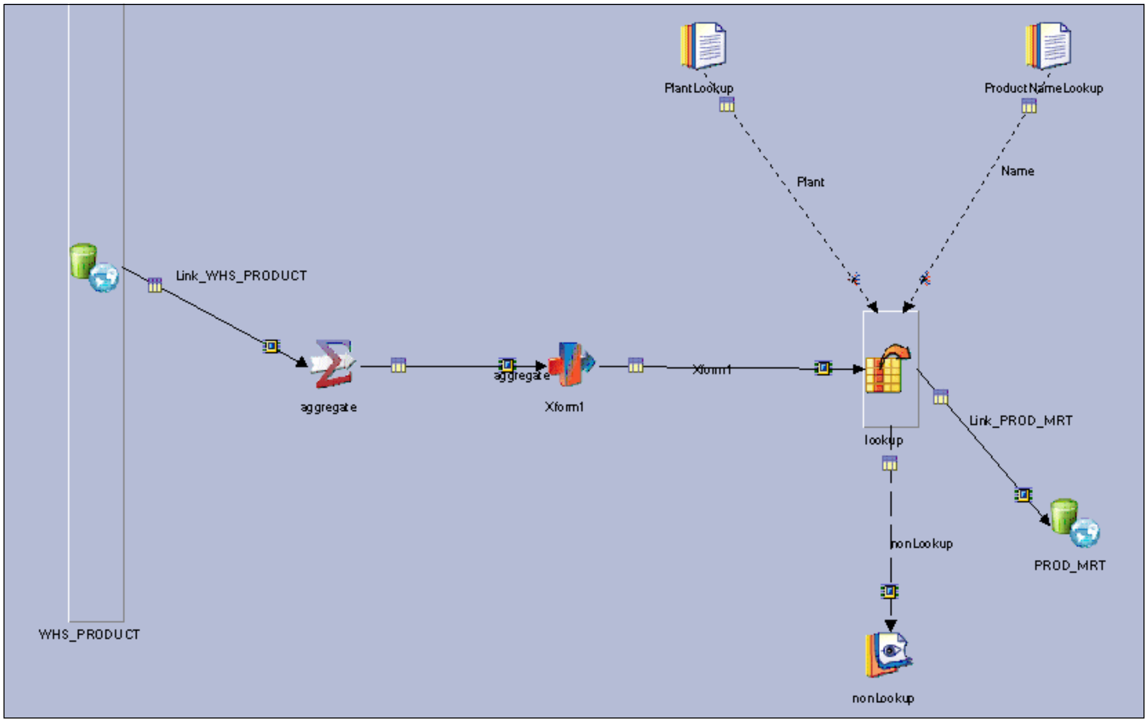
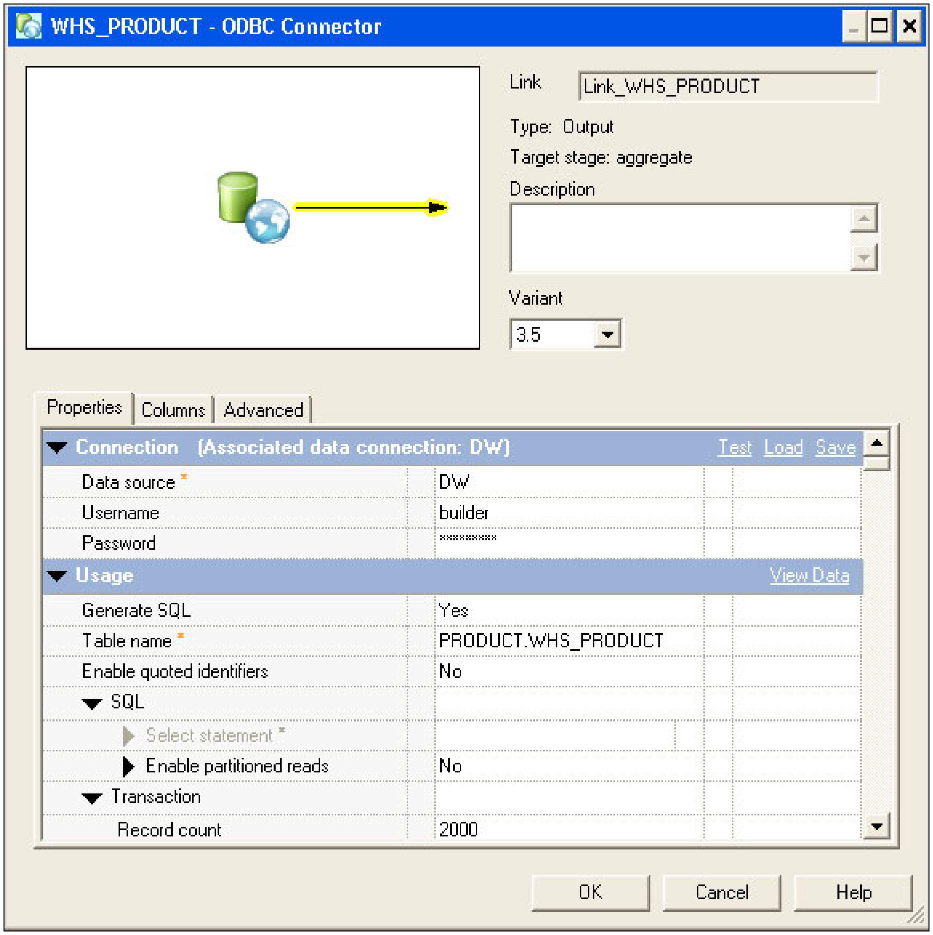


Figure 12-4 InfoSphere DataStage job design view

Design lineage and analysis refers to the projected flow of information across different InfoSphere DataStage and InfoSphere QualityStage jobs where the input and output of such jobs share a common source. The source does not need to reflect a physical data source that was previously imported into InfoSphere Information Server repository, but rather might reflect a reference based on the design information of the job. The parameters and connectors defined on the stage are used in determining such commonality. A projected InfoSphere Metadata Workbench design flow transverses InfoSphere DataStage and InfoSphere QualityStage jobs.

The design parameters (Figure 12-5) include the database or server connection name, database schema, database table, Structured Query Language (SQL) command, data file name, and data file location. The design parameters differ according to the job stage.



WHS_PRODUCT - ODBC Connector

Link: Link_WHS_PRODUCT

Type: Output

Target stage: aggregate

Description

Variant: 3.5

Properties | Columns | Advanced

Connection (Associated data connection: DW)

Data source *	DW
Username	builder
Password	*********

Usage

Generate SQL	Yes
Table name *	PRODUCT.WHS_PRODUCT
Enable quoted identifiers	No
SQL	
Select statement *	
Enable partitioned reads	No
Transaction	
Record count	2000

Figure 12-5 Job design parameters for reading database table information

The process for capturing design lineage includes the following tasks:

1. Importing the environment variables
2. Mapping the database aliases
3. Starting Manage Lineage Services
4. Starting manual binding
5. Maintenance and life cycle

Each of these tasks is explored in the following sections, except for step 5 regarding maintenance and the life cycle.

Importing the environment variables

InfoSphere DataStage and InfoSphere QualityStage jobs often include variables as references for various required parameters, including database name, server name, user name, and file locations. The use of such variables reduces error and promotes data reuse during job development. This usage also benefits the analysis services of InfoSphere Metadata Workbench.

Such variables can be defined locally within an InfoSphere DataStage and InfoSphere QualityStage job. They can also be defined within an InfoSphere DataStage and InfoSphere QualityStage project to be shared across multiple jobs. With the InfoSphere DataStage Administrator tool, global user and system variables, known as *environment variables*, can be declared for a given project. These variables are stored in the InfoSphere DataStage project file system. Such variables are beneficial in that a singular reference to a database storage system can be defined, assuring consistency and quality results.

InfoSphere Metadata Workbench analysis services require the variable names and values to reconcile and link the InfoSphere DataStage and InfoSphere QualityStage job with the referenced data sources.

The IBM InfoSphere Metadata Workbench includes a procedure that reads the environment variables that are defined for InfoSphere DataStage and InfoSphere QualityStage projects. This procedure then stores the values within the InfoSphere Information Server repository for use by the InfoSphere Metadata Workbench analysis services. The procedure is installed by default when you install InfoSphere Metadata Workbench. You must run this procedure for all InfoSphere DataStage and InfoSphere QualityStage projects where environment variables have been included.

The procedure is invoked as a shell script (the `ProcessEnvVariable.sh` shell script) or Microsoft Windows batch file and requires the passing of parameters when it is run. You must schedule the procedure to run regularly to ensure availability of the added or modified variables.

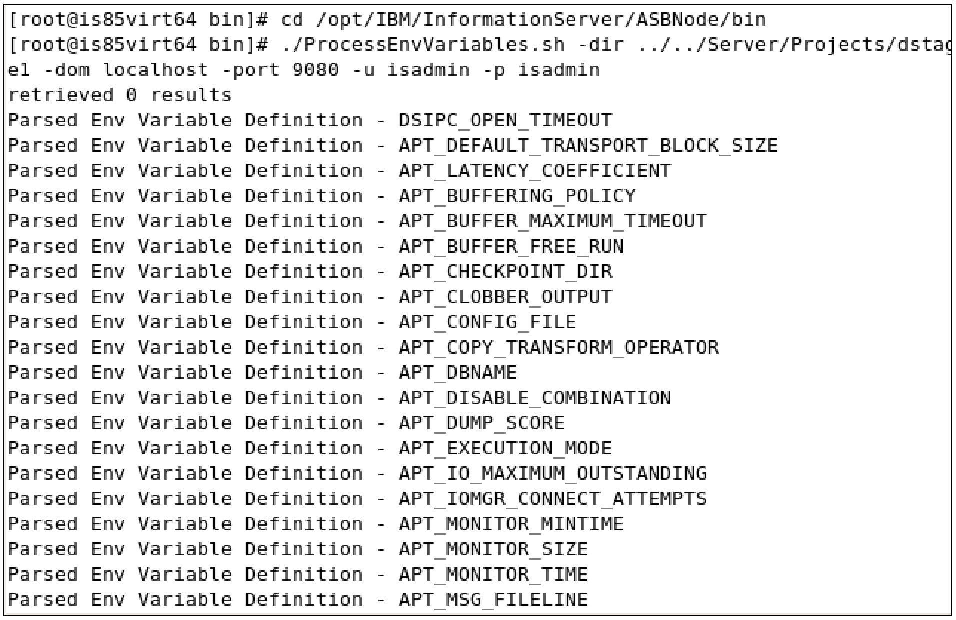
Running the environment variables import script

To run the import script from a Linux or UNIX shell command, use the script shown in Example 12-1.

Example 12-1 Environment variables procedure script

```
cd /opt/ibm/InformationServer/ASBNode/bin ./ProcessEnvVariable.sh -dir
../../Server/Projects/myProject -dom localhost -port 9080 -u username -p
password
```

Figure 12-6 shows the results of running the script.



```
[root@is85virt64 bin]# cd /opt/IBM/InformationServer/ASBNode/bin
[root@is85virt64 bin]# ./ProcessEnvVariables.sh -dir ../../Server/Projects/dstag
e1 -dom localhost -port 9080 -u isadmin -p isadmin
retrieved 0 results
Parsed Env Variable Definition - DSIPC_OPEN_TIMEOUT
Parsed Env Variable Definition - APT_DEFAULT_TRANSPORT_BLOCK_SIZE
Parsed Env Variable Definition - APT_LATENCY_COEFFICIENT
Parsed Env Variable Definition - APT_BUFFERING_POLICY
Parsed Env Variable Definition - APT_BUFFER_MAXIMUM_TIMEOUT
Parsed Env Variable Definition - APT_BUFFER_FREE_RUN
Parsed Env Variable Definition - APT_CHECKPOINT_DIR
Parsed Env Variable Definition - APT_CLOBBER_OUTPUT
Parsed Env Variable Definition - APT_CONFIG_FILE
Parsed Env Variable Definition - APT_COPY_TRANSFORM_OPERATOR
Parsed Env Variable Definition - APT_DBNAME
Parsed Env Variable Definition - APT_DISABLE_COMBINATION
Parsed Env Variable Definition - APT_DUMP_SCORE
Parsed Env Variable Definition - APT_EXECUTION_MODE
Parsed Env Variable Definition - APT_IO_MAXIMUM_OUTSTANDING
Parsed Env Variable Definition - APT_IOMGR_CONNECT_ATTEMPTS
Parsed Env Variable Definition - APT_MONITOR_MINTIME
Parsed Env Variable Definition - APT_MONITOR_SIZE
Parsed Env Variable Definition - APT_MONITOR_TIME
Parsed Env Variable Definition - APT_MSG_FILELINE
```

Figure 12-6 Running the import environment variables script

Parameters of the environment variables import script

The environment variables import script uses the following parameters:

Project Folder Name The full name of the InfoSphere DataStage project folder. The default location of the project folder is `/Server/Projects` within the installation directories of the InfoSphere Information Server domain tier. This folder contains individual subfolders for all InfoSphere DataStage and InfoSphere QualityStage projects that have been defined within this domain. The parameter must reference the name of the project folder and its complete or relative path.

Host The computer name or address where WebSphere Application Server was installed. This information can also be taken from the URL of InfoSphere Information Server Web Console.

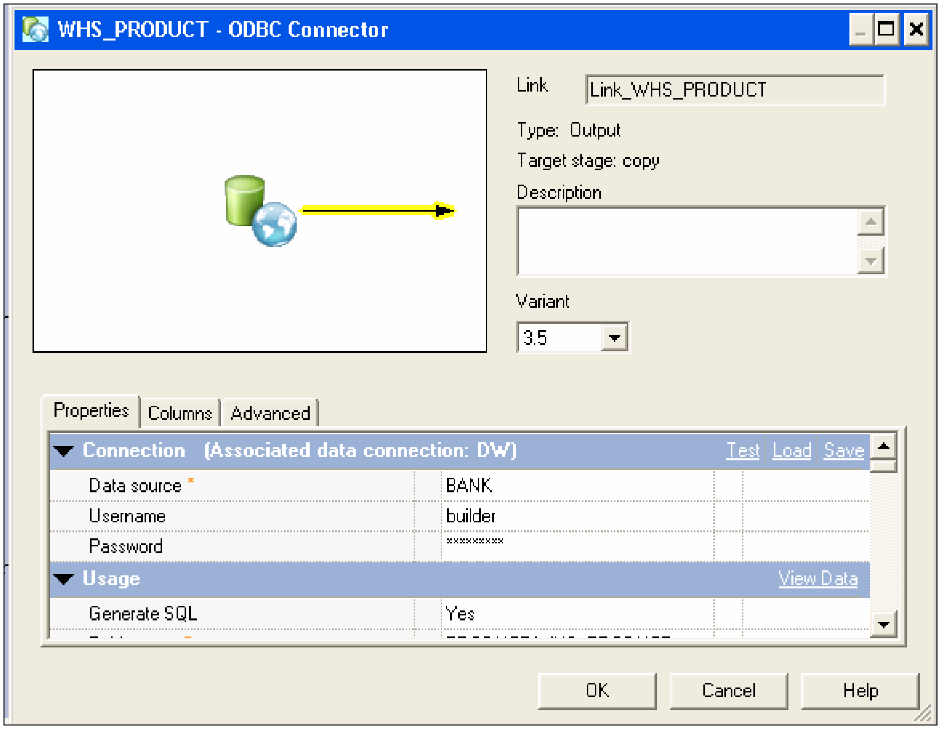
Port The communication port with which the host is communicating. This information can also be taken from the URL of InfoSphere Information Server Web Console.

Username An InfoSphere Information Server user, defined within InfoSphere Information Server Web console. The user requires the suite role.

Password The password credentials for the user.

Mapping the database aliases

InfoSphere DataStage and InfoSphere QualityStage jobs define data connections, such as Open Database Connectivity (ODBC) connectors, to access database storage systems when reading or writing information. Figure 12-7 shows an example for an ODBC connector.



WHS_PRODUCT - ODBC Connector

Link: Link_WHS_PRODUCT

Type: Output

Target stage: copy

Description

Variant: 3.5

Connection (Associated data connection: DW)

Data source *	BANK
Username	builder
Password	*********

Usage

Generate SQL	Yes

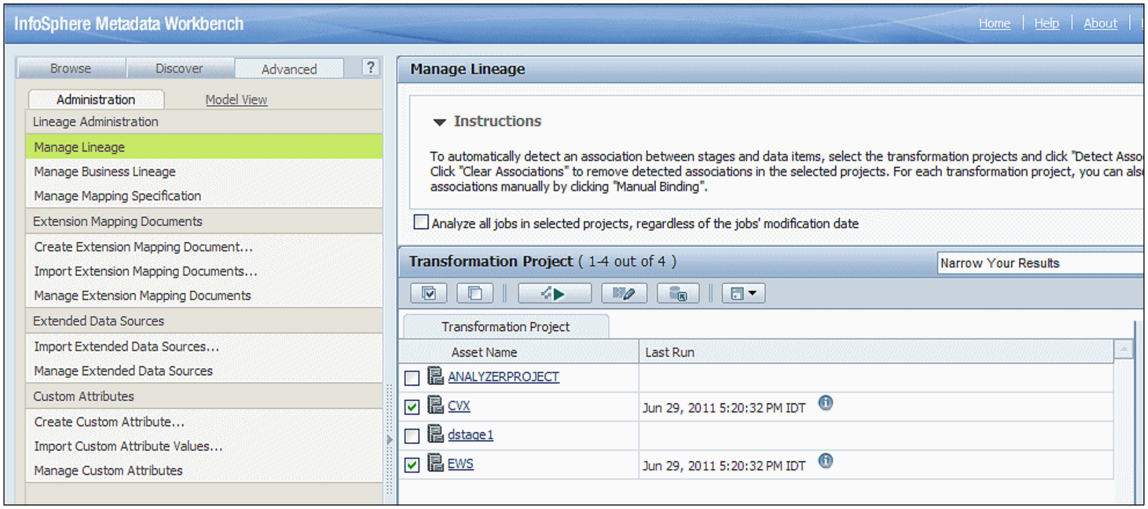
Figure 12-7 Sample InfoSphere DataStage job referencing an ODBC data source

A database Alias references data connections and maps those connectors to the represented host system and database that is imported into InfoSphere Information Server. The IBM InfoSphere Metadata Workbench uses such references in determining the links and associations between InfoSphere DataStage and InfoSphere QualityStage jobs and the data sources.

Database aliases are mapped within InfoSphere Metadata Workbench.

To create database alias mapping, complete these steps:

1. From the left navigation pane, on the **Advanced** tab, click **Manage Lineage**.
2. In the Manage Lineage pane (Figure 12-8), click the **Database Alias Mapping** icon from the toolbar.


InfoSphere Metadata Workbench
Home | Help | About |
Browse
Discover
Advanced
?
Administration
Model View
Lineage Administration
Manage Lineage
Manage Business Lineage
Manage Mapping Specification
Extension Mapping Documents
Create Extension Mapping Document...
Import Extension Mapping Documents...
Manage Extension Mapping Documents
Extended Data Sources
Import Extended Data Sources...
Manage Extended Data Sources
Custom Attributes
Create Custom Attribute...
Import Custom Attribute Values...
Manage Custom Attributes
Manage Lineage
Instructions

To automatically detect an association between stages and data items, select the transformation projects and click "Detect Asso
Click "Clear Associations" to remove detected associations in the selected projects. For each transformation project, you can als
associations manually by clicking "Manual Binding".

□ Analyze all jobs in selected projects, regardless of the jobs' modification date

Transformation Project (1-4 out of 4)

Narrow Your Results

Transformation Project

	Asset Name	Last Run
□	ANALYZERPROJECT	
☑	CVX	Jun 29, 2011 5:20:32 PM IDT
□	dstage1	
☑	EWS	Jun 29, 2011 5:20:32 PM IDT

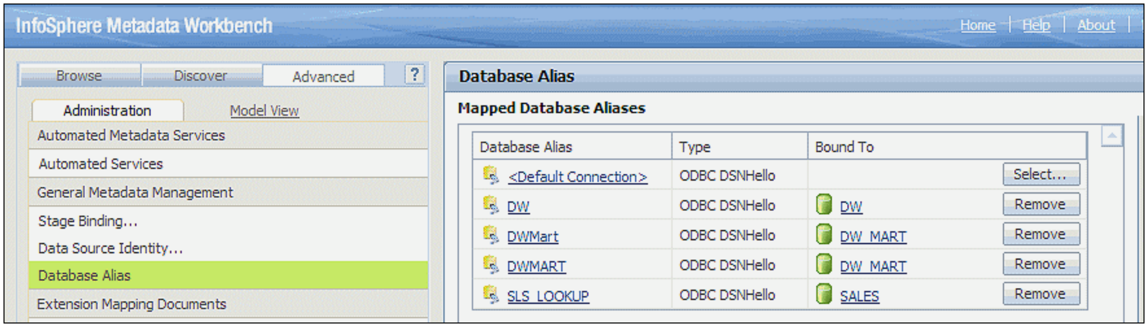
Figure 12-8 Manage Lineage pane

3. In the Manage Database Alias pane, click **Add** to browse and select a corresponding database asset to map to the selected alias.

4. In the Database Alias pane (Figure 12-9; where you can select a database), complete these steps:

   a. Enter the name, or partial name, of the database to be mapped to the alias. Click **Find** to return a list of matched results.

   b. Select the requested database, and click **Select** to assign the database to the alias.



Mapped Database Aliases

Database Alias	Type	Bound To	
<Default Connection>	ODBC DSNHello		Select...
DW	ODBC DSNHello	DW	Remove
DWMart	ODBC DSNHello	DW_MART	Remove
DWMART	ODBC DSNHello	DW_MART	Remove
SLS_LOOKUP	ODBC DSNHello	SALES	Remove

Figure 12-9 Database Alias pane

5. Click **Save** to complete the process.

The IBM InfoSphere Metadata Workbench includes such relationships between the InfoSphere DataStage and InfoSphere QualityStage jobs and database tables when rendering data lineage and impact analysis reports.

Starting Manage Lineage Services

Manage Lineage Services is a set of analyses that scan and read the metadata from all InfoSphere DataStage and InfoSphere QualityStage jobs for the selected projects. The service attempts to link the input and output stages to the previously imported data source assets, database tables, data file elements, or other InfoSphere DataStage jobs. A link is created when the property or parameter values of the stage match the data source name. Further columns defined within the stage are matched to database fields or data file attributes when the column names match.

With InfoSphere DataStage jobs, you can generate the default SQL or define custom SQL statements for reading and writing data sources. The analysis service parses all SQL statements to extract database schema, owner, table, and column information. In some cases, it is possible that the analysis service is unable to parse SQL statements. In such cases, InfoSphere Metadata Workbench offers manual binding services.

Database aliases referenced within InfoSphere DataStage jobs must previously have been mapped by using the Database Alias mapping service of InfoSphere Metadata Workbench. These aliases are used when defining a connection between an InfoSphere DataStage job and a database system, such as through an ODBC connection. In such cases, the connection defined on the job is a reference to the physical database table that has been imported into InfoSphere Information Server.

Cross-job data lineage is accomplished when the input of a given job is matched to the output of another job. In some cases, the jobs are linked to a previously imported data source. When such a data source has not been imported, these jobs are linked directly, as shown in Figure 12-10.

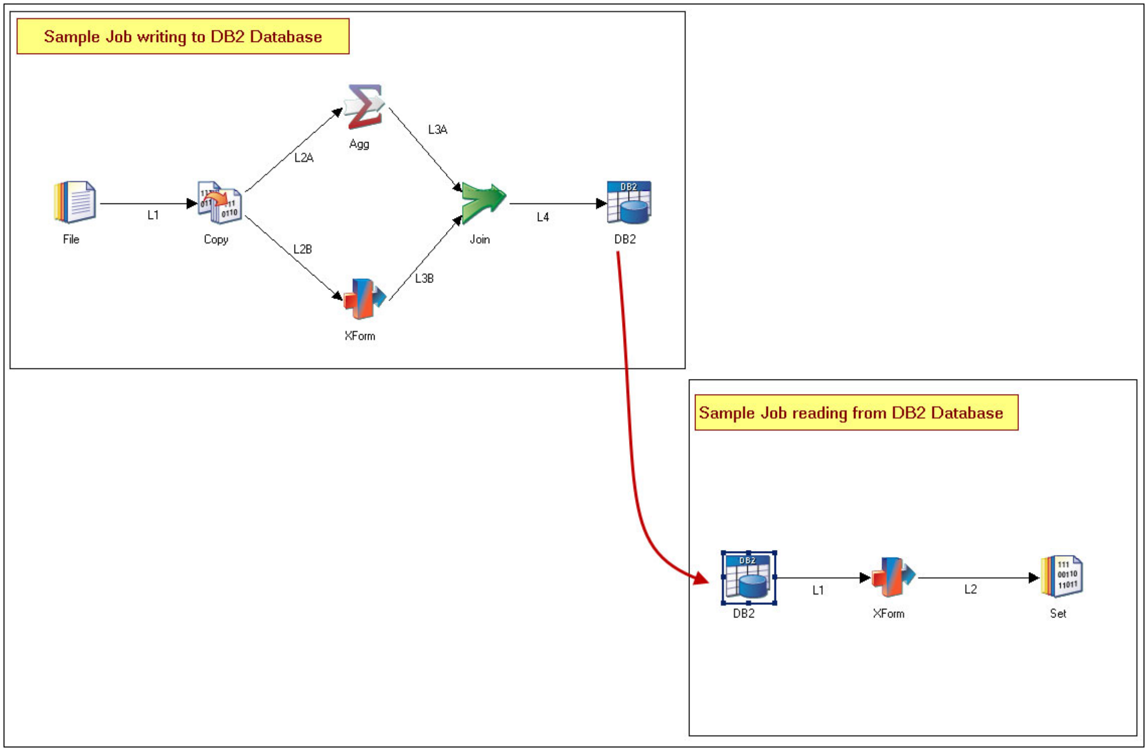

Sample Job writing to DB2 Database
File
L1
Copy
L2A
Agg
L3A
L2B
XForm
L3B
Join
L4
DB2
Sample Job reading from DB2 Database
DB2
L1
XForm
L2
Set

Figure 12-10 Expected result of the analysis services

The Manage Lineage Services of InfoSphere Metadata Workbench can analyze a single InfoSphere DataStage and InfoSphere QualityStage project or multiple projects. InfoSphere DataStage projects that have not been selected are not analyzed, and previous analysis results and linkages are removed.

Project selection is beneficial when multiple projects have been defined, including non-development or archived projects for which there is no requirement to provide data lineage and impact analysis reports.

To start Manage Lineage Services, complete these steps:

1. From the left navigation pane, on the **Advanced** tab, select **Manage Lineage Services**.
2. In the Manage Lineage Services pane, select the InfoSphere DataStage project on which to invoke the service. Then click **Run** to complete the process, which might take some time to complete.

When invoking the service, only those InfoSphere DataStage and InfoSphere QualityStage jobs or data sources that have been created or modified since the previous analysis are scanned. Therefore, when initially selecting a project, the analysis service can run for several minutes, depending on the number of jobs and the complexity of each job, where, subsequent analysis completes more quickly.

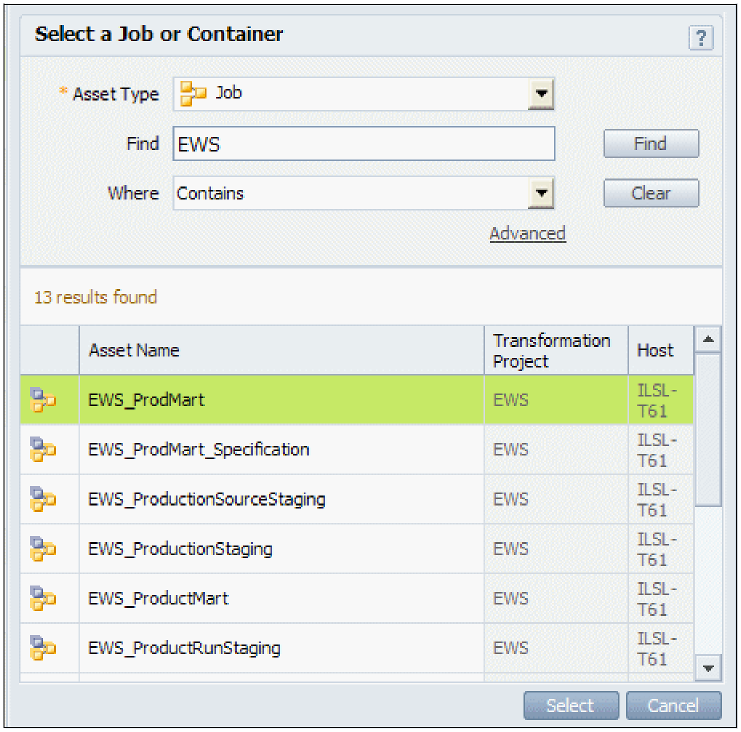
The service can be started from within InfoSphere Metadata Workbench or as a shell script or Windows batch file. The service must be regularly scheduled to ensure current and accurate information and resulting data lineage and impact analysis reports.

Starting manual binding

The IBM InfoSphere Metadata Workbench includes the ability to manually bind an InfoSphere DataStage and InfoSphere QualityStage job to a different job or to a previously imported database or data file asset. Manual binding might be required several reasons, including when custom stages are defined for reading or writing to the data sources.

The links created by manual binding are included when generating data lineage analysis reports. Such links define the data flow between jobs, data sources, and BI reports.

Figure 12-11 shows an example of manual binding.



Select a Job or Container

* Asset Type: Job

Find: EWS

Where: Contains

13 results found

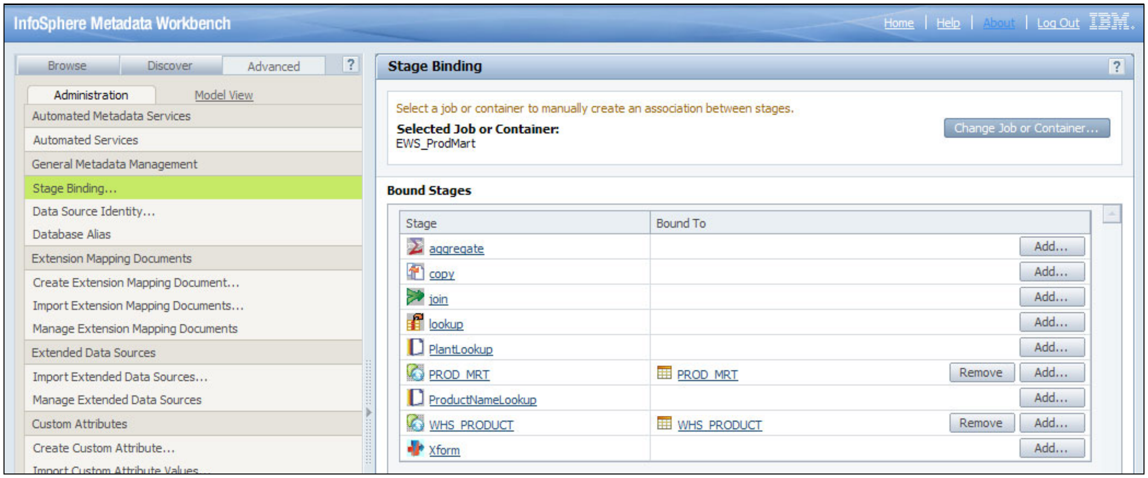
Asset Name	Transformation Project	Host
EWS_ProdMart	EWS	ILSL-T61
EWS_ProdMart_Specification	EWS	ILSL-T61
EWS_ProductionSourceStaging	EWS	ILSL-T61
EWS_ProductionStaging	EWS	ILSL-T61
EWS_ProductMart	EWS	ILSL-T61
EWS_ProductRunStaging	EWS	ILSL-T61

Figure 12-11 Stage Binding select dialog box

To start manual binding, complete these steps:

1. From the left navigation pane, on the **Advanced** tab, click **Manage Lineage**.
2. In the Manage Lineage pane, click the **Stage Binding** button ( ) from the toolbar.
3. In the Select a Job or Container pane (Figure 12-11), complete these steps:
   a. Select the type of asset that you want to bind. Select either **Job** or **Shared container**.
   b. Enter the name or partial name of the asset to be bound. Click **Find**.
   c. From the list of matched results, select the requested asset, and click **Select**.

4. From the list of the stages that can be bound (Figure 12-12), for a given stage, click **Add** to bind the stage.


InfoSphere Metadata Workbench
Home | Help | About | Log Out IBM.
Browse
Discover
Advanced
?
Administration
Model View
Automated Metadata Services
Automated Services
General Metadata Management
Stage Binding...
Data Source Identity...
Database Alias
Extension Mapping Documents
Create Extension Mapping Document...
Import Extension Mapping Documents...
Manage Extension Mapping Documents
Extended Data Sources
Import Extended Data Sources...
Manage Extended Data Sources
Custom Attributes
Create Custom Attribute...
Import Custom Attribute Values...
Change Job or Container...

Stage Binding

Select a job or container to manually create an association between stages.

Selected Job or Container:
EWS_ProdMart

Bound Stages

Stage	Bound To		
aggregate			Add...
copy			Add...
join			Add...
lookup			Add...
PlantLookup			Add...
PROD_MRT	PROD_MRT	Remove	Add...
ProductNameLookup			Add...
WHS_PRODUCT	WHS_PRODUCT	Remove	Add...
Xform			Add...

Figure 12-12 Stage binding mapping

5. In the window that opens, select a job, database table, or data file:
   a. Select the type of asset to which you want to bind the stage. Select stage, database table, or data file element.
   b. Enter the name or partial name of the asset to be bound. Click **Find**.
   c. From the list of matched results, select the requested asset, and click **Select** to create a binding between the stage and selected asset.
6. Click **Save** to complete the process.

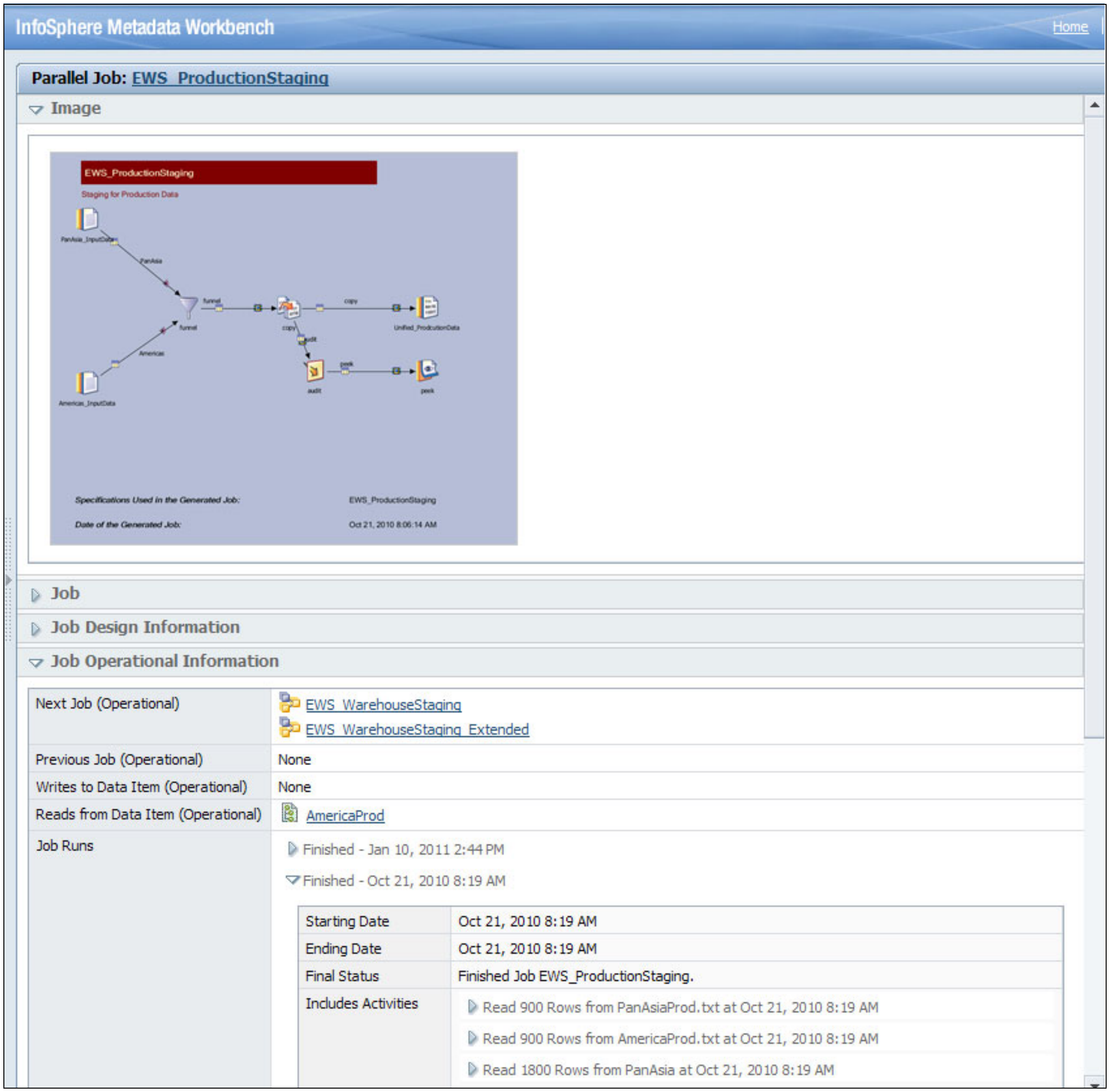
12.2.2 Operational lineage

Operational metadata explains how data was created or transformed rather than its intended design. It is further used to describe the events and processes that occur and when an InfoSphere DataStage and InfoSphere QualityStage job is run.

The following information about the job run is captured and available from InfoSphere Metadata Workbench, further benefitting users:

- When did the job run start and when did it complete?
- What was the job run status?
- How many rows were read or written?
- Which tables or files were read from or written to?

When investigating data lineage from a BI report, you can ascertain when the InfoSphere DataStage job last ran to update the data sources referenced by the report. Therefore, a user can determine the accuracy and quality of the BI report, (Figure 12-13).

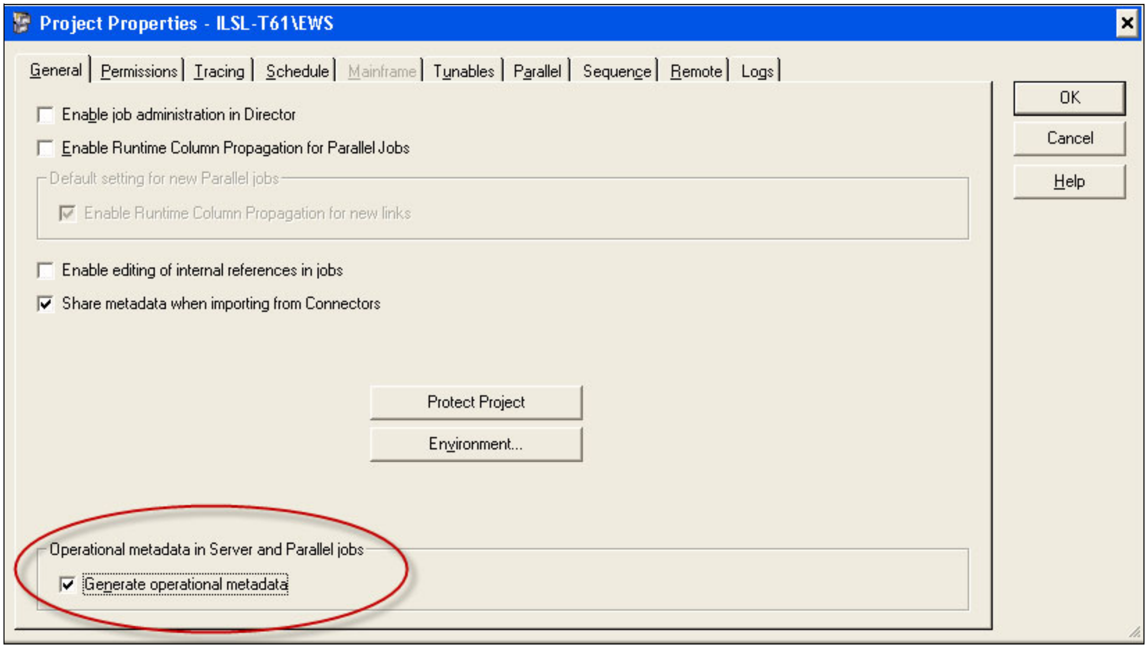


Next Job (Operational)	EWS_WarehouseStaging EWS_WarehouseStaging_Extended
Previous Job (Operational)	None
Writes to Data Item (Operational)	None
Reads from Data Item (Operational)	AmericaProd
Job Runs	Finished - Jan 10, 2011 2:44 PM Finished - Oct 21, 2010 8:19 AM

Starting Date	Oct 21, 2010 8:19 AM
Ending Date	Oct 21, 2010 8:19 AM
Final Status	Finished Job EWS_ProductionStaging.
Includes Activities	Read 900 Rows from PanAsiaProd.txt at Oct 21, 2010 8:19 AM Read 900 Rows from AmericaProd.txt at Oct 21, 2010 8:19 AM Read 1800 Rows from PanAsia at Oct 21, 2010 8:19 AM

Figure 12-13 Display of InfoSphere DataStage job run information in InfoSphere Metadata Workbench

Operational metadata is generated when an InfoSphere DataStage job is run. By using InfoSphere DataStage and InfoSphere QualityStage Administrator tools, you can specify operational metadata to be generated for all jobs of a given project. Alternately, when running a specific InfoSphere DataStage job, you can indicate to also generate operational metadata as shown in Figure 12-14.



□ Enable job administration in Director

□ Enable Runtime Column Propagation for Parallel Jobs

Default setting for new Parallel jobs

☑ Enable Runtime Column Propagation for new links

□ Enable editing of internal references in jobs

☑ Share metadata when importing from Connectors

Operational metadata in Server and Parallel jobs

☑ Generate operational metadata

Figure 12-14 InfoSphere DataStage Administrator project setting window

Operational metadata is generated in the form of an XML file, which is created for each unique job run. The operational metadata files are saved to the file system of the InfoSphere DataStage engine in the `IBM/Information Server/ASBNode/conf/etc/XmlFiles` directory.

Operational metadata files must be imported into InfoSphere Information Server so that they can be shared and viewed within InfoSphere Metadata Workbench.

Importing operational metadata

Load the operational metadata files into InfoSphere metadata repository by using the provided import script. The procedure imports all operational metadata files and removes them from the file system of the engine.

The procedure is installed by default when installing the InfoSphere DataStage engine. The procedure is started as a shell script or Windows batch file and requires parameters when run. The procedure must be regularly scheduled to ensure the availability of operational metadata within InfoSphere Metadata Workbench.

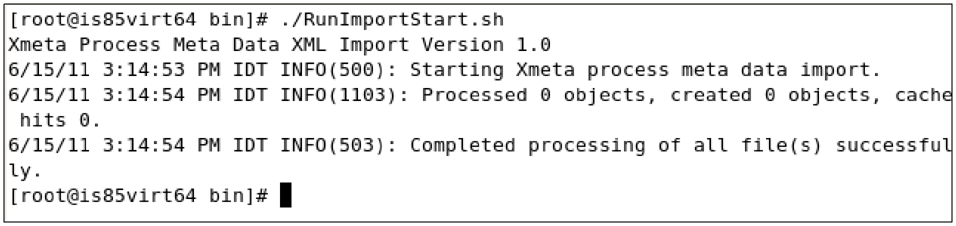
Running the import script

To run the import script from a Linux or UNIX shell command, use the **RunImportStart** shell script shown in Example 12-2.

Example 12-2 Import script

```
cd /opt/ibm/InformationServer/ASBNode/bin ./RunImportStart.sh
```

No parameters are required when starting the procedure. However, you must set the server and user name credentials within the `runimport.cfg` file in the `/opt/IBM/InformationServer/ASBNode/conf` directory. Figure 12-15 shows the import after running the script.



```
[root@is85virt64 bin]# ./RunImportStart.sh
Xmeta Process Meta Data XML Import Version 1.0
6/15/11 3:14:53 PM IDT INFO(500): Starting Xmeta process meta data import.
6/15/11 3:14:54 PM IDT INFO(1103): Processed 0 objects, created 0 objects, cache
 hits 0.
6/15/11 3:14:54 PM IDT INFO(503): Completed processing of all file(s) successful
ly.
[root@is85virt64 bin]# █
```

Figure 12-15 Import of operational metadata after running the RunImportStart.sh script

After the metadata is imported into the repository, you must invoke the lineage services of InfoSphere Metadata Workbench to associate the operational run information with the corresponding InfoSphere DataStage job.

Employ proper lifecycle management to only capture, within InfoSphere Information Server repository, the current and necessary operational metadata.

12.3 Support for external processes

External processes, such as stored procedures and scripts, additionally extract, load, and transform data much in the same way as InfoSphere DataStage and InfoSphere QualityStage. Such processes must be documented and represented within InfoSphere Metadata Workbench to provide support for the required data flow and lineage reporting requirements, for enterprise metadata management, and for the information integration solution.

Extension mapping documents are a defined mapping between a source and target asset and can represent these processes. For more information about extension mapping documents, see Chapter 7, “Source documentation” on page 175.

12.4 Support for InfoSphere FastTrack mapping specifications

IBM InfoSphere FastTrack allows for the documentation and specification of data integration processes. Such specifications include the source and target assets, the intended business rule, applied function, and high-level description of the data transformation process. Furthermore, InfoSphere FastTrack allows for the assignment of glossary terms to the source or target asset. It additionally offers the ability to search glossary terms to find the relevant source and target assets.

The documentation is stored within a mapping specification document. The document acts in a similar way to extension mapping documents in InfoSphere Metadata Workbench. Therefore, the document contains data flow information that can benefit the users of InfoSphere Metadata Workbench when either viewing or analyzing information.

A user within InfoSphere Metadata Workbench can browse mapping specifications of InfoSphere FastTrack and view the detailed mapping between source and target assets. A user can also view the InfoSphere DataStage job that is generated from the specification.

InfoSphere Metadata Workbench also allows consideration of the documented data flow within the mapping specification when rendering data lineage reports. Data lineage depicts the flow of the data source to the target asset, as designed and implemented within the mapping specification of InfoSphere FastTrack.

A user within InfoSphere Metadata Workbench identifies which mapping specifications are considered for data lineage.

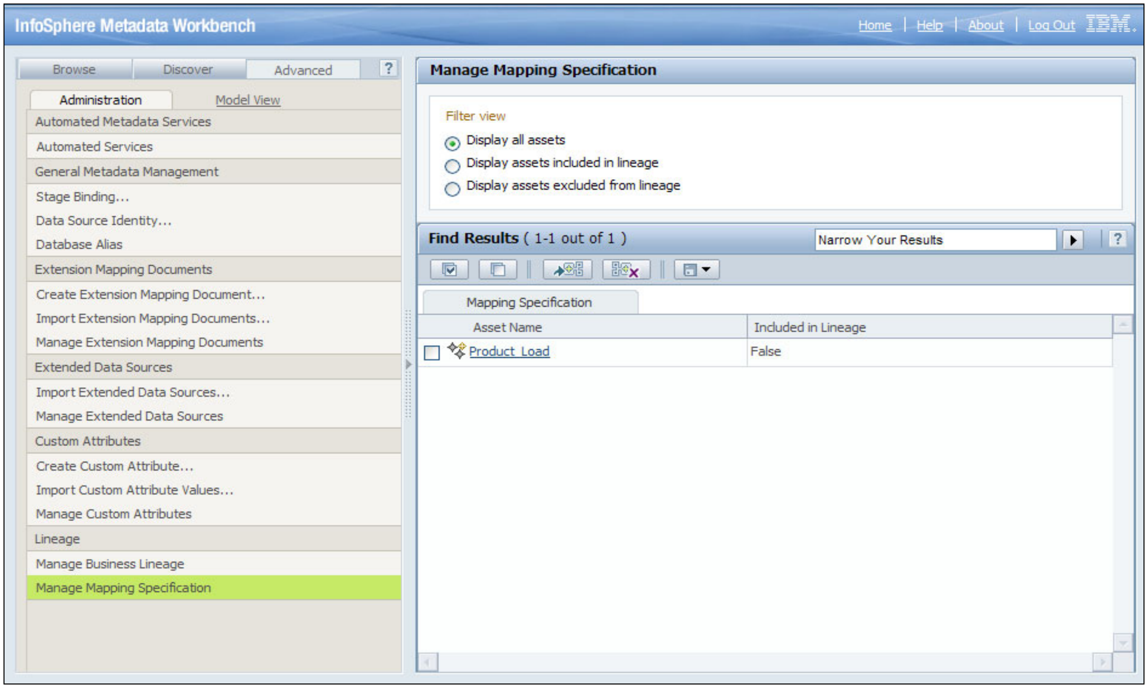
To identify mapping specifications, complete these steps:

1. From the left navigation pane, on the **Advanced** tab, click **Manage Mapping Specification**.
2. In the page, where you see the list of all InfoSphere FastTrack mapping specifications, complete the following tasks to filter the list to view only those mapping specifications that have (or have not been) included.
   a. Select the individual mapping specification documents to include. Alternately click **Select All** from the toolbar.

b. Choose one of the following options:

   - Click **Include in Lineage** to consider all selected mapping specifications when rendering data lineage reports.
   - Click **Exclude from Lineage** to not consider and display the documented source to target mapping when rendering data lineage reports.

   Figure 12-16 shows the Manage Mapping Specification pane where you can filter the results.


InfoSphere Metadata Workbench
Home | Help | About | Log Out
Browse
Discover
Advanced
?
Administration
Model View
Automated Metadata Services
Automated Services
General Metadata Management
Stage Binding...
Data Source Identity...
Database Alias
Extension Mapping Documents
Create Extension Mapping Document...
Import Extension Mapping Documents...
Manage Extension Mapping Documents
Extended Data Sources
Import Extended Data Sources...
Manage Extended Data Sources
Custom Attributes
Create Custom Attribute...
Import Custom Attribute Values...
Manage Custom Attributes
Lineage
Manage Business Lineage
Manage Mapping Specification
Manage Mapping Specification
Narrow Your Results
Mapping Specification

Filter view

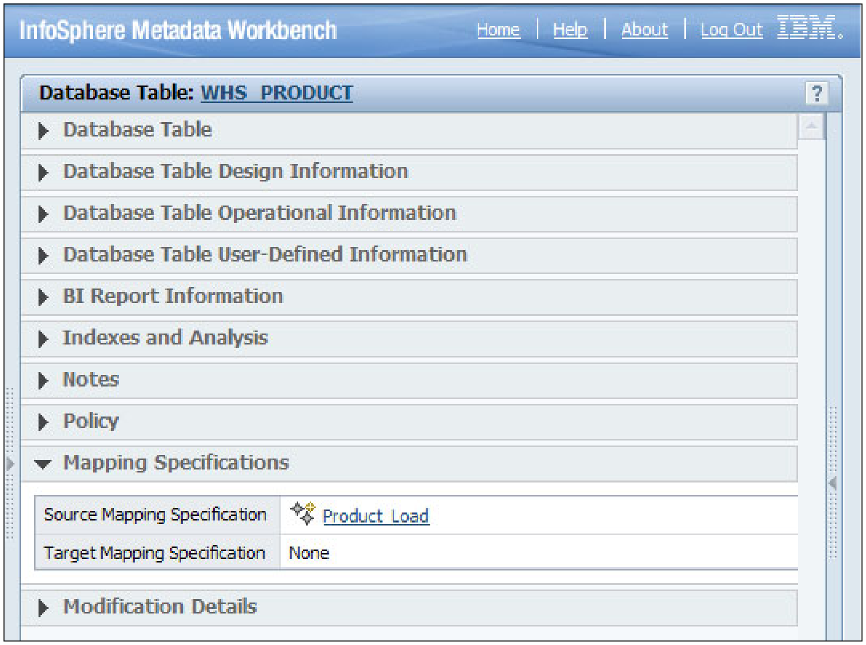
● Display all assets
○ Display assets included in lineage
○ Display assets excluded from lineage

Find Results (1-1 out of 1)

	Asset Name	Included in Lineage
□	Product_Load	False

Figure 12-16 Include mapping specifications for data lineage reports

When complete, browsing a database table (Figure 12-17) shows, for example, the mapping specification that references the table as a source or target asset.

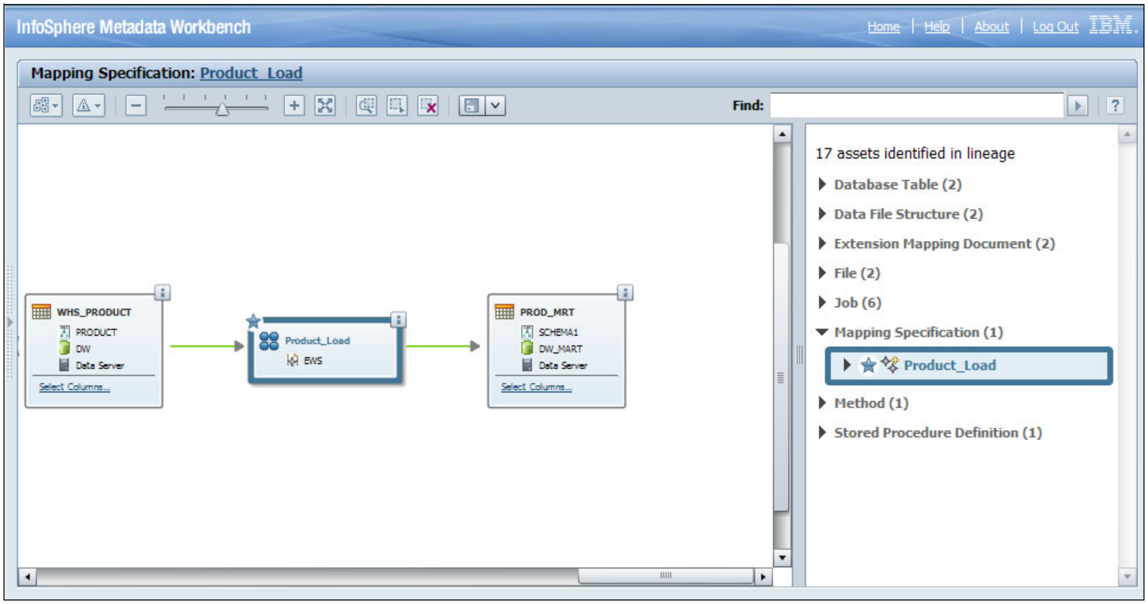

InfoSphere Metadata Workbench
Home | Help | About | Log Out
IBM.
?
Database Table
Database Table Design Information
Database Table Operational Information
Database Table User-Defined Information
BI Report Information
Indexes and Analysis
Notes
Policy
Mapping Specifications
Modification Details

Database Table: WHS_PRODUCT

Source Mapping Specification	Product_Load
Target Mapping Specification	None

Figure 12-17 Association of a database table to a mapping specification within IBM InfoSphere Metadata Workbench

Data lineage reports (Figure 12-18) also include the data flow relationship as described in InfoSphere FastTrack.



Mapping Specification: Product_Load

17 assets identified in lineage

- Database Table (2)
- Data File Structure (2)
- Extension Mapping Document (2)
- File (2)
- Job (6)
- Mapping Specification (1)
  - Product_Load
- Method (1)
- Stored Procedure Definition (1)

Figure 12-18 Data lineage flow through a mapping specification document

12.5 Configuring business lineage

Business lineage offers a high-level view of a data flow, depicting the intersecting data sources. Such lineage provides business users and analyst an understanding of the data flow between source systems and consuming BI reports, so that they can validate the inclusion of such information.

By design, business lineage does not include the InfoSphere DataStage jobs, InfoSphere FastTrack mapping specifications, or extension mapping documents of InfoSphere Metadata Workbench. Such information is not necessary to satisfy the requirements of these users, but rather their interest in visualizing specific data sources.

With IBM InfoSphere Metadata Workbench, a user can configure which assets are included when rendering business lineage reports. Therefore, assets that do not feature within the integrated solution or that act as temporary storage areas can be hidden from the business lineage report.

With InfoSphere Metadata Workbench, the following asset types can be configured:

- Application
- BI Model
- BI Report
- Data File
- File
- Schema
- Stored Procedure Definition

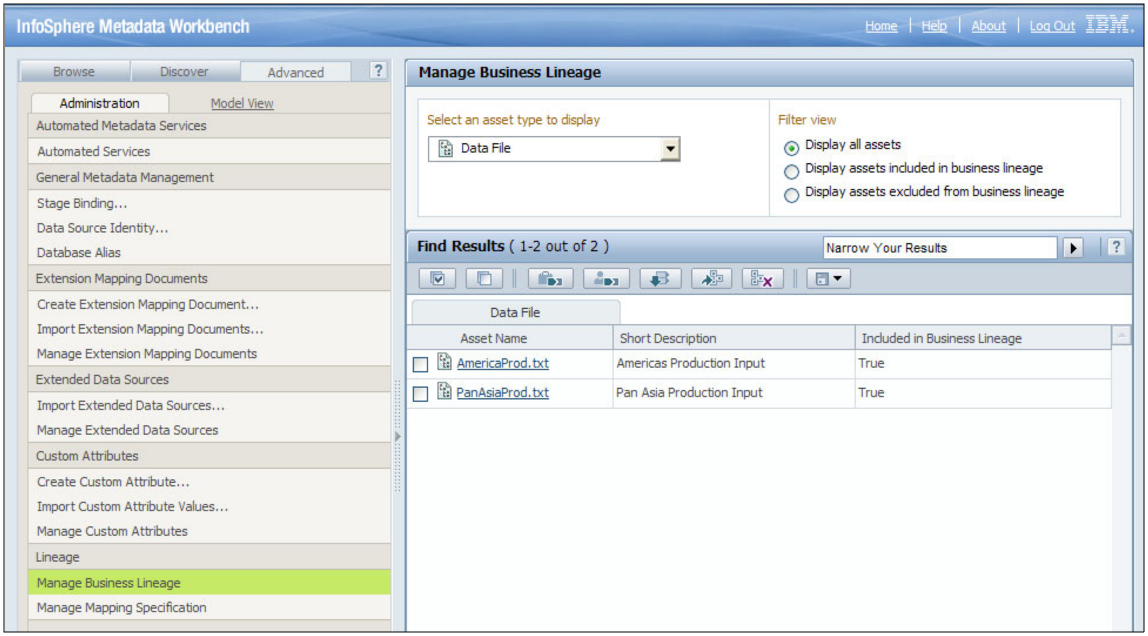
When selecting an asset, all child assets are automatically included. For example, if you select a database schema to exclude from a business lineage, this action also excludes all database tables that are contained by the schema and all database columns that are contained by the tables.

A user within InfoSphere Metadata Workbench continues to be able to browse and view all assets, including assets that were excluded from business lineage. The settings do not affect data lineage nor analysis reports.

To identify mapping specifications, complete these steps:

1. From the left navigation pane, on the **Advanced** tab, click **Manage Business Lineage**.
2. From the list of application assets that is displayed, filter the list to display only those assets that are included for business lineage or those assets that are excluded from business lineage:
   a. Select the type of asset that you want to select and exclude from business lineage. The list of displayed asset refreshes.
   b. Select the individual assets to exclude from business lineage. Alternatively, click **Select All** from the toolbar to select all assets, or **Select None** to clear all assets.
   c. Choose one of the following options:
      - Click **Exclude from Business Lineage** to exclude and not display all selected assets when rendering business lineage reports.
      - Click **Include for Business Lineage** to include and display all selected assets.

Figure 12-19 shows the Manage Business Lineage pane with the filter set to **Display all assets**.


InfoSphere Metadata Workbench
Home | Help | About | Log Out
IBM.
Browse
Discover
Advanced
?
Administration
Model View
Automated Metadata Services
Automated Services
General Metadata Management
Stage Binding...
Data Source Identity...
Database Alias
Extension Mapping Documents
Create Extension Mapping Document...
Import Extension Mapping Documents...
Manage Extension Mapping Documents
Extended Data Sources
Import Extended Data Sources...
Manage Extended Data Sources
Custom Attributes
Create Custom Attribute...
Import Custom Attribute Values...
Manage Custom Attributes
Lineage
Manage Business Lineage
Manage Mapping Specification
Narrow Your Results
Data File

Manage Business Lineage

Select an asset type to display: Data File

Filter view
● Display all assets
○ Display assets included in business lineage
○ Display assets excluded from business lineage

Find Results (1-2 out of 2)

	Asset Name	Short Description	Included in Business Lineage
□	AmericaProd.txt	Americas Production Input	True
□	PanAsiaProd.txt	Pan Asia Production Input	True

Figure 12-19 Managing business lineage from InfoSphere Metadata Workbench

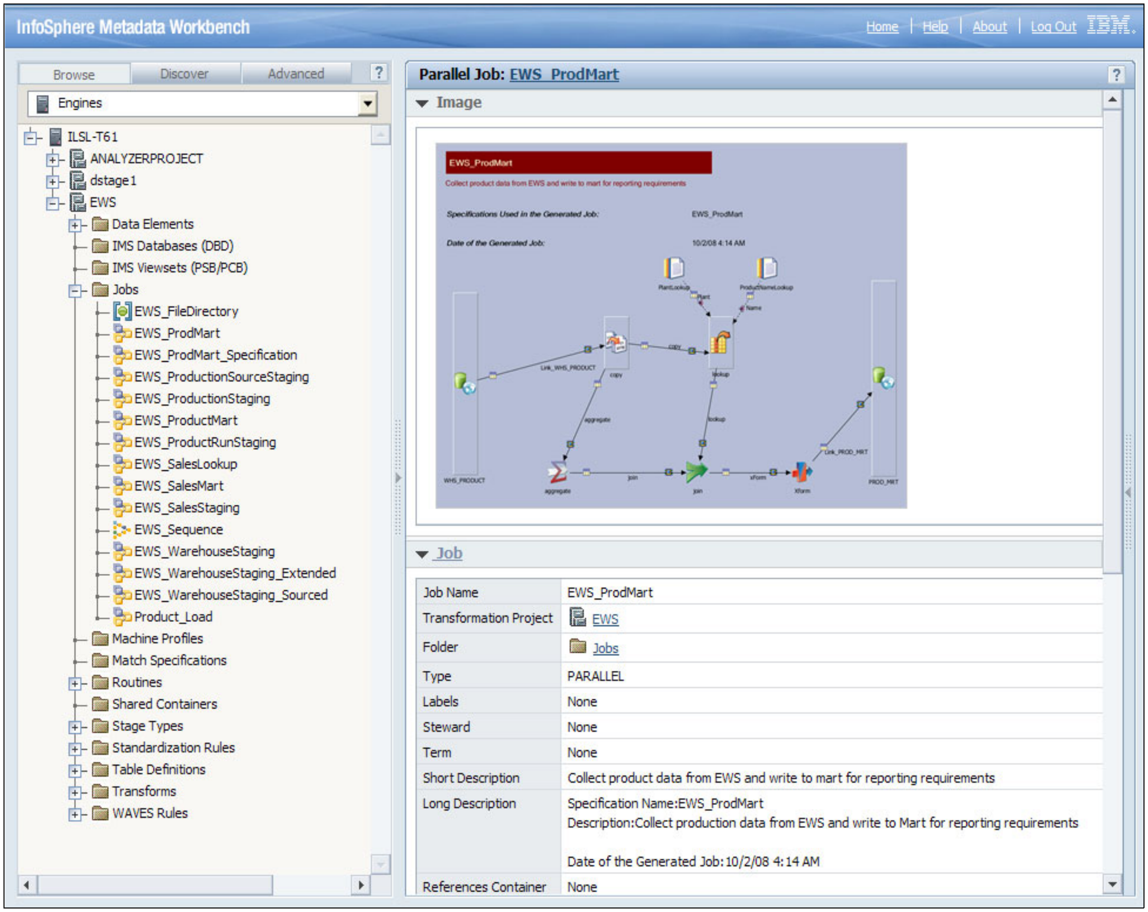
12.6 Search and display

This section explains how to search and display information assets within InfoSphere Metadata Workbench.

12.6.1 Information catalog

A user can browse and navigate information assets in InfoSphere Metadata Workbench. By browsing information, a user can visualize the context and structure of the information, in addition to easily navigating and viewing its details. Furthermore, right-clicking allows for specific actions that are available for the asset, including invoking data lineage or assignment to a glossary term.

From the left navigation pane, click **Browse** to view the assets (Figure 12-20). This view is also available when browsing to the home page of InfoSphere Metadata Workbench.



Job Name	EWS_ProdMart
Transformation Project	EWS
Folder	Jobs
Type	PARALLEL
Labels	None
Steward	None
Term	None
Short Description	Collect product data from EWS and write to mart for reporting requirements
Long Description	Specification Name:EWS_ProdMart Description:Collect production data from EWS and write to Mart for reporting requirements Date of the Generated Job:10/2/08 4:14 AM
References Container	None

Figure 12-20 Browse InfoSphere DataStage engine within InfoSphere Metadata Workbench

Engines

Engines are the host of InfoSphere DataStage and InfoSphere QualityStage projects and reflect the location where the InfoSphere Information Server engine is installed. Projects contain InfoSphere DataStage and InfoSphere QualityStage jobs, routines, quality specifications, table definitions, and other such information.

Information is displayed according to the engine, project, and the folder structure as defined within the InfoSphere DataStage and InfoSphere QualityStage project. A developer can easily navigate the folders of a given project and select a job, viewing its details and inspecting its dependency upon data sources.

Implemented data resources

Data resources are the containers of databases and data files as created during the import process of such assets. Information is displayed according to the structure of the databases or data files. Databases include schemes, which further include tables and fields. Data files include data file structures and data file fields.

A user who navigates to a database table selects the database and database schema and views the contained columns of the table. The user further inspects the database table details, including profiling summaries, job, or mapping usage and term definitions.

Logical data models

Logical data models are the containers of logical entities and attributes. These models offer the design specification for physical models and implemented database systems. Logical data models are created during the import process of such assets.

Information is displayed according to the structure of the model, which further includes entities and attributes. A user who navigates to an entity selects the model and views the entity details, including the subject area and implemented database table.

Physical data models

Physical data models are the containers of design tables and columns and offer the implementation design for database systems. Physical data models are created during the import process of such assets.

Information is displayed according to the structure of the model, which further includes design tables and columns. A user who navigates to a design column selects the model and table and views the implemented database table, column type, and column length.

Importing and exporting logical and physical data models: You can use InfoSphere Metadata Asset Manager to import logical and physical data models into the metadata repository from design tools, such as InfoSphere Data Architect, CA Erwin, ER/Studio, and Sybase. You can use Import Export Manager to export logical and physical data models to InfoSphere Data Architect.

Glossary

Glossary categories define the structure and backbone of InfoSphere Business Glossary. Categories can contain additional categories or terms. A user can browse the structure of categories, understanding their semantic meaning and definition. A user selects a category to view its contained or referenced terms, definition, and steward.

Mapping projects

Mapping projects are defined within IBM InfoSphere FastTrack and contain mapping specifications. Mapping specifications are the defined business logic documenting source to target data movement and can be used to generate InfoSphere DataStage jobs.

Information is displayed according to the project, mapping specification, and included mapping. Selecting a specification shows the source and target asset mapping and the generated InfoSphere DataStage job.

BI reports and models

BI reports allow for the distribution and display of information. BI servers are the container of reports. BI reports, servers, and models are created during the import process of such assets.

Information is displayed according to the structure of a BI server, which includes report packages, reports, and queries. A user navigating to a report browses the BI server and packages, selecting the report to view or the data flow to analyze.

Applications

Applications represent extended data sources that are documented and created within IBM InfoSphere Metadata Workbench. Applications contain object types, which further contain methods and parameters.

Information is displayed according to the structure of the application. A user who navigates to a parameter browses the application structure and selects the parameter to view more information.

Stored procedure definitions

Stored procedure definitions represent extended data sources that are documented and created in InfoSphere Metadata Workbench. Stored procedure definitions contain parameters.

Information is displayed according to the structure of the stored procedure definition. A user who navigates to a parameter browses the stored procedure structure and selects the parameter to view more information.

Files

Files represent extended data sources that are documented and created within InfoSphere Metadata Workbench. Files are displayed as a list of all such assets.

A user who navigates to a file browses the list of files and selects a file to view more information.

Extension mapping documents

Extension mapping documents represent the documented source to target mapping that is authored within InfoSphere Metadata Workbench. These documents are used to document data lineage. Extension mapping documents are stored in a folder so that mapping documents can be grouped according to project, area, and so on.

Information is displayed according to the defined folder structure, so that subfolders or extension mapping documents are shown. A user who navigates to an extension mapping document browses the folder hierarchy and selects a document to further analyze or view its details.

12.6.2 Find and search

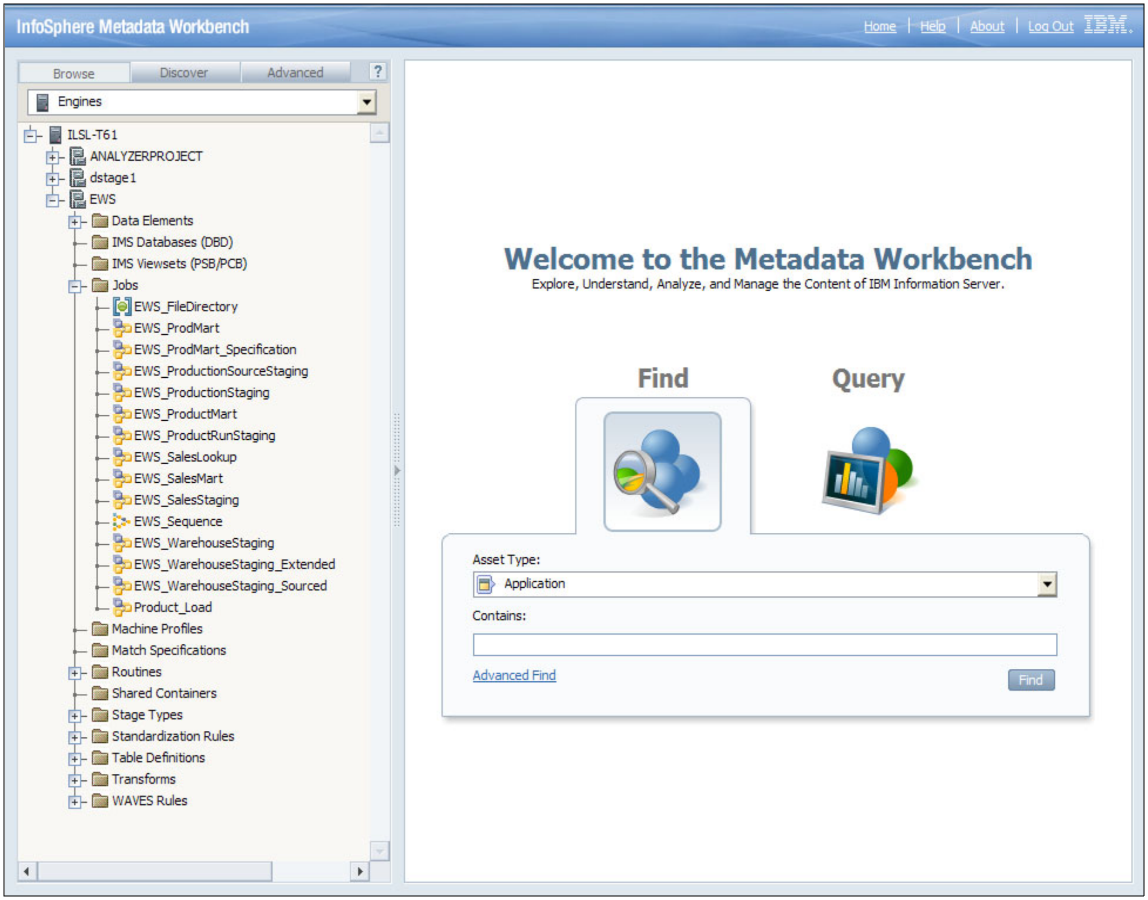
A user of InfoSphere Metadata Workbench might want to quickly locate a particular asset, for further analysis or understanding. The Find feature helps a user to search all assets of a particular type, including adding contextual filters to further refine search results. The resulting list of assets can further be searched to precisely locate the relevant results.

Searching an asset from the home page

To search an asset from the home page, complete these steps:

1. Select the asset type that you want to search.
2. Enter the name or partial name on which to search.

Figure 12-21 shows the home page where you can search for an asset. A list of assets is returned based on the search criteria.



Engines

Welcome to the Metadata Workbench

Explore, Understand, Analyze, and Manage the Content of IBM Information Server.

Asset Type: Application

Contains:

Figure 12-21 Home page of InfoSphere Metadata Workbench

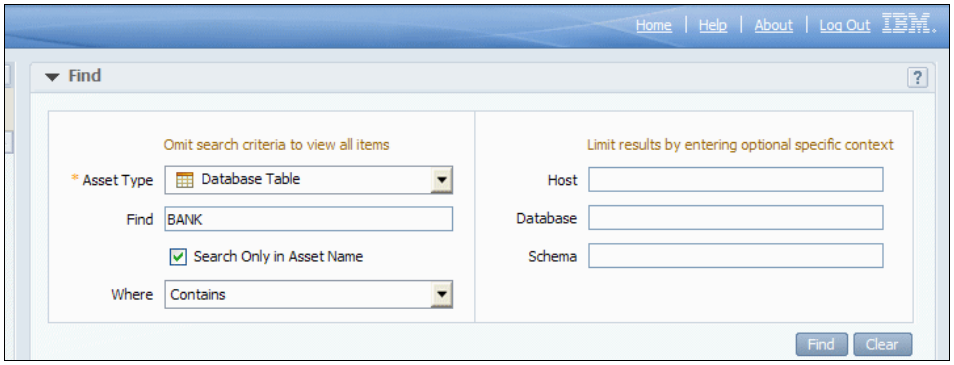
Performing a refined search with the advanced find function

To perform a more refined search by using the advanced find function, follow these steps:

1. Click the **Advanced Find** link to view the advanced find search options.
2. Select the asset type for the asset you want to search.
3. Enter the name or partial name on which to search.
4. Optional: Select **Search Only in Asset Name** to only search against the name of the asset. Otherwise the description fields are also searched.

5. Optional: Enter the name or partial name of the contextual items.

In the example shown in Figure 12-22, the selected asset type is a Database Table. A user can restrict the results by entering a name or partial name of the containing host, database, or schema.



Omit search criteria to view all items

* Asset Type: Database Table

Find: BANK

☑ Search Only in Asset Name

Where: Contains

Limit results by entering optional specific context

Host:

Database:

Schema:

Figure 12-22 Advanced search for database table from InfoSphere Metadata Workbench

Results

Whether you use the home page or the advanced search function, a list of assets that matches the selected criteria is returned. Select any asset to further investigate or analyze the asset details. Right-lick any asset to view a list of tasks, including the ability to invoke data lineage or assign to term.

From the result set, the user can also reposition and highlight the list of assets according to a particular term, by entering a term in the Jump To search field. Further, a user can navigate to a particular result page, print or save the results, and invoke any asset-specific actions, including assigning to term or assigning to steward.

12.7 Querying and reporting

This section explains how you can see reports and use ad-hoc query to obtain information.

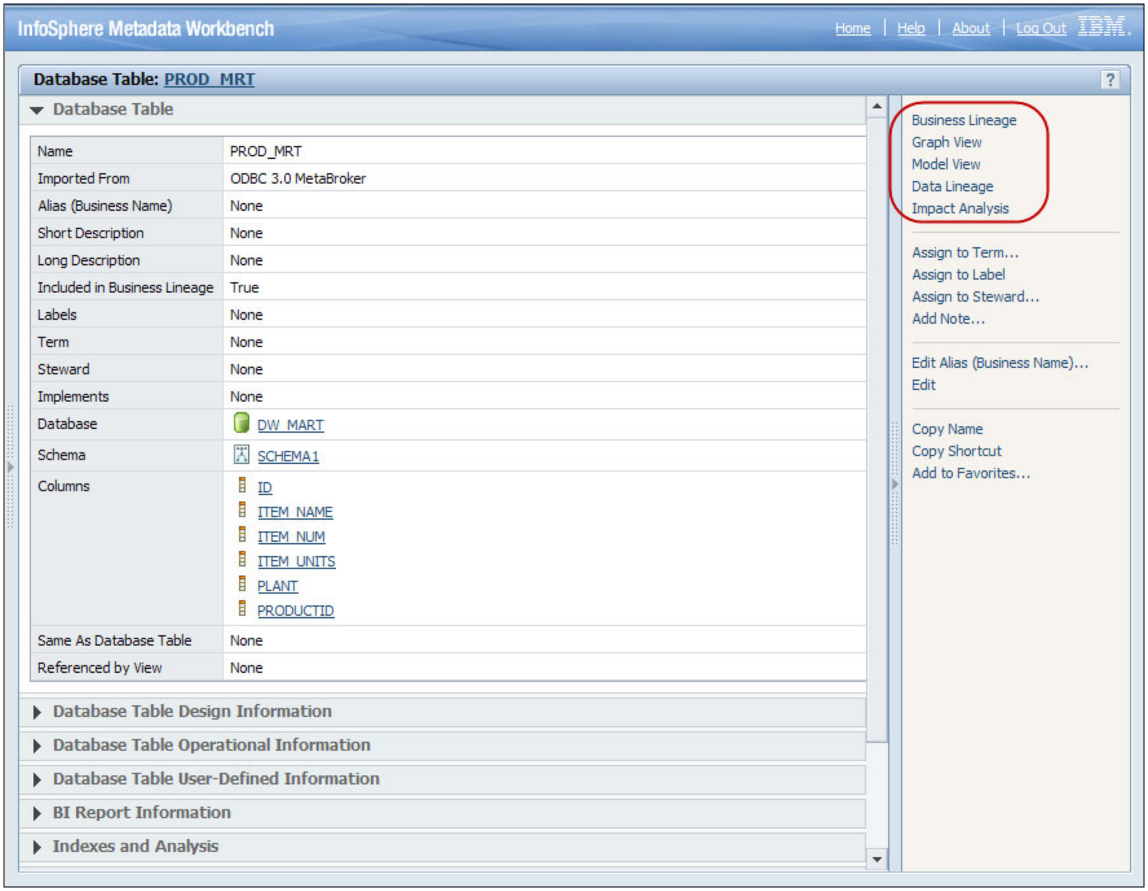
12.7.1 Reports

Lineage reports detail the flow of information, where impact analysis reports detail the dependencies of information, in a graphical representation. A user can further investigate each individual asset or print a report of all included assets.

Reports can be filtered to display only detected relationships based upon design metadata, operational metadata, or user-defined mapping, which include extended data sources and InfoSphere FastTrack mapping specifications.

Graphical reports depict and display information by using the recognized icons of the data sources, BI reports, and jobs. You can start reports in one of the following ways:

- When viewing a list of items, right-click the item and select an item from the menu that opens. Analysis reports that are relevant to the object are displayed.
- When viewing an item, analysis reports that are relevant for that item are displayed in the right-navigation action pane (Figure 12-23).



Name	PROD_MRT
Imported From	ODBC 3.0 MetaBroker
Alias (Business Name)	None
Short Description	None
Long Description	None
Included in Business Lineage	True
Labels	None
Term	None
Steward	None
Implements	None
Database	DW_MART
Schema	SCHEMA1
Columns	ID ITEM_NAME ITEM_NUM ITEM_UNITS PLANT PRODUCTID
Same As Database Table	None
Referenced by View	None

Figure 12-23 Starting analysis reports from InfoSphere Metadata Workbench

The following reports are available:

- *Data lineage reports* (Figure 12-24) on the flow of information, from source to target, through InfoSphere DataStage jobs and extension mapping documents. All analysis on data relationships, design, and operational and user-defined data are included by default in the report.

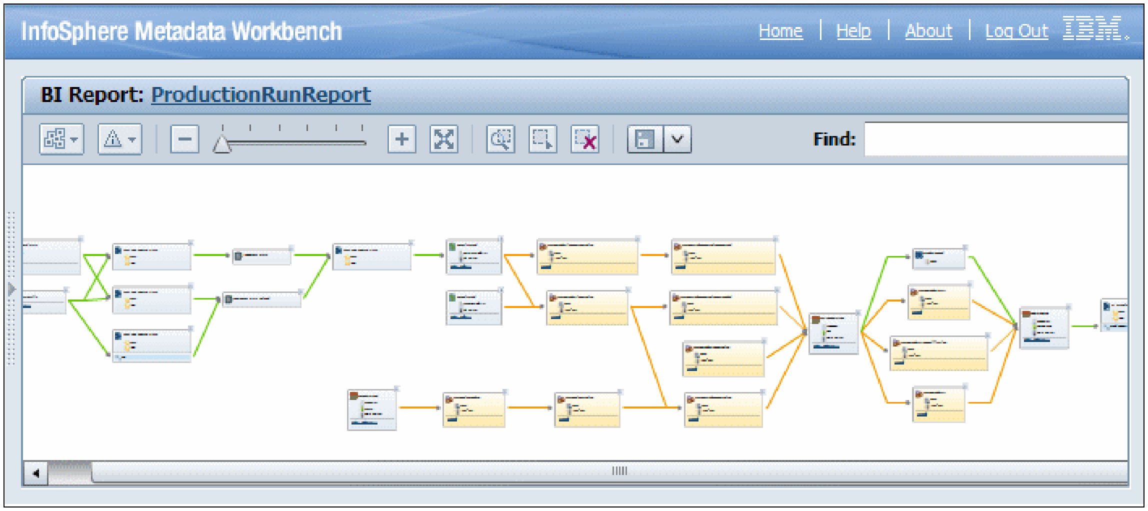

InfoSphere Metadata Workbench
Home | Help | About | Log Out
IBM.
BI Report: ProductionRunReport
Find:

Figure 12-24 Sample data lineage from InfoSphere Metadata Workbench

- *Business lineage reports* on the intersection of data sources, such as database tables, data files and BI reports, as they flow from source to target.
- *Impact analysis reports* on which assets depend on a particular item.

- *Model view reports* (Figure 12-25) on the interdependency between logical, physical, and implemented database tables, depicting their implementation relationships.

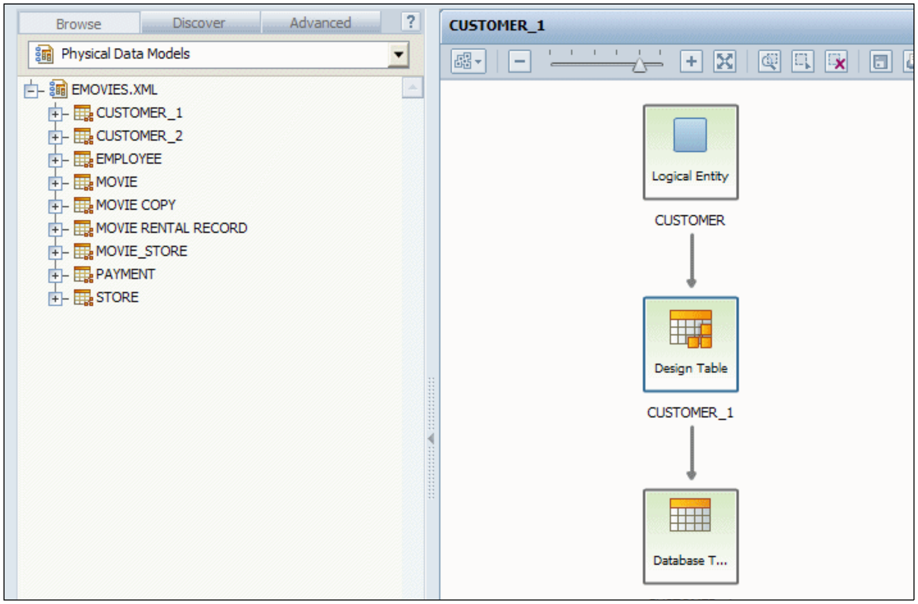


Figure 12-25 Sample model view from InfoSphere Metadata Workbench

- *Graph view reports* (Figure 12-26) on the relationships of a specific asset, such as assigned terms, mapped data sources, or container assets. Select any asset to traverse the relationships and to view the asset details.

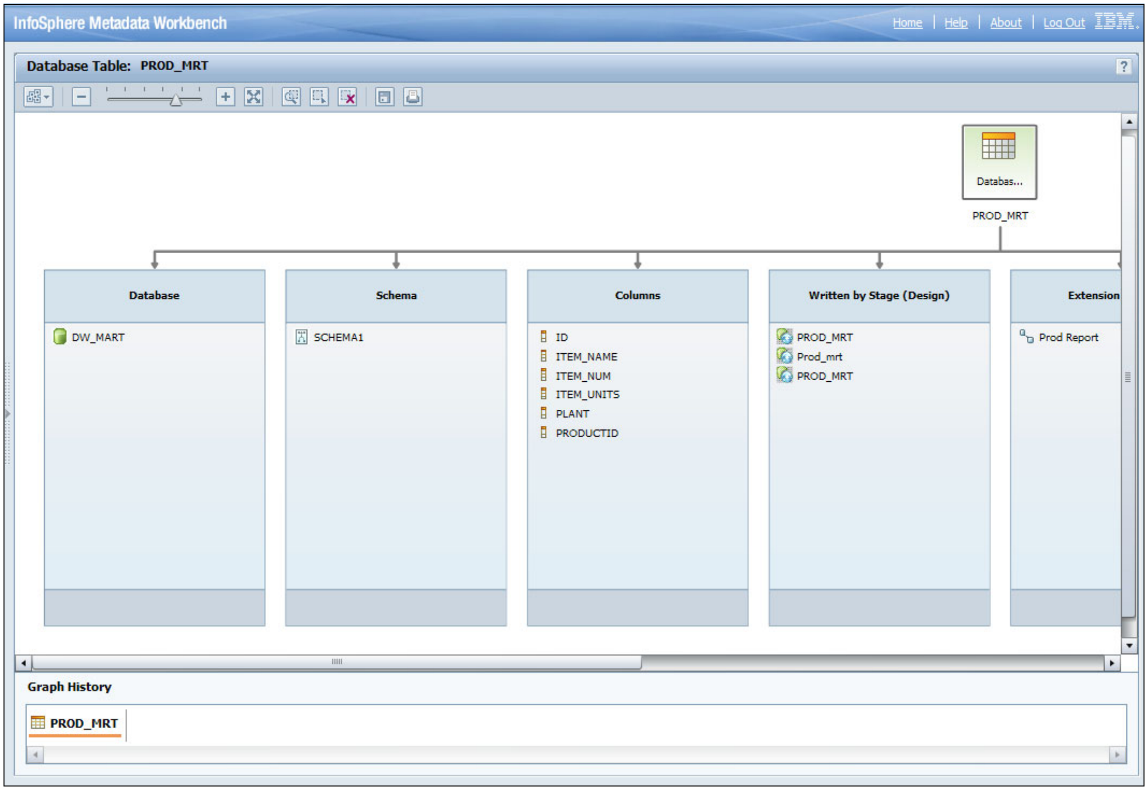


Figure 12-26 Sample graph view of a database table in InfoSphere Metadata Workbench

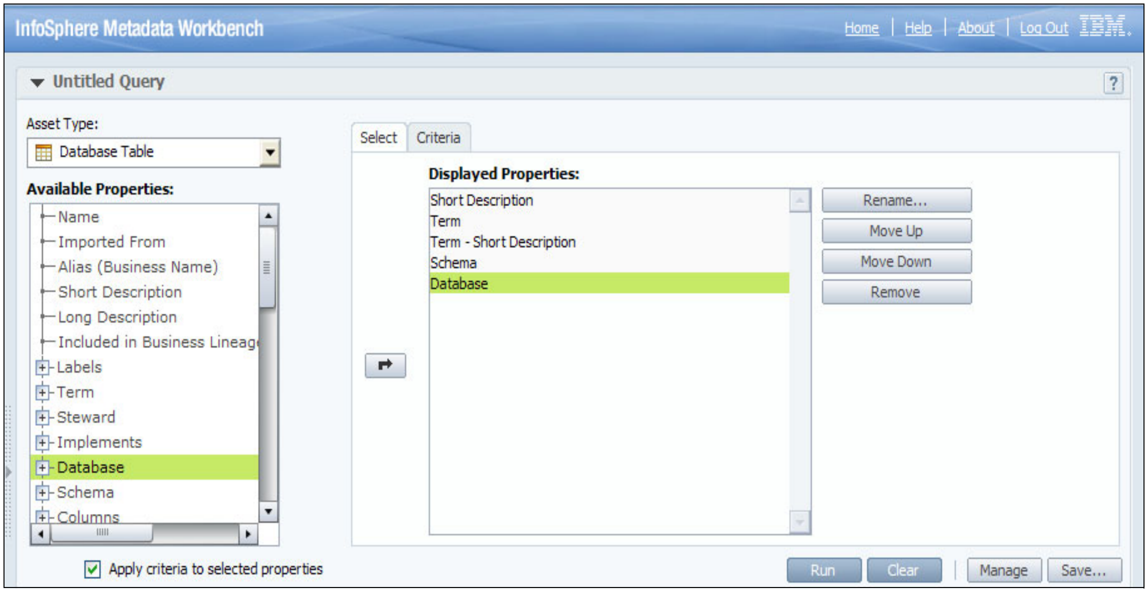
When selecting any analysis report, a graphical display of the data flow or impact analysis is immediately displayed, depicting the source or impacted data sources, and the target or impacted BI reports.

12.7.2 Querying

With InfoSphere Metadata Workbench, casual users can create ad-hoc reports or queries against the information of InfoSphere Information Server. Administrators can further publish such queries, making them available for all users of InfoSphere Metadata Workbench.

Several prebuilt queries are delivered with InfoSphere Metadata Workbench. With these queries, a user can easily report on InfoSphere DataStage jobs, BI reports, and database or model assets.

Figure 12-27 shows the window where you build the query.



Asset Type: Database Table

Displayed Properties:

Short Description
Term
Term - Short Description
Schema
Database

☑ Apply criteria to selected properties

Figure 12-27 Creating a query from InfoSphere Metadata Workbench

To build a query, complete the following steps:

1. From the home page of InfoSphere Metadata Workbench, click **Query**. Select the **New Query** link.

   Alternatively, from the left navigation pane, on the **Discover** page, click **New Query**.

2. Select the asset type of information that you want to query. For example, select a database table to query upon the description, term assignment, and dependent job of the table.

3. From the list of available properties, select the properties to include within the query results.

   a. Double-click a property, or select a property and click the **Add** button to include the selected items.

      The list of available properties also includes relationships. You can expand relationships to browse and selects its properties. In this example, a database table includes glossary terms.

   b. Expand the property to select and include the properties of a term, including the description or category.

4. Manage the view of the properties (Figure 12-27 on page 429):

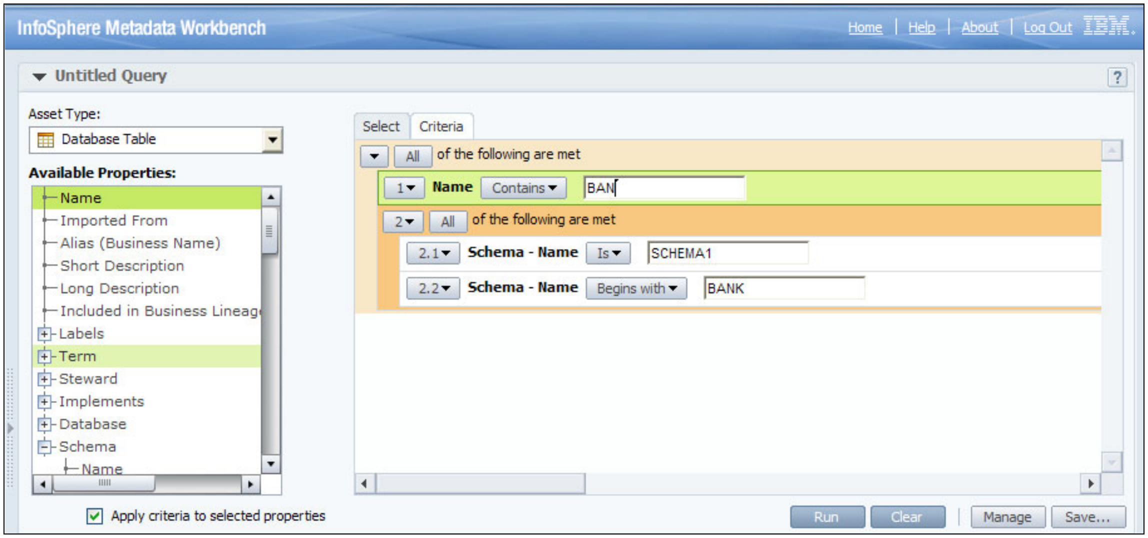
   – Click **Move up** or **Move down** to re-order the display of the properties.
   – Click **Rename** to alter the display name of the output column header.
   – Click **Remove** to remove a property from the output display.

5. Select the **Criteria** tab to add conditions that limit the results of the query. From the toolbar menu, click **Add New Condition**.

   By default, all conditions must match so that the query returns results. Click **All** to change this logic to return results if any of conditions match. For example, query on all database tables of a particular schema or of multiple schemes. Figure 12-28 on page 431 shows the **Criteria** tab where you can add conditions to your criteria.

   Conditions can be nested, as necessary. Multiple values can be evaluated for a property. The following conditions can be used to evaluate a property:

   – Contains or Does Not Contain
   – Contains any Word
   – Begins With or Does Not Begin With
   – Ends With or Does Not End With
   – Is or Is Not
   – Equals or Does Not Equal



Asset Type: Database Table

All of the following are met
1 Name Contains BAN
2 All of the following are met
2.1 Schema - Name Is SCHEMA1
2.2 Schema - Name Begins with BANK

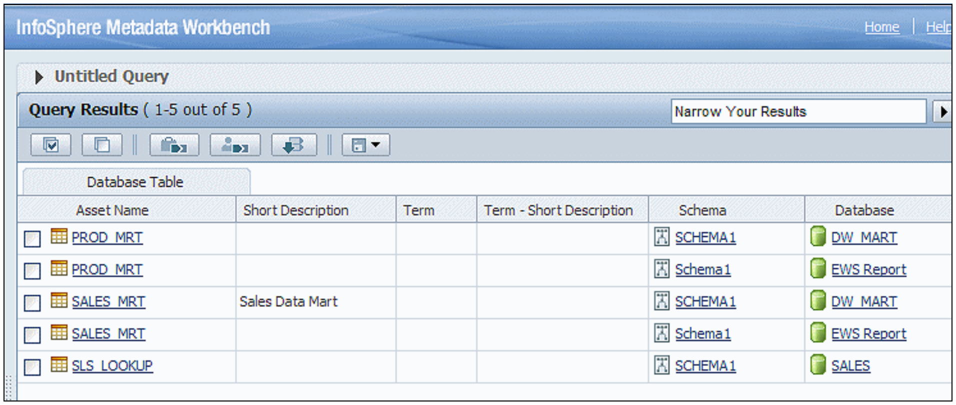
☑ Apply criteria to selected properties

Figure 12-28 Adding conditions to a sample query

6. Click **Run** to execute and display the query results
7. Click **Save** to save the query for future use. Upon saving, enter a description to identity the usage and output of the query.

An administrator of InfoSphere Metadata Workbench can then publish the query, enabling all users of InfoSphere Metadata Workbench to execute this query. Queries that are not published are only visible to the user who created them.

Figure 12-29 shows a query result.



Query Results (1-5 out of 5)

	Asset Name	Short Description	Term	Term - Short Description	Schema	Database
□	PROD_MRT				SCHEMA1	DW_MART
□	PROD_MRT				Schema1	EWS Report
□	SALES_MRT	Sales Data Mart			SCHEMA1	DW_MART
□	SALES_MRT				Schema1	EWS Report
□	SLS_LOOKUP				SCHEMA1	SALES

Figure 12-29 Sample query results from InfoSphere Metadata Workbench

For additional instructions and details about building a query, see InfoSphere Metadata Workbench online help.

12.8 Conclusion

To conclude this book, it is critical for companies to manage their metadata for information governance. Metadata management means providing the tools, processes, and environment to enable companies to easily and conveniently locate and retrieve information about their data with trust.

Many companies have some form of information integration solutions that integrate data from a disparate source system to one location, to generate reports and analysis that help them to make business decisions and set the right business strategy. During this type of implementation process, it is even more important to provide metadata management. This practice ensures that the reports and analysis you get from the solution are from the right data sources and contain the complete set of required data, with quality and accuracy.

IBM InfoSphere Information Server helps companies to achieve this goal and other business objectives and initiatives. It provides a single, unified platform and a collection of product modules and components so that companies can understand, cleanse, transform, and deliver trustworthy and context-rich information.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- *IBM InfoSphere DataStage Data Flow and Job Design*, SG24-7576
- *Information Server: Installation and Configuration Guide*, REDP-4596
- *InfoSphere DataStage Parallel Framework Standard Practices*, SG24-7830

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks publications, at this Web site:

ibm.com/redbooks

Online resources

These Web sites are also relevant as further information sources:

- IBM InfoSphere Information Server

  http://www.ibm.com/software/data/integration/info_server/
- IBM InfoSphere DataStage

  http://www.ibm.com/software/data/infosphere/datastage/
- IBM InfoSphere QualityStage

  http://www.ibm.com/software/data/infosphere/qualitystage/
- IBM InfoSphere FastTrack

  http://www.ibm.com/software/data/infosphere/fasttrack/
- IBM InfoSphere Information Analyzer

  http://www.ibm.com/software/data/infosphere/information-analyzer/

- IBM InfoSphere Discovery

  http://www.ibm.com/software/data/infosphere/discovery/
- IBM InfoSphere Business Glossary

  http://www.ibm.com/software/data/infosphere/business-glossary/
- IBM InfoSphere Metadata Workbench

  http://www.ibm.com/software/data/infosphere/metadata-workbench/
- IBM InfoSphere Data Architect

  http://www.ibm.com/software/data/optim/data-architect/

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Metadata Management with IBM InfoSphere Information Server

(0.5” spine)
0.475”<->0.875”
250 <-> 459 pages

Metadata Management with IBM InfoSphere Information Server

Information governance and metadata management

Typical information integration implementation process

InfoSphere Information Server modules and components

What do you know about your data? And how do you know what you know about your data?

Information governance initiatives address corporate concerns about the quality and reliability of information in planning and decision-making processes. Metadata management refers to the tools, processes, and environment that are provided so that organizations can reliably and easily share, locate, and retrieve information from these systems.

Enterprise-wide information integration projects integrate data from these systems to one location to generate required reports and analysis. During this type of implementation process, metadata management must be provided along each step to ensure that the final reports and analysis are from the right data sources, are complete, and have quality.

This IBM Redbooks publication introduces the information governance initiative and highlights the immediate needs for metadata management. It explains how IBM InfoSphere Information Server provides a single unified platform and a collection of product modules and components so that organizations can understand, cleanse, transform, and deliver trustworthy and context-rich information. It describes a typical implementation process. It explains how InfoSphere Information Server provides the functions that are required to implement such a solution and, more importantly, to achieve metadata management.

This book is for business leaders and IT architects with an overview of metadata management in information integration solution space. It also provides key technical details that IT professionals can use in a solution planning, design, and implementation process.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7939-00 ISBN 0738435996