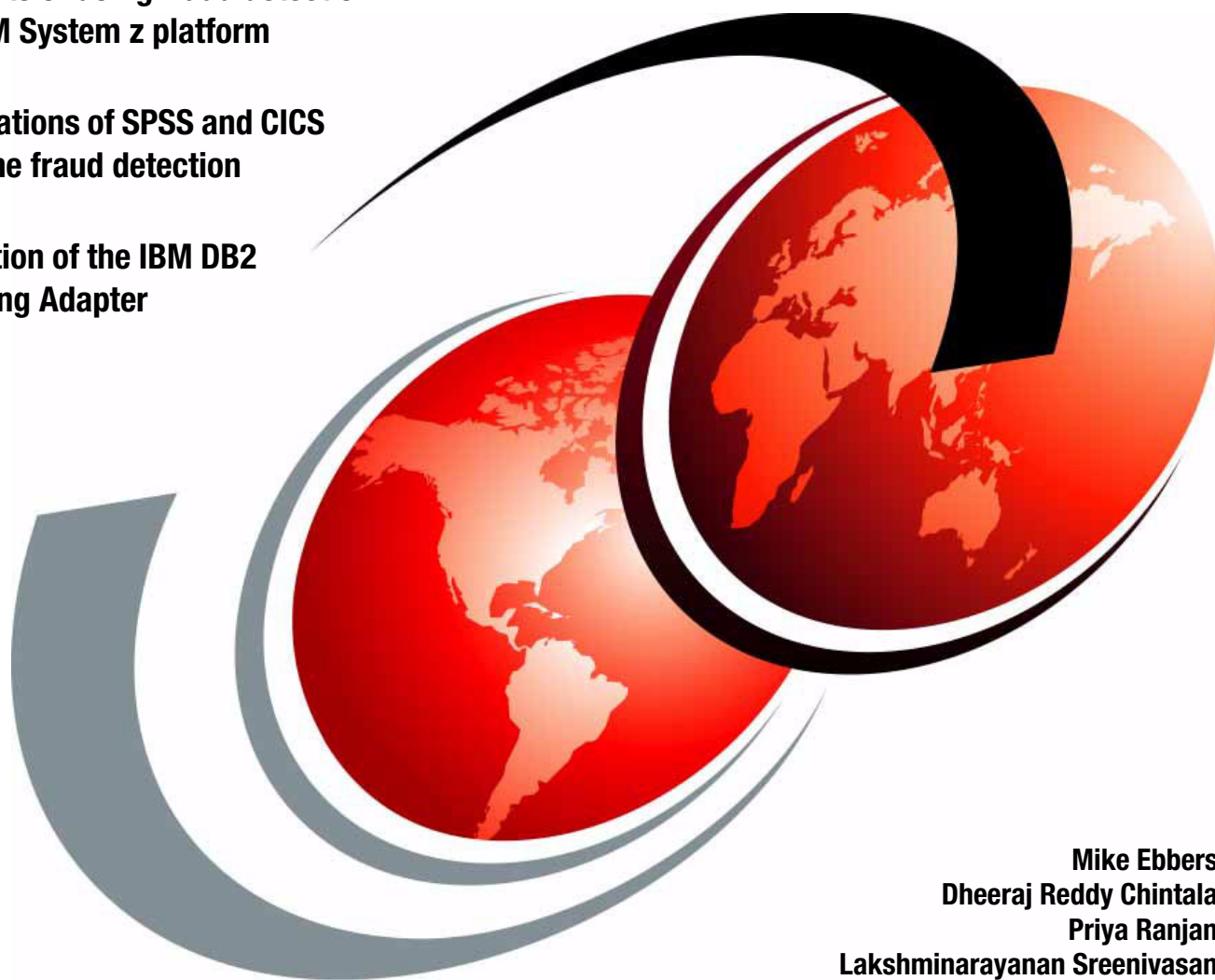


# Real-time Fraud Detection Analytics on IBM System z

The benefits of doing fraud detection  
on the IBM System z platform

Demonstrations of SPSS and CICS  
in real-time fraud detection

A description of the IBM DB2  
UDF Scoring Adapter



Mike Ebbers  
Dheeraj Reddy Chintala  
Priya Ranjan  
Lakshminarayanan Sreenivasan

**Redbooks**





International Technical Support Organization

**Real-time Fraud Detection Analytics on IBM System z**

April 2013

**Note:** Before using this information and the product it supports, read the information in “Notices” on page v.

**First Edition (April 2013)**

This edition applies to Version 10 of IBM DB2 for z/OS, Version 4.1 of IBM CICS, and Version 15 of SPSS.

**© Copyright International Business Machines Corporation 2013. All rights reserved.**

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

# Contents

<b>Notices</b> .....	v
Trademarks .....	vi
<b>Preface</b> .....	vii
The team who wrote this book .....	vii
Now you can become a published author, too! .....	viii
Comments welcome .....	viii
Stay connected to IBM Redbooks .....	ix
<b>Chapter 1. Introduction</b> .....	1
1.1 Fraud as a cross-industry problem .....	2
1.2 Fraud solutions .....	2
1.2.1 Why fraud detection is difficult .....	3
1.2.2 The new fraud solution .....	3
1.3 Fraud in the health insurance industry .....	4
1.3.1 Detecting suspicious transactions .....	4
1.3.2 Analyzing historical and real-time data .....	5
1.3.3 Integrating fraud detection with case management .....	5
1.3.4 Optimizing software on System z .....	5
1.4 Fraud in property and casualty insurance .....	6
1.4.1 Recognizing suspicious transactions and behaviors .....	6
1.4.2 Analyzing data .....	6
1.4.3 Integrating with case management .....	7
1.4.4 Optimizing System z software .....	7
<b>Chapter 2. Solution overview</b> .....	9
2.1 Business need for this solution .....	10
2.2 Fraud detection process flow .....	12
2.3 More benefits .....	13
<b>Chapter 3. IBM Scoring Adapter</b> .....	15
3.1 IBM SPSS Modeler Server Scoring Adapter .....	16
3.2 About scoring .....	16
3.3 Scoring adapter for various database products .....	17
3.4 How the DB2 scoring adapter works on z/OS .....	18
<b>Chapter 4. Installation and configuration</b> .....	21
4.1 IBM SPSS Modeler Premium 15.0 .....	22
4.2 Installing SPSS Modeler Premium 15.0 .....	22
4.3 Installing Data Access Pack 6.1 .....	22
4.3.1 Deploying Data Access Technology .....	23
4.3.2 ODBC data sources .....	23
4.3.3 Installing .....	23
4.4 Setting up the ODBC connection to host database .....	23
4.5 Configuring the IBM SPSS Modeler Client 15.0 .....	24
4.6 Installing the SPSS Modeler Server Scoring Adapter 15 for DB2 for z/OS .....	24
4.7 Configuring the IBM SPSS Modeler 15 Scoring Adapter for DB2 for z/OS .....	25
<b>Chapter 5. Building a scenario</b> .....	27

5.1 Overview . . . . .	28
5.2 Understanding and preparing data . . . . .	28
5.3 Training and modeling . . . . .	29
5.3.1 Basics of neural networks . . . . .	31
5.4 Evaluation and deployment . . . . .	31
5.4.1 Publishing an antifraud model to the DB2 scoring adapter on System z . . . . .	32
5.5 Business rules logic program . . . . .	32
5.6 CICS front-end application . . . . .	33
<b>Chapter 6. Use case model . . . . .</b>	<b>35</b>
6.1 Use case . . . . .	36
6.2 Setting up the transaction . . . . .	36
6.3 Database design . . . . .	37
6.3.1 Static tables . . . . .	37
6.3.2 Dynamic tables . . . . .	37
6.4 A real-time illustration . . . . .	38
6.4.1 Logging on and installing code in the CICS region . . . . .	38
6.4.2 Starting the transaction . . . . .	40
6.5 Processing flow . . . . .	42
<b>Appendix A. Scenario predictors . . . . .</b>	<b>45</b>
<b>Appendix B. Transaction processing tables . . . . .</b>	<b>47</b>
<b>Related publications . . . . .</b>	<b>53</b>
IBM Redbooks . . . . .	53
Other publications . . . . .	53
Online resources . . . . .	53
Help from IBM . . . . .	54

# 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 grant 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites 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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

CICS®	IBM Watson™	SPSS®
Cognos®	InfoSphere®	System z®
DB2®	Redbooks®	z/OS®
i2®	Redbooks (logo)  ®	zEnterprise®
IBM®	Smarter Analytics™	

The following terms are trademarks of other companies:

Netezza, and N logo are trademarks or registered trademarks of IBM International Group B.V., an IBM Company.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.



# Preface

Payment fraud can be defined as an intentional deception or misrepresentation that is designed to result in an unauthorized benefit. Fraud schemes are becoming more complex and difficult to identify. It is estimated that industries lose nearly \$1 trillion USD annually because of fraud. The ideal solution is where you avoid making fraudulent payments without slowing down legitimate payments. This solution requires that you adopt a comprehensive fraud business architecture that applies predictive analytics.

This IBM® Redbooks® publication begins with the business process flows of several industries, such as banking, property/casualty insurance, and tax revenue, where payment fraud is a significant problem. This book then shows how to incorporate technological advancements that help you move from a post-payment to pre-payment fraud detection architecture. Subsequent chapters describe a solution that is specific to the banking industry that can be easily extrapolated to other industries. This book describes the benefits of doing fraud detection on IBM System z®.

This book is intended for financial decisionmakers, consultants, and architects, in addition to IT administrators.

## The team who wrote this book

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

**Mike Ebbers** is a Project Leader and Consulting IT Specialist at the ITSO, Poughkeepsie Center. He has worked for IBM since 1974 in the field, in education, and as a manager, mostly on mainframe systems. He has worked at the ITSO since 1994.

**Dheeraj Reddy Chintala** has over six years of IT experience, mainly in System z. Dheeraj holds a Bachelor's degree in Electronics and Communications Engineering. He has contributed to various IBM forums in the System z area. His expertise is in IBM CICS® and IBM DB2® on System z. He is working on providing solutions for a new data warehouse on System z.

**Priya Ranjan** has 10 years of IT experience in System z technology. He has worked as a Mainframe IT Architect and Data Architect and has expertise in providing cost-saving solutions for System z applications. He specializes in legacy transformation techniques and has developed proof of technology for modernizing core banking applications. He has also worked as a transition lead, where he was responsible for managing System z platform operations. Now, he is responsible for engaging with clients to suggest changes to their systems. He also provides long-term vision towards application modernization.

**Lakshminarayanan Sreenivasan** has over 10 years of IT experience, mainly in the Master Data Management and Business Analytics area. Currently, his focus is on building analytic solutions for various industry verticals to position IBM SPSS® products appropriately. He holds a Master's degree in both Computer technology and in Business Administration. He has contributed to IBM Redbooks initiatives and has been a speaker in various Technical conferences around the Business Analytics area..

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

Rich Conway  
**International Technical Support Organization, Poughkeepsie Center**

Paul DiMarzio  
**IBM Poughkeepsie**

Alex Louwe Kooijmans  
**IBM Poughkeepsie**

Yefim Shuf  
**IBM Watson™**

## 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](http://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](http://ibm.com/redbooks)

- ▶ Send your comments in an email to:

[redbooks@us.ibm.com](mailto: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>





# Introduction

This chapter describes the business process flows of several industries (banking, property/casualty insurance, and tax revenue), where payment fraud is a significant problem. This chapter shows how to modify these processes to incorporate new technological advancements that help you move from a post-payment to pre-payment fraud detection architecture.

Subsequent chapters describe a solution that is specific to the banking industry. These methods and tactics can be easily extrapolated to other industries.

This chapter covers the following topics:

- ▶ Fraud as a cross-industry problem
- ▶ Fraud solutions
- ▶ Fraud in the health insurance industry
- ▶ Fraud in property and casualty insurance

## 1.1 Fraud as a cross-industry problem

Any client interaction that involves a payment is at risk for fraud, waste, or abuse. Fraud can be defined as an intentional deception or misrepresentation that is designed to result in an unauthorized benefit. Fraud schemes are becoming more complex and difficult to identify as defrauders have become adept at hiding criminal instances in the hundreds of millions of payments and claims that are processed annually. It is estimated that industries lose nearly \$1 trillion USD annually because of fraud.

Many organizations that deal with financial payments are under great competitive pressure to make those payments as quickly and easily as a possible, forcing them to use a pay-and-chase strategy to deal with fraud. They make the payment immediately to avoid customer satisfaction issues, and then identify and attempt to recover fraudulent payments retroactively.

However, the pay and chase method includes a serious problem: organizations that make payments under high customer satisfaction pressures are also most likely to be under significant pressure to take cost out across the entire organization. Payments are often viewed under the seemingly opposed lenses of a differentiator (fast and hassle-free) and a cost center. The focus on cost leaves fraud investigation units understaffed and overburdened with case loads, and short of the technological resources that are needed to quickly identify fraud. Even though the payment part of pay-and-chase occurs, the chase part is underfunded. And even when a fraudulent payment is identified, the rate of recovery is inadequate because criminals are good at disappearing.

High volumes, fast payments, and lack of investigative resources allow tax fraud and healthcare fraud to be the top two economic crimes worldwide. You see that the pay-and-chase strategy is not an optimum solution.

## 1.2 Fraud solutions

The ideal solution avoids you making fraudulent payments without slowing down legitimate payments. Such a solution requires that you adopt a comprehensive fraud business architecture that applies advanced predictive analytics systematically towards the goal of reducing fraud, waste, and abuse. This architecture must specify state-of-the-art solutions that combat fraud at all possible points with the following strategies:

- ▶ Identify vulnerabilities. What kinds of schemes were used historically, are in play now, and are likely to be used in the future?
- ▶ Detect transactions. Armed with an understanding of vulnerabilities, create analytic models that detect potentially fraudulent payment requests at the time of intake.
- ▶ Evaluate workloads. As new fraud schemes are uncovered, prioritize and selectively assign cases for follow-up and potential prosecution.
- ▶ Conduct remediation. Efficiently manage cases and enforce compliance.
- ▶ Process appeals. Minimize appeals and efficiently manage the adjudication of those appeals that are brought forward.

Many models that are used to detect fraudulent transactions and to manage the investigation unit work with historical data only. Also, the investigation units are trying to recover payments that were made already. So, the linchpin of success for fraud business architecture, that is, detecting potentially fraudulent payment requests at the time of intake, was missing until now.

## 1.2.1 Why fraud detection is difficult

Virtually all large-scale and high-volume payment systems are handled through an online transaction processing (OLTP) system and the data of record is kept in a relational database (RDB). This setup continues to be the most efficient means of processing payments, both in terms of speed and cost. Unfortunately, the technical design point that makes an OLTP/RDB system efficient for processing transactions also makes it inefficient for analyzing those transactions. OLTP transactions are short-running, while analysis is long-running. RDBs are organized by rows, while analysis examines columns.

These conflicting design points resulted in the traditional approach of taking a copy of the OLTP data offline, prepping it for analysis, and then performing analysis on a different system so as not to affect the operational OLTP system. Hence, the pay-and-chase model is dominant, where analysis is done apart from the payment system while you work with data that represent payments that were made already.

## 1.2.2 The new fraud solution

To move from post-payment to pre-payment detection requires that the detection is moved closer to the payment system. Until recently, the state-of-the-art solution was to inject an interrupt into the transactional system that would make a callout to the analytic system for a judgment on the transaction; a good solution but one fraught with issues. The major issue is the latency that is injected into the process because of the resources and time that is used moving back and forth from one system to another. Such latency can break the service level agreements in high-volume payment scenarios. Because of this limitation, most payers can perform this type of in-line detection only on a small sample of payments while potentially allowing some fraudulent payments through. Another problem is that stale data feeds the models because this solution relies on snapshots of the data.

However, two recent technological breakthroughs have combined to finally make real-time and in-transaction pre-payment fraud detection a reality. In 2011, IBM released the IBM DB2 Analytics Accelerator (IDAA), which directly links the analytics data with the operational data for payers using DB2 for IBM z/OS®, significantly increasing the freshness of the data that is used to build the predictive models and removing many of the burdens that are associated with the traditional offboard data warehouse. In 2012, IBM released IBM SPSS Modeler 15 Real-time Scoring with DB2 for z/OS, which allows the scoring of a payment to be made directly within the OLTP system, with only a small latency penalty versus making calls for scoring at run time to Web Services.

This combination of IDAA and SPSS real-time scoring with DB2 for z/OS can lead to breakthrough results in all payment processing industries where fraud is a significant problem. This fraud detection architecture offers the following advantages:

- ▶ Provides pre-payment fraud detection with an adaptive system that learns from the latest data.
- ▶ Detects suspicious transactions using predictive models and smarter business rules.
- ▶ Harnesses analytics to recommend the best method of interaction for each transaction.
- ▶ Dramatically reduces fraud and abuse costs.
- ▶ Pays valid and non-risky transactions faster and with greater certainty.
- ▶ Allows more efficient use of investigative resources, thus reducing costs and increasing the rate of return.
- ▶ Identifies potential fraud before payment, thus reducing expensive recovery actions.
- ▶ Detects multiple behaviors and schemes simultaneously.

- ▶ Analyzes claim and entity level information to detect suspicious patterns.
- ▶ Provides evidence-based identification of target cases for investigation.

## 1.3 Fraud in the health insurance industry

The annual loss because of healthcare fraud is tremendous, with fraudulent claims hidden among the hundreds of millions of claims that are submitted annually. Costs because of fraud and abuse continue to rise because healthcare organizations are pressured to pay claims quickly and investigators might lack the resources and technology to effectively identify and combat fraud.

The IBM Smarter Analytics™ Signature solution for healthcare on IBM zEnterprise® helps you detect fraud by using adaptive systems that learn from new data. The solution can identify suspicious transactions before payment, minimize losses, and recommend intervention methods. The solution simultaneously analyzes a range of behaviors at the claim and provider level to quickly identify suspect behavior. It also analyzes past behaviors to flag suspicious patterns. Because it can be integrated with your case management system, health plans can more effectively prioritize claims that are based on value, the likelihood of appeal, and other factors. You can also use the solution to move from post-payment analysis to pre-payment fraud detection without affecting efficiency or claim processing time, resulting in fewer erroneous payments and a higher return on investment (ROI).

Using sophisticated analytics, the solution recommends the most effective remedy for each case, which optimizes resources. In addition, the solution provides the following advantages:

- ▶ Detects suspicious transactions.
- ▶ Analyzes historical and real-time data.
- ▶ Integrates fraud detection with case management.
- ▶ Benefits from optimized software that is deployed on System z.

### 1.3.1 Detecting suspicious transactions

Before a claim is paid, the solution identifies potentially fraudulent and abusive behavior by using a unique combination of fraud detection models that contain thousands of scheme classifications across more than 20 medical areas. Data mining along with predictive analysis tools and reports displays a providers' past behaviors, flags suspicious patterns, and predicts the likelihood of future fraud.

The dashboard view displays summary data on the entire inventory of cases and comparative trends from fraud analyses. You also can view summary information about the results of fraud investigations, including the money that is saved, the number of cases that are closed, and other key performance indicators (KPIs).

The solution improves fraud detection ratio by simultaneously identifying multiple fraudulent behaviors and schemes. Antifraud investigators and auditors can zero in on questionable behavior and focus on the most egregious offenders by drilling down into detailed information about each provider or claim.



### **1.3.2 Analyzing historical and real-time data**

The solution embeds advanced algorithms directly into business processes, which allows you to detect fraud in real time and before funds are paid. The solution learns from the latest data, which protects against new approaches to fraud.

Using sophisticated analytics, the solution optimizes resources by recommending the most effective remedy for each case. For example, the system can recommend that a letter that requests payment be sent to resolve one case while recommending that a full investigation be opened in another case.

### **1.3.3 Integrating fraud detection with case management**

When you house operational data and the data warehouse together, you can take advantage of zEnterprise workload optimized fraud detection capabilities. Running analytics close to the data avoids the proliferation of data across multiple discrete servers. By performing analytics where the data is, you enable faster response times with fewer compute resources, and reduce potential security exposures and outages.

The tight integration with transactional systems, where claims data and processes typically are, also results in performance benefits with better resource management and workload assessment. By integrating fraud detection with case management, you can effectively prioritize claims that are based on value, the likelihood of appeal, and other factors. You can move from post-payment analysis to pre-payment fraud detection without affecting efficiency and performance targets for claims adjudication rates. In addition, using the solution on zEnterprise, you can enhance the mission-critical claims application with predictive analytics building blocks without needing to extract, transform, and load the data onto another operating environment.

### **1.3.4 Optimizing software on System z**

The solution embeds industry-proven fraud detection models and integrates the capabilities of the IBM Fraud and Abuse Management System (FAMS), IBM SPSS predictive analytics software, and IBM System z servers. It delivers an optimized analytics environment that takes advantage of new IBM DB2 on IBM z/OS and SPSS Modeler functionality and close proximity to data that is on System z.

The z/OS DB2 technologies for scoring and temporal data combined with SPSS scoring capabilities allow scoring to be updated at the time of the transaction. Data from current transactions can be factored into the analysis along with updated historical data. Also, the feature set that is used in the call to a scoring adapter can be updated.

Combining SPSS predictive analytics capabilities with System z means that a higher percentage of fraudulent claims can be detected before payment without negatively impacting claims processing efficiency, which reduces recovery costs. The SPSS scoring algorithms perform mathematical calculations that require an advanced floating point architecture. The zEnterprise system is designed to optimize this type of analytic computation with improved hardware, compilers, and processors.

The zEnterprise system also supports clustering technologies that allow multiple servers to work together to provide 24x7 operations, which translates into fewer planned or unplanned server outages. Clustered servers can be situated in multiple locations and the solution can float across the clustered systems, which improves efficiency and further reduces disruptions and outages.

## 1.4 Fraud in property and casualty insurance

Fraudulent claims constitute a significant percentage of all claims annually, making cost recovery costs a key goal for insurers. At the same time, fraud methods, such as billing for more extensive services than required and the staging of accidents, have become more sophisticated. Such fraud results in revenue losses for insurers and leads to higher premiums for consumers. Insurers need fraud detection and prevention that detects new fraudulent schemes, recognizes patterns of non-compliant behavior, and identifies who is likely to commit future fraud. The IBM Smarter Analytics Solution on zEnterprise for insurance uses sophisticated predictive analytics to identify potentially fraudulent claims before payment and to reduce the number of false claims that are paid.

Unlike point solutions that address only a single step in the process with a simple score, the solution integrates multiple capabilities to combat fraud across the entire claim lifecycle. The solution combines several analytical technologies that allow insurance organizations to perform the following tasks:

- ▶ Prevent fraud at policy submission.
- ▶ Predict fraud at claim intake.
- ▶ Identify fraud during adjudication.
- ▶ Discover fraud by examining patterns in data.
- ▶ Investigate fraud more efficiently by reducing false positives and accelerating the investigation process.
- ▶ Visualize trends and hotspots to continuously improve antifraud efforts.

### 1.4.1 Recognizing suspicious transactions and behaviors

Adjusters generally are able to detect fraud in only the most obvious cases, because of heavy caseloads and the requirement for fast claim processing. The IBM solution examines public and internal data in the background to provide real-time alerts whenever questionable behaviors, communications, or relationships are detected. The solution uses a wide array of tools to perform the following task:

- ▶ Validate the identities of all parties that are involved.
- ▶ Analyze the relationships among parties, including parties that are involved in other claims.
- ▶ Scrutinize the structured and unstructured data that is associated with the event.
- ▶ Monitor social media to identify inconsistencies with claim details reported to the carrier.

### 1.4.2 Analyzing data

The Smarter Analytics solution embeds advanced algorithms directly into business processes, which allows insurers to detect fraud in real time and before funds are paid. The solution also learns from the latest data. This protects against new fraud schemes and approaches.

After claims are identified as fraudulent, you can examine all the data that is associated with these claims to discover common patterns in content and relationships. By using the IBM Loss Analysis and Warning System (LAWS) and SPSS predictive analytics software, insurance companies can discover these patterns and apply them to the entire population of claims, increasing the potential to discover fraudulent ones.

### 1.4.3 Integrating with case management

Integrating insurance fraud detection with case management is similar to the process that is described for the healthcare industry in 1.3.3, “Integrating fraud detection with case management” on page 5. When you house operational data and the data warehouse together, you can take advantage of zEnterprise workload optimized fraud detection capabilities. Running analytics in close proximity to the data avoids the proliferation of data across multiple discrete servers. By performing analytics where the data is, you enable faster response times with fewer compute resources, and reduce potential security exposures and outages. Data that is used for real-time analysis is more accurate, resulting in fewer false positives. Compliance issues can arise if claim payments are stopped without sufficient reason. Reducing false positives improves confidence in the claims that are stopped for further investigation of potential fraud.

The tight integration with transactional systems, where claims data and processes typically are, also results in performance benefits resulting in better resource management and workload assessment. By integrating fraud detection with case management, you can effectively prioritize claims that are based on value, the likelihood of appeal, and other factors. You can move from post-payment analysis to pre-payment fraud detection without affecting efficiency and performance targets for claims adjudication rates. In addition, using the solution on zEnterprise, you can enhance the mission-critical claims application with predictive analytics building blocks without needing to extract, transform, and load the data onto another operating environment.

### 1.4.4 Optimizing System z software

This solution embeds industry-proven fraud detection models that contain thousands of scheme classifications. It integrates IBM fraud analysis and scoring models, System z, DB2 for z/OS, SPSS, IBM Cognos®, IBM i2®, and Entity Analytics. All these factors combine to deliver a highly optimized analytics environment that takes advantage of new DB2 for zOS and SPSS Modeler functionality and of the close proximity to data that is on System z.

The advantages of the z/OS DB2 technologies for scoring and temporal data that is combined with SPSS scoring capabilities, along with SPSS predictive analytics capabilities combined with System z, are similar to those advantages described for the healthcare industry in 1.3.4, “Optimizing software on System z” on page 5. The same situation applies for System z support of clustering technologies.





## Solution overview

The new technology in fraud detection architecture enables real-time scoring of the transactional data in DB2 for z/OS, which delivers more profitable decisions at the point of customer impact.

This chapter covers the following topics:

- ▶ Business need for this solution
- ▶ Fraud detection process flow
- ▶ More benefits

## 2.1 Business need for this solution

Organizations can use real-time scoring to directly incorporate the newest and most relevant data into the decision making process in real time. You can use this scoring to proactively and repeatedly reduce costs and increase productivity. For example, you can complete the following tasks:

- ▶ Improve fraud identification and prevent disasters by identifying and denying transactions with a high probability of fraudulency, such as insurance claims, credit card purchases, and immigration.
- ▶ Increase revenue per customer ratio by improving marketing campaign success with higher response rates, better cross-up-sell rates, and lower mailing costs.
- ▶ Heighten customer retention with more applicable offers.
- ▶ Improve customer service with visibility into the current inventory. You can proactively assess and demand requirements to ensure that products are available on demand.

Real-time scoring capability on DB2 for z/OS, combined with System z service quality, offers improved accuracy, speed, and performance while reducing the overall cost and complexity. The timeliness of real-time scoring increases efficiency and reliability; also, it is easily implemented.

This solution uses IBM SPSS Modeler Client 15.0 to develop a modeler stream that is based on a business scenario and requirements. For example, a model can be developed for credit card fraud detection that can predict the probability of a credit card transaction being fraudulent. This model references historical data and predicts a probability score that can be used to decide whether to accept, reject, or keep a transaction on hold. After development, the model is published to the Scoring Adapter for DB2 on z/OS. Then, the model is started with a native SQL call to the user-defined function that returns the predicted score that you can use to make business decisions. Figure 2-1 shows this process.

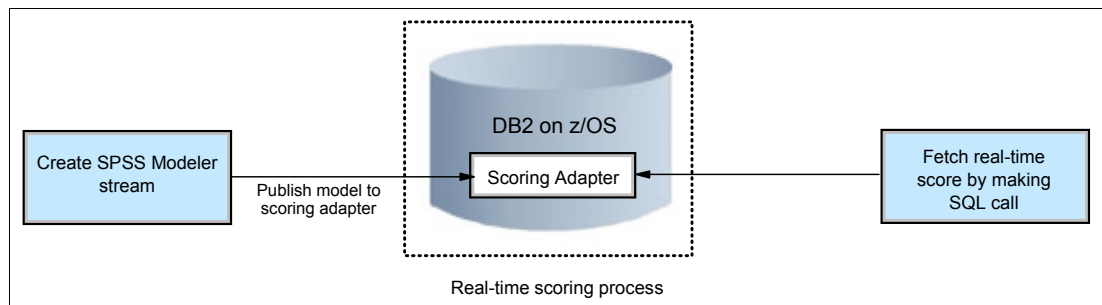


Figure 2-1 Block diagram of real-time scoring by using the DB2 Scoring adapter

With an alternative scoring method, the stream is developed by using IBM SPSS Modeler Client and the model is published to IBM SPSS Collaboration and Deployment Services (C&Ds) running on Linux. This model is called with a web service call from an application program and then fetches historical data from the database. The model processes the data and produces a real-time score that you can use to make business decisions. Figure 2-2 shows this alternative solution.

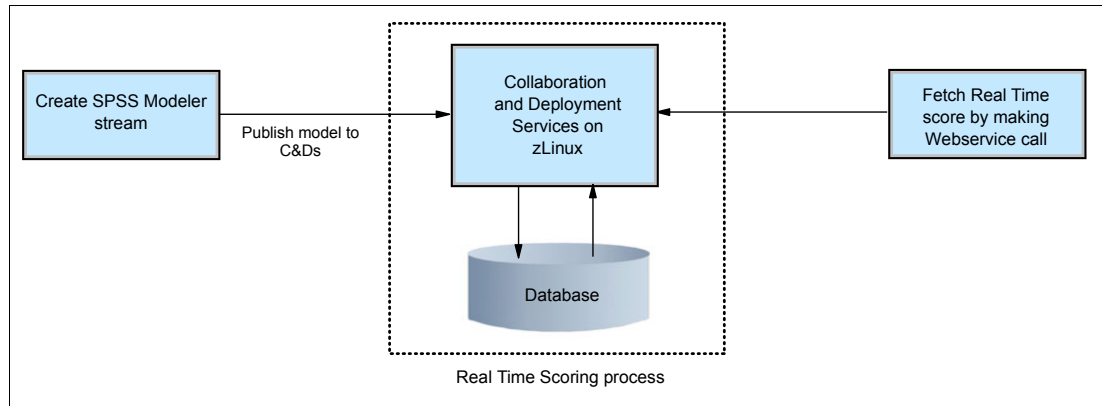


Figure 2-2 Block diagram of real-time scoring by using C&Ds

In the first approach, the model is run and historical data is processed in the DB2 environment. In the alternative approach, the process occurs in the C&Ds environment. Historical data is outside of a C&Ds environment and must travel from database to C&Ds for processing, which causes this approach to be slower than the DB2 scoring adapter approach. This book examines and describes the first approach.

## 2.2 Fraud detection process flow

This book describes an example of transactional fraud that shows how predictive analytics can continually identify new fraud patterns. This task is achieved by building a rules algorithm application to predict the potential fraud for a transaction, which then informs real-time systems to stop fraud before it happens, as shown in Figure 2-3.

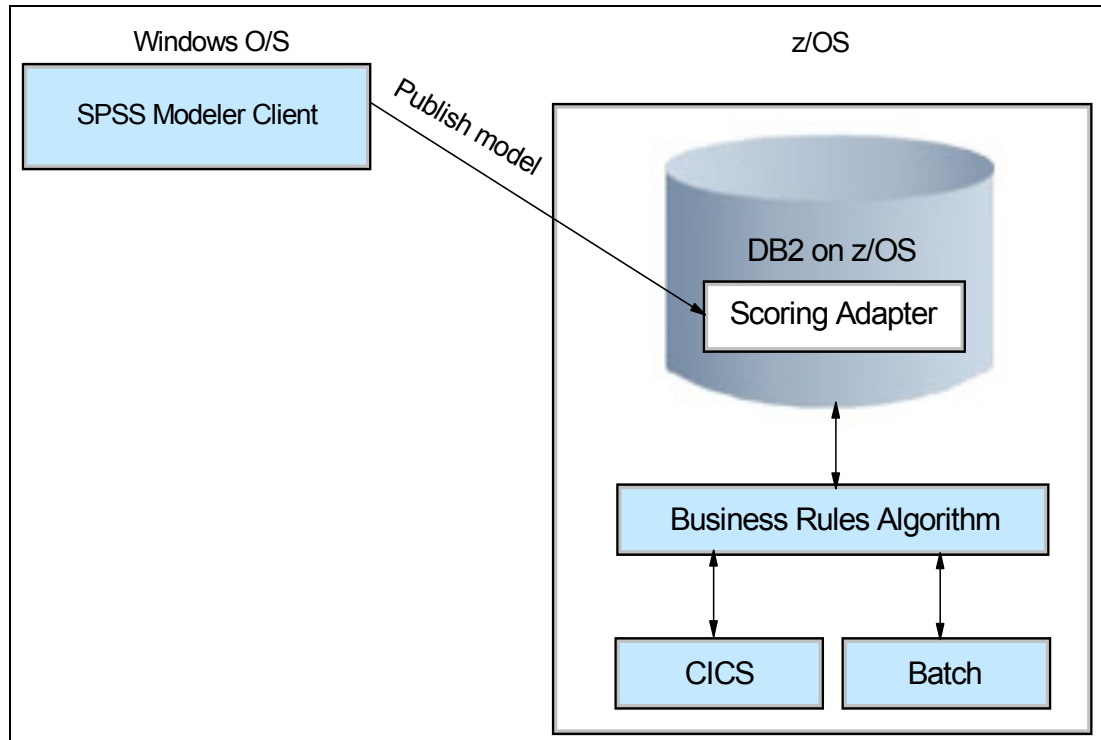


Figure 2-3 Transactional fraud process in DB2 on System z

Here are the processes:

1. Gather all data from the appropriate data sources in DB2 for z/OS.
2. For the response model, look at all customers in a certain time period who made fraudulent and non-fraudulent transactions, train the data set, and then build patterns and scores.
3. The customer-based scores are processed on modeler streams and then published by using the DB2 Scoring Adapter on System z.
4. An alert message is displayed on the custom center application. The alert is based on calculated propensity scores and the business rules algorithm, which is developed in COBOL.

A feedback loop is created to complete the analytical cycle, with models that are refined with newly captured information, all of which updates the customer profiles in real time.



Figure 2-4 shows how scoring from the business rule algorithm is configured and implemented in a real time, or batch, fraud system. The customer makes a credit card purchase through a custom application. The customer transaction does not trigger a fraudulent alert, so the status is GREEN and the payment is successful.

```
Payment Screen

Card Type      : CC (CC)-Credit Card (CO)-Common Card
Card Number    : 4444333322221111
Password       :
Expiration Date : 01 / 13 MM / YY
CVV Code       : 123
Payment Amount : 100.00
Merchant Code   : 12345
Merchant Name   : ABC Inc
Merchant Type   : 5211_

Press ENTER to Submit

Please Enter Card Number
```

Figure 2-4 Business rule algorithm scoring and configuration

## 2.3 More benefits

The new fraud detection process in DB2 on System z provides more benefits:

- ▶ It reduces data movement by scoring on the same System z where the transactional application is running.
- ▶ It improves the accuracy of a business decision by scoring current relevant data.
- ▶ It increases the speed of a business decision and reduces transaction response time by scoring in the online transactional processing (OLTP) database, which is a more efficient process than sending data to an external scoring service and holding a transaction while you wait for a score to come back.
- ▶ It scales to larger transaction volumes.





# IBM Scoring Adapter

This chapter describes the IBM SPSS Modeler Server Scoring Adapter for DB2 on z/OS. This chapter reviews the concept of scoring, describes scoring adapters for various databases, and demonstrates the functions of the DB2 scoring adapter on z/OS.

This chapter covers the following topics:

- ▶ IBM SPSS Modeler Server Scoring Adapter
- ▶ About scoring
- ▶ Scoring adapter for various database products
- ▶ How the DB2 scoring adapter works on z/OS

## 3.1 IBM SPSS Modeler Server Scoring Adapter

IBM SPSS Modeler 15.0 is a high-performance predictive and text analytics workbench that provides unprecedented insight in to your data and delivers a positive return on investment (ROI) by embedding predictive analytics into business processes. Its new features include entity analytics that further extend capabilities to resolve relationships between data that originates from different sources and that address issues with data quality and matching.

IBM SPSS Modeler Server Scoring Adapters allow data to be scored by the generated models within the database, which improves performance.

## 3.2 About scoring

Scoring data is defined as deploying a predictive model on new data with an unknown outcome. This predictive model processes incoming data and produces a predictive score about the likelihood or probability of an event. For example, when an online payment transaction takes place, a predictive model processes the input data and provides a predictive score that gives a probability of the transaction being either genuine or fraudulent.

Figure 3-1 shows how a predictive scoring model works.

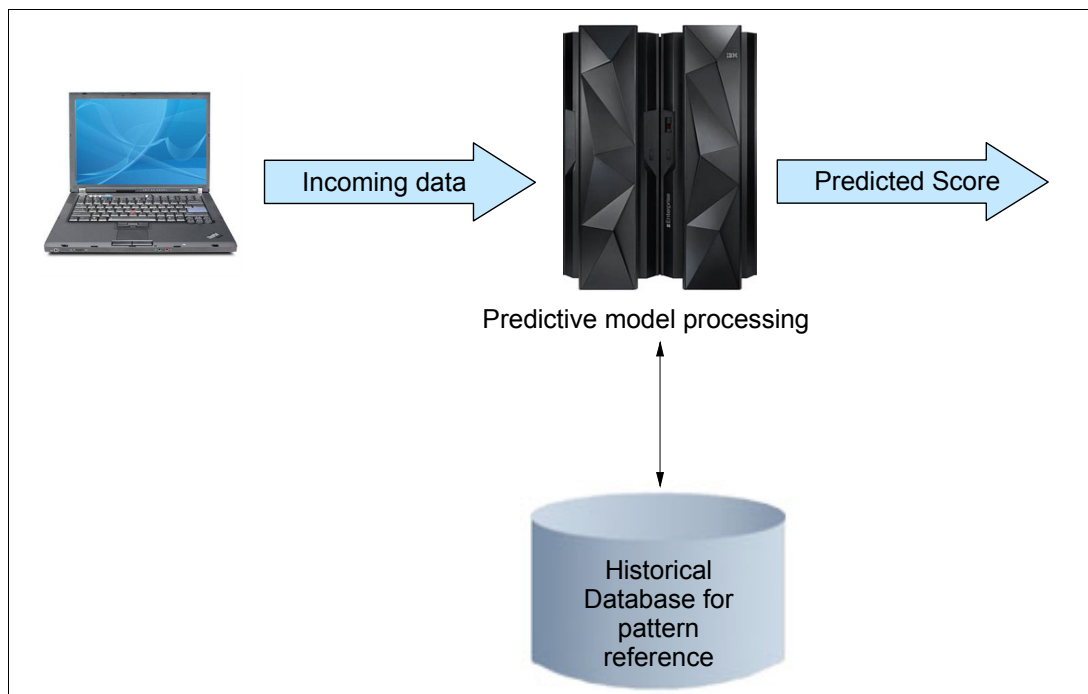


Figure 3-1 Predictive model process

Figure 3-1 illustrates that when a predictive model receives incoming data, it evaluates the input by using a historical data reference from a database and creates an output of a predicted score. This score provides a probability about an event for which a predictive analysis model is built.

Figure 3-2 shows a predictive model process with a scoring adapter. You can use the scoring adapter to evaluate each record and produce a score, or prediction, in the database without needing to export the data from database, run it through the model, and import it again.

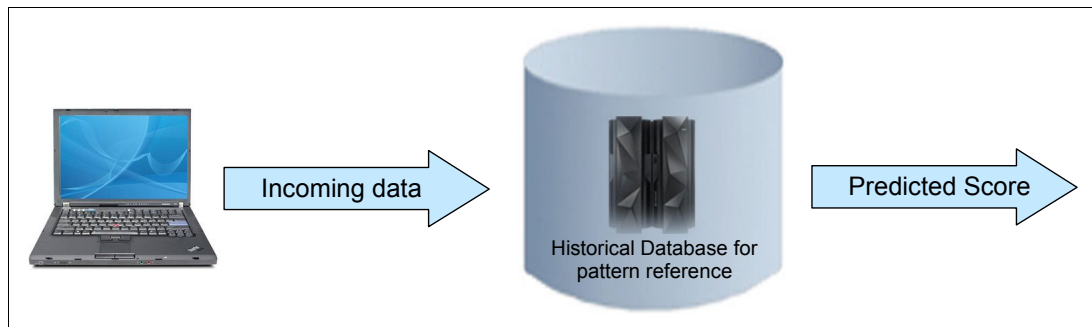


Figure 3-2 Predictive model process with a scoring adapter

### 3.3 Scoring adapter for various database products

Some databases allow SQL pushback of most of the SPSS Modeler model nuggets. In these cases, model scoring can be performed within the database, avoiding the need to extract the data before scoring. This pushback can use either native SQL within SPSS Modeler or, where available, use additional SQL scoring adapters that are tailored for different databases.

**Nuggets:** A model nugget is a container for a model. It is the set of rules, formulas, or equations that represent the results of your model building operations in IBM SPSS Modeler. The main purpose of a nugget is for scoring data to generate predictions, or to allow further analysis of the model properties. Opening a model nugget in the window enables you to see various details about the model, such as the relative importance of the input fields in creating the model. To view the predictions, you must attach and run a further process or output node.

When you successfully run a modeling node, a corresponding model nugget is placed on the stream canvas, where it is represented by a gold, diamond-shaped icon (hence the name “nugget”).

For more information about nuggets, go to:

[http://pic.dhe.ibm.com/infocenter/spssmodl/v15r0m0/index.jsp?topic=%2Fcom.ibm.spss.modeler.help%2Fgenerated\\_models.htm/](http://pic.dhe.ibm.com/infocenter/spssmodl/v15r0m0/index.jsp?topic=%2Fcom.ibm.spss.modeler.help%2Fgenerated_models.htm/)

IBM SPSS Modeler enables integration with IBM and non-IBM databases, and allows models to be deployed faster and with greater efficiency. User-defined functions (UDFs) can also be used in real time against transactional data, such as high volume sales, credit card payment, and customer service and claims transactions. Specifically, the new database integration includes the following items:

- ▶ In-transaction scoring that is embedded in IBM DB2 for z/OS
- ▶ In-database scoring that is embedded in IBM Netezza®
- ▶ In-database scoring that is embedded in Teradata
- ▶ IBM Infosphere algorithm synchronization
- ▶ IBM Netezza algorithm synchronization
- ▶ Multiple scoring input records
- ▶ SQL pushback enhancements

### 3.4 How the DB2 scoring adapter works on z/OS

You can use IBM SPSS Modeler Server 15, together with SPSS Modeler Server Scoring Adapter 15 for DB2 on z/OS, to add predictive analytics to OLTP applications that are running on z/OS. You use SPSS Modeler Server to create and train the models and publish them into DB2 on z/OS.

The scoring adapter for DB2 on z/OS provides a scoring engine that runs the UDF run time. The adapter defines a UDF that applications can start by using SQL to run the scoring models synchronously, in-line within their transactions, by using live transaction data as the input for scoring to maximize the effectiveness of scoring results.

Figure 3-3 shows a sample SPSS Modeler stream. You must publish a model nugget, for example, NN\_Is\_Fraud, to the scoring adapter.

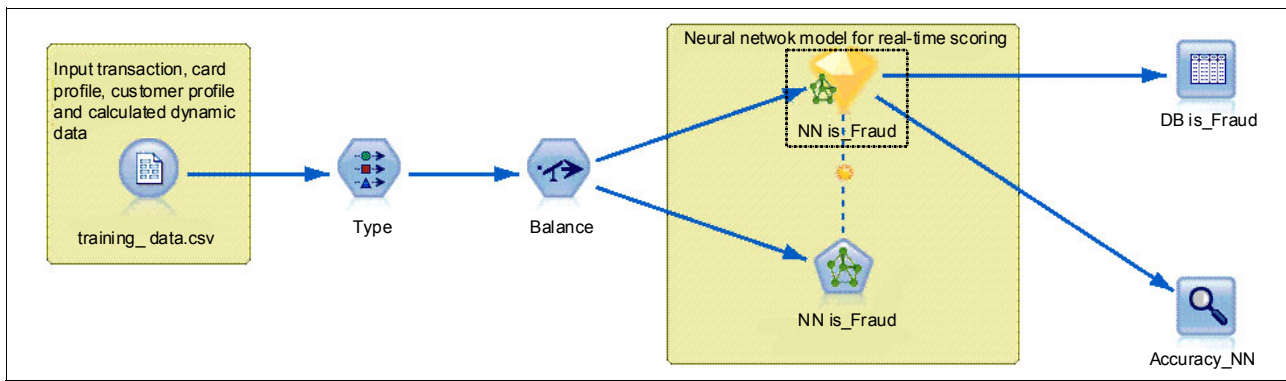


Figure 3-3 SPSS Modeler stream

You must establish a connection to the database before you publish the model nugget. After the connection is set up, publish the nugget to the scoring adapter by clicking **File** → **Publish for server scoring adapter**, as shown in Figure 3-4.

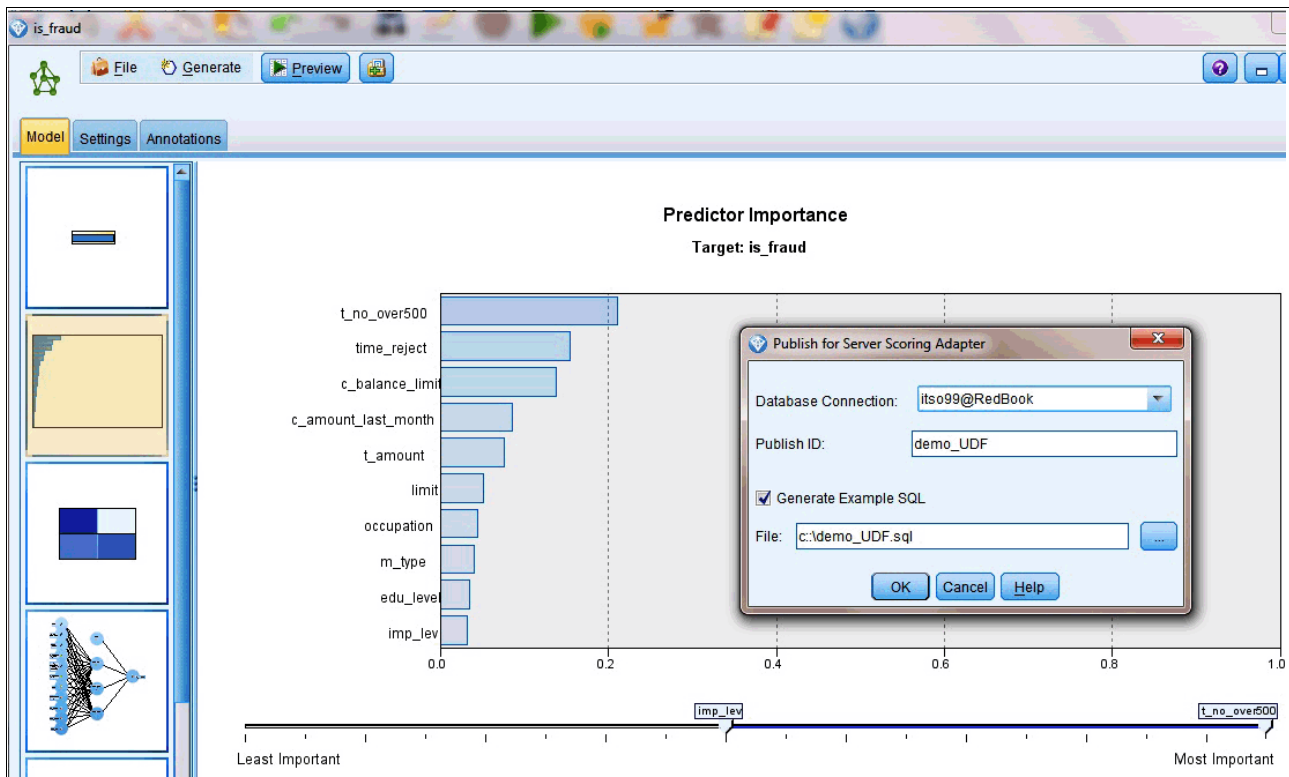


Figure 3-4 Publish for server scoring adapter option

When a model is published to a server scoring adapter, it generates a sample SQL statement. This SQL statement uses UDFs to start the SPSS model that was built earlier and generates a predictive score that can be used by a decision management system.

Example 3-1 shows a sample SQL statement for a scoring adapter.

*Example 3-1 Sample SQL statement for a scoring adapter for DB2 on z/OS*

---

```
SELECT
  UNPACK
    (HUMSPSS.SCORE_COMPONENT('P',
      'demo_UDF',
      PACK(CCSID 1208, TO.C0,TO.C1,TO.C2,TO.C3,TO.C4,TO.C5,TO.C6,TO.C7,
        TO.C8,TO.C9,TO.C10,TO.C11,TO.C12,TO.C13,TO.C14,
        TO.C15,TO.C16,TO.C17,TO.C18,TO.C19,TO.C20,TO.C21,
        TO.C22,TO.C23
      )
    )
  ).* AS
  (
    C24 BIGINT,C25 DOUBLE,C26 DOUBLE
  )

FROM (
  SELECT
    TO."CARD_ID" AS C0,
    TO."T_AMOUNT" AS C1,
    TO."T_NO_OVER500" AS C2,
    TO."C_BALANCE_LIMIT" AS C3,
    TO."C_AMOUNT_LAST_MONTH" AS C4,
    TO."TIME_NO_3HOUR" AS C5,
    TO."TIME_AMOUNT_3HOUR" AS C6,
    TO."M_HISTORY" AS C7,
    TO."M_TYPE" AS C8,
    TO."T_TIME" AS C9,
    TO."E_TIME_LAG" AS C10,
    TO."TIME_REJECT" AS C11,
    TO."E_REJECT" AS C12,
    TO."LIMIT" AS C13,
    TO."GENDER" AS C14,
    TO."EDU_LEVEL" AS C15,
    TO."MAR_STAT" AS C16,
    TO."IMP_LEV" AS C17,
    TO."OCCUPATION" AS C18,
    TO."ECO_CAT" AS C19,
    TO."ANNUAL_SALARY" AS C20,
    TO."OWN_HOU_FLAG" AS C21,
    TO."VENDORS_IN_30_MINUTES" AS C22,
    TO."MERCHANT_COUNTRY" AS C23
  FROM ${TABLE0} TO
) AS TO
```

---

In the example, the SQL query returns Score(C24), Confidence(C25), and Normalized Propensity(C26) as output predicted scores that can be used by a decision management system for making runtime decisions. Running the scoring adapter SQL within the DB2 environment provides scalability and performance similar to DB2 for z/OS. This situation makes it possible to handle large transaction volumes and heavy workloads while you meet stringent response time requirements and SLAs.





# Installation and configuration

This chapter provides installation and configuration information for the IBM SPSS Modeler Client 15.0 on Windows, Data Access Pack V6.0, and IBM SPSS Modeler 15 Scoring Adapter for DB2 on z/OS.

This chapter covers the following topics:

- ▶ Installing the IBM SPSS Modeler Client 15.0 on Windows
- ▶ Installing the IBM SPSS Data Access Pack V6.1
- ▶ Setting up the ODBC connection to a host database
- ▶ Configuring the IBM SPSS Modeler Client 15.0
- ▶ Installing the IBM SPSS Modeler 15 Scoring Adapter for DB2 on z/OS
- ▶ Configuring the IBM SPSS Modeler 15 Scoring Adapter for DB2 on z/OS

## 4.1 IBM SPSS Modeler Premium 15.0

*IBM SPSS Modeler* is a data mining workbench that supports all the steps in the data mining process. IBM SPSS Modeler can work in local (client only) mode or distributed (client/server) mode. In our example, we use the local mode to develop the stream and use the UDF to publish to the Scoring Adapter that is in the z/OS system.

The client software of IBM SPSS Modeler Premium V5 is installed on the user's computer, where it provides the user interface and displays the data mining results. The client is a complete installation of the IBM SPSS Modeler software that runs only on the Windows operating system.

IBM SPSS Modeler is easy to learn because of its intuitive visual interface and because it requires no programming. The SPSS Modeler workbench offers a comprehensive range of data mining functions with powerful automation, including automated data preparation and multi-model creation and evaluation.

The SPSS Modeler is based on an open and scalable architecture that allows for the following functions:

- ▶ SQL pushback support
- ▶ Maximized use of infrastructure with multithreading, clustering, and usage of embedded algorithms (in-database mining)
- ▶ Integration with IBM technologies, such as IBM Cognos and IBM InfoSphere® Warehouse

## 4.2 Installing SPSS Modeler Premium 15.0

To install SPSS Modeler Premium 15.0, complete the following steps:

1. Log on to the system with administrative privileges.
2. Go to the location where the installation files were extracted and double-click `setup.exe`.
3. Follow the displayed instructions. You can modify the location when you are prompted for an installation directory, and then continue.

For more information, see the *IBM SPSS Modeler Client* PDF file that is included with the product package.

## 4.3 Installing Data Access Pack 6.1

The Data Access Pack deploys DataDirect Connect and ConnectXE for ODBC, which provides a comprehensive set of individual and database-specific drivers that use ODBC to deliver reliable connectivity to all major data stores, from relational databases to flat-file data.

**Closed source:** IBM SPSS Data Access Pack, DataDirect Connect, and ConnectXE for ODBC are closed, that is, they can be used only with IBM SPSS products. If you want to access databases with other applications, you need a more general solution.

### 4.3.1 Deploying Data Access Technology

Either the administrator or the user can install the appropriate Connect ODBC drivers. (Connect ODBC does not have a server component.) The drivers must be installed on the computer that is accessing the data (the computer where the IBM SPSS server product is running), or on the user's desktop computer, or both.

### 4.3.2 ODBC data sources

When you install Connect ODBC, you also install one or more ODBC drivers. Before you can use an installed driver, you must create and configure an ODBC data source for that driver.

An ODBC data source consists of two essential pieces of information:

- ▶ The ODBC driver that is used to access the data.
- ▶ The location of the database that you want to access. This database can be on any networked computer.

The ODBC driver and data source must be on the computer that is accessing and processing the data. Depending on the type of IBM SPSS application that you are using, that computer can be either a user's desktop computer or a remote server.

### 4.3.3 Installing

Drivers for all supported operating systems are available in an eAssembly that is provided with your IBM SPSS product. They are also available on a separate DVD. In our example, we use a Windows OS. To install the drivers, download the IBM SPSS Data Access Pack and extract the downloaded file by completing the following steps:

1. Using Windows Explorer, browse to the location of the executable (.exe) file for the IBM SPSS Data Access Pack.
2. Right-click the file and select **Run as Administrator**.
3. Follow the instructions that appear to complete the Installation.

For more information, see the *IBM SPSS Data Access Pack Installation Instructions* PDF, which comes with the product.

## 4.4 Setting up the ODBC connection to host database

To set up the ODBC connection to the host database, create an ODBC data source for the FRAUD database on the Modeler Client machine as a system DSN and name the DNS "FRAUD". Complete the following steps:

1. Open the ODBC Data Source Administrative window. On the Control Panel, click **Administrative Tools**, and then select **Data Sources (ODBC)**.
2. Create a System DSN. Click the **System DSN** tab and click **Add**.
3. On the drivers windows, select **IBM SPSS OEM 6.1 DB2 Wire Protocol**. You do not see this driver if the Data Access Pack is not installed.

4. On the ODBC DB2 Wire Protocol Driver Setup window, enter FRAUD as the DSN name, and enter other details to connect to the DB2 DB that is on the zOS system.
5. Test the connection. You should see a message that states that a connection is established.

## 4.5 Configuring the IBM SPSS Modeler Client 15.0

After you install the SPSS Modeler Client and the required Data Access Pack, you should be able to connect to the database to System z, extract the data, and build the models. (For details about building models, see Chapter 5, “Building a scenario” on page 27.) Complete the following steps:

1. On the All Programs window, open the IBM SPSS Modeler 15.0 and select **IBM SPSS Modeler 15.0**. Validate that the license is required and continue to open the SPSS Modeler Workbench.
2. In the Modeler Workbench, use the Database node from the Sources tab.
3. Select **Data Source as Fraud** and use the table view TRAINING\_DATA\_VIEW, and then select **Preview** to check that the connection is successful and that data is populated correctly.

## 4.6 Installing the SPSS Modeler Server Scoring Adapter 15 for DB2 for z/OS

The SPSS Modeler Server Scoring Adapter 15 for DB2 for z/OS is delivered as a non-priced feature of the IBM DB2 Accessories Suite for DB2 for z/OS Version 2 Release 2, which itself is a non-priced product. It is an SMP/E installable feature that runs as a z/OS UNIX application within the Workload Manager (WLM) application environment on the DB2 for z/OS UDF run time. To install this feature, you also must configure z/OS UNIX and a WLM application environment.

The scoring adapter depends on the PACK/UNPACK SQL that is provided by DB2 for z/OS V10 in authorized program analysis reports (APARs) PM55928 and PM56631. Ensure that you applied them to your DB2 subsystem.

The z/OS system administrator must complete the following steps to install the scoring adapter:

- ▶ Order function modification identifier (FMID) HHUMF00 for the relevant product ID (PID) (for example, 5697-Q02).
- ▶ Follow the installation instructions in the program directory by using SMP/E.

Table 4-1 lists the resulting SMP/E installed data sets and their contents:

Table 4-1 SMP/E installed data sets and their contents

Data set name	Member name	Member description
SHUMSAMP	HUMBIND	Contains statements to bind the modeler server adapter for DB2 zOS packages and grant accesses.
	HUMFREE	Contains a statement to free packages and plan for a scoring adapter.
	HUMSCFDB	Contains SQL statements to define the modeler server adapter for the DB2 zOS database and required tables.
	HUMUDFS	Contains statements to define scoring adapter UDFs to run SPSS published scoring models.
	HUMWLMP	Contains a PROC for Work Load Manager (WLM) Analytics application environment.
	HUMWLMA	Defines the core DB2 WLM environments for use by IBM SPSS Modeler adapter for DB2 for z/OS. It uses the DB2 DSNTWLMB batch program to install the definitions and activate them by using the current active service policy.
SHUMLOAD		Shared libraries (DLLs) and UDF executable.
SHUMHFS		Default Mount at path /usr/lpp/spss/cfscoring_<n.n>, where <n.n> is the version number of SPSS Modeler Server. All dynamically loaded and long named DLLs are externally linked from HFS to the HUMLOAD short name member during SMP/E installation.
SHUMDBRM	HUMSCRSQ	DB2 package.

## 4.7 Configuring the IBM SPSS Modeler 15 Scoring Adapter for DB2 for z/OS

To configure the scoring adapter, you must modify the configuration jobs that are provided in SHUMSAMP. You must make the changes that are specified in the job for your particular installation of the scoring adapter. Complete the following steps:

1. Use HUMSCFDB to create the database and tables that are needed by the scoring adapter and to grant their use.
  - COMPONENTS and PUBLISHED\_COMPONENTS tables should be defined by using HUMSPSS as the owner.
  - The COMPONENTS table has a BLOB column named DATA. Ensure that you modify the HUMSCFDB member to include a LOB table space definition and an auxiliary table definition to hold BLOB data.
2. Use HUMWLMP to set up the WLM PROC that the scoring adapter uses. Ensure that the SHUMLOAD data set is APF-authorized.
3. Use HUMWLMA to define and activate the WLM application environment for the scoring adapter.
  - Do not use a general WLMA application environment.
  - Do not share this WLM application environment with any other application.

4. Use HUMUDFS to create the scoring adapter UDFs. Use HUMSPSS as the owner of the UDFs.
5. Use HUMBIND to bind the scoring adapter packages and plan, and to grant its use.

After you complete these steps, the scoring adapter is ready to receive work. Use the IBM SPSS Modeler to create the models and publish them into DB2 for z/OS for the scoring adapter.



## Building a scenario

This chapter describes the steps that are needed to build an antifraud stream by using IBM SPSS Modeler. This chapter also covers how to train and publish the antifraud stream to the DB2 scoring adapter. Then, this chapter describes the steps that are needed to run the antifraud stream by using the DB2 scoring adapter that is described in Chapter 6, “Use case model” on page 35.

This chapter covers the following topics:

- ▶ Overview
- ▶ Understanding and preparing data
- ▶ Training and modeling
- ▶ Evaluation and deployment
- ▶ Business rules logic program
- ▶ CICS front-end application

## 5.1 Overview

For a typical transactional fraud detection business case, assume that a customer is making a credit card payment. At the time of payment, the bank analyzes the payment pattern on that particular credit card to detect the possibility of fraud. Depending on the analysis, the bank authorizes the transaction, keeps it on hold, or declines it, all in real time.

These steps are not performed during run time. You must build, train, and publish a real-time antifraud stream only once. Then, during run time, the antifraud stream that is deployed onto the scoring adapter runs and fetches the predicted score for the transaction.

You must periodically retrain the antifraud stream and republish it to DB2 scoring adapter to keep the stream intelligent, which ensures that the antifraud stream incorporates any changes in the historical data pattern, increasing the accuracy of predicted value.

## 5.2 Understanding and preparing data

To understand and identify the data that is required to build a real-time antifraud analytical solution from the historical data set, see Appendix A, “Scenario predictors” on page 45 and Appendix B, “Transaction processing tables” on page 47. The flow of preparing data is shown in Figure 5-1.

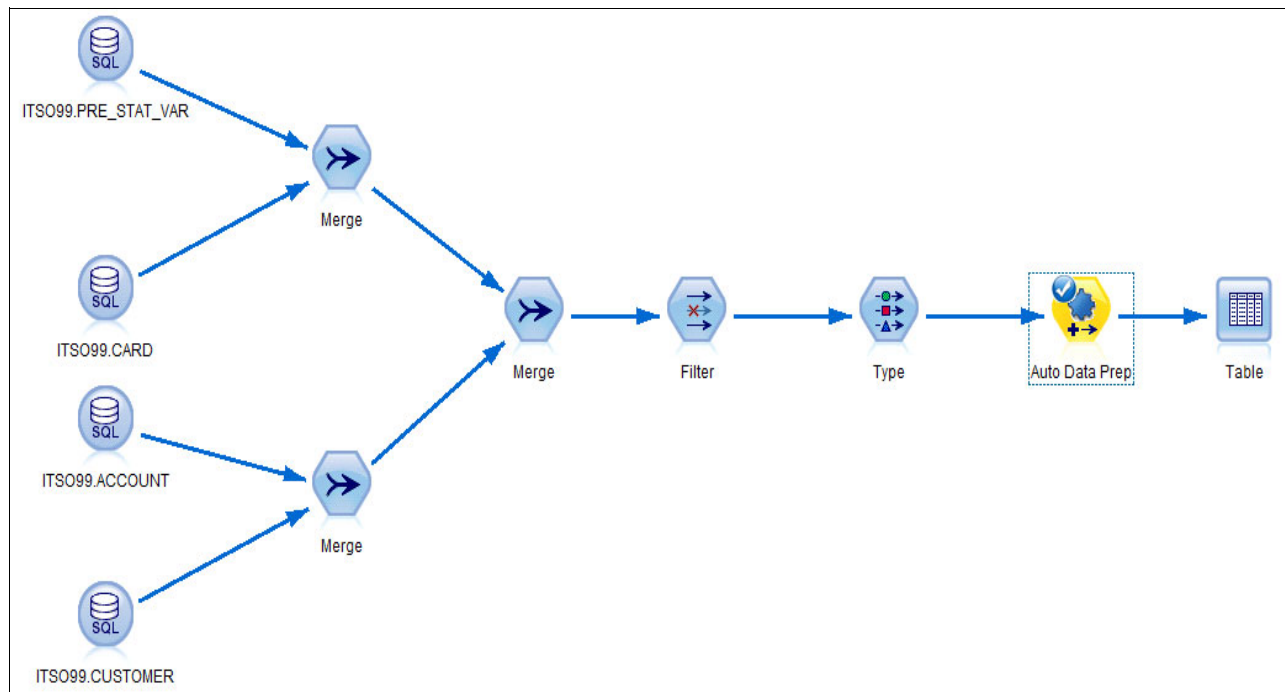


Figure 5-1 Data preparation



Preparing data for analysis is one of the most important steps in any data-mining project and, traditionally, one of the most time consuming steps. The Automated Data Preparation (ADP) node, which is shown in Figure 5-2, handles the task, by analyzing the data and identifying fixes, screening out fields that are problematic or not likely to be useful, deriving new attributes when appropriate, and improving performance through intelligent screening techniques. You can use the ADP node to quickly prepare data for data mining, without requiring prior knowledge of the statistical concepts involved.

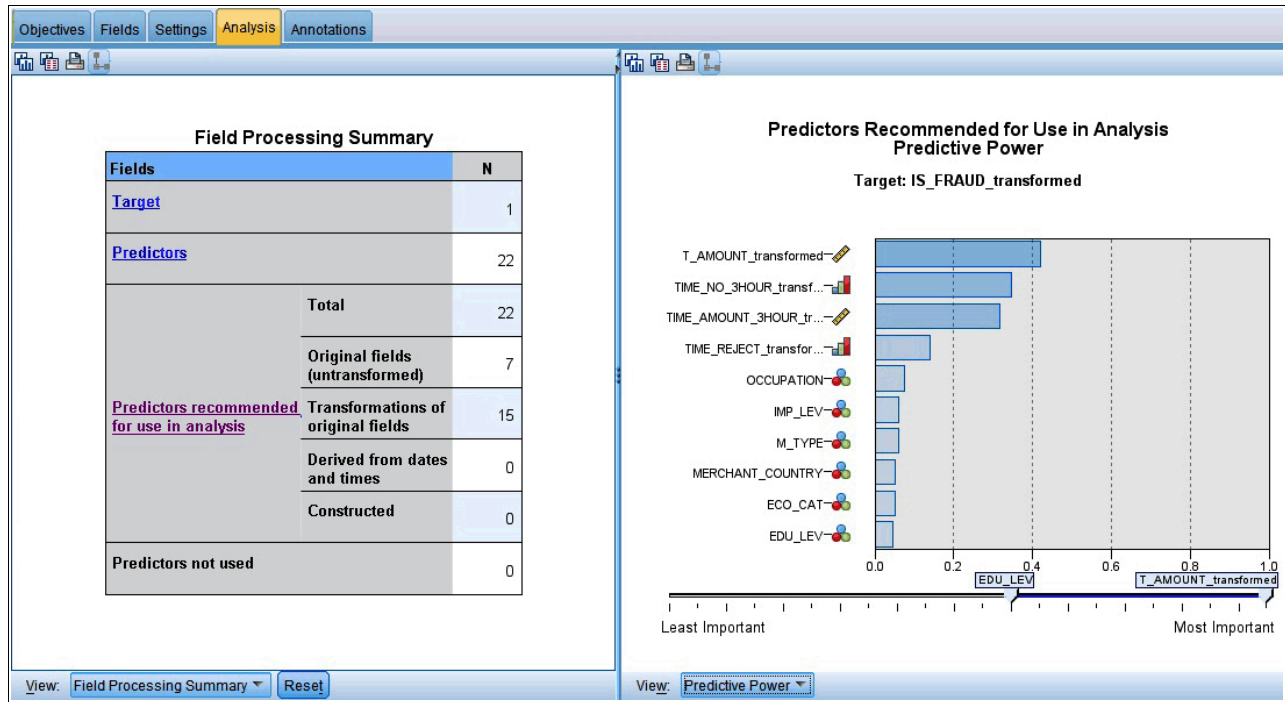


Figure 5-2 ADP analysis

For our example analysis, we weighed bank account details, customer profiles, and card details. The merging process involves merging two or more data sets with similar records but different attributes. The data is merged by using the same key identifier (card ID) for each record. The resulting data increases in columns or characteristics. We used the filter operation to remove or rename columns. The type node helps you identify and set the role and data type for an attribute. We were able to build a more accurate model with little direct data manipulation by running the ADP node to fine-tune the data processing.

You can work directly with the model settings if you must prove a certain theory, or want to build specific models. However, using the ADP node provides an advantage if you have limited time or a large amount of data to prepare.

### 5.3 Training and modeling

After the antifraud stream is built, it must be trained with historical transactional data so that it becomes intelligent in predicting fraudulent behavior. The historical data that is used for training the stream is known as *training data*. In our example, we defined a view of real-time transaction tables that we refer to as Training View.

To keep this stream intelligent and adaptive to changing pattern of transactions, this stream must be trained periodically with training data.

Use the historical analytical data, as outlined in the following steps, to build and train the antifraud stream. The stream detects a pattern of fraudulent credit card transactions, as shown in Figure 5-3.

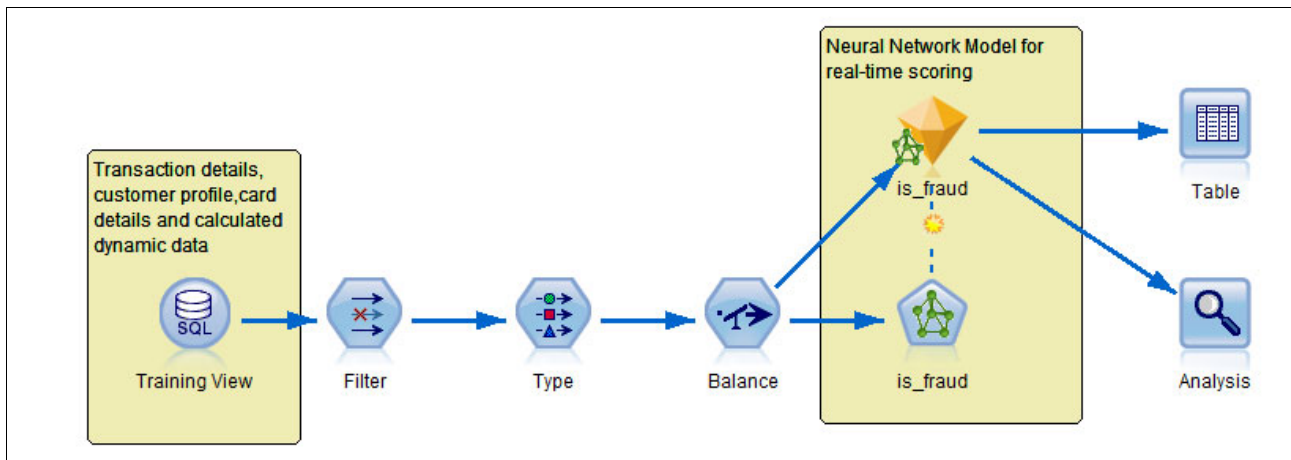


Figure 5-3 Antifraud training stream

1. Connect data to the analytical data view, which is a specific predictor data set view for building your model.
2. Use a filter node to standardize or remove columns.
3. The Type node specifies field metadata and properties, including which data is used as input and which data item is used for prediction. For example, you can specify a measurement level (continuous, nominal, ordinal, or flag) for each field, set options for handling missing values and system nulls, set the role of a field for modeling purposes, specify field and value labels, and specify values for a field.
4. The Balance node corrects imbalances in a data set, so it conforms to a specified condition. The balancing directive adjusts the proportion of records where a condition is true by the factor specified.

The data is now ready for modeling.

Modeling is an iterative process where data miners run several models by using the default parameters and then fine-tune the parameters, or revert to the data preparation phase to complete the adjustments that are required by the selected model. Usually, a data mining question is not answered by creating and running a single model. There are many ways to look at a problem, and SPSS Modeler offers a various tools to perform mining.

Typically, you test the results of several models before you decide which ones to deploy. In our example, we used the Neural Networks Model, after testing with other classification models, because it appeared to be a good model for determining antifraud patterns. You can use the output nodes, such as table and accuracy, to understand the model scores.

Streams and related antifraud prediction models are shown in Table 5-1.

Table 5-1 Antifraud prediction model summary table

<b>Stream name</b>	Anti-Fraud Prediction Model.str
<b>Purpose</b>	Predict the fraudulent transactions that are based on the historical information
<b>Input</b>	Bank account details, customer profile, and card details
<b>Target</b>	Is_Fraud
<b>Model</b>	Neural Network

### 5.3.1 Basics of neural networks

A neural network is a simplified model of how the human brain processes information. The model works by simulating many interconnected processing units that resemble abstract versions of neurons.

The processing units are arranged in layers. There are typically three parts in a neural network: an input layer, with units that represent the input fields, one or more hidden layers, and an output layer, with units that represents the target fields. The units are connected with varying connection strengths, or weights. Input data is presented to the first layer, and values are propagated from each neuron to every neuron in the next layer. Eventually, a result is delivered from the output layer.

The network learns by examining individual records, generating a prediction for each record, and adjusting the weights whenever the network makes an incorrect prediction. This process is repeated many times, and the network continues to improve its predictions until one or more of the stopping criteria are met.

## 5.4 Evaluation and deployment

To deploy the antifraud stream, use the trained model nugget with the complete data set and then publish the models to the Scoring Adapter, as shown in Figure 5-4. The operational data does not have the target field *is\_fraud used* in the Analytical data; here the scoring is done based on the identified fraudulent transactions.

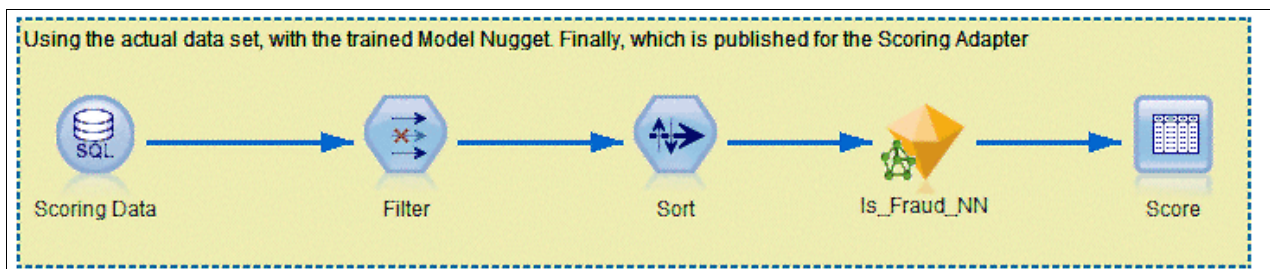


Figure 5-4 Anti fraud deployment stream

Complete the following steps:

1. Connect data to the Operational data view to score the probability of a fraudulent transaction.
2. Use a filter node to standardize and remove the columns.

3. Use the sort node to sort the data in a preferred order, for example, based on card-id in ascending order.
4. The trained model nugget node is copied and pasted to this stream, so the identified fraudulent pattern is used for scoring.
5. Use a table output node to view the scored data on the Modeler Workbench window. You can also use the export nodes to export the scoring to an external source.

### 5.4.1 Publishing an antifraud model to the DB2 scoring adapter on System z

To publish an antifraud model to the DB2 scoring adapter on System z, complete the following steps:

1. Right-click the Is\_Fraud\_NN golden nugget and select the **Edit** option.
2. On the Edit window, click **File** → **Publish for Server Scoring Adapter** to open the window that is shown in Figure 5-5.

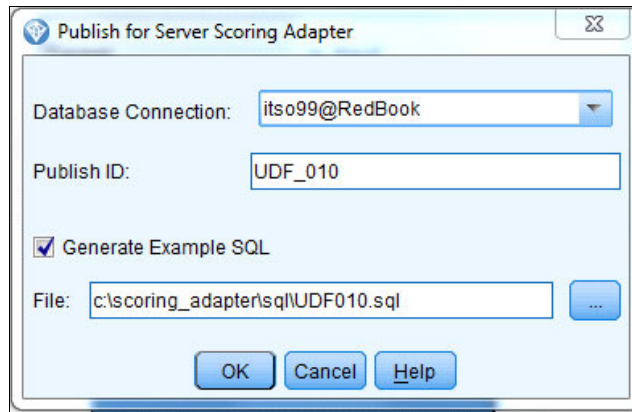


Figure 5-5 Publish for server scoring adapter

3. Enter the database connection (verify that you can establish a connection) details. Use a Publish ID and, if needed, generate the SQL statements in to a file by checking **Generate Example SQL** and specifying a file location. For more details, see Chapter 3, “IBM Scoring Adapter” on page 15.

The generated SQL statement uses UDFs to start an SPSS model stream that generates a predictive score that can be used by any decision management system.

## 5.5 Business rules logic program

A business rules logic program is built in System z by using scores that are generated from IBM SPSS Modeler. The decision making algorithm reads the score and decides the future course of action, for example, whether to hold the transaction or send it for further review. Further details are described in Chapter 6, “Use case model” on page 35.

## 5.6 CICS front-end application

A sample front-end user interface application was developed to start the business rules logic program that is built on System z. This front-end application receives inputs, such as a card number, payment amount, and CVV code. It declares whether the transaction is successful or whether it needs further verification. Further details are described in Chapter 6, “Use case model” on page 35.





## Use case model

This chapter illustrates a real-time situation and a sample transaction that includes the practical scenario that used for the demonstration, the steps that are involved in setting up the transaction, and a sample database design.

This chapter covers the following topics:

- ▶ Use case
- ▶ Setting up the transaction
- ▶ Database design
- ▶ A real-time illustration
- ▶ Processing flow

## 6.1 Use case

After you configure a scoring service, call the service in real time from a CICS transaction. The use case that we use involves a customer that is making an online credit card transaction with CICS as the front end for the transaction.

The customer initiates the transaction to buy something online. He enters the credit card details and continues. After he presses the Enter key, the fraud detection application reads the input data. It starts the UDF, which has the new the scoring adapter for DB2 for z/OS by using SQL in DB2. The predictive model uses the information that is provided by the customer and analyzes the history of the credit card, the previous patterns of the transactions, and other historical data, and then returns the score. A decision making algorithm reads the score and decides the future course of action.

We examine three possible situations in our sample transaction:

- ▶ **Successful transaction:** The transaction is fine with a low fraud probability. The decision making algorithm identifies a transaction as a good transaction if the score returned by the scoring adapter is less than 0.4.
- ▶ **Transaction on hold:** The transaction is doubtful with some probability of fraud. The decision making algorithm identifies this transaction if the score returned by the scoring adapter is more than 0.4 and less than 0.7. The application returns a message “Call customer care for more details”.
- ▶ **Transaction declined:** The transaction is a fraud. The decision making algorithm identifies a transaction as fraudulent if the score returned by the scoring adapter is more than 0.7.

## 6.2 Setting up the transaction

To install a CICS screen and integrate it with the scoring adapter, complete the following steps, as shown in Figure 6-1.

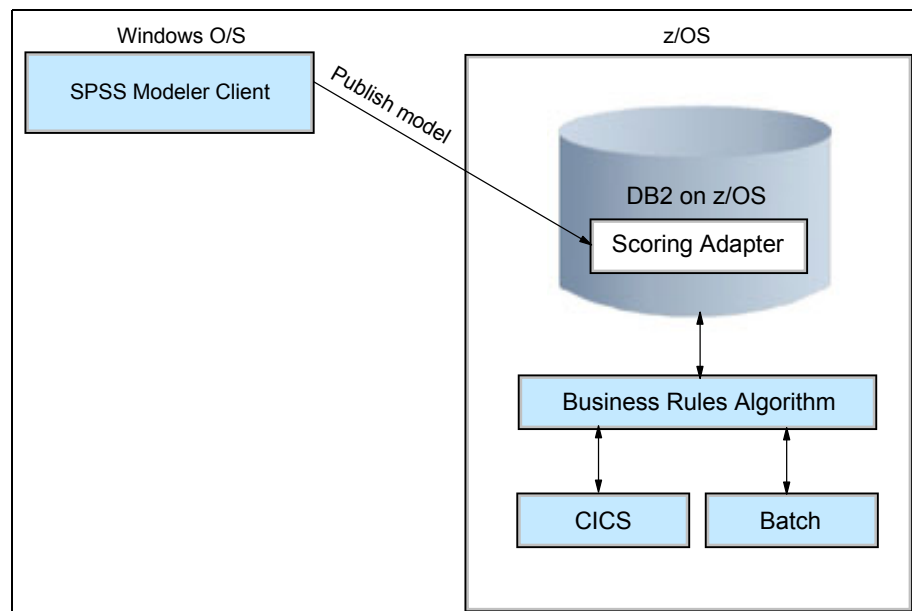


Figure 6-1 Transactional fraud process in DB2 on System z



1. Use SPSS Modeler to create a modeler stream and generate a predictive model from business data that is stored in a DB2 database.
2. Publish the model to the DB2 Scoring Adapter on System z.
3. Create a business rules algorithm by using COBOL. This program decides whether a transaction should be allowed or rejected or if further analysis is needed based on the score.
4. The COBOL program calls the scoring UDF to get the score for a particular transaction of a credit card.

The first two steps are described in previous chapters, particularly in Chapter 5, “Building a scenario” on page 27. Steps 3 and 4 are described in this chapter.

## 6.3 Database design

For demonstration purposes, we use a database that is filled with sample data for the scoring adapter to use and to return a score. The database design contains two types of tables: static and dynamic.

### 6.3.1 Static tables

Static tables are constant and do not change with a transaction. Static data includes a customer profile, an account profile, and the card profile that describes customer and account general information. There are three static tables: customer, account, and card; all table layouts are shown in Appendix B, “Transaction processing tables” on page 47.

- ▶ Customer table: Contains a customer profile, gender, education level, marriage status, occupation, and annual salary.
- ▶ Account table. Contains an account profile, number of cards, and account status.
- ▶ Card table. Contains a card profile, card type, card issue date, and card status.

### 6.3.2 Dynamic tables

Dynamic tables are updated in real time when a transaction is made. For table layouts, see Appendix B, “Transaction processing tables” on page 47. Dynamic tables include the following tables:

- ▶ AUTHORIZATIONS table: All the transactions. An entry is made in this table whenever a transaction is made. The table is updated with the credit card number, transaction amount, transaction time, merchant ID, country, and so on.
- ▶ FRAUD SCORING table: An entry is made in this table for each transaction with the score that is generated from the scoring adapter.
- ▶ CASE table: This table is updated for each transaction with a unique case ID that tracks the closure of a transaction if the transaction is placed under hold.
- ▶ PRE\_STAT\_VAR - Pre scoring table: This table is used for scoring by the adapter. Some columns, such as the amount of a transaction, are updated before you call the UDF to get a score because the transaction amount is important in calculating the score. Other columns are updated after a transaction is complete.

## 6.4 A real-time illustration

This section shows how you can log on, start a credit card transaction, and see the results of a sample transaction.

### 6.4.1 Logging on and installing code in the CICS region

Install all the programs and transactions in the CICS region. Start a 3270 terminal session to the mainframe that you use to log on to the CICS region, and then complete the following steps:

1. Open IBM Personal Communications and set up the Host Parameters that are shown in Figure 6-2.

	Host Name or IP Address	LU or Pool Name	Port Number
Primary	wtsc90.itso.ibm.com		23
Backup 1			23
Backup 2			23

Connection Options

Connection Timeout: 6 Seconds

Auto-reconnect

Try connecting to last configured host infinitely

Keep Alive

Enable Telnet Keep Alive

Keep Alive Time Out: 180 Seconds

Figure 6-2 Link parameter setup screen

2. Log on to CICS by entering the CICS region name, then enter your z/OS ID and password, as shown in Figure 6-3 and Figure 6-4.

```

MSG10
- International Technical Support Organization - ITS0

Enter: citso40

SCxxTS - TSO on SCxx (fill in the "xx")
CITS0xx - CICS systems (fill in the "xx")

Your IP Address: 9.79.255.8          Your Telnet Port: 03253
-----Last Command:
LU: SC90TC04      Sense Code:      Date: 11/14/12 Time: 22:16:30

```

Figure 6-3 Logon screen

```

CICS ITS040 REGION

Type your userid and password, then press ENTER:

Userid . . . .      Groupid . . . .
Password . . . .
Language . . . .
New Password . . . .

Signon to CICS
APPLID CITS040

```

Figure 6-4 CICS logon screen

## 6.4.2 Starting the transaction

After you install the transaction and corresponding programs in CICS, continue to start the transaction, as shown in Figure 6-5 and Figure 6-6 on page 41.

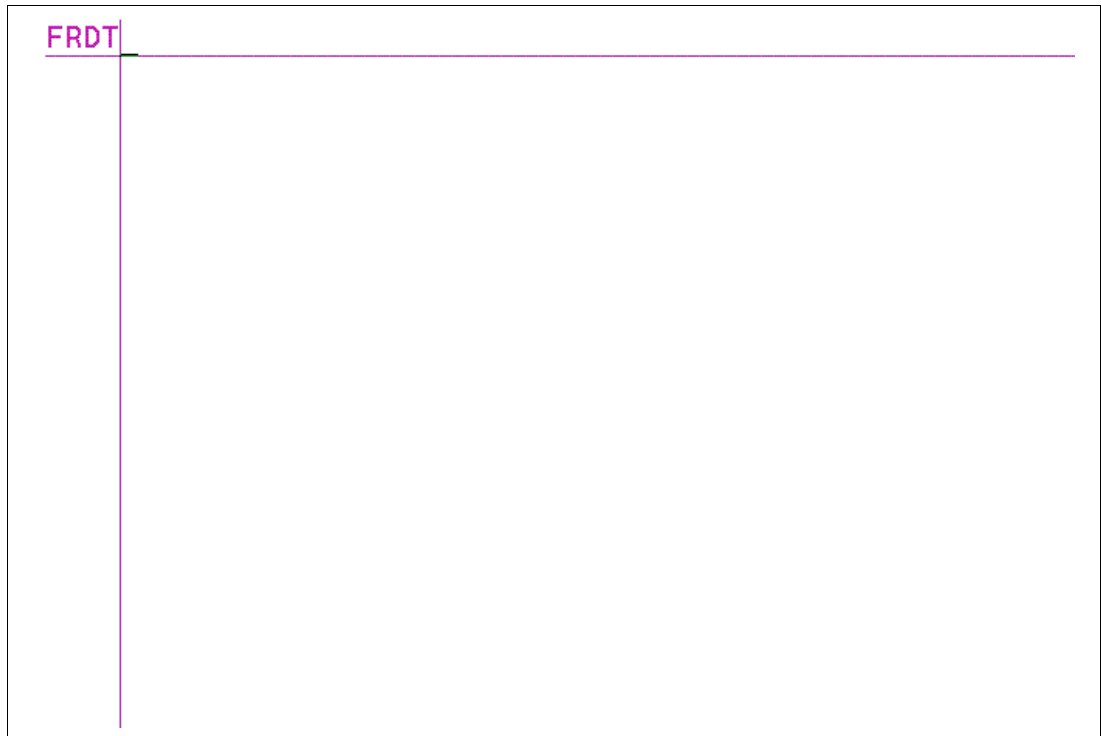


Figure 6-5 Starting the transaction

<u>Payment Screen</u>	
Card Type	: <u>CC</u> (CC)-Credit Card (CO)-Common Card
Card Number	: _____
Password	: _____
Expiration Date	: ___ / ___ MM / YY
CVV Code	: _____
Payment Amount	: 0.00
Merchant Code	: _____
Merchant Name	: _____
Merchant Type	: _____
Press ENTER to Submit	
Please Enter Card Number	
PF3=Exit PF4=Clear	

Figure 6-6 CICS scoring interface

Enter the details, as shown in Figure 6-7.

<u>Payment Screen</u>	
Card Type	: <u>CC</u> (CC)-Credit Card (CO)-Common Card
Card Number	: <u>4392268302891000</u>
Password	: _____
Expiration Date	: <u>12</u> / <u>15</u> MM / YY
CVV Code	: <u>1234</u>
Payment Amount	: <u>505.00</u>
Merchant Code	: <u>541233</u>
Merchant Name	: <u>IBM India Pvt Ltd</u>
Merchant Type	: <u>5065</u>
Press ENTER to Submit	
Please Enter Card Number	
PF3=Exit PF4=Clear	

Figure 6-7 Credit card transaction

## 6.5 Processing flow

After you enter the transaction details, complete the following steps:

1. Update the PRE\_STAT\_VAR table with the current transaction details that are required for the scoring.

2. Start the DB2 UDF on the Scoring Adapter to get the score of the current transaction.

At this point, it is decided whether the transaction is allowed, declined, or further information is needed.

For this user case, the following logic is used for decision making:

- Successful transaction: If the score is less than 0.4, the transaction is considered genuine and is allowed.
- Transaction on hold: If the score is greater than 0.4 and less than 0.7, the transaction is considered doubtful and is placed on Hold. The customer is asked to call customer care for details.
- Transaction declined: If the score is greater than 0.7, the transaction is considered a fraud and is declined.

3. Enter the transaction details into the AUTHORIZATIONS table.
4. Enter the score details into the FRAUD table.
5. Make an entry into the CASE table.
6. If the transaction is successful, update the PRE\_STA\_VAR table with all the statistics of the card for the current transaction. If the transaction is Declined or On Hold, then this table is not updated.
7. Display the score and the transaction status.

Figure 6-8 shows the steps in graphical form.

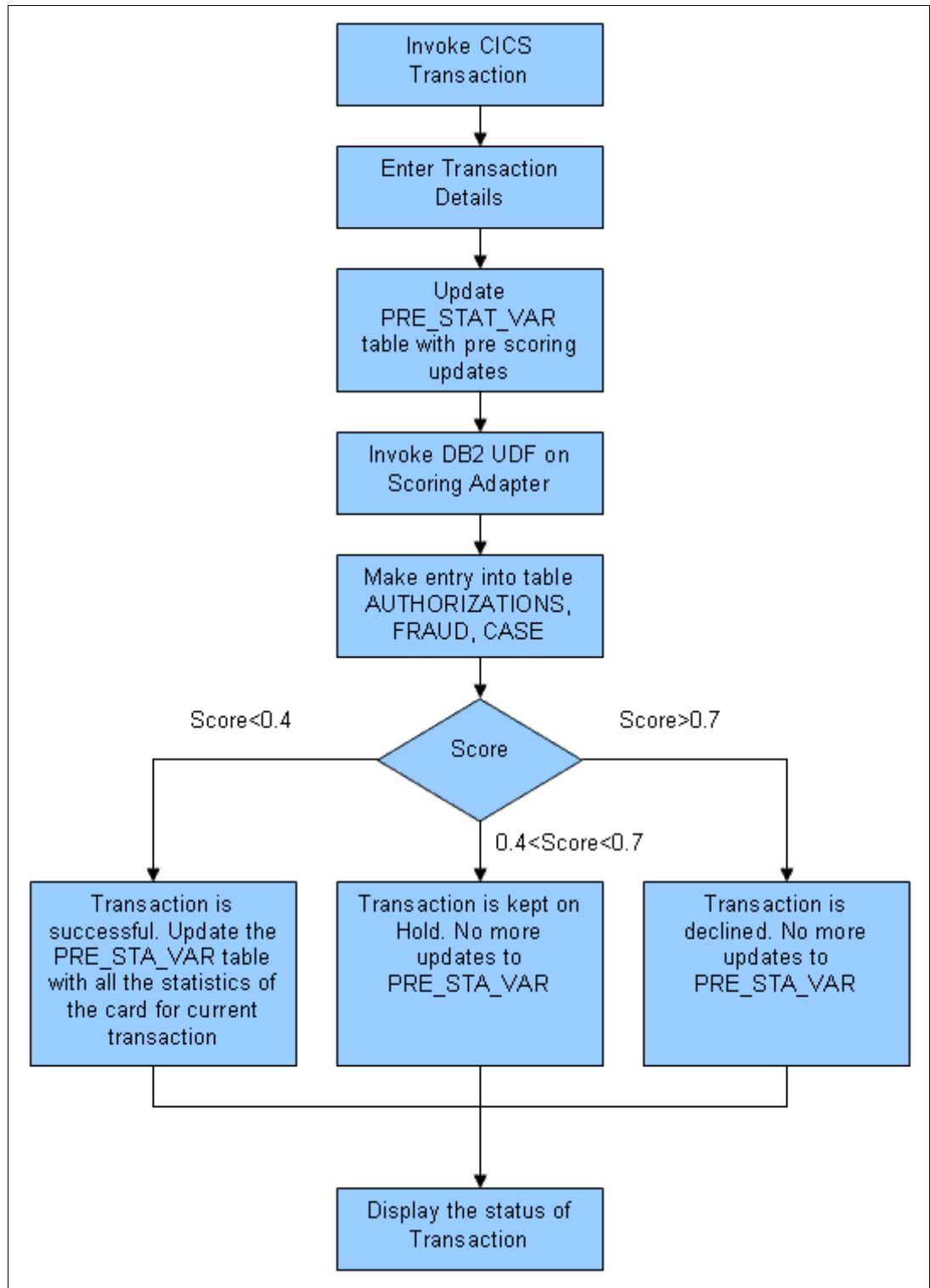


Figure 6-8 Transaction process flow

Figure 6-9 shows a transaction on hold. Figure 6-10 shows an example of a successful transaction.

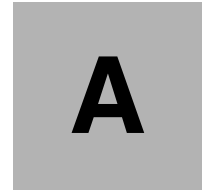
<u>Payment Screen</u>	
Card Type	: <u>CC</u> (CC)-Credit Card (CO)-Common Card
Card Number	: <u>4392268302891000</u>
Password	:
Expiration Date	: <u>12</u> / <u>15</u> MM / YY
CVV Code	: <u>1234</u>
Payment Amount	: <u>0.00</u>
Merchant Code	: <u>541233</u>
Merchant Name	: <u>IBM INDIA PVT LTD</u>
Merchant Type	: <u>5065</u>
Press ENTER to Submit	
0.5402292923101820	
CALL CUSTOMER CARE	
PF3=Exit PF4=Clear	

Figure 6-9 Transaction on hold

<u>Payment Screen</u>	
Card Type	: <u>CC</u> (CC)-Credit Card (CO)-Common Card
Card Number	: <u>4392268302891000</u>
Password	:
Expiration Date	: <u>12</u> / <u>15</u> MM / YY
CVV Code	: <u>1234</u>
Payment Amount	: <u>0.00</u>
Merchant Code	: <u>541233</u>
Merchant Name	: <u>IBM INDIA PVT LTD</u>
Merchant Type	: <u>5065</u>
Press ENTER to Submit	
0.2509803921568630	
SUCCESSFUL	
PF3=Exit PF4=Clear	

Figure 6-10 Successful transaction





## Scenario predictors

Table A-1 lists the predictors that are used in our scenario. These predictors are included in the Predictive Enterprise View (PEV) tables that are used as data sources in IBM SPSS Modeler.

*Table A-1 Scenario predictors*

Name	Type	Measurement	Description
card_id	varchar(20)	Continuous	The credit card ID.
t_amount	decimal(10,2)	Continuous	The amount of the current transaction.
t_no_over500	integer	Continuous	The number of transactions with an amount greater than \$500.
c_balance_limit	double	Continuous	The ratio of the current account balance and the credit limit.
c_amount_last_month	decimal(10,2)	Continuous	The total amount of the previous billing cycle.
time_no_3hour	integer	Continuous	The number of transactions within 3 hours.
time_amount_3hour	decimal(10,2)	Continuous	The total transaction account within 3 hours.
m_history	Integer	Flag	Whether the card holder has previously bought something with the merchant.
m_type	varchar(10)	Nominal	The merchant type.
t_time	Integer	Continuous	The transaction time.
e_time_lag	Integer	Continuous	The interval since the last transaction.

<b>Name</b>	<b>Type</b>	<b>Measurement</b>	<b>Description</b>
time_reject	Integer	Flag	Use the previous month's transaction data as a sample space to test whether the amount of the current transaction is an exception point.
e_reject	Integer	Flag	Use the past 10 transaction data as a sample space to test whether the amount of the current transaction is an exception point.
limit	decimal(10,2)	Continuous	The credit limit of the card.
gender	char(5)	Flag	Gender
edu_level	char(1)	Ordinal	Education level
mar_stat	char(1)	Nominal	Marriage status
imp_lev	char(1)	Ordinal	Important level
occupation	varchar(50)	Nominal	Occupation
eco_cat	char(1)	Ordinal	Economic conditions
annual_salary	decimal(10,2)	Continuous	Annual salary
own_hou_flag	string	Flag	Is house owner flag
vendors_in_30_minutes	integer	Continuous	The number of vendors that the card bought from within 30 minutes.
merchant_country	varchar(50)	Nominal	The country code that the merchant belongs to.
is_fraud	string	Flag	Fraud flag.



# Transaction processing tables

This appendix shows the transaction processing tables that are used in our use case in Chapter 6, “Use case model” on page 35. For more information about these tables, see 6.3, “Database design” on page 37.

Table B-1 shows the customer table.

*Table B-1 Customer table*

Column name	Data type	Length
CUST_ID	VARCHAR	10
CUST_TYPE	CHAR	1
NAME	VARCHAR	20
ADDRESS	VARCHAR	50
CITY	VARCHAR	30
POST_CODE	VARCHAR	10
START_DATE	DATE	4
NATIONALITY	VARCHAR	20
ETHNIC	VARCHAR	10
GENDER	SMALLINT	2
DOB	DATE	4
EDU_LEV	CHAR	1
MAR_STAT	CHAR	1
IMP_LEV	CHAR	1
LANG	VARCHAR	10
PHONE	VARCHAR	10

E_MAIL	VARCHAR	50
COM_NAME	CHAR	50
ECO_CAT	CHAR	1
HOUSE_PRICE	FLOAT	8
OCCUPATION	VARCHAR	50
ANNUAL_SALARY	FLOAT	8
OWN_HOU_FLAG	SMALLINT	2

Table B-2 shows the account table.

*Table B-2 Account table*

Column name	Data type	Length
ACCT_ID	VARCHAR	20
OPEN_DATE	DATE	4
NUM_CARD	INTEGER	4
ACCT_STATUS	CHAR	1
STATUS_DATE	DATE	4
ACCT_TYPE	CHAR	1
CL_CHANGE_DATE	DATE	4
CYC_DELQ	CHAR	1
EVER_DELQ	CHAR	1
CUM_CYC_DELQ	INTEGER	4
CUST_ID	VARCHAR	10
LIMIT	FLOAT	8

Table B-3 shows the authorizations table.

*Table B-3 Authorizations table*

Column name	Data type	Length
AUTH_ID	BIGINT	8
AU_MERCH_AMT	FLOAT	8
AU_MERCH_CURR_CODE	CHAR	5
AU_TRAN_TYPE	CHAR	5
AU_CARD_PRESENT	CHAR	1
AU_CAT	CHAR	1
AU_TERM_PIN_ENTRY_CAP	CHAR	1
AU_POS	CHAR	2

AU_CUST_PRESENT	CHAR	1
AU_CVV_CVC	CHAR	1
AU_CVV2_PSNT	CHAR	1
AU_CVV2_RESP	CHAR	1
AU_WHICH_CARD	CHAR	1
AU_TRACK1_PRESENT	CHAR	1
AU_TRACK2_PRESENT	CHAR	1
AU_SUB_TRAN_TYPE	CHAR	1
AU_CHIP_DATA_STATUS	CHAR	1
AU_MER_ID	VARCHAR	10
AU_MER_NAME	VARCHAR	20
AU_MER_TYPE	VARCHAR	10
AU_MER_CITY	VARCHAR	20
AU_MER_STATE	VARCHAR	20
AU_MER_POST_CODE	VARCHAR	10
AU_MER_CNTRY_CODE	CHAR	3
AU_MER_MCC	VARCHAR	20
C_CARD_ID	VARCHAR	20
CREATE_TIME	TIMESTAMP	10
AU_MER_TYPE_DESC	VARCHAR	50
T_TIME	INTEGER	4
IS_FRAUD	SMALLINT	2

Table B-4 shows the fraud table.

*Table B-4 Fraud table*

<b>Column name</b>	<b>Data type</b>	<b>Length</b>
FRAUD_ID	BIGINT	8
SCORE	FLOAT	8
SEVERITY	CHAR	1
AUTH_ID	BIGINT	8
CARD_ID	VARCHAR	20
ACCOUNT_ID	VARCHAR	20
CUSTOMER_ID	VARCHAR	10
PAYMENT_AMOUNT	FLOAT	8
MERCHANT_ID	VARCHAR	10

MERCHANT_TYPE	VARCHAR	10
IS_FRAUD	SMALLINT	2
CREATE_TIME	TIMESTAMP	10
STATUS	CHAR	1
CLOSE_TIME	TIMESTAMP	10
RULE_NAME	VARCHAR	800
RULE_DESCRIPTION	VARCHAR	800

Table B-5 shows the case table.

*Table B-5 Case table*

Column name	Data type	Length
CASE_ID	BIGINT	8
FRAUD_ID	BIGINT	8
AUTH_ID	BIGINT	8
CARD_ID	VARCHAR	20
ACCOUNT_ID	VARCHAR	20
CUSTOMER_ID	VARCHAR	10
URGENCY	CHAR	1
CASE_ACTION	VARCHAR	10
ACCOUNT_ACTION	VARCHAR	10
IS_FRAUD	SMALLINT	2
COMMENTS	VARCHAR	200
STATUS	VARCHAR	10
STATUS_TIME	TIMESTAMP	10
CREATE_TIME	TIMESTAMP	10

Table B-6 shows the card table.

*Table B-6 Card table*

Column name	Data type	Length
C_CARD_ID	VARCHAR	20
C_ISS_DATE	DATE	4
C_EXP_DATE	DATE	4
C_LST_EXP_DATE	DATE	4
C_ISS_TYPE	CHAR	1
C_MEDIA	CHAR	1

C_CARD_USE	CHAR	1
C_CARD_TYPE	CHAR	1
C_CARD_STATUS	CHAR	1
C_STATUS_DATE	DATE	4
ACCT_ID	VARCHAR	20

Table B-7 shows the PRE\_STAT\_VAR table.

*Table B-7 PRE\_STAT\_VAR table*

<b>Column name</b>	<b>Data type</b>	<b>Length</b>
CARD_ID	VARCHAR	20
T_AMOUNT	FLOAT	8
T_NO_OVER500	INTEGER	4
C_BALANCE_LIMIT	FLOAT	8
C_AMOUNT_LAST_MONTH	FLOAT	8
TIME_NO_3HOUR	INTEGER	4
TIME_AMOUNT_3HOUR	FLOAT	8
M_HISTORY	INTEGER	4
M_TYPE	VARCHAR	10
T_TIME	INTEGER	4
E_TIME_LAG	INTEGER	4
TIME_REJECT	INTEGER	4
E_REJECT	INTEGER	4
VENDORS_IN_30_MINUTES	INTEGER	4
MERCHANT_COUNTRY	VARCHAR	50
IS_FRAUD	SMALLINT	2
C_BAL_LIMIT	FLOAT	8





# Related publications

The publications that are listed in this section are considered suitable for a more detailed discussion of the topics that are covered in this book.

## IBM Redbooks

The following IBM Redbooks publications provide more information about the topic in this document. Some publications referenced in this list might be available in softcopy only.

- ▶ *DB2 10 for z/OS Technical Overview*, SG24-7892
- ▶ *Using zEnterprise for Smart Analytics: Volume 2 Implementation*, SG24-8008

You can search for, view, download, or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

[ibm.com/redbooks](http://ibm.com/redbooks)

## Other publications

These publications are also relevant as further information sources:

- ▶ *DB2 10 for z/OS Installation and Migration Guide*, GC19-2974
- ▶ *DB2 for z/OS Application Programming Topics*, SG24-6300
- ▶ *IBM SPSS Modeler Server Scoring Adapter for DB2 on z/OS License Information*, GC19-3721
- ▶ *IBM SPSS Modeler Server Scoring Adapter for DB2 on z/OS Program Directory*, GI10-8919

## Online resources

These websites are also relevant as further information sources:

- ▶ Business Analytics on System z - IBM  
<http://www.ibm.com/software/os/systemz/badw/>
- ▶ IBM - DB2 Accessories Suite for z/OS - Software  
<http://www.ibm.com/software/data/db2imstools/db2tools/accessories-suite/>
- ▶ IBM SPSS software  
<http://www.ibm.com/software/analytics/spss/>

## Help from IBM

IBM Support and downloads

[ibm.com/support](https://ibm.com/support)

IBM Global Services

[ibm.com/services](https://ibm.com/services)









# Real-time Fraud Detection Analytics on IBM System z



**The benefits of doing fraud detection on the IBM System z platform**

**Demonstrations of SPSS and CICS in real-time fraud detection**

**A description of the IBM DB2 UDF Scoring Adapter**

Payment fraud can be defined as an intentional deception or misrepresentation that is designed to result in an unauthorized benefit. Fraud schemes are becoming more complex and difficult to identify. It is estimated that industries lose nearly \$1 trillion USD annually because of fraud. The ideal solution is where you avoid making fraudulent payments without slowing down legitimate payments. This solution requires that you adopt a comprehensive fraud business architecture that applies predictive analytics.

This IBM Redbooks publication begins with the business process flows of several industries, such as banking, property/casualty insurance, and tax revenue, where payment fraud is a significant problem. This book then shows how to incorporate technological advancements that help you move from a post-payment to pre-payment fraud detection architecture. Subsequent chapters describe a solution that is specific to the banking industry that can be easily extrapolated to other industries. This book describes the benefits of doing fraud detection on IBM System z.

This book is intended for financial decisionmakers, consultants, and architects, in addition to IT administrators.

**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](http://ibm.com/redbooks)**

SG24-8066-00

ISBN 0738437638