IBM DB2 Intelligent Miner for Data

Using the Intelligent Miner for Data

Version 8 Release 1

SH12-6750-00

IBM DB2 Intelligent Miner for Data

Using the Intelligent Miner for Data

Version 8 Release 1

SH12-6750-00

Note

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 357.

First Edition, November 2002

This edition applies to

- version 8, release 1 of IBM DB2 Intelligent Miner for Data, 5724-B93
- version 6, release 1, modification level 1 of IBM DB2 Intelligent Miner for Data for iSeries, 5733-IM3
- version 6, release 1, modification level 1 of IBM DB2 Intelligent Miner for Data for OS/390, 5655-IM3

and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SH12-6394-00.

Contents

Figures

Tables

About this book

The IBM® DB2® Intelligent Miner™ for Data Version 8 Release 1, referred to as the Intelligent Miner in this book, is a suite of statistical, preprocessing, and mining functions that you can use to analyze large databases. It also provides visualization tools for viewing and interpreting mining results.

The server software runs on AIX®, OS/400®, OS/390®, Solaris Operating Environment, and on the following Windows® operating systems:

- Windows NT®
- Windows 2000
- Windows XP

In this book, the term Windows refers to all of the supported Windows operating systems.

You can use AIX and Windows clients. Unless otherwise stated, all functions that are described apply to all supported servers and clients.

This book helps you install the Intelligent Miner on your system. It also describes how to set up your system environment to run the Intelligent Miner. The book then goes on to describe the various functions of the Intelligent Miner. It gives examples of business problems that can be solved, shows you how to preprocess and select your data, and provides sample parameter settings. Moreover, it shows how to use the visualization tools and how to interpret the results. A tutorial-like scenario is provided at the end of the book.

Who should read this book

This book is intended for business, financial, and database analysts, and for programmers and application builders.

You should be familiar with the client and server operating systems that you are using. You should also know about database concepts and terminology. In addition, you should be familiar with statistics.

How to read syntax diagrams

This chapter explains how to read syntax diagrams as these are used under "Running the IBM DB2 Intelligent Miner for Data V6.1.1 server for iSeries" on page 39 and "Running the partitioning program" on page 122 to illustrate the programming syntax.

To use a diagram, follow a path from left to right, top to bottom, adding elements as you go. In these diagrams, all spaces and other characters are significant.

Each diagram begins with a double right arrowhead and ends with a right and left arrowhead pair. Lines beginning with single right arrowheads are continuation lines.

```
►►──keyword──variable_value──────────────────────────────────────────────►◄
```

Keywords are all in lowercase, but can be entered in uppercase or in lowercase. Variable values that you provide are shown in *italics* and are usually in lowercase. Where values are shown in uppercase, they should be entered as they appear.

In a choice of items, the default item is always shown above the main line:

```
              ┌─default_value─┐
►►──keyword───┼─other_value───┼──────────────────────────────────────────►◄
              └─other_value───┘
```

Optional syntax elements are shown below the main line:

```
►►──keyword───┬───────┬──────────────────────────────────────────────────►◄
              └─value─┘
```

In some cases, when an item has additional items associated with it, an additional syntax diagram is shown that represents the full syntax of that item. For example, in the following syntax diagram, additional information that can or must be specified for ITEM1 appears in the "ITEM1 Variables" syntax diagram.

```
►►──keyword──keyword_name──┤ ITEM1 ├──ITEM2──────────────────────────────►◄
```

ITEM1 Variables:

```
├──┬─variable1─┬──────────────────────────────────────────────┤
   ├─variable2─┤
   └─variable3─┘
```

Sample Syntax Diagram

The following is a sample syntax diagram. It shows the expressions that you can form with the hello command.

Hello Command

```
►►──hello──┬────────┬──┬────────────┬─────────────────────────►◄
           └┤ Name ├┘  └┤ Greeting ├┘
```

Name

```
├── name_of_person──────────────────────────────────────────────┤
```

Greeting

```
├──, how are you?───────────────────────────────────────────────┤
```

Valid versions of the hello command are:

```
hello
hello name
hello, how are you?
hello name, how are you?
```

Note that the space before the *name_of _person* value is significant and that, if you leave out a value for *name_of _person*, you still code the comma before **how are you?**.

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other Intelligent Miner documentation, you can contact us in the following ways:

- Send your comments by e-mail to swsdid@de.ibm.com. Be sure to include the name of the book, the part number of the book, the version of Intelligent Miner, and, if applicable, the specific location of the text you are commenting on (for example, a page number or table number).

- Fill out one of the forms at the back of this book and return it by mail, by fax, or by giving it to an IBM representative. The mailing address is on the back of the Readers' Comments form. The fax number is +49-7031-16-4892.
- You can also contact us when you receive an error message by clicking the **Web help** push button in the error message window. The Web page that opens in your default browser either provides help information to solve your problem or prompts you to submit a problem report. If you fill in a problem report, the Intelligent Miner support team will investigate your problem and respond to you by e-mail.

How to use this book

This book is intended to help you get started using Intelligent Miner. Chapter 1, "Introducing the Intelligent Miner" on page 1 introduces you to data mining using the Intelligent Miner main window and wizards. More detailed information on how to use the Intelligent Miner functions is provided by the wizards and online help.

This book provides a conceptual overview of the Intelligent Miner functions and tells you how to install and run the Intelligent Miner on your system. In addition, it describes the visualization tools. See Chapter 20, "Viewing results" on page 257 for information on interpreting the results of the individual functions.

For an in-depth discussion of data mining, underlying concepts, areas of application, and business-related case studies, see the *Intelligent Miner for Data Applications Guide*.

You find other valuable information and last-minute corrections in the README.TXT file on your client or server CD-ROM. Read this file before you install the Intelligent Miner.

The glossary contains terms and definitions. Related publications are listed in the bibliography.

This book is also available in HTML and in Portable Document Format (PDF) on your client CD-ROM.

What is new in version 8 of the Intelligent Miner

New features in this version include:

- New visualizers for the Associations mining function, Tree Classification mining function, and Demographic Clustering mining function.
- New tool to display Gains charts
- Predictive Model Markup Language (PMML) model support

- Radial Basis Function (RBF) model export to C code
- DB2 Universal Database™ (UDB) V8.1 support

Chapter 1. Introducing the Intelligent Miner

This chapter introduces data mining technology and the data mining process of the Intelligent Miner. It also explains in general the statistical, preprocessing, and mining functions that are provided by the Intelligent Miner.

Data mining with the Intelligent Miner

Information technology has developed rapidly over the last three decades. Many organizations store increasingly large volumes of data on their computer systems. Useful information might be hidden in the data in the form of implicit patterns and connections that are not easy to discern using conventional data queries and statistical calculations.

Data mining is the process of discovering valid, previously unknown, and ultimately comprehensible information from large stores of data. You can use extracted information to form a prediction or classification model, or to identify similarities between database records. The resulting information can help you make more informed decisions.

The Intelligent Miner supports a variety of data mining tasks. For example, a retail store might use the Intelligent Miner to identify groups of customers that are most likely to respond to new products and services or to identify new opportunities for cross-selling. An insurance company might use the Intelligent Miner with claims data to isolate likely fraud indicators.

For a quick overview, visit the following Web site to download a ScreenCam presentation of the IBM DB2 Intelligent Miner for Data:

http://www.software.ibm.com/data/iminer/fordata/downloads.html

The data mining process

Data mining is an iterative process that typically involves selecting input data, transforming it, running a mining function, and interpreting the results. The Intelligent Miner assists you with all the steps in this process. You can apply the functions of the Intelligent Miner independently, iteratively, or in combination. Mining functions use elaborate mathematical techniques to discover hidden patterns in your data. After interpreting the results of your data-mining process, you can modify your selection of data, data processing and statistical functions, or mining parameters to improve and extend your results. Figure 1 shows the basic data mining process.

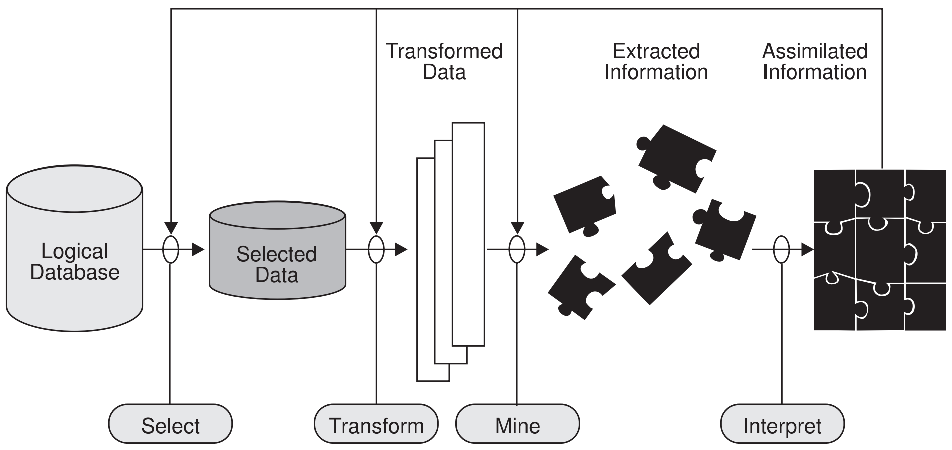


Figure 1. The data mining process

Selecting the input data

The first step in data mining is to specify the input data that you want to mine and analyze. A data source might not contain all the data that you want to use for a specific data mining objective, or it might include irrelevant data. The data that you want to mine might be in one or more database tables, views, or flat files.

The Intelligent Miner helps you select specific data from a variety of data types and sources to create input data to which you can apply preprocessing and mining functions. For example, you can omit data fields or select a subset of records in one table and then combine them with data selected from another table.

Exploring the data

At any point in the process, you can use statistical functions to explore and analyze the data. You might want to apply statistical analysis as you consider input data variables for a mining function. You can also use statistics functions to transform data to create input fields for mining. In addition, these functions are useful for evaluating the output data generated by the mining functions.

See Chapter 18, “The statistical functions” on page 241 and the online help for more information.

Transforming the data

After you specify the input data, you can transform it using the Intelligent Miner preprocessing functions. Processing functions such as discretization, filtering, and joining help you organize your data so that you can mine it effectively.

For example, if your data contains the fields Salary and Commission, you might aggregate the values of these fields and create a data field named Total_Salary. You might also use an Intelligent Miner function to remove Null values from your input data so that they do not affect the results of the data mining process.

See Chapter 19, "The preprocessing functions" on page 249 and the online help for more information.

Mining the data

Transformed data is subsequently mined using one or more mining functions. The Intelligent Miner has the following types of mining functions:

- Associations
- Neural Classification
- Tree Classification
- Demographic Clustering
- Neural Clustering
- Sequential Patterns
- Similar Sequences
- Neural Prediction
- Radial Basis Function (RBF)-Prediction

See the chapters on these functions in this book and the online help for more information.

Interpreting the results

You can analyze the results of the data mining process with respect to your decision-support objectives. Visualization tools allow you to view the results and identify important information laid bare by the mining process. You can export the results to a remote workstation so that they can be viewed at a different location. You can also copy certain results to the clipboard to make them available for other tools, such as spreadsheets or statistical applications. Moreover, you can print the results.

See Chapter 20, "Viewing results" on page 257 for more information about the visualization features.

Data mining can be an iterative process. When you look at previous results, you might want to adjust the mining settings for a further mining run to improve the result quality.

Suppose that you want to use direct marketing to offer new services to your customers and want to target only those customers most likely to be interested.

Example: A first step might be to use the Intelligent Miner to discover categories or clusters in your customer database. You can then look for group characteristics that might make customers more or less likely to accept the new services. The process of clustering your customer database might include the following steps:

1. Select relevant customer data as input data.
2. Transform the data by reorganizing it, eliminating duplicate records, or converting it from one form to another.
3. Specify the location of resulting output data.
4. Specify the parameters for the Demographic Clustering mining function.
5. Run the Demographic Clustering mining function (clustering mode).
6. Visualize the resulting data.
7. Analyze and interpret the results.
8. Repeat steps 3 to 6 until the results satisfy your needs.

You can then apply the clustering model you have just created to new data (application mode).

A scenario using the Demographic Clustering mining function is provided in Appendix A, "Intelligent Miner tutorial" on page 309.

Overview of the Intelligent Miner components

This section provides a high-level overview of the Intelligent Miner architecture. See the *IBM DB2 Intelligent Miner for Data: Application Programming Interface and Utility Reference* for more detailed information about the architecture and APIs for the Intelligent Miner.

The Intelligent Miner communicates between mining and preprocessing functions on the server, as well as between the administrative and visualization tools on the client. The client component includes a user interface from which you can invoke functions on an Intelligent Miner server. The results are returned to the client where you can visualize and analyze them. The client components are available for AIX and Windows operating systems.

The server software is available for AIX, OS/390, iSeries™, Solaris Operating Environment, and Windows systems. In addition, on AIX, Solaris Operating Environment, and Windows systems, the server software supports parallel mining with multiple processors. You can have client and server components on the same machine.

Figure 2 shows the client and server components of the Intelligent Miner:

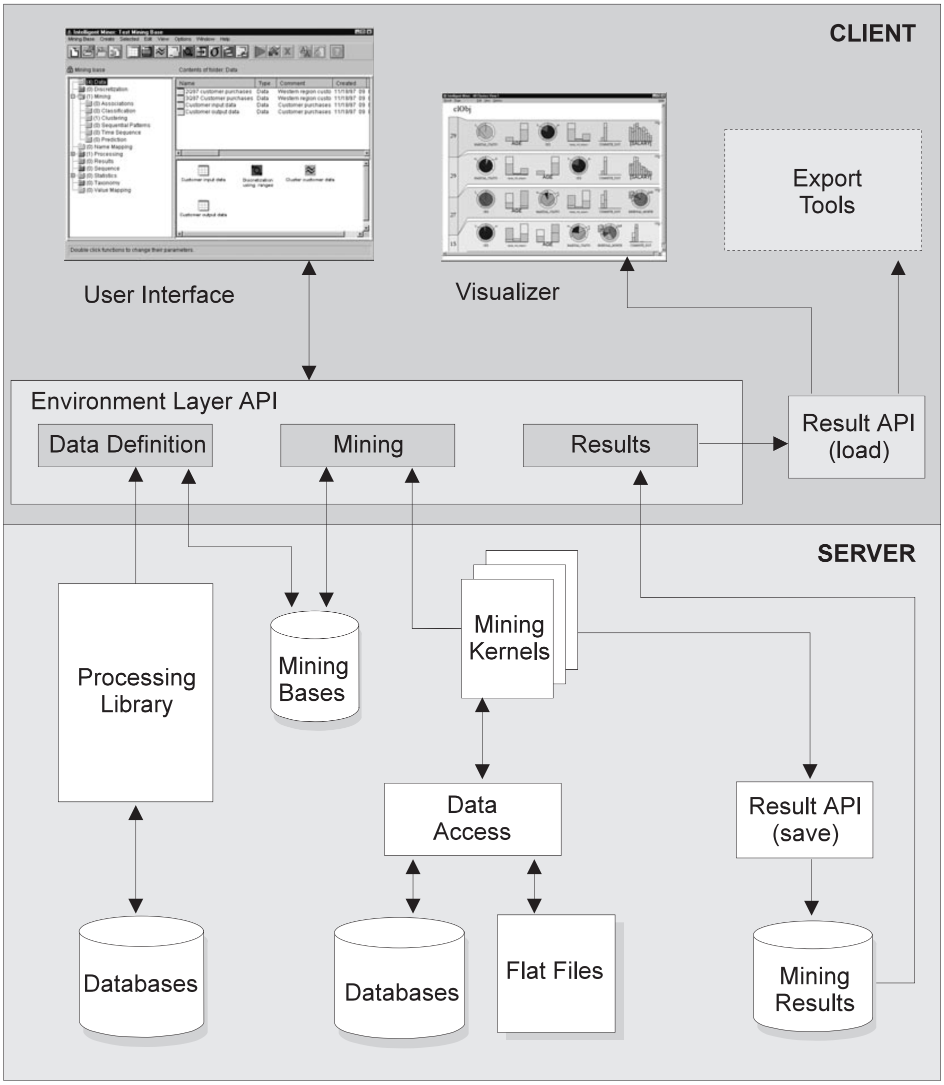


Figure 2. The Intelligent Miner architecture

User interface
A program that allows you to define data mining functions in a graphical environment. You can define preferences for the user interface, which are stored on the client. See “Understanding basic concepts” on page 77.

Environment layer API
A set of API functions that control the execution of mining runs and results. Sequences of functions and mining operations can be defined and executed using the user interface through the environment layer API. The environment layer API is available on all server operating systems.

Visualizer
A tool that displays the results produced by a mining or statistical function. The Intelligent Miner provides a rich set of visualization tools. You can also use other visualization tools.

Data access
Data access to flat files, database tables, and database views.

Database tables and flat files
The data types that you can process. The Intelligent Miner components work directly with data stored in a relational database or in flat files. The data need not be copied to a special format. You define input and output data objects that are logical descriptions of the physical data. This logical description allows you to change the physical location of the data without affecting objects that use the data. Only the logical descriptions must be changed. The change might be as simple as changing the name of a database table.

Processing library
A library that provides access to database functions.

Mining bases
A collection of data mining objects used for a mining objective or business problem. Mining bases are stored on the server, which allows access from different clients. See “Creating and using mining bases” on page 79.

Mining kernels
The algorithms brought into operation when you run a data mining or statistical function.

Mining results, result API, and export tools
The data extracted by running a mining or statistics function. These components allow you to visualize results at the client. Results can be exported for further processing or use with visualization tools.

Chapter 2. Installing Intelligent Miner servers

This chapter lists the hardware and the software requirements for the supported server operating systems and operating environments. It also describes how to install and remove the Intelligent Miner servers. Table 1 shows where to find information on installing the Intelligent Miner on a particular server operating system.

Table 1. Server installation sections in this chapter

Operating system	Section in this chapter
AIX	"Installing the Intelligent Miner server for AIX"
OS/400	"Installing the Intelligent Miner V6.1.1 server for iSeries" on page 13
OS/390	"Installing the IBM DB2 Intelligent Miner for Data V6.1.1 server for OS/390" on page 15
Solaris Operating Environment	"Installing the Intelligent Miner server for the Solaris Operating Environment" on page 17
Windows NT, Windows 2000, Windows XP	"Installing the Intelligent Miner server for Windows" on page 21

Depending on the IBM DB2 Intelligent Miner for Data product that you purchased, the product package contains software for one or more Intelligent Miner servers. Select the appropriate server CD-ROM and read the corresponding sections in this chapter.

Installing the Intelligent Miner server for AIX

This section describes the hardware requirements, software requirements, and installation process for the Intelligent Miner server for AIX. It also describes how to create links to executable files and how to remove the Intelligent Miner.

Hardware requirements

The Intelligent Miner server for AIX runs on processors of the RS/6000® Systems family and the @server pSeries™ family including the following systems:

- Uniprocessor systems
- Symmetric multiprocessor systems (SMP)
- Multiple parallel processor systems (MPP)

To run the Intelligent Miner for AIX, use the environment layer API or install one of the supported clients, either on the same machine or on a remote system. For more information on how to install the clients, see the appropriate sections in this book:

- "Installing AIX clients" on page 27
- "Installing Windows clients" on page 31

The required storage space varies with the amount of data processed per run. Table 2 shows the storage space that you need.

Table 2. Storage space for AIX server

Storage type	For demonstration	Required	Recommended
RAM	64 MB	128 MB	512 MB to 2 GB
Disk space for AIX server	75 MB	100 MB	100 MB + 200% of data
Disk space for AIX server including client (local mode)	105 MB	130 MB	130 MB + 200% of data
Additional disk space for toolkit	4 MB	4 MB	4 MB
Additional disk space for parallel server	25 MB	25 MB	25 MB

Software requirements

Table 3 lists the required and optional software needed for the AIX server.

Table 3. Software requirements for the AIX server

Software	Version	Required?
IBM AIX	4.2.1 (or higher) or 5.0 (or higher)	Yes
One of the following DB2 Universal Database (DB2 UDB) servers is required:		Yes
IBM DB2 Universal Database Enterprise Edition	7.2	
IBM DB2 Universal Database Enterprise Extended Edition	7.2	
IBM DB2 Universal Database Enterprise Server Edition	8.1	

To run the client on the same machine as the AIX server, you must install the Java™ Runtime Environment. See "Software requirements" on page 28.

Notes:

1. DB2 Universal Database Enterprise Server Edition V8.1 is provided with the Intelligent Miner for use with this product.
2. If you want to use a DB2 UDB database on a server that is different from the Intelligent Miner server, install only the respective DB2 UDB client supported by DB2 UDB on your Intelligent Miner server instead of the complete DB2 UDB server software.

To connect to other relational databases, for example, Oracle, you must install one of the following data access products of DB2:

- DB2 DataJoiner® V2.1.1
- DB2 Relational Connect V7.2 or higher

Additional software requirements for parallel processing: Table 4 lists the additional software requirements for parallel processing on AIX.

Table 4. Additional software requirements for parallel processing on AIX

Software	Version	Required
Parallel System Support Programs (PSSP) for AIX	2.2 (or higher)	Only for SP™ systems
Only together with AIX 5L: Parallel System Support Programs (PSSP) for AIX	3.4 (or higher)	Only for SP systems
IBM Parallel Environment (PE) for AIX	2.2 (or higher)	Yes
Only together with AIX 5L: IBM Parallel Environment (PE) for AIX	3.2 (or higher)	Yes

Software packages

The *IBM DB2 Intelligent Miner for Data* CD-ROM for the AIX server contains the following software packages:

- Installation package for the AIX server

  Contains file sets (installp images) for the AIX server.
- Installation package for the AIX parallel server

  Contains file sets (installp images) for the AIX parallel server. Before you install this package, install the server package.
- Installation package for the AIX client

  Contains file sets (installp images) for the AIX client.
- Installation package for HTML files supporting the AIX client

  Contains file sets (installp images) that provide online help and other text in U.S. English to assist users of AIX clients.
- Additional installation packages for HTML files in various languages

Contains file sets (installp images) that provide online help and other text in the supported languages. For a list of the supported languages, see the README file.

- Installation package for the Application Development Toolkit (ADT)

  Contains header files required to write programs using the Intelligent Miner application programming interfaces. Before you install the ADT, make sure that either the client or server software is already installed.

You might want to install DB2 Universal Database, which is part of your IBM DB2 Intelligent Miner for Data product package. Use the *DB2 Universal Database Enterprise Server Edition Version 8* CD-ROM.

Important:

1. You can only use the DB2 Universal Database in your product package with a licensed version of the Intelligent Miner.
2. You must install DB2 before you install the Intelligent Miner. However, you can upgrade your version of DB2 as usual. See Chapter 6, "Working with databases" on page 61 for more information.

Installing the Intelligent Miner

Use smit or smitty to install the Intelligent Miner. To install the Intelligent Miner server for AIX using smit, follow these steps:

1. Log in as user root.
2. Insert the Intelligent Miner for Data for AIX CD-ROM in the CD-ROM drive.
3. Enter `smit` on the AIX command line.
4. You might need to mount the CD-ROM as a file system. Mount the file system into a directory, for example, `/cdrom`.
5. Run the standard software installation procedure from smit for Custom Install and select:
   a. Software Installation and Maintenance
   b. Install and Update Software
   c. Install/Update Selectable Software (Custom Install)
   d. Install Software Products at Latest Level
   e. Install New Software Products at Latest Level

   These steps might differ slightly, depending on the AIX version installed on your machine.
6. Select the CD-ROM drive from the list of input devices.
7. Display the list of the licensed software available to install. The list contains the following software packages:
   - IMiner.server

     Intelligent Miner for Data - Serial Server

- IMiner.parallel

  Intelligent Miner for Data - Parallel Server
- IMiner.common

  Intelligent Miner for Data - Common Function
- IMiner.client

  Intelligent Miner for Data - Client

  Intelligent Miner for Data - Images
- IMiner.html.en_US

  Intelligent Miner for Data - Help Navigation - U.S. English

  Intelligent Miner for Data - Examples and Concepts - U.S. English

  Intelligent Miner for Data - Wizard Texts - U.S. English

  Intelligent Miner for Data - Glossary Terms - U.S. English

  Intelligent Miner for Data - Valid Values - U.S. English
- Additional installation packages containing translated HTML files for each language supported. The names of these packages are in the format IMiner.html.<lang>, where <lang> is the language identifier.
- IMiner.toolkit

  Intelligent Miner for Data - Toolkit (Header Files)
- IMVisualization.java

  Intelligent Miner Visualization

  Intelligent Miner Visualization - Documentation - German

  Intelligent Miner Visualization - Documentation - U.S. English

  Intelligent Miner Visualization - Documentation - Spanish

  Intelligent Miner Visualization - Documentation - French

  Intelligent Miner Visualization - Documentation - Hungarian

  Intelligent Miner Visualization - Documentation - Japanese

  Intelligent Miner Visualization - Documentation - Korean

  Intelligent Miner Visualization - Documentation - Portuguese

  Intelligent Miner Visualization - Documentation - Russian

  Intelligent Miner Visualization - Documentation - Simplified Chinese

  Intelligent Miner Visualization - Documentation - Traditional Chinese
- IMinerX.conversion

  Intelligent Miner - Conversion
- IMinerX.symblnk

  Intelligent Miner - Symbolic Links

When you select one of the group headings to select it for installation (for example, IMiner.server), all the components under the heading are installed automatically. You do not need to select the components individually. Selecting all the components has the same effect as selecting the group heading only. Although it is possible to select only one component from a package, always install the complete package.

The IMiner.html.en_US package requires the IMiner.client package. The IMiner.toolkit package requires the IMiner.server or IMiner.client package. You cannot install the HTML files or the toolkit independently.

To install the Intelligent Miner server, select **IMiner.server**. To install the Intelligent Miner parallel server, select **IMiner.server** and **IMiner.parallel**.

If you want to write Intelligent Miner applications of your own, also select **IMiner.toolkit**.

8. Install the AIX client software. This step is not required unless you use the Intelligent Miner in local mode. However, it is recommended that you install the client software to verify that the Intelligent Miner server is installed properly. See "Installing AIX clients" on page 27 for more information.
9. Follow the instructions on the smit installation menus to complete the installation.

Facilitating the start of executable files

Optional: You can use one of the following methods to enable the start of executable files irrespective of your current directory:

- Create links in the /usr/bin directory to the executable files in the /usr/lpp/IMiner/bin directory.
- Add the path that contains the executable files to your system environment.

To create links to the executable files, log in as user root and enter the following command:

```
/usr/lpp/IMiner/bin/imln
```

To add the path to your environment, append the /usr/lpp/IMiner/bin directory to the search path of your profile. Edit one of the following profiles, depending on the shell that you use on your AIX system:

For sh and ksh	`$HOME/.profile`
For csh and tcsh	`$HOME/.login`
For desktop	`$HOME/.dtprofile`

For example, if you use the Korn shell (sh or ksh), the appropriate command is:

```
export PATH=/usr/lpp/IMiner/bin:$PATH
```

Removing the Intelligent Miner

To remove the Intelligent Miner from your system:

1. Log in as user root.
2. Run the standard software and maintenance procedure from smit or smitty and select:
   a. Software Installation and Maintenance
   b. Software Maintenance and Utilities
   c. Remove Installed Software
3. List the installed software.
4. Follow the directions on the smit menus to remove the product from your system.

Installing the Intelligent Miner V6.1.1 server for iSeries

This section describes the hardware and software requirements and the installation process for the Intelligent Miner V6.1.1 server on an iSeries computer. It also describes how to remove the Intelligent Miner.

Hardware requirements

The IBM DB2 Intelligent Miner for Data V6.1.1 for iSeries requires an iSeries Advanced Series RISC System.

To run the Intelligent Miner, use the environment layer API or install one of the supported clients. For more information on how to install the clients, see the appropriate sections in this book:

- "Installing AIX clients" on page 27
- "Installing Windows clients" on page 31

Software requirements

To install the Intelligent Miner on an iSeries server, one of the operating systems listed in Table 5 is required.

Table 5. Software requirements for iSeries servers

Operating system	Version
IBM OS/400	V4R4
IBM OS/400	V4R5
IBM OS/400	V5R1
IBM OS/400	V5R2

Installing the Intelligent Miner

Install the Intelligent Miner from the CD-ROM. If you have a previous release of the Intelligent Miner installed on your iSeries system, remove it before you proceed. For more information, see "Removing the Intelligent Miner" on page 15.

If you use OS/400 V4R4, you must install the following PTFs before you install the Intelligent Miner:

- MF21545
- SF56446

Without these PTFs, the Intelligent Miner server does not work properly.

To install the Intelligent Miner on your iSeries system:

1. Sign on to the iSeries system as the security officer QSECOFR or a user profile with similar authority.
2. Ensure that the system operator message queue is in *BREAK mode. To do this, enter the following command:
   ```
   CHGMSGQ QSYSOPR *BREAK
   ```
3. Insert the Intelligent Miner distribution disk into the disk drive and ensure that the drive is online.
4. To install the Intelligent Miner with the proper language settings, use the Restore Licensed Program (RSTLICPGM) command in one of the following ways:
   - To install both program objects and language objects in the primary language of your system, enter:
     ```
     RSTLICPGM LICPGM(5733IM3) DEV(OPT01)
     ```
   - To install program objects and language objects in a language other than the primary language of your system, enter:
     ```
     RSTLICPGM LICPGM(5733IM3) DEV(OPT01) LNG(xxxx)
     ```

     where `xxxx` is the language identifier of the desired language.
   - To install an additional language in a multiple language environment, specify the desired language and restore only the language objects. Enter:
     ```
     RSTLICPGM LICPGM(5733IM3) DEV(OPT01) LNG(xxxx) RSTOBJ(*LNG)
     ```

     where `xxxx` is the language identifier of the desired language.

   The installation of the base product creates the QIDM library on your system. Integrated file system objects are created in the Qibm/ProdData/IMiner directory. Any user profile or job that uses the Intelligent Miner API must have the QIDM library in its library list.

5. If you use OS/400 V5R1 or higher, you must define and set the environement variable IDM_OSVERSION to 5 before you start the Intelligent Miner server. Otherwise the Intelligent Miner server does not work properly.

   For OS/400 Version 4, this environement variable must not be defined.

Removing the Intelligent Miner

To remove the Intelligent Miner from an iSeries system:

1. Sign on to the iSeries system as the security officer QSECOFR or a user profile with similar authority.
2. Ensure that the system operator message queue is in *BREAK mode. To do this, enter the following command:
   ```
   CHGMSGQ QSYSOPR *BREAK
   ```
3. Remove the Intelligent Miner using the Delete Licensed Program (DLTLICPGM) command:
   ```
   DLTLICPGM LICPGM(5733IMx)
   ```

   where *x* is the version of the Intelligent Miner.

Installing the IBM DB2 Intelligent Miner for Data V6.1.1 server for OS/390

This section describes the hardware and software requirements for the IBM DB2 Intelligent Miner for Data V6.1.1 for OS/390 server. It also describes how to customize the security environment.

Hardware requirements

The IBM DB2 Intelligent Miner for Data V6.1.1 for OS/390 runs on System/390® processors.

To run the IBM DB2 Intelligent Miner for Data V6.1.1 for OS/390, you need one of the supported clients. For more information, see the appropriate section in this book:

- "Installing AIX clients" on page 27
- "Installing Windows clients" on page 31

Temporary files are created and deleted when a mining run is completed. As a guideline, approximately 30% of your total disk space should be available for these temporary files.

Software requirements

Table 6 on page 16 lists the required and optional software for the OS/390 server and the clients.

Table 6. Required and optional software for OS/390 server and clients

Software	Version	Required?
IBM OS/390	2.4.0 (or higher)	Yes
Security Server or RACF®	2.1	No
DB2 for MVS/ESA™	4.1 (or higher)	No

Installing the Intelligent Miner

The IBM DB2 Intelligent Miner for Data V6.1.1 for OS/390 is provided on a tape in SMP/E format with the Program Directory. The Program Directory describes the installation prerequisites and how to install the Intelligent Miner on System/390 processors.

Because the IBM DB2 Intelligent Miner for Data V6.1.1 for OS/390 uses facilities of OS/390 OpenEdition®, a hierarchical file system (HFS) must be available before you begin the installation.

The following steps provide a general overview of the installation process, which is described in detail in the Program Directory:

1. Unload the installation JCL library from the distribution tape provided with the Intelligent Miner.
2. Submit a series of jobs that perform the following tasks:
   - Allocate SMP/E data sets
   - Allocate target and distribution libraries
   - Initialize SMP/E
   - Receive SMP/E
   - Allocate and initialize the files contained in the HFS
   - Apply SMP/E
   - Accept SMP/E
   - Bind the plans with a database management system (optional)
3. Run a sample program to verify that the Intelligent Miner is installed correctly.

Important: The IBM DB2 Intelligent Miner for Data V6.1.1 for OS/390 runs in the OpenEdition environment of OS/390. It must be installed, started, and used by TSO users who are also OpenEdition users, otherwise errors will occur during mining runs. You must observe the security instructions documented in the Program Directory to provide the necessary access rights for the server program and for the clients working with the IBM DB2 Intelligent Miner for Data V6.1.1 for OS/390.

Customizing the security environment

The Intelligent Miner server receives requests from workstation clients. The Intelligent Miner server runs programs and accesses data sets on behalf of the clients. To prevent unauthorized access to server data such as MVS™ data sets or database tables, the server program must run in a protected environment.

You can use the following security systems to protect the Intelligent Miner server:

- Resource Access Control Facility (RACF) or an equivalent security product
- OpenEdition

When a mining run is performed, the server starts a child process that runs under the identity of the client user. This means that the child process has the same rights as the client TSO user ID for program execution, data set access, and database access.

See the Program Directory for more information about customizing the security environment.

Installing the Intelligent Miner server for the Solaris Operating Environment

This section describes the hardware requirements, software requirements, and the installation process for the IBM DB2 Intelligent Miner for Data on Solaris Operating Environment servers. It also describes how to create links to executable files and how to remove the Intelligent Miner.

Hardware requirements

The Intelligent Miner server for the Solaris Operating Environment runs on systems with Sun SPARC processors.

To run the Intelligent Miner in the Solaris Operating Environment, you must install one of the supported clients. For more information, see the appropriate section of this book:

- “Installing AIX clients” on page 27
- “Installing Windows clients” on page 31

The required storage space varies with the amount of data processed per run. Table 7 on page 18 shows the storage space that you need.

Table 7. Storage space for the Solaris Operating Environment server

Storage type	For demonstration	Required	Recommended
RAM	128 MB	128 MB	512 MB to 2 GB
Disk space for the Solaris Operating Environment	80 MB	80 MB	80 MB + 200% of data
Additional disk space for toolkit	4 MB	4 MB	4 MB
Additional disk space for parallel server	25 MB	25 MB	25 MB

Software requirements

Table 8 lists the software requirements for the Solaris Operating Environment server.

Table 8. Software requirements for the Solaris Operating Environment server

Software	Version
Solaris Operating Environment	2.7, 2.8, 2.9 (or higher)
One of the following DB2 Universal Database (DB2 UDB) servers is required:	
IBM DB2 Universal Database Enterprise Edition	7.2
IBM DB2 Universal Database Enterprise Extended Edition	7.2
IBM DB2 Universal Database Enterprise Server Edition	8.1

Notes:

1. DB2 Universal Database Enterprise Server Edition V8.1 is provided with the Intelligent Miner for use with this product.
2. The Sun Solaris version of the Intelligent Miner requires *iconv* modules for the UTF-8 locale. Therefore, verify that the SUNWuiu8 package is installed on the server.
3. If you want to use a DB2 UDB database on a server that is different from the Intelligent Miner server, install only the respective DB2 UDB client supported by DB2 UDB on your Intelligent Miner server instead of the complete DB2 server software.

To connect to other relational databases, for example, Oracle, you must install one of the following data access products of DB2:

- DB2 DataJoiner V2.1.1
- DB2 Relational Connect V7.2 or higher

Additional software requirements for parallel processing: Table 9 on page 19 lists the additional software requirements for parallel processing on Sun

Solaris.

Table 9. Additional software requirements for parallel processing on Sun Solaris

Software	**Version**
Sun HPC Software Foundation Package	2.0
or	
Sun HPC Cluster Tools	3.0

Software packages

The *IBM DB2 Intelligent Miner for Data for Sun Solaris (Server)* CD-ROM contains the following software packages:

- Installation package for the Sun Solaris server

  Contains all files for the Sun Solaris server.
- Installation package for the Sun Solaris parallel server

  Contains all files for the Sun Solaris parallel server. Before you install this package, install the server package.
- Installation package for the Application Development Toolkit (ADT)

  Contains header files and libraries required to write programs using the Intelligent Miner application programming interfaces. Before installing the ADT, install the server software.
- Installation packages for demonstration data

  Contain the demonstration data in various languages. You install the demonstration data in languages other than English independently of the other software components. For information on how to install the demonstration data, see "Using the demonstration data on Solaris Operating Environment servers" on page 314.

Installing the Intelligent Miner

Before you install the Intelligent Miner, follow these steps:

1. Install DB2.
2. Create a DB2 instance.

   Users need a DB2 instance to start the Intelligent Miner. See "IBM DB2 UDB publications" on page 369 for more information about the DB2 library.
3. Include the DB2 profile in the profiles of users who start the Intelligent Miner server.

To install the Intelligent Miner, follow these steps:

1. Log in as user root.
2. Insert the server CD-ROM in the CD-ROM drive.

If the Volume Manager (vold) is installed, the CD-ROM is automatically mounted as:

```
/cdrom/IBMIMD
```

If the Volume Manager is not installed, mount the CD-ROM by entering the following commands:

a. `mkdir -p /cdrom/IBMIMD`
b. `mount -f ufs -r /dev/dsk/c0t60s2 /cdrom/IBMIMD`

3. Change to the `/cdrom/IBMIMD` directory by entering the following command:

   ```
   cd /cdrom/IBMIMD
   ```

4. Enter one of the following commands to install the Intelligent Miner server, the parallel server, the Application Development Toolkit (ADT), or all three components:

`pkgadd -a ./admin -d IMiner`	Installs the Intelligent Miner server only.
`pkgadd -a ./admin -d IMinerP`	Installs the parallel server after you installed the Intelligent Miner server.
`pkgadd -a ./admin -d IMinerTK`	Installs the ADT after you installed the Intelligent Miner server.
`./iminstall`	Installs the Intelligent Miner server and the ADT.

Facilitating the start of executable files

Optional: You can use one of the following methods to enable the start of executable files irrespective of your current directory:

- Create links in the /usr/bin directory to the executable files in the /opt/IMiner/bin directory.
- Add the /opt/IMiner/bin path to your system environment.
- Change to the /opt/IMiner/bin directory before you start the executable files.

To create links to executable files, log in as user root and enter the following command:

```
/opt/IMiner/bin
```

To add the path to your environment, append the /opt/IMiner/bin directory to the search path of your profile. Edit one of the following profiles, depending on the shell that you use on your Sun Solaris system:

For sh and ksh	`$HOME/.profile`

For csh and tcsh	`$HOME/.login`

For example, if you use the Korn shell (sh or ksh), the appropriate command is:

```
export PATH=/opt/IMiner/bin:$PATH
```

Removing the Intelligent Miner

To remove the Application Development Toolkit (ADT), the Intelligent Miner server, or both:

1. Log in as user root.
2. Change to the /opt/IMiner directory by typing the following command:
   ```
   cd /opt/IMiner
   ```
3. Enter one of the following commands:

`pkgrm -a ./admin IMinerTK`	This command removes the ADT.
`pkgrm -a ./admin IMinerP`	This command removes the Intelligent Miner parallel server.
`pkgrm -a ./admin IMiner`	This command removes the Intelligent Miner server. If you installed the parallel server, the ADT, the demonstration data, or all these components, remove them before you enter this command. Otherwise the Intelligent Miner server is not removed.
`./imremove`	This command removes the Intelligent Miner server and the ADT. If you installed the parallel server, the demonstration data, or both, remove these components before you enter this command. Otherwise the Intelligent Miner server is not removed.

Installing the Intelligent Miner server for Windows

This section describes the hardware requirements, software requirements, and the installation process for the Intelligent Miner on Windows servers. It also describes how to remove the Intelligent Miner.

Hardware requirements

The Intelligent Miner server for Windows runs on systems with 300 MHz processors or higher.

To run the IBM DB2 Intelligent Miner for Data on Windows, you must install one of the supported clients, either on the same machine or on a remote machine. For more information, see the appropriate section of this book:

- "Installing AIX clients" on page 27
- "Installing Windows clients" on page 31

The required storage space varies with the amount of data processed per run. Table 10 shows the storage space that you need.

Table 10. Storage space for Windows server

Storage type	For demonstration	Required	Recommended
RAM	128 MB	128 MB	512 MB to 2 GB
Disk space for Windows server	130 MB	130 MB	130 MB + 200% of data
Disk space for Windows server including client (local mode)	200 MB	200 MB	200 MB + 200% of data
Disk space for Intelligent Miner toolkit	150 MB	150 MB	> 150 MB
Additional disk space for parallel server	25 MB	25 MB	25 MB

The disk space needed for demonstration purposes depends on the type of your hard drive partition. The values shown in the above table are valid for NTFS partitions. If you use a FAT partition, you need about twice as much disk space.

Software requirements

Table 11 lists the software requirements for the Windows servers.

Table 11. Software requirements for Windows server

Software	Version
Microsoft® Windows NT	4.0 SP6
Microsoft Windows 2000	SP2
Microsoft Windows XP	-
One of the following DB2 Universal Database (DB2 UDB) servers is required:	
IBM DB2 Universal Database Enterprise Edition	7.2
IBM DB2 Universal Database Enterprise Extended Edition	7.2
IBM DB2 Universal Database Enterprise Server Edition	8.1

Notes:

1. DB2 Universal Database Enterprise Server Edition V8.1 is provided with the Intelligent Miner for use with this product.
2. If you want to use a DB2 UDB database on a server that is different from the Intelligent Miner server, install only the respective DB2 UDB client supported by DB2 UDB on your Intelligent Miner server instead of the complete DB2 UDB server software.

To connect to other relational databases, for example, Oracle, you must install one of the following data access products of DB2:

- DB2 DataJoiner V2.1.1
- DB2 Relational Connect V7.2 or higher

When you complete the installation, verify that the Intelligent Miner works correctly by using the sample mining bases.

Additional software requirements for parallel processing: Table 12 lists the additional software requirements for parallel processing on Windows.

Table 12. Additional software requirements for parallel processing on Windows

Software	Version
Critical Software WMPI	1.54

Software packages

The Intelligent Miner server for Windows CD-ROM contains the following software packages:

- Intelligent Miner for Data for Windows server

  Contains all the files for the Windows server.
- Intelligent Miner for Data for Windows parallel server

  Contains all the files for the parallel server. You can only install this package in addition to the Intelligent Miner for Data for Windows server.
- Intelligent Miner for Data for Windows Application Development Toolkit

  Contains header files and libraries required to write programs using the Intelligent Miner application programming interfaces.
- Intelligent Miner for Data for Windows client

  Contains all the files for the Windows client.

You can first install the server package and later add the client and the toolkit.

The packages cannot be reinstalled. If you installed a package, this package is no longer displayed as a selectable component when you run the installation program again.

Attention: Do not use different program versions for the server and the client, or the Intelligent Miner might become unusable.

Installing the Intelligent Miner

Before you install the Intelligent Miner, follow these steps:

1. Install DB2.
2. Create a DB2 instance.

   Users need a DB2 instance to start the Intelligent Miner. See "IBM DB2 UDB publications" on page 369 for more information about the DB2 library.

To install the Intelligent Miner:

1. Log on to the system with administrator authority.
2. Insert the server CD-ROM in the CD-ROM drive.
3. Use the Windows Explorer to switch to the CD-ROM drive.
4. Change to the xx directory, where xx denotes the appropriate language directory.

   For example, to install the Intelligent Miner client in English, change to the \EN directory. For a list of available languages, see the README file.
5. Run SETUP.EXE by double-clicking on the file icon.

   A startup message is displayed. The Main window opens and displays a Welcome dialog box.

You are guided through the installation process. You can choose to install any or a combination of the following packages:

- Server
- Parallel server
- Client
- Application Development Toolkit

Installing the server as a Windows service

The Intelligent Miner server is installed as a Windows service. The name of this service is **IBM Intelligent Miner**.

Important: You must install the Intelligent Miner server on a local drive. The IBM Intelligent Miner service does not run if you install the program files on a network drive.

You can also install the IBM Intelligent Miner service after the installation by typing the following command on the command line of a Command Prompt window:

```
idmd -install
```

The IBM Intelligent Miner server is installed using the Local System user ID. If you experience problems when you start the service, change the user ID of the IBM Intelligent Miner service on the Windows control panel. Make sure that the following is true for the user who starts the Intelligent Miner server:

- The user must belong to the Administrator group.
- The user must have all rights including advanced user rights.

To verify the rights of a user by using the Windows NT User Manager, follow these steps:

1. Click **Start** → **Programs** → **Administrative Tools** → **User Manager**. The User Manager window opens.
2. Click **Policies** → **User Rights**.
3. Check the **Show advanced user rights** box.
4. Select a user group in the **Grant To** list.
5. Scroll through the **Rights** drop-down list. If a right is missing, add it to the selected user group.

To verify the rights of a user or a user group on Windows 2000 or on Windows XP, follow these steps:

1. Click **Start** → **Settings** → **Control Panel** → **Administrative Tools** → **Local Security Policy**. The Local Security Settings window opens.
2. Double-click **Local Policies**.
3. Double-click **User Rights Assignment**. On the right pane, you can see the user rights (policies) and the users or groups the policies are assigned to.
4. If a policy for a user or a group is missing, add it to the users or the group by double-clicking on the policy name.

Removing the IBM Intelligent Miner service

To remove the IBM Intelligent Miner service, enter the following command in a Command Prompt window:

```
idmd -remove
```

If the IBM Intelligent Miner service is running at the time you enter this command, it is not removed until you restart the system.

Removing the Intelligent Miner

To remove the Intelligent Miner:

1. Log on to the server with system administrator authority.
2. Double-click the **Uninstall** icon in the Intelligent Miner folder.

Chapter 3. Installing Intelligent Miner clients

This chapter lists the hardware and the software requirements for the supported client operating systems. It also describes how to install and remove Intelligent Miner clients.

The *IBM DB2 Intelligent Miner for Data Clients* CD-ROM is part of all IBM DB2 Intelligent Miner for Data product packages. Read the appropriate section for your client operating system.

Table 13 shows where to find information on installing the Intelligent Miner on a particular client operating system.

Table 13. Client installation sections in this chapter

Operating system	**Section in this chapter**
AIX	"Installing AIX clients"
Windows	"Installing Windows clients" on page 31

Installing AIX clients

This section describes the hardware and software requirements and the installation process for the Intelligent Miner AIX client. It also explains how to remove the AIX client.

The *IBM DB2 Intelligent Miner for Data Clients* CD-ROM contains the following software packages for AIX:

- Installation package for the AIX client.

  Contains file sets (installp images) for the AIX client.
- Installation package for text files supporting the AIX client

  Contains file sets (installp images) that provide online help and other text in U.S. English to assist users of AIX clients.
- Additional installation packages for text files in various languages

  Contains file sets (installp images) that provide online help and other text in the supported languages. For a list of the supported languages, see the README file.

  The files for user assistance in U.S. English are always installed. You can add support files for other languages to your installation. This allows you to use the software in more than one language. The language that is actually used depends on the language settings of the individual shell.

- Installation packages for the AIX Application Development Toolkit (ADT)

  These packages contain header files required to write programs using the Intelligent Miner application programming interface. Before installing the ADT, you must install the client software.

Hardware requirements

The Intelligent Miner client for AIX runs on processors of the RS/6000 Systems family and the IBM eServer pSeries family including the following systems:

- Uniprocessor systems
- Symmetric multiprocessor systems (SMP)
- Multiple parallel processor systems (MPP)

The required storage space varies with the amount of data processed per run. Table 14 shows the storage space that you need for an AIX client.

Table 14. Required storage space for AIX client

Storage type	For demonstration	Required	Recommended
RAM	32 MB	64 MB	64 MB
Disk space	85 MB	100 MB	> 100 MB
Additional disk space for the Intelligent Miner Toolkit	4 MB	4 MB	4 MB

Software requirements

Table 15 lists the required and optional software needed for an AIX client.

Table 15. Required and optional software for AIX client

Software	Version	Required?
IBM AIX	4.3.3 (or higher)	Yes
Java Runtime Environment from Java Development Kit (JDK) for AIX	1.3 (or higher)	Yes
Netscape Navigator for AIX to display the online help system	4.7 (or higher)	Yes (required to display online help)

Installing the Intelligent Miner

Use smit or smitty to install the Intelligent Miner. To install the Intelligent Miner client using smit:

1. Log in as user root.
2. Insert the Intelligent Miner for Data for AIX CD-ROM in the CD-ROM drive.
3. Enter `smit` on the AIX command line to start smit.

smit provides a menu-driven environment for system administration tasks. For more information on smit, see your AIX operating system documentation.

4. You might need to mount the CD-ROM as a file system. Mount the file system into a directory, for example, `/cdrom`.
5. Run the standard software installation procedure from smit for Custom Install and select:
   a. **Software Installation and Maintenance**
   b. **Install and Update Software**
   c. **Install/Update Selectable Software (Custom Install)**
   d. **Install Software Products at Latest Level**
   e. **Install New Software Products at Latest Level**

   These steps might differ slightly, depending on the AIX version installed on your machine.
6. Select the CD-ROM drive from the list of input devices.
7. Display the list of the licensed software available to install. The list contains the following software packages:
   - IMiner.client

     Intelligent Miner for Data - Client

     Intelligent Miner for Data - Images
   - IMiner.common

     Intelligent Miner for Data - Common Function
   - IMiner.html.en_US

     Intelligent Miner for Data - Help Navigation - U.S. English

     Intelligent Miner for Data - Examples and Concepts - U.S. English

     Intelligent Miner for Data - Wizard Texts - U.S. English

     Intelligent Miner for Data - Glossary Terms - U.S. English

     Intelligent Miner for Data - Valid Values - U.S. English
   - Additional installation packages containing translated text files for each language supported. The names of these packages are in the format IMiner.html.<lang>, where <lang> is the language identifier.
   - IMiner.toolkit

     Intelligent Miner for Data - Toolkit (Header Files)
   - IMVisualization.java

     Intelligent Miner Visualization

     Intelligent Miner Visualization - Documentation - German

     Intelligent Miner Visualization - Documentation - U.S. English

     Intelligent Miner Visualization - Documentation - Spanish

     Intelligent Miner Visualization - Documentation - French

  Intelligent Miner Visualization - Documentation - Hungarian
  Intelligent Miner Visualization - Documentation - Japanese
  Intelligent Miner Visualization - Documentation - Korean
  Intelligent Miner Visualization - Documentation - Portuguese
  Intelligent Miner Visualization - Documentation - Russian
  Intelligent Miner Visualization - Documentation - Simplified Chinese
  Intelligent Miner Visualization - Documentation - Traditional Chinese
- IMinerX.conversion
  Intelligent Miner - Conversion
- IMinerX.symblnk
  Intelligent Miner - Symbolic Links
- IMiner.server
  Intelligent Miner for Data - Serial Server
- IMiner.parallel
  Intelligent Miner for Data - Parallel Server

When you select one of the group headings (for example, IMiner.client), all the components under the heading are installed automatically. You do not need to select the components individually. Selecting all the components has the same effect as selecting the group heading only. Although it is possible to select only one component from a package, always install the complete package.

The IMiner.html.<lang> and IMiner.toolkit packages require the IMiner.client package. You cannot install the text files or the toolkit independently. To install the Intelligent Miner client, select:
- **IMiner.client**
- One or more text file packages

If you want to write application programs, also select **IMiner.toolkit**.

8. Follow the instructions on the smit installation menus to complete the installation.

Removing the Intelligent Miner

To remove the Intelligent Miner:
1. Log in as user root.
2. Run the standard software and maintenance procedure from smit or smitty and select:
   a. **Software Installation and Maintenance**
   b. **Software Maintenance and Utilities**
   c. **Remove Installed Software**

3. List the installed software.
4. Follow the directions on the smit menus to remove the product from your system.

Installing Windows clients

This section describes the hardware and software requirements and the installation process for the Intelligent Miner on Windows NT, Windows 2000, and Windows XP. It also explains how to remove a Windows NT, Windows 2000, or Windows XP client.

The *IBM DB2 Intelligent Miner for Data Clients* CD-ROM contains the following software packages for Microsoft Windows:

- Installation package for Windows NT, Windows 2000, and Windows XP clients

  The directories for these Windows clients contain a subdirectory for each supported language. To install the client software in a certain language, you must change to the appropriate subdirectory and start the installation procedure from there. You can install the Windows client software in one language only.
- Installation package for the Windows Application Development Toolkit (ADT)

  Contains header files and libraries required to write programs using the Intelligent Miner application programming interface.

Hardware requirements

The Intelligent Miner client for Windows runs on a workstation with a 300 MHz processor or higher.

The required storage space varies with the amount of data processed per run. Table 16 shows the storage space that you need for a Windows client.

Table 16. Required storage space for Windows client

Storage type	For demonstration	Required	Recommended
RAM for Microsoft Windows client	64 MB	64 MB	64 MB
Disk space	50 MB	50 MB	> 60 MB
Disk space for toolkit	60 MB	60 MB	> 60 MB
Disk space for client including toolkit	80 MB	80 MB	> 80 MB

The disk space needed for demonstration purposes depends on the type of your hard drive partition. The values shown in the above table are valid for NTFS partitions. If you use a FAT partition, you need about twice as much disk space.

Software requirements

Table 17 lists the required and optional software needed for a Windows client.

Table 17. Required and optional software for Windows client

Software	Version	Required?
One of the following operating systems:		Yes
Microsoft Windows NT	4.0 SP6	
Microsoft Windows 2000	SP	
Microsoft Windows XP	n/a	
Web browser capable of frames, such as:		Yes (required to display online help)
Microsoft Internet Explorer	3.0 (or higher)	
Netscape Navigator for Windows	4.7 (or higher)	

The Java Runtime Environment, which is also required, is part of your Intelligent Miner client package for Windows and is installed automatically.

Installing the Intelligent Miner

The Intelligent Miner Windows clients are available on the clients CD-ROM. Ensure that the software listed under "Software requirements" is installed on your system.

To install the client package from the CD-ROM:

1. Insert the Intelligent Miner Clients CD-ROM in the CD-ROM drive.
2. Use the Windows Explorer to switch to the CD-ROM drive.
3. Change to the xx directory, where xx is the appropriate language directory.

   For example, to install the Intelligent Miner client in English, change to the EN directory. For a list of available languages, see the README file.
4. Run SETUP.EXE by double-clicking on the file icon.

   A startup message is displayed. The Main window opens and displays a Welcome dialog box.
5. Click **Next**.

   The Select Components window opens.
6. In the list box, select the components that you want to install.

7. Click **Browse** if you want to select a directory other than the default directory c:\Program Files\IBM\IM. This directory must have at least 50 MB of free space.
8. Click **OK** to return to the Select Components window.
9. Click **Next**.

   The installation program checks if the minimum amount of space required on the target system is available. If there is not enough space, a warning message is displayed. If there is enough space, the file transfer process to the selected directory is started. A progress indicator displays the progress of the file transfer process.

   During the installation process, the LANG environment variable is set to the appropriate language.

   The installation program automatically creates program folders and icons on the desktop of the target system.
10. Click **OK**.

    The message `Do you want to read the README file?` is displayed.
11. Click **OK**.

    A Notepad window opens and displays the README file.
12. Close the Notepad window after reading the README file.
13. Click **OK** to exit the installation tool.

    The Restart Windows window opens, asking you if you want to restart your machine.
14. Select **Yes, I want to restart my computer now** and click **OK**.
15. When your system has restarted, double-click the IMiner Windows Client icon to start the Intelligent Miner client.

Removing the Intelligent Miner

To remove the Intelligent Miner for Windows:

1. Click the Add/Remove Programs icon on the Control Panel. A list of programs that can be removed is displayed.
2. Select **Intelligent Miner**.
3. Click **Add/Remove**. The Intelligent Miner is removed from your system.

You might need to manually remove the directory that contained the Intelligent Miner and the program group that contained the Intelligent Miner icons. For more information, see your Windows documentation.

Chapter 4. Running Intelligent Miner servers

This chapter describes how to run Intelligent Miner servers in the supported operating environments. It also describes how to set up the parallel operating environment for AIX, the Solaris Operating Environment, and Windows operating systems.

Figure 3 shows the different client/server configurations supported by the Intelligent Miner.

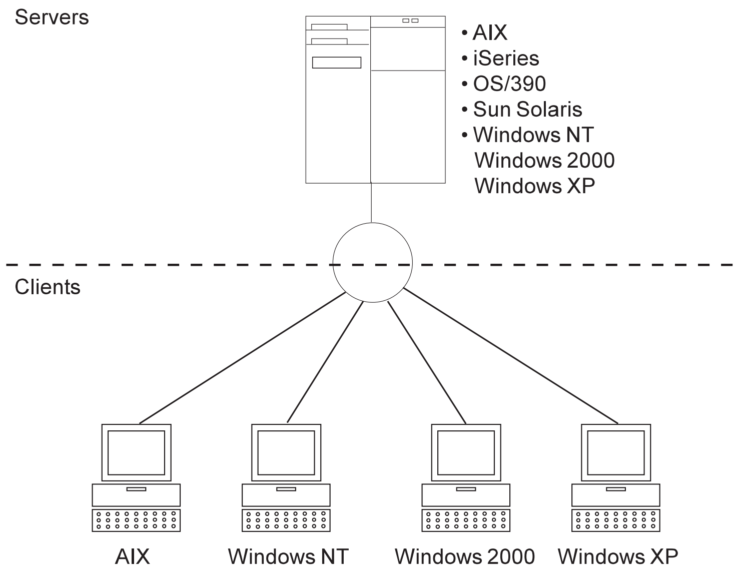


Figure 3. The client/server combinations

You can run the Intelligent Miner server on AIX, iSeries, OS/390, Solaris Operating Environment, and Windows servers.

- On AIX and Windows, you can run the Intelligent Miner in local mode, that is, you can run the client and the server software on the same workstation.

  The Windows versions also provide a standalone mode, which means that you run the Intelligent Miner on a single workstation without a server process. However, you must install the client and the server software on that workstation.

  In client/server mode, you can connect a client to the server.

- On iSeries, OS/390, and in the Solaris Operating Environment, you must connect one of the supported clients to the server.

Alternatively, you can write applications of your own using the Intelligent Miner application programming interface (API). See the *IBM DB2 Intelligent Miner for Data: Application Programming Interface and Utility Reference*.

Running the Intelligent Miner server for AIX

The information in this section applies to the IBM DB2 Intelligent Miner for Data AIX server. It describes how multiple users can share mining bases and result files, how to start and stop the Intelligent Miner, and how to unlock mining bases.

Using shared mining bases and result files

The Intelligent Miner uses the environment variables `IDM_MNB_DIR` and `IDM_RES_DIR`, which point to the directories that contain the mining bases and result files respectively.

If you do not set these variables and use the default values, directories named idmmnb and idmres are created in each user's home directory.

Before starting the Intelligent Miner server, you can set the environment variable `IDM_MNB_DIR` to point to one common directory if you want multiple users to share the same mining bases.

For example, if you want all users to share the mining bases in the idmmnb subdirectory of a directory named miningbases, enter the following command on the Korn shell:

```
export IDM_MNB_DIR=/miningbases
```

All Intelligent Miner clients connected to this server now share the same mining bases.

Restriction: Only one user at a time can have write access to a shared mining base.

Permission problems might occur if a user tries to save a mining base that another user created because the creator is also the owner of the mining base. To avoid permission problems, assign all users accessing the shared mining bases to the same group and grant write permission to that group. For that purpose, change the file mode creation mask by entering the following command:

```
umask 02
```

You can set the environment variable IDM_RES_DIR in a similar way if you want all users to write their result files to a common directory.

For example, if you want all users to share the result files in the idmres subdirectory of a directory named resultfiles, enter the following command on the Korn shell:

```
export IDM_RES_DIR=/resultfiles
```

You can specify other environment variables. For a complete list, see Appendix C, "Environment variables" on page 349.

Starting the Intelligent Miner server for AIX

To start the AIX server, enter one of the following commands on the command line:

idmstart	This command starts the AIX server.
idmstart -d	This command starts the AIX server and shows the tracing information.

Now multiple clients can access the Intelligent Miner server. See Chapter 5, "Running Intelligent Miner clients" on page 57 for more information.

Tips:

1. You can add the `idmstart` command to the /etc/inittab file of your system setup to start the Intelligent Miner automatically after an operating-system failure.
2. You can use the `crontab` command to create cron jobs for idmstart and idmstop. These jobs make the Intelligent Miner automatically available for certain time slots in your system setup.

Stopping the Intelligent Miner server for AIX

To stop AIX servers:

1. Log in using the user ID that started the server or the user ID root.
2. Enter `idmstop` on the command line.

Unlocking mining bases from the AIX server

Normally, a mining base is locked when you open it, and unlocked when you close it. This ensures that only one person at a time can make changes to the mining base and save these changes.

If a mining run that was started from a client ends abnormally, the loaded mining base might is also locked. For example, if you shut down Windows and allow the operating system to stop the Intelligent Miner client rather than stopping all Intelligent Miner applications before this, the routine that unlocks the mining base is not started. If this is the case, the following message is displayed on the client when you open the mining base:

```
The mining base was loaded in READ-ONLY mode. To save changes, you must
save the mining base under a different name.
```

You must unlock the mining base manually before you can open it again by using the following procedure:

1. Sign on to the server with the same user profile as the client.
2. Start the idmclear program, which resides in the /usr/lpp/IMiner/bin directory.

   If IDM_MNB_DIR is set, the idmclear program unlocks mining bases in the directory identified by this environment variable.

   If IDM_MNB_DIR is not set, the idmclear program unlocks mining bases in the $HOME directory.

   Attention: The idmclear program unlocks all mining bases in a user's home directory or in a common directory identified by IDM_MNB_DIR. This includes mining bases that are locked because a user opened them. Therefore, make sure that all mining bases in the common directory or user's home directory are closed before you run idmclear.

See "Unlocking mining bases from a client" on page 59 for information on how to unlock mining bases from a client workstation.

Parallel processing on AIX

This section provides information about parallel processing on AIX. Parallel processing works on servers with multiple SMP or SP processors.

If you run the Intelligent Miner on parallel nodes, the performance of the mining runs improves significantly.

You can run most of the mining functions and the Copy Records to File preprocessing function on parallel nodes. The mining functions that you cannot run in parallel mode are listed here:

- Similar Sequences
- RBF Prediction in:
  - Training mode
  - Test mode

Before you can run the Intelligent Miner on parallel nodes, you must specify the nodes that you want to use and partition the input data. See "Specifying parallel parameters" on page 118 and "Partitioning the input data" on page 119 for more information.

Setting up the parallel operating environment

To improve the performance when you run the IBM DB2 Intelligent Miner for Data in parallel mode on a system with SP processors, use the High Performance Switch feature.

Use the switch communication to communicate among parallel tasks. For that purpose, use the switch-related host names of the SP nodes in the idmhost.list file. If multiple users work with the same SP nodes simultaneously, make sure that the environment variable MP_EUILIB is set as follows:

```
MP_EUILIB=ip
```

For more information on using the High Performance Switch feature, see *IBM Parallel Environment for AIX: Operation and Use*.

Specifying the parallel nodes

To determine the nodes for a parallel mining run, the Intelligent Miner uses the information stored in a host-list file. To specify the names of the nodes:

1. Create a host-list file with the name idmhost.list.
   - If users need individually configured host-list files, proceed as follows:
     a. Make sure that `IDM_MNB_DIR` is not set for the server process.
     b. Create a file idmhost.list in each user's home directory on the server.
   - If you want to create a common host-list file for all users, store idmhost.list in the directory identified by `IDM_BIN_DIR` on the server. This is, by default, the `/usr/lpp/IMiner/bin` directory.

     Restriction: This file is not used if one of the following conditions applies:
     - `IDM_MNB_DIR` is set and a file named idmhost.list exists in the directory that this environment variable points to.
     - `IDM_MNB_DIR` is not set and a file named idmhost.list exists in a client user's home directory on the server. The client user in question will use this host-list file instead of the common host-list file.
2. Specify the nodes in the host-list file as pool numbers or host names. You can specify pool numbers only if the input data resides in a file system that is shared by all hosts.

Running the IBM DB2 Intelligent Miner for Data V6.1.1 server for iSeries

Before starting the iSeries server for the first time, check the Program Directory or the QIDM/QYDMREADME file for a list of required program temporary fixes (PTFs). Issue the **DSPPTF** command to see which PTFs are installed on your system.

TCP/IP must be configured and running on the iSeries server. To check the configuration:

1. Enter `CFGTCP`.
2. Select option 12 to display the domain name and the local host name.
3. Make sure both host table entries (option 10) include an entry for the long and the short form of your system's IP address. For example:

   ```
   Local domain name            italy.ibm.com

   Local host name              nc400
   ```

   If you get this information from option 12, you should have two host-name entries for your IP address in option 10:

   ```
   Internet Opt Address         Host Name

   9.87.129.105                 NC400

                                NC400.ITALY.IBM.COM
   ```

If the server does not start TCP/IP automatically, issue the **STRTCP** or **STRTCPSVR** commands. To change TCP/IP configuration settings, you must first stop and then restart the TCP/IP jobs that you were running.

To start the RPC server, your user profile must have *IOSYSCFG authority. Also, it must be included in the system distribution directory. If necessary, use the **WRKDIRE** or **ADDDIRE** commands to add your user profile to this directory. Start the RPC server with the following command:

```
STRNFSSVR *RPC
```

Starting the IBM DB2 Intelligent Miner for Data V6.1.1 server for iSeries

To start the IBM DB2 Intelligent Miner for Data V6.1.1 for iSeries server:

1. Sign on to your system with a user profile that has *JOBCTL authority.
2. Enter `STRIDM` and one of the optional parameters on the command line. The authority to this command follows the AS/400 security conventions.

   The parameter that you choose determines whether the Intelligent Miner server creates a trace file. The trace file is stored in the same location as the spooled files of the user who started the Intelligent Miner server.Syntax of the **STRIDM** command:

   ```
   ►►──STRIDM──┬─*NO──┬──────────────────────────────────────────────►◄
               └─*YES─┘
   ```

   Parameters

   ***NO**
   Disables the trace facility of the Intelligent Miner server.

***YES**
Enables the trace facility of the Intelligent Miner server.

If you are not familiar with syntax diagrams, see "How to read syntax diagrams" on page xiv for more information.

When you issue the **STRIDM** command to run the Intelligent Miner, a job called QYDMIDMD is started in the QSYSWRK subsystem. The QYDMIDMD job runs a program named IDMD, which uses the QGPL/QDFTJOBD job description. The user profile from which the QYDMIDMD job is started must be able to read and write files to the home directory specified in a client user profile. Therefore, to start the Intelligent Miner server, use a profile with *ALLOBJ authority or a profile that was given specify authority using the **CHGAUT** command.

After you issued **STRIDM** command, multiple clients can access the Intelligent Miner server. See Chapter 5, "Running Intelligent Miner clients" on page 57 for information on how to start one of the supported clients.

Recommendation: For automatic restarts, you must add the **STRIDM** command to the startup program. To find the name of the startup program for your system, use the **DSPSYSVAL QSTRUPPGM** command. Remember that the RPC network file server must be started before the Intelligent Miner server is started. Therefore, check whether your startup program contains the **STRNFSSVR *RPC** command. If not, add this command to your startup program. Make sure that it comes before the **STRIDM** command.

You can customize the startup program by setting environment variables listed in Appendix C, "Environment variables" on page 349. This appendix also explains how to view and set environment variables.

Stopping the IBM DB2 Intelligent Miner for Data V6.1.1 server for iSeries

To stop the IBM DB2 Intelligent Miner for Data V6.1.1 for iSeries server:

1. Sign on to your system with a user profile that has *JOBCTL authority.
2. Enter **ENDIDM** and any of the optional parameters on the command line. The syntax of the command is:

```
►►──ENDIDM──┬─*CNTRLD──┬───────┬─┬──────────────────────────────────────────►◄
            │          └─DELAY─┘ │
            └─*IMMED─────────────┘
```

Parameters

Use the optional parameters to specify whether the Intelligent Miner server processes and mining runs are stopped in a controlled manner or immediately.

***CNTRLD**
Stops all Intelligent Miner server operations after a specified delay period has elapsed.

DELAY
If you select controlled ending, enter the number of seconds to delay before stopping the Intelligent Miner server operations.

***IMMED**
Stops all Intelligent Miner server operations immediately.

Unlocking mining bases from the iSeries server

Normally, a mining base is locked when you open it, and unlocked when you close it. This ensures that only one person at a time can make changes to the mining base and save these changes.

If a mining run that was started from a client ends abnormally, the loaded mining base might is also locked. For example, if you shut down Windows and allow the operating system to stop the Intelligent Miner client rather than stopping all Intelligent Miner applications before this, the routine that unlocks the mining base is not started. If this is the case, the following message is displayed on the client when you open the mining base:

```
The mining base was loaded in READ-ONLY mode. To save changes, you must
save the mining base under a different name.
```

You must unlock the mining base manually before you can open it again by using the following procedure:

- Sign on to the server using the same user profile as the client. Issue the **CALL QIDM/QYDMCLEAR** command to unlock mining bases in your home directory.

  For example, if a mining base with the name dmtksample in the subdirectory /MYPROF of your home directory is locked, enter this on the command line:

  ```
  call qidm/qydmclear '/home/MYPROF/dmtksample'
  ```

  Attention: This command unlocks all mining bases in the specified home directory, including mining bases that are regularly locked because a user opened them. Therefore, make sure that all mining bases in the common directory or user's home directory are closed before you run idmclear.
- Use the **WRKLNK** command to display your home directory, and delete any files with an extension of lck in the /idmmnb subdirectory.

See "Unlocking mining bases from a client" on page 59 for information on how to unlock mining bases from a client workstation.

Customizing OS/400 user profiles

On OS/400 operating systems, the root directory is the home directory by default.

To set the home directory, enter `CHGPRF HOMEDIR(/myhomedirectory)`.

If you use the default values, the idmmnb and idmres directories are created in the home directory of each user.

On an iSeries server, you can specify the same environment variables that are used on an AIX server. See "Running the Intelligent Miner server for AIX" on page 36 for more information.

To display, create, or change environment variables for the iSeries server, use the **WRKENVVAR** command.

Customizing the security environment

Each client user connected to an iSeries server needs a user profile with *JOBCTL authority to start and stop mining runs.

In addition to this requirement, you must provide users with various object authorities so that they can use common functions on the Intelligent Miner iSeries server. Table 18 shows the authorities that are required for these objects.

Table 18. Object authorities required to use common functions on the Intelligent Miner iSeries server

Profile	Function	Object	Authority
Client	Set profile	QWTSETP	*USE
Server	Set profile	QWTSETP	*USE and OBJMGT
Server	Get profile handle	QSYGETPH	*USE
Server	Check user authority	QSYCUSRA	*USE
Server	N/A	Client user profile	*USE
Server	Release profile handle	QSYRLSPH	*USE

Tips:

1. If you provide the user profile on the server with *ALLOBJ authority, the client user profile only needs *JOBCTL authority.
2. You can provide each client user with *USE authority for the QWTSETP object if you change the authority of *PUBLIC accordingly. This is not a security risk because you cannot use the *get profile handle* function without proper authority for the QSYGETPH object. To grant *USE authority to *PUBLIC, enter the following command:

   ```
   EDTOBJAUT OBJ(QWTSETP) OBJTYPE(*PGM)
   ```

If *PUBLIC is already on the list of users, change the object authority by typing *USE in the appropriate object authority field.

To add *PUBLIC to the list of users, press the F6 key.

Running the IBM DB2 Intelligent Miner for Data V6.1.1 server for OS/390

The IBM DB2 Intelligent Miner for Data V6.1.1 for OS/390 uses environment variables that point to the files needed to complete a specific task.

You can set environment variables in the server startup job. If these variables are omitted, the default values are used. See Appendix C, "Environment variables" on page 349 for a complete list of environment variables.

Using shared mining bases and result files

To allow users to share the same mining bases or result files, set the following environment variables to define common HFS directories:

IDM_MNB_DIR	If you set this variable, all mining bases are located in a common HFS directory. All clients connected to the server can access this directory and share the same mining bases. However, only one client at a time can make changes to a particular mining base and save these changes.
IDM_RES_DIR	If you set this variable, result files are written to a common HFS directory. All clients connected to the server can access this directory and share the result files. However, only one client at a time can make changes to a particular result file and save these changes.

Recommendation: When you use a common directory, any user can open and change any file in that directory. Thus there is a high risk of overwriting files accidentally. You can exclude this risk by using the default values. By default, the mining bases and result files are stored in the home directories of the clients.

Starting the IBM DB2 Intelligent Miner for Data V6.1.1 server for OS/390

To start the Intelligent Miner server program, submit the batch job provided during the installation. The program performs a loop while waiting for requests from a client. The program does not stop.

Multiple clients can now access the Intelligent Miner server. See Chapter 5, "Running Intelligent Miner clients" on page 57 for information on starting one of the supported clients.

Stopping the IBM DB2 Intelligent Miner for Data V6.1.1 server for OS/390

To stop the Intelligent Miner server program, cancel the batch job provided during the installation.

Unlocking mining bases from the OS/390 server

Normally, a mining base is locked when you open it, and unlocked when you close it. This ensures that only one person at a time can make changes to the mining base and save these changes.

If a mining run that was started from a client ends abnormally, the loaded mining base might is also locked. For example, if you shut down Windows and allow the operating system to stop the Intelligent Miner client rather than stopping all Intelligent Miner applications before this, the routine that unlocks the mining base is not started. If this is the case, the following message is displayed on the client when you open the mining base:

```
The mining base was loaded in READ-ONLY mode. To save changes, you must
save the mining base under a different name.
```

You must unlock the mining base manually before you can open it again by using the following procedure:

1. Sign on to the server with the same user ID as the client.
2. If you set the environment variable IDM_MNB_DIR in the Intelligent Miner startup job, you must also set it for the unlock program. Enter the following command:

   ```
   export IDM_MNB_DIR=/path
   ```

   where *path* is the path to your mining base directory.
3. Start the idmclear program, which resides in the /usr/lpp/IMiner/bin HFS directory.

   If IDM_MNB_DIR is set, the idmclear program unlocks mining bases in the directory identified by this environment variable.

   If IDM_MNB_DIR is not set, the idmclear program unlocks mining bases in the $HOME/idmmnb directory.

   Attention: The idmclear program unlocks all mining bases in a user's home directory or in a common directory identified by IDM_MNB_DIR. This includes mining bases that are regularly locked because a user opened them. Therefore, make sure that all mining bases in the common directory or user's home directory are closed before you run idmclear.

See "Unlocking mining bases from a client" on page 59 for information on how to unlock mining bases from a client workstation.

Running the Intelligent Miner server for the Solaris Operating Environment

The Intelligent Miner uses the environment variables `IDM_MNB_DIR` and `IDM_RES_DIR`, which point to the directories that contain the mining bases and result files respectively.

If you do not set these variables and use the default values, directories named idmmnb and idmres are created in each user's home directory.

Using shared mining bases and result files

Before starting the Intelligent Miner server, you can set the environment variable `IDM_MNB_DIR` to point to one common directory if multiple users share the same mining bases.

For example, if you want all users to share the mining bases in the idmmnb subdirectory of a directory named miningbases, enter the following command on the Korn shell:

```
export IDM_MNB_DIR=/miningbases
```

All Intelligent Miner clients connected to this server now share the same mining bases.

Restriction: Only one user at a time can have write access to a shared mining base.

You can set the environment variable `IDM_RES_DIR` in a similar way if you want all users to write their result files to a common directory.

For example, if you want all users to share the result files in the idmres subdirectory of a directory named resultfiles, enter the following command on the Korn shell:

```
export IDM_RES_DIR=/resultfiles
```

You can specify other environment variables. For a complete list, see Appendix C, "Environment variables" on page 349.

Starting the Intelligent Miner server for the Solaris Operating Environment

To start the Solaris Operating Environment server, enter one of the following commands on the command line:

idmstart This command starts the Solaris Operating Environment server.

idmstart -d	This command starts the Solaris Operating Environment server and shows the tracing information.

Now multiple clients can access the Intelligent Miner server. See Chapter 5, "Running Intelligent Miner clients" on page 57 for more information.

Stopping the Intelligent Miner server for the Solaris Operating Environment

To stop the Solaris Operating Environment server:

1. Log in by using the same user ID that started the server or by using the user ID root.
2. Enter `idmstop` on the command line.

Tips:

1. You can add the `idmstart` command to the /etc/inittab file of your system setup to start the Intelligent Miner automatically after an operating-system failure.
2. You can use the `crontab` command to create cron jobs for idmstart and idmstop. These jobs make the Intelligent Miner automatically available for certain time slots in your system setup.

Unlocking mining bases from the Solaris Operating Environment server

Normally, a mining base is locked when you open it, and unlocked when you close it. This ensures that only one person at a time can make changes to the mining base and save these changes.

If a mining run that was started from a client ends abnormally, the loaded mining base might is also locked. For example, if you shut down Windows and allow the operating system to stop the Intelligent Miner client rather than stopping all Intelligent Miner applications before this, the routine that unlocks the mining base is not started. If this is the case, the following message is displayed on the client when you open the mining base:

```
The mining base was loaded in READ-ONLY mode. To save changes, you must
save the mining base under a different name.
```

You must unlock the mining base manually before you can open it again by using the following procedure:

1. Sign on to the server with the same user profile as the client.
2. Start the idmclear program, which resides in the /opt/IMiner/bin directory.

   If IDM_MNB_DIR is set, the idmclear program unlocks mining bases in the directory identified by this environment variable.

   If IDM_MNB_DIR is not set, the idmclear program unlocks mining bases in the $HOME directory.

Attention: The idmclear program unlocks all mining bases in a user's home directory or in a common directory identified by IDM_MNB_DIR. This includes mining bases that are regularly locked because a user opened them. Therefore, make sure that all mining bases in the common directory or user's home directory are closed before you run idmclear.

See "Unlocking mining bases from a client" on page 59 for information on how to unlock mining bases from a client workstation.

Parallel processing in the Solaris Operating Environment

This section provides information about parallel processing in the Solaris Operating Environment using the IBM DB2 Intelligent Miner for Data.

If you run the Intelligent Miner on parallel processing units, the performance of the mining runs improves significantly.

You can run most of the mining functions and the Copy Records to File preprocessing function on parallel processing units. The mining functions that you cannot run in parallel mode are listed here:

- Similar Sequences
- RBF-Prediction in:
  - Training mode
  - Test mode

Before you can run the Intelligent Miner on parallel nodes, you must specify the nodes that you want to use and partition the input data. See "Specifying parallel parameters" on page 118 and "Partitioning the input data" on page 119 for more information.

Setting up the parallel operating environment for Solaris Operating Environment

To run the Intelligent Miner in parallel mode on a Solaris Operating Environment server, you must install and configure Sun HPC Software Foundation Package Version 2.0 or Sun HPC Cluster Tools Version 3.0, abbreviated in this chapter to *Sun HPC 2.0* and *Sun HPC 3.0*. You must also start the run-time environment. The run-time environment for Sun HPC 2.0 is called RTE; the run-time environment for Sun HPC 3.0 is called CRE. To perform these tasks, you must have system administrator's rights on the Solaris Operating Environment server. For more information, see the Sun HPC software documentation.

Specifying the parallel nodes on Solaris Operating Environment

The run-time environment of the Sun HPC software organizes the nodes in the Sun HPC system into logical sets, which are called partitions, and distributes the parallel processes to the nodes in a particular partition. To specify the partition that you want to use in a Sun HPC 2.0 environment, you can set the TMRUN_FLAGS environment variable. In a Sun HPC 3.0 environment, you can set an environment variable called MPRUN_FLAGS. If you do not set the appropriate variable for your Sun HPC environment, the Intelligent Miner uses the default partition defined by the system administrator. For more information, see the Sun HPC software documentation.

Running the Intelligent Miner server for Windows

The Intelligent Miner uses the environment variables `IDM_MNB_DIR` and `IDM_RES_DIR`, which point to the directories that contain the mining bases and result files respectively.

If you do not set these variables and use the default values, directories named idmmnb and idmres are created in the directory specified by the `IDM_HOME_DIR\<userid>` environment variable.

Using shared mining bases and result files

If you want multiple users to share the same mining bases, you can set the system variable `IDM_MNB_DIR` on the Windows Control Panel to one common directory.

Restriction: Only one user at a time can have write access to a shared mining base.

You can set the environment variable `IDM_RES_DIR` in a similar way if you want all users to write their result files to a common directory.

You can specify other environment variables. For a complete list, see Appendix C, “Environment variables” on page 349.

Starting the Intelligent Miner server for Windows

You start the Intelligent Miner Windows server as a native Windows service called IBM Intelligent Miner. The IBM Intelligent Miner service starts automatically when the system starts.

To start the IBM Intelligent Miner service manually on Windows NT:

1. Open the Windows NT Control Panel and double-click the **Services** icon. The Services window opens.
2. Select IBM Intelligent Miner Service from the list.
3. Click **Start**.
4. Click **Close** to close the Services window.

To start the IBM Intelligent Miner service manually on Windows 2000 and on Windows XP:

1. Click **Start** → **Settings** → **Control Panel** → **Administrative Tools** → **Services**. The Services window opens.
2. Select IBM Intelligent Miner service from the list.
3. From the **Context** menu, select **Start**.
4. Click **Close** to close the Services window.

Alternatively, you can enter `idmd -start` in a Command Prompt window to start the IBM Intelligent Miner service.

Portmapper: The installation process installs an NT service called *NobleNet Portmapper for TCP*, which is required to run the server.

If you have other software installed that makes use of portmappers, such as NFS, make sure that the NobleNet Portmapper for TCP service starts before the other programs. Otherwise the Intelligent Miner server might not work.

By default, this service is installed so that it starts automatically when the system starts.

To start or stop the NobleNet Portmapper for TCP service manually on Windows NT:

1. Open the Windows Control Panel and double-click the **Services** icon. The Services window opens.
2. Select the NobleNet Portmapper for TCP service from the list.
3. Click **Start** or **Stop**.
4. Click **Close** to close the Services window.

To start or stop the NobleNet Portmapper for TCP service manually on Windows 2000 and on Windows XP:

1. Click **Start** → **Settings** → **Control Panel** → **Administrative Tools** → **Services**. The Services window opens.
2. Select NobleNet Portmapper for TCP service from the list.
3. From the **Context** menu, select **Start** or **Stop**.
4. Click **Close** to close the Services window.

You can run the server without the portmapper by setting the environment variable IDM_SERVER_PORT. When you set this variable, the client/server communication bypasses the portmapper so that you can stop the portmapper service. You must set the IDM_SERVER_PORT variable on the server and on each client workstation connected to the server. See Appendix C, “Environment variables” on page 349 for more information.

To set the IDM_SERVER_PORT environment variable on Windows NT, follow these steps:

1. Open the Windows Control Panel.
2. Double-click the **System** icon.
3. Select the Environment page.
4. Set the variable to a port and add it to the list.

To set the IDM_SERVER_PORT environment variable on Windows 2000 and on Windows XP, follow these steps:

1. Click **Start** → **Settings** → **Control Panel** → **System**. The System Properties Window opens.
2. Click **Advanced** → **Environment Variables**. The Environment Variables Window opens.
3. Select **New** in the System Variables section. A pop-up window is displayed where you can specify the variable name and the port number.
4. To close all windows, click **OK** in each window.

Stopping the Intelligent Miner server for Windows

If the server was started by the IBM Intelligent Miner service, you can stop the service manually.

On Windows NT, follow these steps:

1. Open the Windows Control Panel and double-click the **Services** icon. The Services window opens.
2. Select the IBM Intelligent Miner service from the list.
3. Click **Stop**.
4. Click **Close** to close the Services window.

On Windows 2000 and on Windows XP, follow these steps:

1. Click **Start** → **Settings** → **Control Panel** → **Administrative Tools** → **Services**. The Services window opens.
2. Select the IBM Intelligent Miner service from the list.
3. From the **Context** menu, select **Stop**.
4. Click **Close** to close the Services window.

Alternatively, you can enter `idmd -stop` in a Command Prompt window to stop the IBM Intelligent Miner service.

Unlocking mining bases from the Windows server

Normally, a mining base is locked when you open it, and unlocked when you close it. This ensures that only one person at a time can make changes to the mining base and save these changes.

If a mining run that was started from a client ends abnormally, the loaded mining base might is also locked. For example, if you shut down Windows and allow the operating system to stop the Intelligent Miner client rather than stopping all Intelligent Miner applications before this, the routine that unlocks the mining base is not started. If this is the case, the following message is displayed on the client when you open the mining base:

```
The mining base was loaded in READ-ONLY mode. To save changes, you must
save the mining base under a different name.
```

You must unlock the mining base manually before you can open it again by using the following procedure:

1. Log on to the Windows server with the same user ID as the client.
2. Open a Command Prompt window.
3. Change to the directory identified by the %IDM_BIN_DIR% environment variable. This is the bin directory of your Intelligent Miner installation, for example, C:\im\bin.
4. Enter `idmclear`.

   If IDM_MNB_DIR is set, the idmclear program unlocks mining bases in the directory identified by this environment variable.

   If IDM_MNB_DIR is not set, the idmclear program unlocks mining bases in the directory identified by %IDM_HOME_DIR%\%USERNAME%.

   Attention: The idmclear program unlocks all mining bases in a user's home directory or in a common directory identified by IDM_MNB_DIR. This includes mining bases that are regularly locked because a user opened them. Therefore, make sure that all mining bases in the common directory or user's home directory are closed before you run idmclear.

See "Unlocking mining bases from a client" on page 59 for information on how to unlock mining bases from a client workstation.

Parallel processing on Windows

This section provides information about parallel processing on Windows using the IBM DB2 Intelligent Miner for Data.

If you run the Intelligent Miner on parallel processing units, the performance of the mining runs improves significantly.

You can run most of the mining functions and the Copy Records to File preprocessing function on parallel processing units. The mining functions that you cannot run in parallel mode are listed here:
- Similar Sequences
- RBF-Prediction in:

  – Training mode
  – Test mode

Before you can run the Intelligent Miner on parallel nodes, you must specify the nodes that you want to use and partition the input data. See “Specifying parallel parameters” on page 118 and “Partitioning the input data” on page 119 for more information.

Setting up the parallel operating environment for Windows

The IBM DB2 Intelligent Miner for Data supports SMP architectures and SMP cluster configurations. This allows you to run the Intelligent Miner on multiple processors or workstations simultaneously. To exploit the IBM DB2 Intelligent Miner for Data’s SMP capabilities, you must install the *Critical Software WMPI* software and the Intelligent Miner parallel server component.

The Critical Software WMPI product is not part of the Intelligent Miner package. You must obtain an appropriate license from Critical Software SA. For more information, visit their Web site at http://www.criticalsoftware.com/

To run the Intelligent Miner on a computer with more than one processor, install the Critical Software WMPI product and the parallel server component of the IBM DB2 Intelligent Miner for Data on that computer.

To run the Intelligent Miner in a cluster environment consisting of multiple workstations, you must install the Critical Software WMPI product and the parallel server component of the IBM DB2 Intelligent Miner for Data on each workstation involved. You can run the mining functions and the statistics functions on different workstations within the cluster. You can also specify the number of nodes that each workstation employs for that purpose. To exploit the parallel capabilities of the Intelligent Miner in such an environment, follow these instructions:

- Install the Windows server software on each workstation that you want to use for parallel mining runs.
- On each machine, install the server software in the same directory, that is, the name of and the path to the directory must be the same.
- In a cluster involving multiple workstations with one or more processing nodes, one of the nodes is the submit node, on which the Intelligent Miner starts the mining processes. The submit node coordinates the distribution of the various parallel processes to the other processing nodes. To start the initial processes on the submit node, the Intelligent Miner employs the Intelligent Miner for Data Windows service. Make sure that this Windows service runs properly on the submit node.
- The Critical Software WMPI service establishes the communication between the submit node and the other processing nodes. The Critical Software

WMPI service initiates and controls the processes on all the nodes. Hence Intelligent Miner processes are started under the same user ID as the Critical Software WMPI service. Therefore, make sure that you provide the user who starts the Critical Software WMPI service with sufficient rights. The user's authority must include the right to open all files and DB2 tables that are usually accessed from the workstations involved. To change a user ID or its access rights, open the Services window from the Windows Control Panel.

Specifying the parallel nodes on Windows

In a cluster environment, the Intelligent Miner uses information stored in a host-list file to determine the nodes for a parallel mining run. To specify the nodes, proceed as follows:

1. Create a host-list file named idmhost.list in the bin directory, which is a subdirectory of the directory in which you installed the Intelligent Miner. It is identified by the IDM_BIN_DIR environment variable. So if you are not sure where this directory is located, check to which path the IDM_BIN_DIR environment variable is set.
2. In the idmhost.list file, list the host names of the workstations including the nodes that you want to use. See the sample host-list file below.

```
#"local" specifies the local workstation
#
#Allocate 3 additional processes to local machine (Intelligent Miner
#always allocates 1 process to the local machine. So in this example, 4
#processes run on the local machine)
local
local
local
#
#Allocate 2 processes to zeus
zeus
zeus
#
#Allocate 1 process to artemis
artemis
```

Lines starting with a # are comments that do not affect the process. If you list a host name more than once, an according number of processes is started on the corresponding workstation. The Intelligent Miner reads the idmhost.list file from top to bottom. The file must contain at least one entry per process. Surplus entries are ignored. You must group multiple entries for the same machine together. Otherwise, only the first entry or group of entries is used.

In the example, the Intelligent Miner uses four nodes on the local workstation, two nodes on the workstation with the host name *zeus*, and one node on the workstation named *artemis* for the parallel mining run.

Important: List only those nodes in the host-list file for which you obtained a valid license of Critical Software WMPI.

Chapter 5. Running Intelligent Miner clients

The following chapter describes how to run the Intelligent Miner client software on AIX and Windows operating systems. It also describes how to unlock mining bases from a client workstation.

Running AIX clients

The following sections describe how to set the timeout variable and how to start and stop an Intelligent Miner AIX client. The timeout variable is one of the environment variables that you can set to adjust the behavior of the Intelligent Miner. For a complete list of environment variables, see Appendix C, "Environment variables" on page 349.

Setting the timeout variable

The timeout variable determines the time period within which the server must respond to a client request. If the server does not respond within this time period, the request is canceled, and an error message is displayed on the client workstation. Increase the timeout value if unusually many requests are canceled because they exceed the timeout limit.

To set the timeout variable to 10 seconds on AIX, use the following command on the ksh shell:

```
export IDM_CS_TIMEOUT=10
```

Note: Mining runs can last hours or even days if they are started asynchronously on the server. The timeout variable does not correlate to the run time of your mining runs. The timeout variable depends only on your network speed.

To run the AIX client, first start a server. For more information, see Chapter 4, "Running Intelligent Miner servers" on page 35.

Starting AIX clients

To start the Intelligent Miner, your system must be in graphical mode.

To start the AIX client, enter `im` on the command line. The Intelligent Miner main window opens.

If the Intelligent Miner server is on a machine other than the client, the Preferences notebook opens and shows the Server logon page. Connect to a server as described in the wizard.

Stopping AIX clients

To stop the Intelligent Miner client, click the **Close system** icon.

Running Windows clients

The following sections describe how to set the timeout variable and how to start and stop an Intelligent Miner Windows client. The timeout variable is one of the environment variables that you can set to adjust the behavior of the Intelligent Miner. For a complete list of environment variables, see Appendix C, “Environment variables” on page 349.

You can specify environment variables by using the Windows system settings.

Setting the timeout variable

The timeout variable determines the time period within which the server must respond to a client request. If the server does not respond within this time period, the request is canceled, and an error message is displayed on the client workstation. Increase the timeout value if unusually many requests are canceled because they exceed the timeout limit.

For example, to set the timeout variable to 10 seconds, you can:

- Add `set IDM_CS_TIMEOUT=10` to the IM.BAT file.
- Use the Windows system settings to set the environment variable IDM_CS_TIMEOUT to 10. For more information, see your Windows documentation.

Note: Mining runs can last hours or even days if they are started asynchronously on the server. The timeout variable does not correlate to the run time of your mining runs. The timeout variable depends only on your network speed.

To run the Windows client, first start a server. For more information, see Chapter 4, “Running Intelligent Miner servers” on page 35.

Starting Windows clients

To start a Windows client, you can use one of the following methods:

- Enter `IM.BAT` in a Command Prompt window.
- Double-click the **Intelligent Miner** icon.

The Intelligent Miner main window opens. Then the Preferences notebook opens, showing the Server logon page. Connect to a server as described in the wizard.

Stopping Windows clients

To stop the Windows client, select **Mining Base → Exit** from the menu bar in the Intelligent Miner main window.

Unlocking mining bases from a client

Normally, a mining base is locked when you open it, and unlocked when you close it. This ensures that only one person at a time can make changes to the mining base and save these changes.

If a mining run that was started from a client ends abnormally, the loaded mining base might is also locked. For example, if you shut down Windows 95 and allow the operating system to stop the Intelligent Miner client rather than stopping all Intelligent Miner applications before this, the routine that unlocks the mining base is not started. If this is the case, the following message is displayed on the client when you open the mining base:

```
The mining base was loaded in READ-ONLY mode. To save changes, you must
save the mining base under a different name.
```

You must unlock mining bases manually. To unlock mining bases from a client:

1. Enter the appropriate command to change to the directory in which you installed the Intelligent Miner.
2. Change to the bin directory.
3. Enter the following command, which works for all client operating systems:

   ```
   idmclmnb <hostname> <uid>
   ```

   For <hostname> and <uid>, use the values that you entered in the fields Server and USERID in the Preferences notebook. To see these fields, click the Server logon tab in the Preferences notebook.

   You are then prompted to enter the password for that server. Enter the password. The password is not displayed.

Attention: The idmclmnb program unlocks all mining bases in a user's home directory or in a common directory identified by IDM_MNB_DIR. This includes mining bases that are regularly locked because a user opened them. Therefore, make sure that all mining bases in the common directory or user's home directory are closed before you run idmclmnb.

Normally, a mining base is locked when you open it, and unlocked when you close it. This ensures that only one person at a time can make changes to the mining base and save these changes.

Chapter 6. Working with databases

Most functions of the Intelligent Miner can use input from either flat files or database tables. To take advantage of the preprocessing functions, however, you must have access to a database management system.

On AIX, in the Solaris Operating Environment, and on Windows operating systems, you can use IBM DB2 Universal Database Enterprise Server Edition, which is part of the Intelligent Miner product package. For the other operating systems, use an appropriate database management system.

The database can reside on the same server as the Intelligent Miner or on a different server. If it resides on a different server, you must use an appropriate database client product.

Important: If you use IBM DB2, note that the Intelligent Miner does not support the following DB2 data types:
- GRAPHIC
- VARGRAPHIC
- LONG VARGRAPHIC
- CLOB
- BLOB

Using DB2

On AIX, in the Solaris Operating Environment, and on Windows operating systems, you can use the Intelligent Miner with one of the DB2 products listed in Table 19.

Table 19. DB2 products that you can use with the Intelligent Miner on AIX

Software	Version
IBM DB2 Universal Database Enterprise Edition	7.2
IBM DB2 Universal Database Enterprise Extended Edition	7.2
IBM DB2 Universal Database Enterprise Server Edition	8.1

To use the Intelligent Miner with a DB2 database on AIX and in the Solaris Operating Environment:
1. Install your DB2 product on the Intelligent Miner server.

2. Before starting the Intelligent Miner server, set the following environment variable:
```
   DB2INSTANCE=xxxxx
```
   where xxxxx is the instance owner (the DB2 administrator).

   If the DB2INSTANCE variable is not set to the correct instance owner, you cannot access DB2.

To use the Intelligent Miner with a DB2 database on Windows, you must set the DB2INSTANCE variable as system variable.

If you are using the Intelligent Miner with DB2, specify different table names for the input data and the output data.

If the Intelligent Miner and your database reside on different servers, use the following products on the Intelligent Miner server to establish the necessary connection.

Table 20. Required database products when you are using different servers

Database management system (DBMS) on remote server	Required database product on the Intelligent Miner server
DB2 on workstations	DB2 client
DB2 on OS/390	DataJoiner or DB2 Federated
DB2 on iSeries	DataJoiner or DB2 Federated
Oracle or other non-DB2 DBMS	DataJoiner or DB2 Relational Connect

Working with different versions of DB2 on the same AIX server

If you work with more than one version of DB2 on the same AIX server, make sure that the Intelligent Miner accesses the correct run time library. This run time library is called *libdb2.a*.

Unless you set the LIBPATH environment variable or use a link to another directory, the Intelligent Miner tries to access the libdb2.a run time library in the /usr/opt/db2_08_01/lib directory.

To make sure that the Intelligent Miner accesses the correct run time library, follow the instructions in the appropriate sections below.

Running the Intelligent Miner in client/server mode

DB2 is always accessed through the Intelligent Miner server, unless you set the IDM_CLI_USED environment variable on a client workstation. For more information on this environment variable, see “Client variables” on page 351.

If you set the IDM_CLI_USED environment variable on an AIX client workstation, skip the remainder of this chapter and follow the instructions in "Running the Intelligent Miner in local mode" on page 64.

If you did not set the IDM_CLI_USED environment variable, see the appropriate sections that follow to enable access to the DB2 client libraries.

Working with DB2 Version 7.2: To access the correct run time library for DB2 Version 7.2, follow these steps:

1. Log on to the Intelligent Miner server as user root.
2. Set the DB2INSTANCE environment variable to the instance of DB2 Version 7.2 that you want to use.
3. Add /usr/lpp/db2_07_01/lib to your LIBPATH.
4. Call **idmstart** on the Intelligent Miner server.
5. Log off from the Intelligent Miner server.
6. Start the Intelligent Miner client.

Working with DB2 Universal Database Version 8: To access the correct run time library for DB2 Universal Database Version 8, follow these steps:

1. Check whether a link from the /usr/lib directory to a file named libdb2.a exists, where libdb2.a belongs to another DB2 version than DB2 V8.
2. If the link does not exist, proceed with step 3. If the link does exist, follow these steps:
   a. Log on to the Intelligent Miner server as user root.
   b. Set the DB2INSTANCE environment variable to the instance of DB2 Universal Database Version 8 that you want to use.
   c. Add /usr/opt/db2_08_01/lib to your LIBPATH.
   d. Call **idmstart** on the Intelligent Miner server.
   e. Log off from the Intelligent Miner server.
   f. Start the Intelligent Miner client.
   g. Skip step 3.
3. Follow these steps if the link does not exist:
   a. Log on to the Intelligent Miner server as user root or as another user.
   b. Set the DB2INSTANCE environment variable to the instance of DB2 Universal Database Version 8 that you want to use.
   c. Call **idmstart** on the Intelligent Miner server.
   d. Log off from the Intelligent Miner server.
   e. Start the Intelligent Miner client.

Running the Intelligent Miner in local mode

DB2 client calls are used to list tables and schemas in windows of the graphical user interface. Thus DB2 is accessed directly for these types of operations. For all other purposes, DB2 is accessed through the Intelligent Miner server.

To enable access to the correct DB2 run time library when you installed DB2, the Intelligent Miner server, and the Intelligent Miner client on the same AIX computer, follow the instructions in the appropriate sections below.

Working with DB2 Version 7.2: To access the correct run time library for DB2 Version 7.2, follow these steps:

1. Log on as user root.
2. Set the DB2INSTANCE environment variable to the instance of DB2 Version 7.2 that you want to use.
3. Add /usr/lpp/db2_07_01/lib to your LIBPATH.
4. Call **idmstart**.
5. Log off.
6. Log on as the Intelligent Miner user.
7. Set the DB2INSTANCE environment variable to the instance of DB2 Version 7.2 that you want to use.
8. Add /usr/lpp/db2_07_01/lib to your LIBPATH.
9. Start the graphical user interface of the Intelligent Miner.

Working with DB2 Universal Database Version 8: To access the correct run time library for DB2 Universal Database Version 8, follow these steps:

1. Check whether a link from the /usr/lib directory to a file named libdb2.a exists, where libdb2.a belongs to another DB2 version than DB2 V8.
2. If the link does not exist, proceed with step 3 on page 65. If the link does exist, follow these steps:
   a. Log on as user root.
   b. Set the DB2INSTANCE environment variable to the instance of DB2 Universal Database Version 8 that you want to use.
   c. Add /usr/opt/db2_08_01/lib to your LIBPATH.
   d. Call **idmstart**.
   e. Log off.
   f. Log on as the Intelligent Miner user.
   g. Set the DB2INSTANCE environment variable to the instance of DB2 Universal Database Version 8 that you want to use.
   h. Add /usr/lpp/db2_08_01/lib to your LIBPATH.
   i. Start the graphical user interface of the Intelligent Miner.

   j. Skip step 3.
3. Follow these steps if the link does not exist:
   a. Log on as user root or as another user.
   b. Set the DB2INSTANCE environment variable to the instance of DB2 Universal Database Version 8 that you want to use.
   c. Call **idmstart** on the Intelligent Miner server.
   d. Log off.
   e. Log on as the Intelligent Miner user.
   f. Set the DB2INSTANCE environment variable to the instance of DB2 Universal Database Version 8 that you want to use.
   g. Start the graphical user interface of the Intelligent Miner.

Using the parallel versions of DB2

This section contains information about DB2 Universal Database Enterprise Extended Edition and DB2 Enterprise Server Edition in addition to that given in the previous section.

You can use any number of table partitions as input data.

If you run the Intelligent Miner in parallel mode and the number of table partitions equals the number of processing nodes, a single parallel node processes one table partition.

If the number of table partitions is higher than the number of nodes, each parallel node processes more than one table partition.

If you have more nodes than table partitions, the number of nodes involved in the parallel mining run is reduced to the number of table partitions.

Restriction: You cannot use views as input data for a parallel mining run.

The data records used for one of the parallel processes do not necessarily reside on the node the process is running on. Therefore, each DB2 table partition must be accessible from each node specified in the host-list file.

For the Associations and Sequential Patterns mining functions, records that belong to the same transaction or transaction group must be kept together. Therefore, use the following fields as partitioning keys for the tables:

Transaction ID
: For the Associations mining function

Transaction group field
: For the Sequential Patterns mining function

Some of the mining functions optionally generate tables as output data. The creation of these tables follows the rules in Chapter 7, "How the Intelligent Miner creates output data" on page 71.

Using DB2 for iSeries

To access DB2 data on an iSeries server:

1. Check the Program Directory or the QIDM/QYDMREADME file to verify that the program temporary fixes (PTFs) required for the operating system are installed.
2. Enter `DSPNETA` to verify that the system name, local control point name, and the default local location are the same.
3. Make sure that you defined a local database. Issue the **WRKRDBDIRE** command to check whether a relational database with a remote location of *LOCAL exists. The name of the database must match your system name. Issue the **WRKRDBDIRE** command again to define such a database if necessary.
4. Configure the clients so that DB2 CAE is used only when the database that you want to access resides on another server.

Using DB2 for OS/390

You can install the IBM DB2 products specified in the *Program Directory for IBM Intelligent Miner for OS/390*.

The Intelligent Miner server establishes and controls its connections to the database automatically through the Call Attachment Facility (CAF), provided that CAF has access to the DB2 system.

If you want to work with the preprocessing functions of the Intelligent Miner, set the server environment variable IDM_LOCAL_DB2_SSID to the appropriate DB2 subsystem ID. See Appendix C, "Environment variables" on page 349 for more information.

Configuring Intelligent Miner clients to work directly with databases

From an Intelligent Miner client, you can use flat files or database tables for any Intelligent Miner function without installing database software on the client. Access to the database is provided through the Intelligent Miner server. In general, this is the most convenient way to run Intelligent Miner clients.

You might want to connect your client directly to the database. To let the Intelligent Miner client access a database without using the Intelligent Miner communications interface, you must:

1. Enable the communications of the database server to facilitate remote access.

   See the appropriate documentation of your database management system.
2. Install and configure the required database client software.

   The software that you need depends on the operating system of the client and the database server that you are using and is described in the remainder of this chapter.

   Make sure that you catalog all databases that you want to access from the client.

   If your Intelligent Miner server is installed on AIX, the alias you choose on the client must match the alias cataloged on the server.

   If your Intelligent Miner server is installed on OS/390, the alias you choose on the client must match the name of the database subsystem.

   If your Intelligent Miner server is installed on iSeries, the alias you choose on the client must match the name of the database that is shown when you run the **WRKRDBDIRE** command. The name of this database must not be longer than eight characters. Generally, this name is the same as the LCLNETA value, which is shown when you run the **DSPNETA** command.
3. If you use DB2, set the following environment variables before starting the Intelligent Miner client:

   ```
   IDM_CLI_USED
   DB2INSTANCE=xxxxx
   ```

   where xxxxx is the instance owner (the DB2 administrator).

   Important: If the DB2INSTANCE variable is not set to the correct instance owner, the database cannot be accessed. Instance names for the Intelligent Miner client and the Intelligent Miner server can be different.

Software requirements for DB2

The following tables show the client products required for DB2 if you want to connect an Intelligent Miner client directly to the DB2 server rather than using the Intelligent Miner communications interface.

AIX client

Table 21 shows the software that is required if you want to access DB2 directly from an AIX client.

Table 21. Software required to access DB2 directly from an AIX client

DB2 platform	Required software on AIX client
AIX, Solaris Operating Environment, Windows	DB2 Client
iSeries	DB2 Connect

Table 21. Software required to access DB2 directly from an AIX client (continued)

DB2 platform	Required software on AIX client
OS/390	DB2 Connect

Windows client

Table 22 shows the software that is required if you want to access DB2 directly from a Windows client.

Table 22. Software required to access DB2 directly from a Windows client

DB2 platform	Required software on Windows clients
AIX, Solaris Operating Environment, Windows	DB2 Client
iSeries	DB2 Connect
OS/390	DB2 Connect

Accessing remote non-workstation DB2 databases or databases from different database management sytems

If you want want to access remote non-workstation databases, for example, a database on OS/390 or databases from different database management systems (DBMS), for example, Oracle, you must meet one of the following requirements on the Intelligent Miner server machine:

- DataJoiner Version 2.1.1 must be installed and configured.
- A federated DB2 database must be configured.

If you want to use a federated database and if you want to access databases from different DBMS databases, you must also install the DB2 feature DB2 Relational Connect on the Intelligent Miner server machine.

With DB2 V7, you can use a federated database only to read data from remote databases or from different DBMS databases. With DB2 V8, you can use a federated database also to write data to the remote database or to different DBMS databases.

The configuration of DataJoiner or a federated database involves executing SQL statements, for example, CREATE SERVER, CREATE NICKNAME. These statements make the remote database or the different DBMS database and the tables in this database accessible from the DataJoiner or the federated database. See the DataJoiner or the DB2 documentation to get more information about the configuration steps.

After creating nicknames for the remote tables, you can use these tables as input tables for the Intelligent Miner. If you want to use the Intelligent Miner to create output tables in a remote database or in a different DBMS database, write the name of the remote server into the **Tablespace** field during the creation of an output data object. The name of the remote server is the name that you have used in the CREATE SERVER statement.

If you use a federated database, select different names for tablespaces (CREATE TABLESPACE) and servers (CREATE SERVER). If you type a name in the **Tablespace** field, Intelligent Miner looks first for a tablespace with this name before it looks for a server with this name. If the name is identified as a tablespace, the output table is created in this tablespace.

Chapter 7. How the Intelligent Miner creates output data

Output data consists of the values that were calculated when you used a mining function, a preprocessing function, or a statistical function. In contrast to the results, output data contains the values calculated by the various functions. Results contain the data used to generate the graphical views or charts based on the calculated values. Most Intelligent Miner functions allow you to save output data.

The mining and statistical functions create output data as flat files or database tables. Preprocessing functions create output data as database tables or database views.

The following rules apply to all types of output data:
- If you select a data object that is marked as **Read only** on the output data page, the attempt to create output data fails.
- If the output data does not exist, an attempt is made to create it even if you select **Output should be appended to the specified table** on the database table or view page of the Data wizard.

Partitioned output tables

If your database management system supports partitioned tables and you specify output data of the type **Table**, the Intelligent Miner tries to create a partitioned output table provided that any of the following conditions or set of conditions applies:
- You specify a partitioned table space for the output data.
- You specify **Read and write** or **Overwrite existing records** for your output data, but no table space. The output data already exists as a partitioned table.
- You do not specify a table space for the output data and the input data is a partitioned table.

How the partitioning key is created

Partitioned tables require a partitioning key. Normally, this is not a problem. For all mining and statistics functions, the Intelligent Miner adds the appropriate fields automatically if you did not specify them in the lists of input and output fields. However, if a partitioning key cannot be determined for some reason, the attempt to create output data fails. Which partitioning key the Intelligent Miner uses is determined as follows:

- The Intelligent Miner uses the partitioning key of the input data if the following conditions apply:
  - The input data is a partitioned table.
  - All of the input table's partitioning key fields are included in the output data.
  - In the case of DB2 for OS/390, the number of partitions in table space of the input table is the same as the number of partitions in the table space of the output table.
- If the partitioning key of the input data cannot be used, the Intelligent Miner uses the partitioning key of an existing table if the following conditions apply:
  - The output data already exists as a partitioned table.
  - You specified the **Read and write** or **Overwrite existing records** option for the output data.
  - All of the existing table's partitioning key fields are included in the output data.
  - For DB2 for OS/390, the number of partitions in the table space of the existing table is the same as the number of partitions in the table space of the output table.
- If the partitioning key of an existing table cannot be used and the database management system creates default partitioning keys, the Intelligent Miner allows the database management system to create a default partitioning key for the partitioned output table.

Specifying table spaces

If your database management system supports table spaces, you can optionally specify a table space for the output data. Because table spaces apply to database tables only, any table space specification for output data of the type **View** is ignored.

DB2 table spaces on AIX, in the Solaris Operating Environment, and on Windows

If you use one of the DB2 for AIX products and you specify a table space, the table space must exist. If the specified table space does not exist, an output table cannot be created.

If you do not specify a table space, the table space of the output table is determined as follows:

- If the output table already exists and you specified the **Read and write** or **Overwrite existing records** option for the output data, the Intelligent Miner drops the existing table, and the new output table is created in the same table space.

- If the output data does not exist and the input data is a partitioned table, the Intelligent Miner creates the output table using the table space of the input table. As a result, the input table and the output table are collocated.
- In all other cases, the default table space of the database management is used.

DB2 table spaces on OS/390

If you use DB2 for OS/390 and you specify a table space, the table space need not exist.

If you specify an existing table space

To successfully create an output table, the following conditions must be met when you specify an existing table space.

- If the existing table space S1 is partitioned, the following rules apply:
  - The Intelligent Miner must be able to determine the output table’s partitioning key based on the input data or an existing output table as described under “Partitioned output tables” on page 71.
  - If a table T1 exists in the specified table space S1, the existing table must have the same name as the output data. The **Read and write** or **Overwrite existing records** option must be specified. If you selected one of these options, the existing table space S1 and the existing table T1 are dropped. A new table space S1 with the same name is created for the new output table T1.
  - If the output table T1 already exists in a table space S2, which is different from the specified table space, the specified table space S1 must be empty. The **Read and write** or **Overwrite existing records** option must be specified for the output data. If this is the case, the existing table space S2 and the existing table T1 are dropped. A new output table T1 is created in the specified table space S1.
- If the existing table space S1 is not partitioned, a nonpartitioned output table is created in the specified table space S1.

If you specify a table space that does not exist

If the table space S1 does not exist, the Intelligent Miner tries to create a table space S1 for the new output data. The new table space is created as follows:

- The Intelligent Miner uses the table space information of the input data if either of the following conditions applies:
  - The input data is a nonpartitioned table.
  - The input data is partitioned and all of the input table’s partitioning key fields are included in the output data.
- If the table space information of the input data cannot be used, the Intelligent Miner uses the table space information of an existing table space S1 if the following conditions are met:
  - The output data already exists in the specified table space S1.

  - The **Read and write** or **Overwrite existing records** option is specified for the output data.
  - All of the existing table's partitioning key fields are included in the output data.

If you specify a database, but no table space

If you specify a database D1 but do not specify a table space, the table space of the output table is determined as follows:

- If the output table exists in an implicit table space, the database management system creates the output table in the default table space.
- If the output table exists in a table space S1 in a different database D2, the Intelligent Miner attempts to create the output table in table space S1 in database D1.

  If the table space S1 exists in database D1, the table space is determined as described in "If you specify an existing table space" on page 73. Otherwise, the table space is determined as described in "If you specify a table space that does not exist" on page 73.
- If the output table exists in the table space S1 in the specified database D1, the table space is determined as described in "If you specify an existing table space" on page 73.

If you do not specify a database

If you do not specify a database, the Intelligent Miner attempts to create the output table in your default database.

Storage groups

When the Intelligent Miner creates table spaces and indexes for output data, the new table spaces and indexes are allocated to the storage groups whose names correspond to the value that you specify for the environment variable IDM_STOGROUP_PREFIX.

If the new table space or index is partitioned, the Intelligent Miner allocates them to the storage groups whose names begin with the value of the environment variable IDM_STOGROUP_PREFIX. The storage group whose name is equal to the value of the environment variable IDM_STOGROUP_PREFIX is reserved for nonpartitioned table spaces or indexes.

The Intelligent Miner distributes the partitions equally across the storage groups. For example, if there are five table space partitions and four storage groups, the Intelligent Miner allocates the first partition to the first storage group, the second partition to the second storage group, and so on. The fifth partition is allocated to the first storage group again.

The Intelligent Miner uses this procedure in reverse order to allocate partitioned indexes. In the previous example, the first index partition is allocated to the fourth storage group, the second index partition to the third storage group, and so on.

If the new table space or index is not partitioned, the Intelligent Miner allocates them to the storage group whose name is equal to the value of the environment variable IDM_STOGROUP_PREFIX.

Buffer pools

When the Intelligent Miner creates an output table, the DB2 for OS/390 buffer pool is determined as follows:

- The buffer pool of the input table is used if the output table resides on the same database server as the input table.
- The default buffer pool is used in all other cases.

Chapter 8. Performing common tasks

This chapter describes the basic concepts and features of the Intelligent Miner and explains how to perform tasks necessary to achieve most data mining goals.

Understanding basic concepts

In general, data mining in the Intelligent Miner is accomplished through the creation of interrelated objects. The objects are displayed as icons and represent the collection of attributes or settings that define the data or function. You create *settings objects* to perform a particular task. See "Creating and using settings objects" on page 80. Other objects are created by the Intelligent Miner, for example, result objects that contain the findings of a mining run.

You save the objects of a particular data mining project as a group called *mining base*. You can create a mining base for each mining objective or project that you put together.

Getting familiar with the Intelligent Miner main window

Figure 4 on page 78 shows the Intelligent Miner main window.

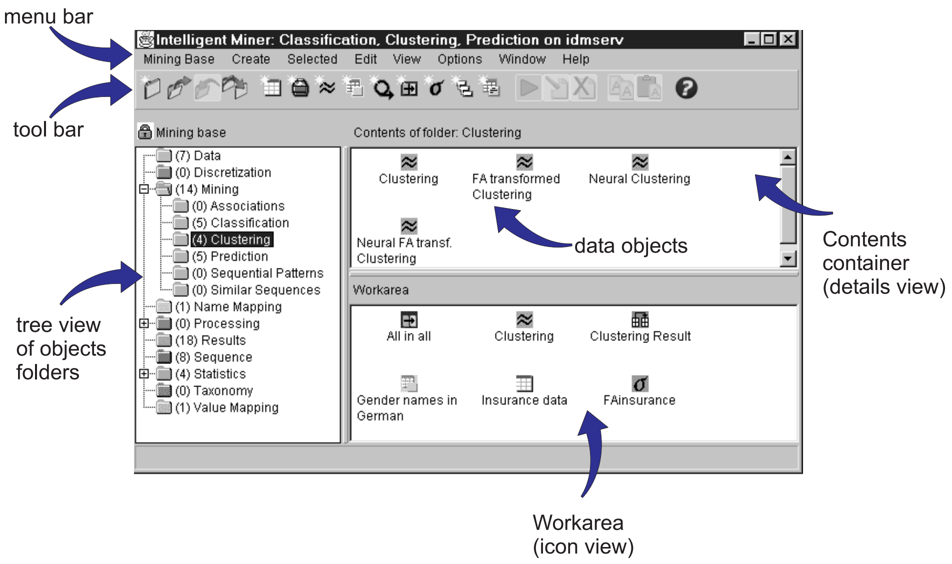


Figure 4. The Intelligent Miner main window

Main window areas

The main window has the following areas:

Mining base container

The mining base container on the left side of the main window shows the types of objects that are stored in a mining base. See Figure 4. There is a folder or subfolder for each type of Intelligent Miner object that you can create. You can expand folders by clicking on the plus sign next to the folder. Similarly, you can collapse folders by clicking on the minus sign next to an expanded folder. The number in parentheses in front of each folder displays the number of objects in that folder or in any of its subfolders. When you click a folder, the objects or subfolders are displayed in the contents container.

Contents container

You can work with settings objects of a given type in the contents container. The contents container is in the upper right corner of the main window. See Figure 4. Use the **View** menu to indicate how you want the objects displayed, for example, by icon or with details. You can select an object and click **Selected → Open** on the menu bar. This opens the settings notebook of the object or, if you selected a result object, the corresponding visualizer. You can also double-click the object.

Work area

The work area (lower right corner) allows you to work with settings objects of different types. See Figure 4 on page 78. You can drag and drop objects from the contents container into the work area to create shortcuts to your objects. This way, you can work with different types of objects without having to click folders and find objects. The work area contents of a mining base are saved as part of your preferences data. It is restored when you restart the client.

When you position the mouse pointer over the toolbar icons, brief explanations are displayed. More help is displayed at the bottom of the main window when you position the mouse pointer over some of the interface elements.

Creating and using mining bases

Mining bases are displayed in the main window of the Intelligent Miner, as shown in Figure 4 on page 78. In the main window, you also work with the objects that belong to the mining bases. The mining base holds descriptive information about the objects. It does not contain the data that is being analyzed.

Creating objects in the mining base is a task that you must perform frequently when using the Intelligent Miner. The task of creating an object is essentially the same, no matter which type of object you create. A wizard guides you through the process.

Before you can manipulate objects in a mining base, you must open it. If your client runs on a system other than the Intelligent Miner server, you must first select a server and provide a user ID and password. These and other preferences can be set in the Preferences notebook. See “Setting preferences for the Intelligent Miner” on page 129.

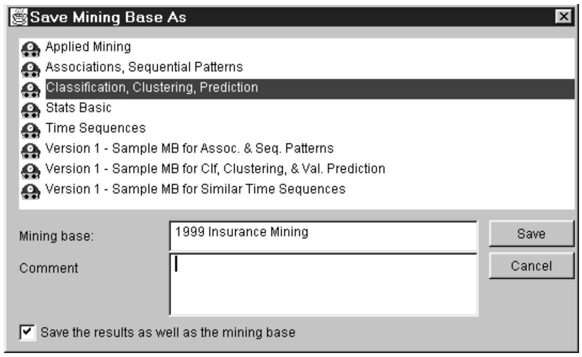
However, when you run the Intelligent Miner in local mode or in stand-alone mode, that is, client and server software are installed on the same machine, you can open mining bases directly.

The main window provides a standard menu bar and taskbar for opening mining bases and creating objects. Use the **Mining Base** menu or taskbar icon to open an existing mining base or create a new one. With a new mining base you can begin creating objects that can be stored as part of the new mining base.

Important: You can have only one mining base open at a time. In general, settings objects in the same mining base can interact with each other, but objects in different mining bases cannot. To overcome this problem, you can use the import facility, which allows you to merge two or more mining bases.

Saving mining bases

Save the current mining base by clicking **Mining Base → Save Mining Base** on the menu bar. Because mining bases are stored on the server, they can be accessed by multiple client workstations. Figure 5 shows the Save Mining Base As window.


Save Mining Base As
Applied Mining
Associations, Sequential Patterns
Classification, Clustering, Prediction
Stats Basic
Time Sequences
Version 1 - Sample MB for Assoc. & Seq. Patterns
Version 1 - Sample MB for Clf, Clustering, & Val. Prediction
Version 1 - Sample MB for Similar Time Sequences
Save
Cancel

Mining base: 1999 Insurance Mining

Comment:

[✓] Save the results as well as the mining base

Figure 5. The Save Mining Base As window

Saving new mining bases

To save a new mining base, follow these steps:

1. Enter a name in the **Mining base** field.

   You can also add a description in the **Comment** field.

   Alternatively, you can select an existing name, and then change the name and its comment.
2. Click the **Save** push button.

Overwriting existing mining bases

To overwrite an existing mining base, follow these steps:

1. Select a mining base in the list.
2. Click the **Save** push button. You are asked to confirm the replacement of the existing mining base.
3. Click the **OK** push button.

Creating and using settings objects

When you work with the Intelligent Miner, creating settings objects is one of the fundamental tasks. Wizards help you define settings objects by prompting you to fill in required fields and attributes. Click an existing object icon to open a notebook in which you can change the current specifications.

Information about the use of settings objects is provided in the topics that follow:

- “Overview of settings objects”
- “Creating settings objects using wizards” on page 82
- “Changing settings objects” on page 85
- “Data objects” on page 85
- “Mining and statistics settings objects” on page 96
- “Preprocessing settings objects” on page 96
- “Sequence settings objects” on page 99
- “Discretization objects” on page 100
- “Name mapping and value mapping objects” on page 103
- “Taxonomies” on page 109
- “Value mapping objects” on page 104

The functions and settings of particular settings objects are described in the wizards and notebook helps for those objects.

Advanced pages and controls

In addition to the basic pages of wizards or settings notebooks, you can use advanced pages and controls with mining, statistics, and data objects.

When using the basic pages, you can only specify or modify basic parameters. The advanced parameters are hidden from view, but their settings are used when you run the object. When using the advanced pages, you can additionally modify the advanced parameters.

To use the advanced pages, click the **Show the advanced pages and controls** check box on the Settings page of a wizard or settings object.

Tip: You can use the advanced pages and controls by default. Click **Options → Preferences** on the menu bar and change the selection of the appropriate check box on the Defaults page of the Preferences notebook.

Overview of settings objects

You create settings objects using wizards and modify them using settings notebooks.

Settings objects have built-in relationships to other objects in the same mining base. For example, a mining settings object requires that you specify the input data, either by creating a new data object or by referring to an existing one. The mining settings object represents parameters that you specified for a mining function. One of these parameters is the name of the data object, which serves as a logical description of the input data.

You must provide a name for settings objects. Objects of the same type must have unique names, but objects of different types can have identical names. For example, a result object is by default given the same name as the mining settings object that produced it to show that the two objects are related.

You can also add a comment to the settings object. You can copy, rename, and delete objects. Select the object and then click the **Selected** menu on the menu bar or right-click the object to see a menu of actions that you can take for that object.

You can run the following settings objects using the **Run** button or the menus:

Mining	Results are stored as a result object. You can also specify that output data be created, which requires an output data object.
Statistics	The results are stored as a result object. You can also specify that output data be created, which requires an output data object.
Preprocessing	Most of the preprocessing settings objects produce DB2 tables or views. Input tables or views are not updated.
Sequence	The results depend on the settings objects that are part of the sequence. See “Sequence settings objects” on page 99 for more information about running multiple settings objects in a sequence.

While a mining, statistics, preprocessing, or sequence settings object is running, the Intelligent Miner progress indicator provides information about the status of the process.

Creating settings objects using wizards

To create a settings object, use the **Create** menu or click a settings-object button on the task bar. A wizard guides you through creating the object.

Figure 6 on page 83 shows the task-bar buttons that represent the Intelligent Miner settings objects.

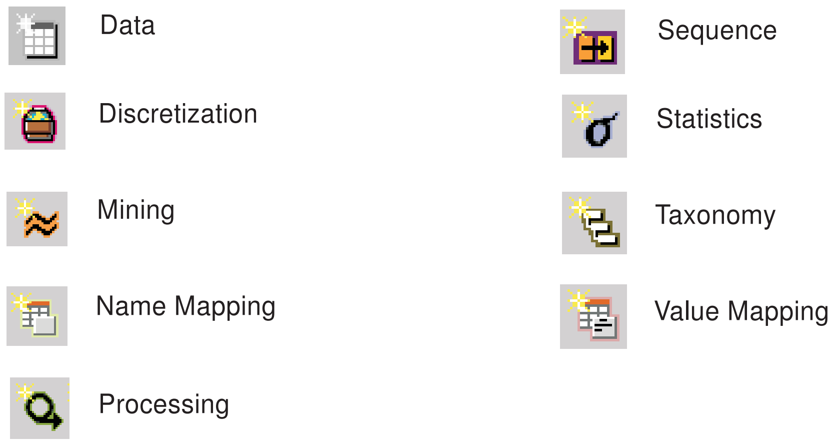


Figure 6. Task-bar buttons representing Intelligent Miner settings objects

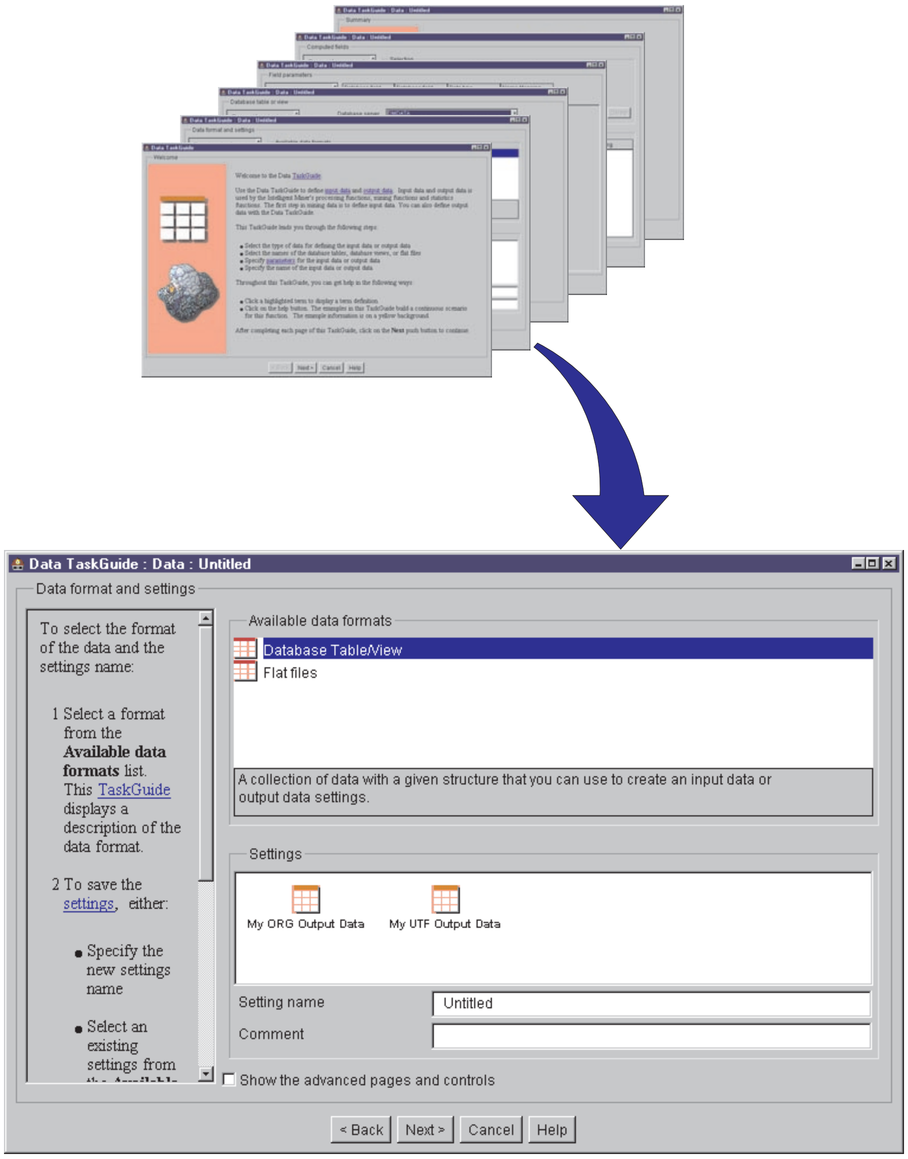
Most wizards and settings notebooks provide links to other wizards. This allows you to create related objects that are frequently used in conjunction with the main object you are creating. The advantage of this method is that you can immediately connect the objects with one another. To follow such a link, you just click a push button, and the wizard for the related object opens. You can easily recognize the type of object that you create when you follow such link because the links use the push buttons that you also find on the task bar.

Each wizard starts with a Welcome page that provides an overview of the type of settings object that you are creating. Advanced users can suppress the display of the Welcome page by checking **Skip the Welcome page when opening a wizard** on the Miscellaneous page in the Preferences notebook. See "Setting preferences for the Intelligent Miner" on page 129 for more information.

Each wizard page provides step-by-step instructions for filling in the fields and making selections that define the settings for the object. You can click a highlighted term to see a short definition of the term.

After you provided the required values, you can click the **Next** push button to navigate to the next wizard page. At any time, you can return to an earlier page by means of the **Back** push button. The last page of every wizard summarizes the settings object that you created. Click the **Finish** button to create the object.

Figure 7 on page 84 shows the wizard for creating a data object.



Data TaskGuide : Data : Untitled

Data format and settings

To select the format of the data and the settings name:

1 Select a format from the **Available data formats** list. This TaskGuide displays a description of the data format.

2 To save the settings, either:

- Specify the new settings name
- Select an existing settings from

Available data formats

Database Table/View

Flat files

A collection of data with a given structure that you can use to create an input data or output data settings.

Settings

My ORG Output Data

My UTF Output Data

Setting name: Untitled

Comment:

Show the advanced pages and controls

< Back | Next > | Cancel | Help

Figure 7. The Data wizard

You can have more than one wizard open at a time. This allows you to temporarily leave a wizard to create another object that is necessary to complete the first wizard. For example, while you are defining a mining function, you might need to define or modify an input data object. You can open a Data wizard to define an input data object, then continue with the mining wizard.

Changing settings objects

To change the specifications of an object, double-click the object icon in the contents container or work area. You can also select the object and click **Selected → Open**. The pages of the settings notebook match those of the wizard for the object, but there is no Welcome page. The values for the object are filled in, and there are tabs on top of each page so that you can go directly to the parameter that you want to check or change.

Click the **OK** push button to apply the changes and close the notebook. Click the **Apply** push button to apply your changes and keep the settings notebook open.

When you change the settings in a notebook, you can also change the settings of related objects from the same notebook. In the selection panes that allow you to connect your current object with another object, you can double-click the appropriate object icon to open the related object's settings notebook.

Data objects

Data objects are logical descriptions of the data that resides in a flat file or database. When you create a data object, you are specifying the layout and the location of the data that you want to use as input or to generate as output.

Creating data objects

Before you can use any data with a mining or statistics function, you must create a data object that refers to that data. You then specify the data object as an input data object or as an output data object of the function that you want to use. To define a data object, follow these instructions:

1. Start the Data wizard by using one of the following methods:
   - Click **Create → Data** on the menu bar in the Intelligent Miner main window.
   - Click the **Create data** push button on the Input data page or Output data page of a wizard or settings notebook.
2. If the Welcome page is displayed, click the **Next** push button to access the Data format and settings page.
3. Depending on the format of your data, select **Database Table/View** or **Flat files** in the **Available data formats** pane.
4. Enter a name for the data object in the **Settings name** field.

   You can also add a description in the **Comment** field.
5. Click the **Next** push button to access the next page.
6. Depending on your selection in step 3, proceed with "Database Table/View" or "Flat files" on page 87.

Database Table/View

If you selected **Database Table/View**, follow these steps:

1. Select a database server from the drop-down list. The available schemata are displayed in the **Schema** list.
2. Select a schema from the **Schema** list. The tables and views that use the selected schema are displayed in the **Database tables and views** list.
3. Select a database table or view from the **Database tables and views list**.
4. Select the use mode. The following use modes are available:

   Read and write
   Select this use mode if you repeatedly want to use this data object as an output data object.

   This use mode is selected by default.

   Read only
   Select this use mode if you do not want an input data object to be used as an output data object. The data is protected against overwriting or modification when accessed through this particular data object.

   Write once
   Select this use mode if you want to append the output data of a particular mining run to the input data and protect the original data and the new data against future modification. Also select this mode if a mining or statistics function creates new data that you only want to use as input later. The data can be modified only once when it is accessed through this particular data object.
5. If you want to use the table as an output data object, check **The specified table does not yet exist** and continue with step 4b on page 127. Otherwise, click the **Next** push button to access the Field parameters page. Settings on this page are optional. You can change the data type of the fields in the selected table and assign name mapping objects to the fields. See "Assigning name mapping objects to fields in data objects" on page 107 for more information on how to assign name mapping objects.
6. Click the **Next** push button to access the Computed fields page. Specifications on this page are optional. You can define fields to be computed when you want the Intelligent Miner to perform discretization, value mapping, or mathematical operations. See "Computed fields" on page 90 for more information.
7. If you use advanced pages and controls, you can access additional pages on which you can specify bucket limits, valid values, and cyclic fields. Click the **Next** push button to navigate through these pages. Click the **Help** push button to get information about the parameters that you can specify.
8. Click the **Next** push button to access the Summary page. Your settings are displayed in the Summary list.
9. Click the **Finish** push button to save the data object.

Flat files

If you selected **Flat files**, follow these steps:

1. Select the directory in which the file that you want to use is located by navigating through the tree view in the upper left corner of the page.
2. Select the file in the list on the right of the tree view.

   Instead of navigating through the tree view and selecting the file in the list, you can also enter the path followed by the file name in the **Path and file name** field.

   Restrictions:

   - **OS/390 server:** If your data is stored in MVS data sets rather than HFS files, a folder tree and a file selection list are not available. To specify an MVS data set, you must enter its TSO name preceded by two slashes in the **Path and file name** field, as in this example:

//'HLQ.xxx.yyy'	Specifies a data set with the high-level qualifier.
//xxx.yyy	Specifies a data set without the high-level qualifier.
//'HLQ.xx.yyy(member1)'	Specifies a partitioned data set with the high-level qualifier.

   - **Windows server:** To switch to another drive on a Windows server, you cannot use the tree view. Enter the drive and the path to the file in the **Path and file name** field.
3. Click the **Add file** push button. To use more than one file, repeat the last three steps. If you use more than one file, the data in any additional file must be organized in the same way because the Intelligent Miner treats all selected files as one large file.
4. Select the use mode as in step 4 on page 86.
5. If you want to use the file as an output data object, you can check the **The specified flat file does not yet exist** box and continue with step 4b on page 127. Otherwise, click the **Next** push button to access the Field parameters page. The Flat file display shows how the data is arranged in the selected flat files.
6. Look at the Flat file display to identify the column range of the field that you want to define.
7. Enter the number of the first column, the range delimiter, and the number of the last column of that field in the **Begin and end position** field, as in this example:

```
11-19
```

Hints:

- You can also specify a field whose information is stored in more than one column. End the definition for the first column with a semicolon, and add the column numbers for any additional column in the same manner. Here is an example:

  ```
  11-19;24-28
  ```
- The above examples use the default range delimiter (dash) and the default list delimiter (semicolon). If you defined other delimiters, use the delimiters you specified rather than the default delimiters.

8. Enter a unique name in the **Field name** field.
9. Select a suitable data type from the **Data type** drop-down list. See “Selecting the data type” on page 89 for a description of the available data types.
10. Optionally, assign a name mapping object to the field. See “Assigning name mapping objects to fields in data objects” on page 107 for more information.
11. Click the **Add** push button. The field is displayed in the list called Matrix at the bottom of the page.

    You can also change an entry in the list. Select the field in the list called Matrix. The field parameters are displayed in the entry fields where you originally defined them, and you can make the necessary changes. Click the **Update** push button to apply your changes.

    To delete a field, select it in the list and click the **Delete** push button.
12. Click the **Next** push button to access the Computed fields page. Settings on this page are optional. You can define fields to be computed when you want the Intelligent Miner to perform discretization, value mapping, or mathematical operations. See “Computed fields” on page 90 for more information.
13. If you use advanced pages and controls, you can access additional pages on which you can specify bucket limits, valid values, and cyclic fields. Click the **Next** push button to navigate through these pages. Click the **Help** push button to get information about the parameters that you can specify.
14. Click the **Next** push button to access the Summary page. Your settings are displayed in the Summary list.
15. Click the **Finish** push button to save the data object.

Selecting the data type

If the data that your data object refers to resides in a database table, Intelligent Miner automatically selects one of the available data types on the basis of the DB2 field type.

You can change the automatically selected data types. However, be aware that you can run into problems when you try to produce an output table. The Intelligent Miner uses the DB2 data types of the input table when it creates the output table, and Intelligent Miner data types when it inserts the records. For example, if you convert the data type from CHAR to *continuous*, the field content is represented by more characters. The field length increases. When the Intelligent Miner generates the output table, the data type of this field will be CHAR, as for the input table. The content of the continuous field is thus inserted into the CHAR field, but the CHAR field is not long enough to hold all the information. In such a case, DB2 issues an error message.

If your data resides in a flat file, you must select a data type for the fields that you define.

In the description below, the various data types are related to the appropriate scales of measurement. Statisticians are usually familiar with these terms, however, for a discussion of the scales of measurement, see “Categorizing your data” on page 242. The following data types are available:

Binary

For data that is measured by the nominal scale of measurement and that can assume only two possible values, 0 and 1.

Categorical

For data that is measured by the nominal scale of measurement.

Continuous

For data that is measured by the interval or ratio scale of measurement.

When you select **Continuous**, the whole value range of the field is split into buckets, to which the individual values are assigned.

By default, a range of approximately ±2 standard deviations from the mean is split into 10 buckets. The exact number of buckets varies with the function that you use. In addition, the visualizers display different numbers of buckets. Each visualizer might display fewer buckets than actually used.

Discrete-numeric

For data that is measured by the interval or ratio scale of measurement.

Each value in the field is treated as it is. There is no additional processing. This data type is helpful if you want to know how many different values you have in a particular field, and how often these values occur. However, it has the disadvantage that even the slightest differences between values are treated as distinct observations. This often leads to very large and detailed results, which in fact blur the information that is in your data.

Numeric

For data that is measured by the interval or ratio scale of measurement.

This data type is a combination of the continuous data type and the discrete-numeric data type. If the field contains up to 50 different values, it is treated as a *discrete-numeric* field. If the field contains more than 50 different values, it is treated as a *continuous* field.

Computed fields

Computed fields are additional fields that hold the information resulting from the following operations:

- Discretization
- Mathematical operations
- Value mapping

They are called computed fields because they are computed by the Intelligent Miner during a mining run.

To define a computed field, follow these instructions:

1. Access the Computed fields page in a Data wizard or settings notebook.
2. Enter a name for the computed field in the **Computed field name** field.
3. Select a computed field technique from the **Computed field technique** drop-down list. The following computed field techniques are available:

   Discretization

   Select this type if you want to use a discretization object. The computed field contains the values resulting from the discretization process, that is, the interval labels assigned to the original values.

   Function

   Select this type to perform mathematical operations on fields in your input data and copy the calculated values to the computed field for further processing.

   Value Mapping

   Select this type if you want to use a value mapping object. The computed field contains the values resulting from the mapping operation, that is, the new values after the conversion.

4. Click the **...** push button under **Formula settings or field names**. Depending on the type of computed field that you selected, this action opens one of the following windows:

   Discretization

   In this window, you select a discretization object to assign it to the computed field. Follow these steps:

   a. Select a discretization object in the **Available discretizations** pane.
   b. Select the field in the input data object that you want to discretize in the **Available fields** list.
   c. Click the > push button. The name of the selected field is displayed in the **Discretize field**.
   d. Click **OK** to return to the Computed fields page of the data object.

   Expression Builder

   In this window, you define a mathematical expression that uses one or more of your input fields as variables. During a mining run, the Intelligent Miner calculates values according to the expression, which are then copied to the computed field for further processing. To define a mathematical expression, see "Defining mathematical expressions" on page 92.

   Value Mapping

   In this window, you select a value mapping object to assign it to the computed field. Follow these steps:

   a. Select a value mapping object in the **Available value mappings** pane.
   b. Select the field in the input data object that contains the values that you want to substitute in the **Available fields** list.
   c. Click the > push button. The name of the selected field is displayed in the **Argument fields** list.
   d. Click **OK** to return to the Computed fields page of the data object.

   Optionally, you can assign a name mapping object to the computed field. Click the **...** button to the right of the **Name Mapping** field and proceed with step 2 on page 108.
5. Click the **Add** button. The computed field and its properties are displayed in the list called Matrix at the bottom of the page.

   You can also change the properties of a computed field. Select the appropriate entry in the list. The field name and its properties are displayed in the fields above the list, where you can make the necessary changes. Click the **Update** push button to apply your changes.

To delete a computed field, select the appropriate entry in the list and click the **Delete** push button.

Repeat steps 2 on page 90 through 5 on page 91 to define additional computed fields.

6. To close and save the data object, click **OK** in a Data settings notebook or **Next** and **Finish** in a Data wizard.

Defining mathematical expressions

This section shows how to define a mathematical expression using the Expression Builder. These expressions determine the values of the newly computed field, which are calculated during a mining run.

1. Select **Computed Functions** in the **Category** list. The available mathematical functions are displayed in the **Value** list. For a description of these functions, see *Built-in functions for computed fields* in *IBM DB2 Intelligent Miner for Data: Application Programming Interface and Utility Reference*.
2. Select a function from the **Value** list. Use the arrow keys to scroll through the list. The selected function is displayed in the pane at the bottom of the window, showing the number of arguments and their required data types in parentheses. The middle of the window shows as many **Arg** push buttons as the function requires arguments. The push buttons are inactive for the moment.
3. Select **Field Names** or **Constants** from the **Category** list.

   If you selected **Field Names**, select a data field from the **Value** list and proceed with step 4.

   If you selected **Constants**, select a constant from the **Value** list or define a constant, following these steps:

   a. Double-click the term **< new constant >** in the **Value** list. The **Value** list changes into a field.
   b. Enter a value in the **Value** field. The value can be any number or character string. The value is displayed in the **Value** list.
   c. Select the constant from the **Value** list.
4. Click **Arg1**, **Arg2**, or any other appropriate argument push button.

   With some functions, the number of arguments is variable. You can add another argument by clicking the **...** push button located on the right in the row of argument push buttons. Doing so creates another argument push button. Select a further constant or field and click the newly created argument push button.

   Make sure that your arguments are in the right order. For example, if you selected **divide**, if you defined the number 5 as a constant, and if you want to divide the expression in the first argument by 5, click **Arg2** after selecting the constant.

Restriction: Most functions require numeric data. You cannot select categorical data fields as arguments for these functions.

The Intelligent Miner data types continuous, discrete-numeric, and numeric are considered to be numeric. The data types binary and categorical are considered to be categorical.

The Intelligent Miner takes the selected constant or field as an argument of the mathematical expression.

5. Repeat steps 3 on page 92 through 4 on page 92 until you provided the selected function with the required number of arguments.
6. Click the **OK** push button to save the function and return to the Computed fields page.

   To make corrections to your mathematical expression, click the **Clear** push button. The function that you defined is deleted from the pane at the bottom of the window. Start anew with step 1 on page 92.

Browsing through your data

The client software of the Intelligent Miner is equipped with a browsing function. To view a small sample of your data in a Web browser, follow these steps:

1. Select a data object in the Intelligent Miner main window.
2. Click **Select → Browse** on the menu bar.

Starting with the first record in your data, the Web browser displays up to 1000 records in sequential order. The size of the generated HTML file is restricted by the IDM_MAX_BROWSE_SIZE environment variable. This variable defines the maximum size of the HTML file in bytes. If this variable is not set, the maximum HTML file size is one MB. See “Client variables” on page 351 for information on how to set the IDM_MAX_BROWSE_SIZE environment variable.

To select a browser of your choice, follow these instructions:

1. In the Intelligent Miner main window, click **Options → Preferences**.
2. Click the Visualizers tab.
3. From the **Result type** drop-down list, select **Data/Browse** to list all visualizers registered for that type of result in the **Result format and visualizer** drop-down list.
4. Select a visualizer from the **Result format and visualizer** drop-down list.
5. Click the **Update** push button.
6. Click the **Apply** push button to save your changes or click the**OK** push button to save your changes and close the Preferences notebook.

You can also change the visualizer by editing the client tool registration file, which is called idmcsctr.dat. In that file, you find a section for each visualizer available. You can identify these sections by searching for the following line:

```
RESULT_TYPE=IDM_DATA_SAMPLE_RESULT
```

The section pertaining to the currently selected browser contains the line `DEFAULT=yes`. To select a different browser, move this line to another section. See *IBM DB2 Intelligent Miner for Data: Application Programming Interface and Utility Reference* for more information.

On Windows client workstations, the default browser is the standard HTML browser used by the operating system. On AIX and OS/2 client workstations, the default browser is Netscape Navigator or Netscape Communicator. If a default visualizer is neither selected in the Preferences notebook, nor in the client tool registration file, the Intelligent Miner randomly selects an appropriate visualizer from the client tool registration file.

Tip: If you do not want to view the first 1000 records, but a random sample of your data or a sample derived from a subset of the available fields, you can use the sampling feature of the Bivariate Statistics function. Activate the sampling feature and disable all other features of the Bivariate Statistics function. After the function completed processing successfully, your default browser displays the sample.

Exploring data with other applications

On client workstations, you can use the exploration function to convert data for use with other applications. The exploration function converts records of the data that a data object refers to. You can produce output in comma-separated variables (CSV) format for use with applications such as:

- Lotus 1-2-3
- Microsoft Excel
- SPLUS
- SPSS

Note: The software needed to process this data is not part of the Intelligent Miner product package, and must be obtained separately.

To convert records “in” a data object, follow these steps:

1. Select a data object in the Intelligent Miner main window.
2. Click **Select → Explore**.

A reproducible sample of 5000 records is converted and written to a temporary file. After you applied the exploration function, the software associated with the output format starts and displays the converted records.

When you launch the function, the Intelligent Miner creates a Bivariate Statistics settings object with the name of the data object. When the function finishes processing, it deletes this settings object and creates a result object with the same name.

Attention:

- If there already is a result object with this name, it is overwritten.
- The result objects are stored in the directory identified by the IDM_RES_DIR environment variable. They are usually very large because they contain copies of the 5000 records that you explored. To avoid disk space shortages, check the result directory periodically, and delete the objects that you do not need anymore.

To select a visualizer of your choice, follow these instructions:

1. In the Intelligent Miner main window, click **Options → Preferences**.
2. Click the Visualizers tab.
3. From the **Result type** drop-down list, select **Exploration sample** to list all visualizers registered for that type of result in the **Result format and visualizer** drop-down list.
4. Select a visualizer from the **Result format and visualizer** drop-down list.
5. Click the **Update** push button.
6. Click the **Apply** push button to save your changes or click the**OK** push button to save your changes and close the Preferences notebook.

You can also change the visualizer by editing the client tool registration file, which is called idmcsctr.dat. In that file, you find a section for each visualizer available. You can identify these sections by searching for the following line:

```
RESULT_TYPE=IDM_DESC_STAT_QUANT_SAMPLE_RESULT
```

The section pertaining to the currently selected browser contains the line `DEFAULT=yes`. To select a different visualizer, move this line to another section. See *IBM DB2 Intelligent Miner for Data: Application Programming Interface and Utility Reference* for more information.

By default, the output is displayed in the Bivariate Statistics visualizer. The visualizer starts automatically after running the exploration function if it is properly installed on your client workstation.

Tip: If you do not want to explore a sample of 5000 records selected by the Intelligent Miner, but some other sample or a sample derived from a subset of the available fields, you can use the sampling feature of the

Bivariate Statistics function. Activate the sampling feature and enter the following expression in the **Power options** field on the Input data page:

```
-explore
```

To enable the **Power options** field, you must check the **Show the advanced pages and controls** box on the settings page.

Mining and statistics settings objects

Mining and statistics settings objects are similar in that they represent analytical functions that you can apply to data. In both cases, you must indicate which data object you want to use.

Some mining and statistics settings objects produce a result object when run. You can view and analyze most result objects with visualization tools. You can also indicate in the settings for these functions that you want to create output data in addition to a result object.

Table 23 shows the mining and statistics functions of the Intelligent Miner.

Table 23. The mining and statistics functions

Mining function	Statistics function
Associations	Bivariate Statistics
Demographic Clustering	Linear Regression
Neural Clustering	Principal Component Analysis
Sequential Patterns	Univariate Curve Fitting
Similar Sequences	Factor Analysis
Tree Classification	
Neural Classification	
RBF-Prediction	
Neural Prediction	

See the chapters on the mining functions, Chapter 18, “The statistical functions” on page 241, and the wizards and online help for more information.

Preprocessing settings objects

Use preprocessing functions to make data suitable for mining or analysis. Preprocessing settings objects apply only to database tables and views because they take advantage of the processing capability of the database engine.

Table 24 shows the preprocessing functions of the Intelligent Miner.

Table 24. The preprocessing functions

Function	
Aggregate Values	Filter Fields
Calculate Values	Filter Records
Clean Up Input Data or Output Data	Filter Records Using a Value Set
Convert to Lowercase or Uppercase	Get Random Sample
Copy Records to File	Group Records
Discard Records with Missing Values	Join Data Sources
Discretization into Quantiles	Map Values
Discretization Using Ranges	Pivot Fields to Records
Encode Missing Values	Run SQL
Encode Nonvalid Values	

Preprocessing settings objects always read input from a database and create output data in a database. The only exception is the Copy Records to File function, which copies data to a file. When you create a preprocessing settings object or update an existing one, you can use a data object to identify input data or output data. Identifying the input or output data in this way simply copies the name of a database table or view to the preprocessing settings object. Subsequent changes to the data object have no effect on the preprocessing settings object.

See Chapter 19, “The preprocessing functions” on page 249 and the online help for more information about functions that you can use to transform data.

Result objects

Result objects represent information that is discovered when you run a mining or statistics settings object. The result objects can be viewed using an Intelligent Miner visualizer, or they can be accessed through API programs for specialized analysis. You cannot create or modify result objects directly as you can with data and other settings objects.

Generally, you can use results in several ways:

- To visualize or access the results of a mining or statistics function
- As input when running a mining function in test mode to validate the predictive model represented by the result
- As input when running a mining function in application mode to apply the model to new data
- As input to provide statistical information for the clustering functions.

Like input and output data objects, result objects can be used as input to the mining function and can be generated as output. However, while data objects represent data that resides in a flat file or database, result objects represent information that is stored in a separate file.

Table 25 shows how the mining and statistics functions use results.

Table 25. Functions and their use of results

Function	Mode/page	Uses input result?	Produces result?
Associations	N/A	Never	Always
Tree Classification Neural Classification	Training	Never	Always
	Test	Always	Always
	Application	Always	Never
Demographic Clustering Neural Clustering	Clustering	Optional	Always
	Application	Always	Never
Sequential Patterns	N/A	Never	Always
Neural Prediction RBF-Prediction	Training	Never	Always
	Test	Always	Always
	Application	Always	Never
Similar Sequences	N/A	Never	Always
Bivariate Statistics	Statistics	Never	Always
	Sampling	Never	Always
	Output only	Never	Never
	In combination	Never	Always
Factor Analysis Linear Regression	Training	Never	Always
	Application	Always	Never
Principal Component Analysis Univariate Curve Fitting	N/A	Never	Always

How result objects are used depends on the mode in which functions are run:

Training mode
In training mode, a mining function builds a model based on the selected input data. This model is stored in the result object.

Clustering mode
In clustering mode, the clustering functions build a model based on the selected input data. This is similar to training mode for the predictive algorithms. Clustering mode offers the choice of using background statistics from the input data or from an input result.

Test mode
In test mode, a mining function uses new data with known results. This allows you to verify that the model created in training mode produces useful results. Result objects are used as input and are created as output.

Application mode
In application mode, a mining or statistics function applies a model created in training mode to new input data.

Sequence settings objects

To run a series of data mining functions in a sequence, you can create a sequence settings object. When you run a sequence, the Intelligent Miner processes each settings object in the sequence. You determine the order in which these settings objects are processed by dragging and dropping object icons in the Sequence wizard or settings notebook. Sequence settings objects enable you to construct sequences that represent complex tasks. Sequences can be part of larger sequences.

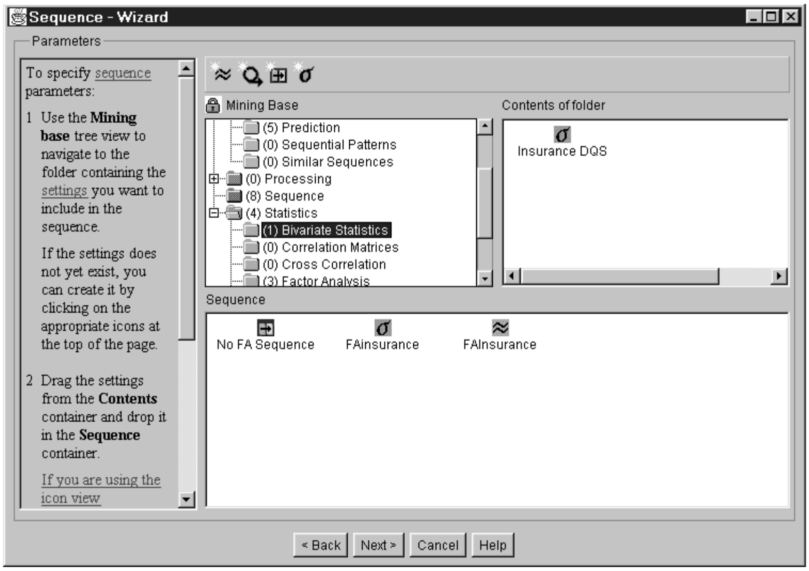
Sequences can contain the following settings objects:

- Mining
- Preprocessing
- Statistics
- Sequences

Settings for sequences include:

- Settings objects and the order in which they are run
- Whether the sequence continues if any settings object fails
- Wether some settings objects in the sequence are inactive.

Figure 8 on page 100 shows the Parameters page of the Sequence Settings wizard. On this page, you can find the settings objects that you want to run in a sequence. You can drag and drop object icons out of the Contents area into the Sequence area to create a sequence. Click one of the create settings object buttons on the toolbar to create an object.


Sequence - Wizard

Parameters

To specify sequence parameters:

1 Use the **Mining base** tree view to navigate to the folder containing the settings you want to include in the sequence.

If the settings does not yet exist, you can create it by clicking on the appropriate icons at the top of the page.

2 Drag the settings from the **Contents** container and drop it in the **Sequence** container.

If you are using the icon view

Figure 8. Sequence settings notebook

Each settings object in a sequence, including all nested settings, must have a unique name. While running a sequence, you cannot run a settings object independently or influence its behavior if this settings object is part of the sequence. Sequences have no parallel processing mode, but objects started by the sequence are run in parallel if you defined the settings object to run in parallel processing mode.

See the online help for more information about sequence settings objects.

Discretization objects

Discretization objects split the value range of a continuous, discrete-numeric, or numeric field into intervals and then map each value to an interval. By using a discretization object, you can thus convert numeric data into categorical data. Discretizing fields often gives you much clearer results even though it is accompanied by loss of detail.

Example: Suppose that you have a field in your input data whose values represent the yearly income of your customers. You want to gain some insight into the consumer behavior of certain income groups. Therefore, the difference between an income of $80000 per year and an income of $81000 per year is of no real interest for you because both incomes can be said to belong to the same income group. For that reason, you split the whole range of the income field into 4 intervals:

Range	Interval label
< $30000	1
$30000 ≤ $50000	2
$50000 ≤ $90000	3
$90000 or more	4

This is how the actual values in your income field would be mapped:

Income	Interval label
38400	2
46500	2
26900	1
78000	3

To use such intervals with the Intelligent Miner, you first define them in a discretization object and later assign that object to a computed field.

Creating discretization objects

You can create a discretization object in the following ways:

- You can define the intervals manually.
- You can use existing data in a database table or a flat file if that data contains the value ranges and the interval labels that you want to use.

Defining intervals manually: To define the intervals in a discretization object manually, follow these instructions:

1. Click **Create → Discretization** on the menu bar in the Intelligent Miner main window.
2. If the Welcome page is displayed, click the **Next** push button to access the Settings page.
3. Select **Manual** in the **Available data formats** pane.
4. Enter a name for your discretization object in the **Settings name** field. You can also add a description in the **Comment** field.
5. Click the **Next** push button to access the Interval parameters page.
6. Enter the upper interval boundary of the interval in the **Interval boundary** field.
7. Select the logical operator < or <= from the **Flag** drop-down list.
8. Enter an interval label in the **Value** field.

   The interval label can be a numeric value or a character string.
9. Click the **Add** button. The interval is displayed in the list named Matrix at the bottom of the page.

You can also change an entry in the list. Select the interval. It is displayed in the **Interval boundary**, **Flag**, and **Value** fields, where you can make the necessary changes. Click the **Update** push button to apply your changes.

To delete an interval, select it in the list and click the **Delete** push button.

Repeat steps 6 on page 101 through 9 on page 101 until your list of intervals is complete.

The first interval includes the field values from the lowest value up to the interval boundary, with or without the interval boundary itself.

If you define another interval, and therefore enter another value in the **Interval boundary** field, the Intelligent Miner automatically recognizes the adjacent boundaries of any preceding or following intervals. Thus both interval boundaries of the new interval are defined correctly. For example, if you enter the value 15 as the interval boundary of a first interval and the value 30 as the interval boundary of a second interval, and for both boundaries select the operator <, the Intelligent Miner automatically detects the correct interval limits for the second interval, that is, 15 and 30.

10. Click the **Next** push button to access the Summary page.
11. Click the **Finish** push button to save the discretization object.
12. To use the discretization object, follow the instructions in "Computed fields" on page 90.

Using existing data: To create a discretization object using existing data, follow these instructions:

1. Click **Create → Discretization** on the menu bar in the Intelligent Miner main window.
2. If the Welcome page is displayed, click the **Next** push button to access the Settings page.
3. Select **Input data** in the **Available data formats** list.
4. Enter a name for your discretization object in the **Settings name** field. You can also add a description in the **Comment** field.
5. Click the **Next** push button to access the Input data page.

If you already created a data object defining your interval boundaries and labels, skip steps 6 through 8.

6. Click the **Create data** push button to open the Data wizard.
7. Create a data object as described in "Data objects" on page 85. The data object must refer to at least three fields. One field must contain the interval boundaries; a second field must contain the logical operators < or <=; a third field must contain the interval labels.
8. Click **Finish** to save the new data object and return to the Input data page of the Discretization wizard.

9. Select the new data object or another data object in the **Available input data** pane.
10. Click the **Next** push button. If your discretization data resides in a flat file, you access the Field parameters page. You do not have to define any field ranges here because you have defined them already in the data object. Click the **Next** push button to access the Interval parameters page.
11. Select the field containing the interval boundaries in the **Available fields** list.
12. Click the **>** push button next to the **Interval boundary field**. The name of the selected field is displayed in that field.
13. Select the field containing the logical operators in the **Available fields** list.
14. Click the **>** push button next to the **Flag field**. The name of the selected field is displayed in that field.
15. Select the field containing the interval labels in the **Available fields** list.
16. Click the **>** push button next to the **Value field**. The name of the selected field is displayed in that field.

    To remove a field name from the **Interval boundary** field, the **Flag field**, or the **Value field**, click the **<** push button next to the field in question.
17. Click the **Next** push button to access the Summary page.
18. Click the **Finish** push button to save the discretization object.
19. To use the discretization object, follow the instructions in "Computed fields" on page 90.

Name mapping and value mapping objects

Name mapping objects or value mapping objects allow you to convert data. The difference between the two types of objects lies in the point of time when the conversion is done:

- Value mapping objects convert values *before* the mining run so that the Intelligent Miner functions use the converted values as input.
- Name mapping objects convert values *after* the mining run completes processing so that the result viewers display the converted values.

The following sections explain when to use which of the two object types.

Name mapping objects

You can use name mappings to give more descriptive names to field values. This is particularly useful if specific items are represented by a numerical code in your data.

Example: A supermarket uses code numbers for its products. Number 43 signifies cheese, and number 67 signifies wine. You want to include the products of the supermarket in a mining run that discovers associations. In the results of the mining run, an assumed association rule for cheese and wine would be displayed in textual format as:

```
When a customer buys 43, then the customer also buys 67 in 32% of cases
```

To avoid that your results are displayed in this way, you can use a name mapping object. In a name mapping object, you can map the value 43 to cheese and the value 67 to wine so that the following, more descriptive association rule is displayed:

```
When a customer buys cheese, then the customer also buys wine in 32% of cases
```

Value mapping objects

Value mappings are particularly useful to correct inconsistent coding. Inconsistent coding can lead to confusing or even wrong results.

Example: Suppose that the product name of a detergent in the database of a supermarket is inconsistently coded as Superclean and Supcln. An Associations mining run detects that this detergent and a particular brand of tissues are frequently bought in common. Instead of a single association rule that shows the relationship between the detergent and the tissues, you would get two rules.

Support	Confidence	Type	Lift		Rule	
2.05	64.0	+	6.2	[Superclean]	==>	[Pearl Tissues]
...	...	...	...	...	...	...
1.74	47.0	+	4.8	[Supcln]	==>	[Pearl Tissues]

Apart from the fact that it can be hard to track such split rules in a long list of association rules, you would not be able to see the strength of the relationship between the two products. Even worse, it might be that one of the rules does not appear at all because it does not reach the minimum confidence or support threshold. This would indeed give you wrong results.

A name mapping object cannot solve this problem. Although its application would correct the inconsistent coding, you would still see two rules because the mapping is done after the calculation. However, by using a value mapping object, you can map all occurrences of Supcln to Superclean before the processing starts so that only one value is used for the detergent during the mining run. This results in only one association rule for the relationship between this detergent and the tissues:

Support	Confidence	Type	Lift		Rule	
3.79	54.9	+	5.3	[Superclean]	==>	[Pearl Tissues]

Creating name mapping or value mapping objects

You can create a name mapping or a value mapping object in the following ways:

- You can define the mappings manually.
- You can use existing data in a database table or a flat file if that data contains the values that you want to map and the names or new values.

Defining name mappings or value mappings manually: To define the name mappings or value mappings in an object manually, follow these instructions:

1. Click **Create → Name Mapping** or **Create → Value Mapping** on the menu bar in the Intelligent Miner main window.
2. If the Welcome page is displayed, click the **Next** push button to access the Settings page.
3. Select **Manual** in the **Available data formats** pane.
4. Enter a name for your name mapping or value mapping object in the **Settings name** field. You can also add a description in the **Comment** field.
5. Click the **Next** push button to access the Parameters page.
6. For name mapping objects, enter a value that you want to substitute in the **Item ID** field. In the example in "Name mapping and value mapping objects" on page 103, this would be one of the code numbers, that is, 43 or 67.

   For value mapping objects, enter a value that you want to substitute in the **Arguments** field. In the example in "Value mapping objects" on page 104, this would be the abbreviated product name Supcln.

   Note: You can also specify more than one value. Then all the values that you specify are substituted. If you specify multiple values, separate them by list delimiters. The default for this delimiter is the semicolon.

   Example: To define a name mapping object or value mapping object that substitutes the values 12, 18, and 81 for another value enter the following string in the **Item ID** field:

   ```
   12;18;81
   ```
7. If you define name mappings, enter the name that you want to map the value to in the **Item description** field. In the example in "Name mapping and value mapping objects" on page 103, this would be one of the product names, that is, cheese or wine.

   If you define value mappings, enter the value that you want to map the initial value to in the **Value** field. In the example in "Value mapping objects" on page 104, this would be the product name Superclean.

8. Click the **Add** button. The name mapping or value mapping is displayed in the list named Matrix at the bottom of the page. Repeat steps 6 on page 105 through 8 until your list of mappings is complete.

   You can also change an entry in the list. Select the mapping. It is displayed in the **Item ID** and **Item description** fields or the **Arguments** and **Value** fields, where you can make the necessary changes. Click the **Update** push button to apply your changes.

   To delete a mapping, select it in the list and click the **Delete** push button.
9. Click the **Next** push button to access the Summary page.
10. Click the **Finish** push button to save the name mapping or the value mapping object.
11. To use a name mapping object, follow the instructions in "Assigning name mapping objects to fields in data objects" on page 107. To use a value mapping object, follow the instructions in "Computed fields" on page 90.

Using existing data: To create a name mapping object or value mapping object using existing data, follow these instructions:

1. Click **Create → Name Mapping** or **Create → Value Mapping** on the menu bar in the Intelligent Miner main window.
2. If the Welcome page is displayed, click the **Next** push button to access the Settings page.
3. Select **Input data** in the **Available data formats** list.
4. Enter a name for your name mapping object in the **Settings name** field. You can also add a description in the **Comment** field.
5. Click the **Next** push button to access the Input data page.

If you already created a data object defining the values that you want to substitute and the values that you want to use instead, skip steps 6 through 8.

6. Click the **Create data** push button to open the Data wizard.
7. Create a data object as described in "Data objects" on page 85. The data object must refer to at least two fields. One field must contain the values that you want to map. The other field must contain the names or values that you want to map the original values to.
8. Click **Finish** to save the new data object and return to the Input data page of the Name Mapping wizard.
9. Select the new data object or another data object in the **Available input data** pane.
10. Click the **Next** push button. If your name mapping data resides in a flat file, you access the Field parameters page. You do not have to define any field ranges here because you have defined them already in the data object. Click the **Next** push button to access the Parameters page.

11. Select the field containing the values that you want to map in the **Available fields** list. If you create a value mapping object, you can select more than one field.

    In the example in "Name mapping and value mapping objects" on page 103, this would be the field containing the code numbers 43 and 67.

    In the example in "Value mapping objects" on page 104, this would be the field containing the abbreviated product name Supcln.
12. Click the **>** push button next to the **Item ID** field or the **Argument fields** list. The name of the selected field appears in the **Item ID** field or in the **Argument fields** list. To remove a field name from the **Item ID** field, click the **<** push button. To remove a field from the **Argument fields** list, select the field and click the **<** push button.
13. Select the field containing the names that you want to map the values to in the **Available fields** list.

    In the example in "Name mapping and value mapping objects" on page 103, this would be the field containing the product names cheese and wine.

    In the example in "Value mapping objects" on page 104, this would be the field containing the product name Superclean.
14. Click the **>** push button next to the **Item description** or **Value** field. The name of the selected field is displayed in that field. To remove a field name from the **Item description** field or **Value** field, click the **<** push button.
15. Click the **Next** push button to access the Summary page.
16. Click the **Finish** push button to save the name mapping or the value mapping object.
17. To use a name mapping object, follow the instructions in "Assigning name mapping objects to fields in data objects". To use a value mapping object, follow the instructions in "Computed fields" on page 90.

Assigning name mapping objects to fields in data objects

To use the name mappings defined in a name mapping object, you must assign the name mapping object to a field in a data object.

Exception: If you use a taxonomy with the Associations or Sequential Patterns mining functions, you must assign the name mapping objects to categories of the taxonomy. See "Assigning name mapping objects to categories of a taxonomy" on page 108 for more information.

To assign a name mapping object to a field in a data object, follow these instructions:

1. Click the **...** button to the right of the **Name Mapping** field on the Field parameters page of the data object to open the Name Mapping window.

2. Select a name mapping object in the **Available name mappings** pane.
3. Click **OK** to return to the Field parameters page of the data object.

Tip: You can also access the data object containing the fields that you want to assign name mapping objects to from a wizard or a settings notebook of an Intelligent Miner function. Double-click a data object icon in the **Available input data** list on the Input data page. This action opens the settings notebook of the data object. You then access the field Parameters page of the notebook and proceed as described in steps 1 on page 107 through 3.

This method allows you to assign a name mapping object to fields in the data object that you use as input when specifying settings for an Intelligent Miner function. The name mapping object becomes active for any settings object that uses this particular data object.

Assigning name mapping objects to categories of a taxonomy

To employ name mappings when using a taxonomy with the Associations or the Sequential Patterns mining functions, you must assign name mapping objects to the categories of the taxonomy rather than to the fields in your input data. Name mapping specifications in a taxonomy override name mapping specifications in a data object. So, if you use a taxonomy and do not assign name mapping objects to the categories, name mappings are not employed, regardless of any name mapping objects that are assigned to fields in the input data.

To assign a name mapping object to a category, follow these instructions:

1. Click the Taxonomy folder icon in the Mining base container of the Intelligent Miner main window. The available taxonomies are displayed in the Contents container.
2. Double-click a taxonomy icon. The settings notebook of the taxonomy opens.
3. Click the Parameters tab.
4. In the Categories pane on the Parameters page, double-click the icon of the category that you want to assign a name mapping object to. The Category window opens.
5. Click the **...** push button to the right of the **Name Mapping** field. The Name Mapping window opens.
6. Select a name mapping object in the **Available name mappings** pane.

   To create a name mapping, click the **Create name mappings** push button. This action takes you to the Name Mapping wizard. See “Defining name mappings or value mappings manually” on page 105 and “Using existing data” on page 106 for more information.

7. Click **OK** to confirm your selection and return to the Category window. The name of the selected name mapping object is displayed in the Name Mapping field.
8. Click **OK** to confirm your selection and return to the Taxonomy settings notebook.
9. Click **Apply** to save the settings or **OK** to save the settings and close the notebook.

Taxonomies

You can use taxonomies with Associations and Sequential Patterns mining settings objects. Taxonomies define a hierarchy between items and categories. Categories usually represent grouped field values, but can also consist of only one value. When you use a taxonomy with a mining function, the Intelligent Miner treats the categories like values in your input fields. The Intelligent Miner searches for relationships between categories, between field values, and between categories and field values.

Note: When a certain field value becomes a member of a category, the Associations and Sequential Patterns mining functions ignore the obvious relationship between the field value and this particular category during a mining run. Thus your results are not diluted by trivial rules.

Example: Suppose that you run the Associations mining function on the transaction database of a supermarket without a taxonomy. Your input data consists of the fields Food and Non-food. The field values are the different food and non-food products sold in the supermarket. Your goal is to find relationships between purchases of food and purchases of non-food products. The results show a number of association rules, but the relationships between the items are weak because the rules have low confidence values. However, by skimming through the list of association rules, you see a number of rules like these:

Support	**Confidence**	**Type**	**Lift**		**Rule**	
1.05	23.0	+	2.6	[Toffees]	==>	[Lipstick]
...	...	...	...	...	...	...
1.02	21.2	+	2.1	[Lollipops]	==>	[Make-up]
0.74	9.0	+	1.4	[Chocolate fudge]	==>	[Eyeliner]
...	...	...	...	...	...	...
0.69	8.2	+	0.9	[Crispy bars]	==>	[Nail varnish]

You know that there is no strong association between the items in these rules, but you assume that there might be a strong relationship between sweets and cosmetics. Hence it makes sense to create a taxonomy. Creating a taxonomy involves the following steps:

1. You create a category named Articles, which represents the product level.
2. You create a category named Article Types, which represents the level above the product level.
3. You invent group names for products belonging together.

   In the example, cosmetics and sweets would be suitable groups.
4. You create a relation between the categories.
5. You assign the products to suitable group names.

After that, you run the Associations mining function again, including the taxonomy that you just defined. The group names in the taxonomy serve as additional input values for the Intelligent Miner.

This time, the results show more association rules, one of which states a relationship between sweets and cosmetics. This rule is stronger because it has a higher support value and a higher confidence value. It might look like this:

Support	Confidence	Type	Lift		Rule	
3.5	61.0	+	5.2	[Sweets]	==>	[Cosmetics]

Creating taxonomy objects

1. Click **Create → Taxonomy** on the menu bar in the Intelligent Miner main window to start the Taxonomy wizard.
2. If the Welcome page is displayed, click the **Next** push button to access the Settings page.
3. Enter a name for your taxonomy object in the **Settings name** field. You can also add a description in the **Comment** field.
4. Click the **Next** push button to access the Parameters page.
5. If you want to use existing categories, skip to step 6 on page 111. Else, create the necessary categories.
   a. Click the **Create Category** push button. The Category window opens.
   b. Enter a name for your category in the **Category Name** field.

      In the example in "Taxonomies" on page 109, the category names would be Articles and Article Types.

      You can also assign a name mapping object to the category. Enter the name of a name mapping object in the **Name Mapping** field or click the **...** push button and proceed with step 6 on page 108 in "Assigning name mapping objects to categories of a taxonomy" on page 108.

   c. Click the **OK** push button to return to the Taxonomy wizard.
6. Drag at least two category icons from the Categories pane to the Taxonomy pane.

   Following the example in "Taxonomies" on page 109, you would drag the two categories Articles and Article Types.
7. Click a child category in the Taxonomy pane with the right mouse button. A pop-up menu opens.
8. Select **Create relation** from the pop-up menu. You see a line originating from the child-category icon that you can move with your mouse.
9. Direct this line to a parent-category icon and click the left mouse button when the line touches it. This action starts the Taxonomy Relation wizard.

   Note: The line must always originate from a child category and point to a parent category. Following the example in "Taxonomies" on page 109, you would right-click the Articles category and drag the line to the Article Types category. If you inadvertently started from the wrong icon, delete the taxonomy relation and re-create it.

   When you want to modify existing taxonomy relations, click the arrow connecting the categories with your right mouse button. The arrow turns from black to blue and a pop-up menu opens. Select **Edit** from the pop-up menu.
10. Assign items to group names as described in "Creating taxonomy relations".

Tip: You can also create a taxonomy object from a wizard or a settings notebook of the Associations or the Sequential Patterns mining function. Click the **Create taxonomy** push button on the Taxonomy page. This action starts the Taxonomy wizard.

Moreover, you can modify an existing taxonomy object from a wizard or a settings notebook of the Associations or the Sequential Patterns mining function. Double-click the icon of the taxonomy object in the **Available taxonomies** list. This action opens the settings notebook of the taxonomy object.

This method allows you to create or modify a taxonomy object and immediately assign that object to an Associations or Sequential patterns settings object.

Creating taxonomy relations

You can create the taxonomy relations in a taxonomy object in the following ways:

- You can define taxonomy relations manually.
- You can define taxonomy relations between fields in existing data. This data can reside in a database table or a flat file.

Defining taxonomy relations manually: To define a taxonomy relation manually, follow these instructions:

1. Select **Manual** in the **Available data formats** list on the Settings page of the Taxonomy Relation wizard.

   Optionally, you can add a comment in the **Comment** field.
2. Click the **Next** push button to access the Parameters page.
3. Enter a group name in the **Parent category field**.

   The group name should describe the items that it represents.

   Following the example in "Taxonomies" on page 109, you would enter `Cosmetics` or `Sweets`.
4. Enter a field value that you want to assign to the group name in the **Child category field**. When defining the base level in the hierarchy of a taxonomy, you must enter a field value from your input data here.
5. Click the **Add** button. The pair of values is displayed in the list named Matrix at the bottom of the page. Repeat steps 3 through 5 until your list of taxonomy relations is complete.

   Table 26 shows the complete list of assignments for the example in "Taxonomies" on page 109.

Table 26. Assigning field values to group names

Parent category field	Child category field
Cosmetics	Lipstick
Cosmetics	Eyeliner
Cosmetics	Make-up
Cosmetics	Nail varnish
Sweets	Toffees
Sweets	Lollipops
Sweets	Chocolate fudge
Sweets	Crispy bars

   You can also change an entry in the list. Select the pair of values. It is displayed in the **Parent category field** and in the **Child category field**, where you can make the necessary changes. Click the **Update** push button to apply your changes.

   To delete an entry, select it in the list and click the **Delete** push button.

6. Click the **Finish** push button to save the taxonomy relation and return to the Parameters page of the Taxonomy wizard.

7. Click the **Next** push button to access the Summary page.
8. Click the **Finish** push button to save the taxonomy object.
9. To use the taxonomy object, select the object icon in the **Available taxonomies** list on the Taxonomy page of an Associations or Sequential Patterns wizard or settings notebook.

   Note: You can only access the Taxonomy page if you select **Show the advanced pages and controls** on the Settings page.

Using existing data: To create a taxonomy relation using existing data, follow these instructions:

1. Select **Input data** in the **Available data formats** list on the Settings page of the Taxonomy Relation wizard.

   Optionally, you can add a comment in the **Comment** field.
2. Click the **Next** push button to access the Input data page.

If you already defined a data object referencing taxonomy data, skip steps 3 through 5.

3. Click the **Create data** push button to open the Data wizard.
4. Create a data object as described in "Data objects" on page 85. The data object must refer to at least two fields. One field must contain the group names that you want to relate the items to; the other field must contain the item names.
5. Click **Finish** to save the new data object and return to the Input data page of the Taxonomy Relation wizard.
6. Select the new data object in the **Available input data** pane.
7. Click the **Next** push button. If your taxonomy relation data resides in a flat file, you access the Field parameters page. You do not have to define any field ranges here because you have defined them already in the data object. Click the **Next** push button to access the Parameters page.
8. Select the field containing the group names in the **Available fields** list.
9. Click the **>** push button next to the **Parent category field**. The name of the selected field is displayed in that field.
10. Select the field containing the item names in the **Available fields** list.
11. Click the **>** push button next to the **Child category field**. The name of the selected field is displayed in that field.

    To remove a field name from the **Parent category field** or the **Child category field**, click the **<** push button next to the field in question.
12. Click the **Finish** push button to return to the Parameters page of the Taxonomy wizard.
13. Click the **Next** push button to access the Summary page.
14. Click the **Finish** push button to save the taxonomy object.

15. To use the taxonomy object, select the object icon in the **Available taxonomies** list on the Taxonomy page of an Associations or Sequential Patterns wizard or settings notebook.

    Note: You can only access the Taxonomy page if you select **Show the advanced pages and controls** on the Settings page.

Selecting an input data object

Whenever you want to run a function of the Intelligent Miner that uses input data, you must specify a data object as input. To specify a data object, select an existing data object or create and then select a data object.

You select a data object from the **Available input data** pane on the Input data page of a wizard or settings notebook of a mining or statistics function.

Optimizing the mining run

By clicking the appropriate radio button on the Input data page of a mining wizard or settings notebook, you can optimize a mining run in one of the following ways:

Optimize mining run for time

Selecting this option often improves the performance of a mining run significantly. In addition, you can ensure that the sequence of the data records remains the same in subsequent data passes when you use DB2 tables as input.

The Intelligent Miner stores any data used during a mining run in a temporary directory. The size of the data stored in this directory during a single run is typically about 75% of the size of your input data. You must therefore ensure that you have sufficient disk space on your server.

Recommendation: By default, the Intelligent Miner server uses a certain directory as the temporary directory. This directory is also used by the operating system and by other programs. To avoid that you run out of storage space during a mining run, change the temporary directory for the Intelligent Miner by following these steps:

1. Choose or create a directory on your Intelligent Miner server to serve as the new temporary directory. Make sure that you allocated enough disk space to this directory.
2. Click **Options → Preferences** on the menu bar in the Intelligent Miner main window. The Preferences notebook opens.
3. On the Diagnostics page, enter the path to the new temporary directory in the **Work directory** field.

Optimize mining run for disk space

If you optimize the mining run for disk space, the Intelligent Miner controls the mining run in a way to prevent you from running out of disk space.

When you select this option, the Intelligent Miner does not use a temporary directory.

Exception: This option is not available for the Tree Classification mining function.

Filtering records

Sometimes you might want to select only records with certain values in some fields.

Filtering the input data helps you to focus on the values of interest. You can filter the input data by specifying a filter records condition that all records must meet.

Example: If you are interested only in purchase transactions by customers who are older than 50 years, who spent more than 100 dollars per transaction, and whose yearly income is higher than 60000 dollars, you might use the following filter records condition:

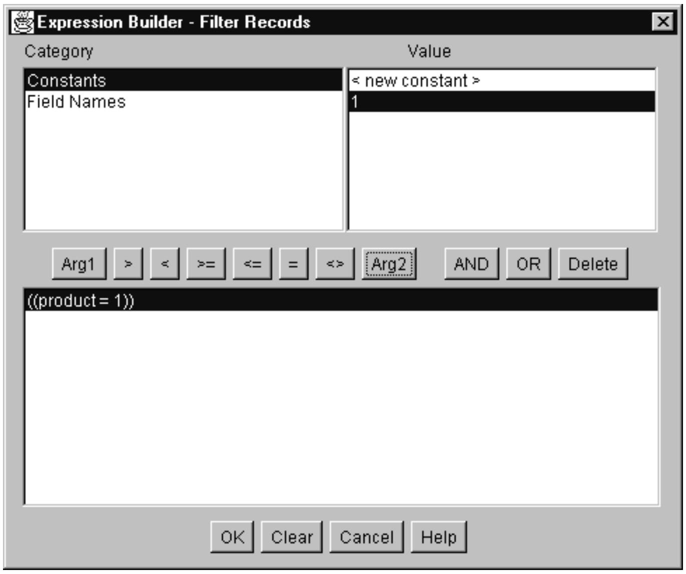
```
"Customer Age">50 AND Income>60000 AND "Amount per Transaction">100
```

This example assumes that the numeric fields *Customer Age*, *Income*, and *Amount per Transaction* exist in your data.

Starting the Expression Builder

Use the Expression Builder to define filter records conditions. To start the Expression Builder, follow these steps:

1. Check the **Show the advanced pages and controls** box on the Settings page of a mining or statistics wizard or settings notebook if this box is not checked.
2. Select the input data.
3. Click the **...** push button next to the **Filter records condition** field. The Expression Builder window opens. Figure 9 on page 116 shows the Expression Builder.



((product = 1))

Figure 9. The Expression Builder

Defining filter records conditions

This section shows you how to define a filter records condition using the Expression Builder. Follow these steps:

1. Click the **AND** push button in the Expression Builder window. The formula `((Arg1 = Arg2))` appears in the lower pane.
2. Select **Field Names** in the **Category** list. The available fields are displayed in the **Value** list.
3. Select the field for which you want to define a filter records condition in the **Value** list.
4. Click the **Arg1** push button. The selected field name replaces `Arg1` in the formula in the lower pane.
5. Select a logical operator by clicking the appropriate push button. The following operators are available:

>	Greater than
<	Less than
>=	Greater than or equal to (≥)
<=	Less than or equal to (≤)
=	Equal to (default)
<>	Not equal to (≠)

The selected operator replaces the operator in the formula.

6. Select a field or a constant as the second argument.

   If you select a field, the Expression Builder compares the values of the first field and the second field in each record and takes only those records as input that meet the filter record condition. You select a field as described in steps 2 on page 116 and 3 on page 116.

   To select a constant, click **Constant** in the **Category** list. Define a new constant or select an existing one:

   - To define a new constant, follow these instructions:
     a. Double-click **< new constant >** in the **Value** list. The **Value** list changes into a field.
     b. Enter a value in the **Value** field. The value can be any number or character string. The value is displayed in the **Value** list.
     c. Select the new constant in the **Value** list.
   - To use an existing constant, select it in the **Value** list.

7. Complete your first expression by clicking the **Arg2** push button. The selected field name or constant replaces `Arg2` in the formula.

To define complex filter records conditions, you can link several expressions together by using Boolean operators. Click the **AND** push button or the **OR** push button to define another expression that is linked with the one before. The sequence of expressions is interpreted according to Boolean logic. See the example in "Filtering records" on page 115.

8. Click the **OK** push button to save your filter records condition and close the Expression Builder.

Using power options

In the **Power options** field on the Input data page of a wizard or settings object, you can specify additional parameters to further influence the behavior of a mining or statistics function.

Attention: When you use power options, no feedback about your actions is given, and no error messages are issued. Make sure that you type in your power options correctly.

Examples for the use of power options are in the following sections of this book:

- "Using power options with tree classification" on page 196.
- "Using power options with neural classification" on page 206.
- "Using power options with RBF prediction" on page 219.
- "Using power options with linear regression" on page 245.
- "Using power options with univariate curve fitting" on page 247.

- “Using power options with bivariate statistics” on page 248.
- “Using power options with demographic clustering” on page 156.

The **Power options** field is displayed only if you use the advanced pages and controls.

Specifying parallel parameters

The following server operating systems allow you to run the IBM DB2 Intelligent Miner for Data in serial mode or in parallel mode:

- AIX
- Solaris Operating Environment
- Windows

Running the Intelligent Miner in serial mode means that the program uses a single processing node. Running the Intelligent Miner in parallel mode means that the program uses more than one processing node simultaneously. Running the Intelligent Miner on several parallel processing nodes improves the performance significantly.

Table 27 shows the functions that you can run in parallel mode.

Table 27. Functions that you can run in parallel mode

Mining functions	Statistics functions	Preprocessing functions
Associations Neural Classification Tree Classification Demographic Clustering Neural Clustering Neural Prediction RBF-Prediction (in application mode only) Sequential Patterns	All Bivariate Statistics functions, except for Quantiles and Sampling.	Copy Records to File

Serial mode

To use the serial mode, select **Run the serial mode of the function** on the Parallel parameters page of an appropriate wizard or settings notebook.

In serial mode, the Intelligent Miner uses the server node. The server node is the processing node of the Intelligent Miner server that you logged on to in the Preferences notebook. If you run the Intelligent Miner in stand-alone mode, it is the node of your local workstation.

Parallel mode

To run a function in parallel mode, select **Run the parallel mode of the function** on the Parallel parameters page of an appropriate wizard or settings notebook.

You can either let the system determine the number of parallel processes or specify a value of your own.

To let the system determine the number automatically, select **Automatically determine the number of parallel processes**.

To specify a value of your own:
- Select **Use this number of parallel processes**.
- Type the number of parallel processes in the field next to this option.

Partitioning the input data

When you use flat-file data as input, you must partition the files in question before you can run the Intelligent Miner in parallel mode. Each of the specified nodes processes a portion of the input data. By partitioning the input data, you define the portions that the Intelligent Miner allocates to the processing nodes. If your input data consists of several files, you must partition all the files.

Partitioning database tables

The Intelligent Miner is able to process partitioned database tables in parallel mode. When you specify such a partitioned table as input, the Intelligent Miner recognizes and allocates the partitions automatically.

If you use less processing nodes than table partitions exist, the Intelligent Miner tries to distribute the workload as equally as possible. Thus one or more nodes process more than one table partition so that the complete table is processed.

If you use more nodes than table partitions exist, the number of parallel tasks is reduced automatically to match the number of input partitions. Consequently, fewer nodes than actually available share the workload.

If you want each node to process only those table partitions that reside in the file system connected to it, make sure that the following conditions apply:
- The Intelligent Miner server runs on the same nodes as DB2 Universal Database Enterprise Extended Edition or DB2 Universal Database Enterprise Server Edition..
- In both files, the host-list file *idmhost.list* and the *db2nodes.cfg* file, the nodes are listed in the same order.

- The db2nodes.cfg file contains only the nodes that are used when a particular table is processed.
- You start as many parallel tasks as table partitions exist.

For information on how to create partitioned tables, see the documentation of your database management system.

Handling of output data: Some of the mining functions optionally generate output data. If the input data consists of a partitioned database table, the output data also consists of a partitioned database table with the same number of partitions.

The same partitioning key is used. Hence output records are processed by the same nodes as the corresponding input records and are written to the same file systems.

Exception: This rule does not apply if your input data consists of flat-file data and you produce a DB2 table as output.

Partitioning flat files

To partition your flat-file input data, run the idmdpart program. The names of the generated partitions consist of the target file name that you specified and an extension. The extension follows the pattern `Ii`, where *i* can be any positive integer or 0. A partition `Ii` is assigned to the parallel task with the number `i`.

The partitioning program also generates two control files. The first control file has the same file name as the target file, but does not have an extension. It contains the first record of the input data.

The second control file contains information about the number of partitions. It has an *N* as its extension.

Example: If you specify a target name part.data and four parallel tasks, idmdpart creates the following files:

part.data.I0
part.data.I1
part.data.I2
part.data.I3
part.data
part.data.N

The node that your client is connected to is called the *server node*. The server node must be able to access the control files to perform checks.

If you do not use a shared file system, the control files are generated in the file system belonging to the first node that is listed in the host-list file. If this node is not the server node, copy the control files to the file system accessed through the server node and ensure that the path is not changed.

When you are specifying input data in a Data Task Guide or settings notebook, specify the name of the first control file. In the example above, this is the part.data file. You do not need to specify the names of the partitions because the Intelligent Miner identifies the correct partition for each parallel task. The partitions can reside in the local file systems belonging to the nodes or in a shared file system.

If the server node cannot access the control files at the time that you create a data settings object for the input data, a warning message is displayed at the client. In this case, you must specify the record length of the partitioned input files because the Intelligent Miner cannot determine it automatically.

When you start a mining run, and the partitions reside in local file systems, you must ensure that the first node in the host-list file can access the file with the extension I0. The second node must be able to access the file with the extension I1, and so on. This is because the parallel operating environment assigns task IDs to the parallel tasks in the order of the nodes in the host-list file.

You can let the Intelligent Miner detect the number of partitions. This value is then taken as the number of nodes. If you do this, be aware that:

- The Intelligent Miner looks for data files with extensions of I0, I1, I2, and so on. If there is a gap in the sequence of extensions, some files are not detected, and thus the number of parallel tasks does not equal the number of partitions. For example, if there are three partitions with extensions of I0, I1, and I3, only the first and the second partitions are detected. Consequently, the value taken as the number of tasks is 2.
- The partitions must reside in a shared file system. Otherwise, the Intelligent Miner cannot automatically determine the number of files. If the partitions are not in a shared file system, you must specify the appropriate parameter of the idmdpart program to determine the correct number of files. See "Running the partitioning program" on page 122 for more information.

  Solaris Operating Environment: The partitions must reside in a file system that is shared by the parallel nodes. For more information on how to specify the parallel nodes, see "Running the Intelligent Miner server for the Solaris Operating Environment" on page 46.

Handling of output data: Some of the mining functions optionally generate output data. If the input data consists of flat files, the output data also consists of flat files.

The name of the output data is used by all parallel mining-run processes. To ensure that the names of the files that are generated by different processes are unique, an extension `Ii` is added, where *i* represents the task ID.

The first process that is listed in the host-list file generates a control file without an extension. The control file contains a sample record of the output data, so that the output data can be used as input data for another mining run.

Running the partitioning program

Run the **idmdpart** command from the command line or the command prompt for the following purposes:
- To partition an input file into a specified number of subfiles.
- To create a control file with the name specified as the target file name.

The following section uses syntax diagrams. If you are not familiar with syntax diagrams, see "How to read syntax diagrams" on page xiv for more information.

When you partition data for the Associations and the Sequential Patterns mining functions, run the **idmdpart** command with the following parameters:

idmdpart

```
►►──idmdpart──sourceFileName──reclen── -id beginPos:endPos──┬──────────────────────────────────────────────┬────────────►◄
                                                            └─targetFileName── -p nbparts── -nstripe n─┘
```

When you partition data for the other mining functions, run **idmdpart** with the following parameters:

idmdpart

```
►►──idmdpart──sourceFileName──reclen──────────────────────────────────────────►

►──┬────────────────────────────────────────────────────────┬────────────────►
   └─targetFileName── -nstripe n──┬─ -p nbparts───────────┬─┘
                                  └─┤ PE_Arguments ├──────┘

►─┤ PE_Arguments for Solaris Operating Environment: ├────────────────────────►◄
```

PE_Arguments for AIX:

```
├──-procs nbProcs──┬──────────────────────────────┬──────────────────────────────────────┤
                   └─┬─ -hfile────┬─hostFileName─┘
                     └─ -hostfile─┘
```

PE_Arguments for Windows:

```
├──-procs nbProcs──────────────────────────────────────────────────────────────────────┤
```

Parameters

sourceFileName
: The name of the data file to be partitioned. Include the full path when specifying the file name.

reclen
: The record length of one record in the source file, including the trailing new-line characters.

-id *beginPos***:***endPos*
: The first and the last column number of the ID field that identifies records that must be kept together in one partition. The data in the source file must be sorted according to the sequence of the ID fields.

  Use the transaction ID as the ID field when partitioning data to discover associations.

  Use the transaction group ID as the ID field when partitioning data to discover sequential patterns.

targetFileName
: The base name of the files that are created after the partitioning. If files with the target file name already exist, they are replaced with the new files. Include the full path when specifying the target file name. An extension `Ii`, where *i* can be any positive integer or 0, is automatically added to the target file name. The partitions can reside in a shared or a local file system. The target file name must be different to the source file name.

  If you do not specify *targetFileName*, the name of the source file is used and a record control file is not created.

-p *nbparts*
: The number of partitions that you want to generate and store in a single file system. This file system can be a shared file system. Specify an integer for *nbparts* to determine the number of partitions.

  Specifying this parameter is unnecessary if you use the *PE_Arguments* parameter to specify the number of parallel tasks. By using *PE_Arguments*, you also specify that the partioned output files are distributed equally among the file systems connected to each node. Thus the value of *PE_Arguments* also determines the number of partitions, that is, one per node or parallel task.

-nstripe *n*

Specifies how many successive input records are written to the same output partition before a split is performed, and the records are written to the next output partition. The default value is 1. For example, if you specify *-nstripe 2*, input records are written to output partitions as shown in Table 28.

Table 28. Method by which input records are written to output partitions

Number of input record	Written to output partition
1	1
2	1
3	2
4	2
5	3
6	3
7	1
8	1

If you specify a value for -nstripe *n* and use the Demographic Clustering function, make sure that you enter the same value in the **Records per data stripe** field. You find this field on the Parallel parameters page of a Demographic Clustering wizard or settings notebook if you switch on the advanced pages and controls.

PE_Arguments

Set of parameters to determine the number of parallel tasks. Which parameters you have to use depends on your operating system. If you do not specify these parameters at all, you must use the -p *nbparts* parameter to determine the number of partitions.

-procs *nbProcs*

The parameters for *PE_Arguments* on AIX.

You need not specify this parameter if you set the environment variable MP_PROCS.

-hostfile *hostFileName*

The host-list file that contains the names of the nodes. If it contains more names than specified by the -procs *nbprocs* parameter, a subset of nodes is used. For example, if you specify -procs *4*, the first four nodes in the host-list file are used.

If only the file name is given, the file must reside in the current directory. To avoid problems, use the host-list file that you also use for the mining run.

You need not specify the host-list file if you set the environment variable MP_HOSTFILE.

Specifying output settings

Many Intelligent Miner functions always produce output data, and many others can produce output data. You might want to use output data as input for further mining runs or keep it for use with other applications. Saving output data is particularly useful if new fields and values are calculated during processing. You can save output data as a database table or as a flat file.

Table 29 shows the functions that produce output data.

Table 29. Functions that produce output data

Function	Mode (if exists)	Produces output data?	Type of output data
Neural Classification Tree Classification Neural Prediction RBF-Prediction	Training	Optional	Database tables Database views Flat files
	Test	Optional	
	Application	Always	
Demographic Clustering Neural Clustering	Clustering	Optional	Database tables Database views Flat files
	Application	Always	
Bivariate Statistics Principal Component Analysis Univariate Curve Fitting	N/A	Optional	Database tables Database views Flat files
Factor Analysis Linear Regression	Training	Optional	Database tables Database views Flat files
	Application	Always	

Table 29. Functions that produce output data (continued)

Function	Mode (if exists)	Produces output data?	Type of output data
Aggregate Values Calculate Values Convert to Lowercase or Uppercase Discard Records with Missing Values Encode Missing Values Encode Nonvalid Value Filter Fields Filter Records Filter Records Using a Value Set Get Random Sample Group Records Join Data Sources Map Values Pivot Fields to Records	N/A	Always	Database tables Database views
Copy Records to File	N/A	Always	Flat files only
Discretization into Quantiles Discretization Using Ranges	N/A	Always	Database tables only

Restriction: You can only create views if you used a view as input.

Before output data can be created, you must select the fields to be written to the output data. If the function produces output data optionally, select **Create output data** on the Output fields page of a wizard or settings notebook. If the function you use always produces output data, you cannot deselect this button.

The output data object contains the same subset of records as are shown in the sampling result. However, you can select different fields for the output data object and the result object. For example, you can generate small sampling results, but large output data objects. Thus the sampling results show only the most important fields, while the output data object is large enough to serve as input for mining functions or other statistics functions.

To make the necessary specifications when output data is created, follow this procedure:

1. To select the fields that are written to the output data, use one of the following methods:
   - To include all the fields from the input data in the output data, click the >> push button on the Output fields page. This action moves all items in the **Available fields** list to the **Output fields** list.

     If you want to remove one or more fields, do either of the following:

  - Select the fields that you want to remove from the **Output fields** list and click the < push button.
  - Click the << push button to remove all fields from the **Output fields** list.
- To include only one or a few of the fields from the input data in the output data, select these fields in the **Available fields** list and click the > push button. This moves the selected fields to the **Output fields** list.

2. If applicable, enter names for fields that are created during the mining run in the appropriate entry fields.
3. If you are using a wizard, click the **Next** push button to access the Output data page. If you are using a settings notebook, click the Output data tab of the notebook to access that page.
4. Specify an existing data object for the output data or create a data object:
   - To specify an existing data object, select a data object in the **Available output data** pane. The name of this settings object is displayed in the **Settings name** field. The comment of this settings object, if any, is displayed in the **Comment** field.
   - To create a data object, click the **Create data** button on the Output data page. The Data wizard opens. Follow these steps:
     a. Continue with step 2 on page 85.
     b. After selecting the use mode, check **The specified flat file does not yet exist** or **The specified table does not yet exist**. Checking the respective option reduces the number of parameters that you must specify on the Data wizard because the Intelligent Miner derives the required information from the input data object and from your output specifications.

        Attention: Be aware that specifying the name of an existing table, view or flat file overwrites that table, view or flat file unless it is write-protected. If it is write-protected, an error message is issued when the attempt to write the table, view, or file fails.

        When using database tables or views, you can additionally check **Output should be appended to the specified table**, but note that the output can only be appended if the existing table or view is not write-protected.
     c. Click the **Finish** push button to close the Data wizard and return to the mining or statistics wizard or settings notebook.

5. If you are using a wizard, click the **Next** push button to access the next page. If you are in using a settings notebook, click the **Apply** push button to save your changes or the **OK** push button to save the changes and close the notebook.

Using additional functions

You can filter the data fields in the **Available fields** list by specifying a field name in the **Filter mask** entry field. You can also use wildcards.

Example: To display only data fields whose names contain the string *customer*, enter `*customer*` in the **Filter mask** field on the Input fields page of a Task Guide or settings notebook.

To redisplay all data fields in the **Available fields** list, delete *customer** and press Enter.

In the **Available fields** list, you can sort the fields in increasing or decreasing alphabetical order. Click the right mouse button and move the pointer over the displayed selection menu.

Size limitations

If the output data is larger than 2 GB, the function splits the file into as many files of 2 GB as required. Each split file retains the name of the specified output file. Numbers are added as extensions in ascending order.

Example: The output data of the Quality food supermarket clusters might be 9 GB. The following files are created:

Quality food supermarket clusters.od
Quality food supermarket clusters.od.1
Quality food supermarket clusters.od.2
Quality food supermarket clusters.od.3
Quality food supermarket clusters.od.4

Exception: The IBM AIX Parallel I/O File System (PIOFS) and DB2 allow output files that are larger than 2 GB. Consequently, the output data is not split.

Renaming result objects

By default, a result object takes on the same name as the mining or statistics settings object that created the result. However, you can specify a different name for the result object.

To rename a result object, follow these steps:

1. Replace the default result name by entering a new name in the **Results name** field. You find the **Results name** field on the Results page of a wizard or settings notebook.
2. Optionally, add a description in the **Comment** field.
3. Click the **Next**, **Apply**, or **OK** push button.

For more information on result objects, see "Result objects" on page 97.

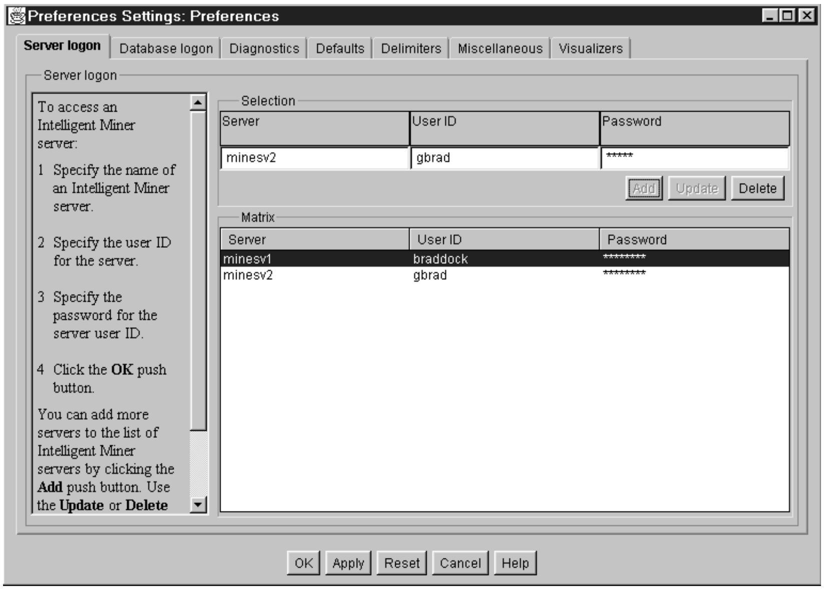
Replacing existing results

To replace an existing result object with the new result object to be created by a mining or statistics function, follow these steps:

1. Select the result-object icon in the **Available results** pane on the Results page of a mining or statistics wizard or settings notebook. The name and the comment of the selected object are copied into the input fields.
2. Check the **If a result with this name exists, overwrite it** box.
3. Click the **Next**, **Apply**, or **OK** push button.

Setting preferences for the Intelligent Miner

Use the Preferences notebook to establish connections to an Intelligent Miner server and to set other preferences for your data mining work. Click **Options → Preferences** on the menu bar in the Intelligent Miner main window. Figure 10 on page 130 shows the Server logon page.



To access an Intelligent Miner server:

1 Specify the name of an Intelligent Miner server.

2 Specify the user ID for the server.

3 Specify the password for the server user ID.

4 Click the **OK** push button.

You can add more servers to the list of Intelligent Miner servers by clicking the **Add** push button. Use the **Update** or **Delete**

Selection

Server	User ID	Password
minesv2	gbrad	*****

Matrix

Server	User ID	Password
minesv1	braddock	********
minesv2	gbrad	********

Figure 10. Server logon page in the Preferences notebook

You might need to connect to an Intelligent Miner server before you can access a particular mining base. The server name must be defined on the TCP/IP name server used by the Intelligent Miner server and the clients. Instead of server names, you can also enter IP addresses.

To establish a server connection for the first time, enter the server name, user ID, and password in the entry fields on the Server logon page. The password will not be displayed.

Click the **Add** button. The server name and the user ID are displayed in the list called Matrix at the bottom of the page. For the password, asterisks are displayed. Repeat this procedure to add further servers to the list.

You can also change an entry in the list. Select the entry. The values are displayed in the fields where you entered them for the first time. Make the necessary changes in the fields and click the **Update** push button to apply your changes.

To delete an entry, select it in the list and click the **Delete** push button.

When you revisit the page to reconnect to the same server, click the server name in the list and enter the password in the **Password** field. The wizard text field on the left tells you how to use this page of the notebook.

If the data you want to mine resides on a database server, connect to such a server by using the Database logon page. Use the entry fields to enter the name of the database server and your user ID and password for that server. The password will not be displayed. This page is similar to the Server logon notebook page.

The following list shows you how you can further control the behavior of the Intelligent Miner by using the Preferences notebook:

- Switch tracing information on or off and determine the trace level on the server
- Set a memory limit on the server
- Specify the work directory on the server
- Specify the default data format for the Data wizard on the client
- Show the advanced pages and controls by default on the client
- Specify that a settings objects is run automatically after you complete a wizard
- Select another list or range delimiter
- Skip the Welcome page when you open a wizard on the client.
- Specify that a visualizer is automatically run on the client when results were produced
- Select the display characteristics for the main window containers of the graphical user interface on the client.
- Enable autosaving of mining bases on the server
- Connect result objects to visualizers of your choice

See the online help for more information.

Viewing trace information

Trace information gives you hints about the nature or cause of errors. The Intelligent Miner provides trace information in the following places:

- In the trace log during a client session
- In files on the server

In addition, you can set the IDM_DEBUG environment variable to `on` on the client. When you run the Intelligent Miner, information about all API calls is displayed in a separate window, including error messages. See “Client variables” on page 351 for more information.

Viewing the trace log

The Intelligent Miner records each mining action that you perform in the trace log. The trace log might give you a hint about the nature or causes of an error if an Intelligent Miner function fails or does not run properly.

To view the trace log, Click **Options → Trace Log** on the menu bar in the Intelligent Miner main window. The trace information is displayed in the Trace Log window.

To remove the current trace information from the window, click the **Clear** push button.

To print the current trace information, perform these steps:

1. Click the **Print** push button.

   The Print Trace Log window opens.
2. Select a printer from the list.
3. Click the **OK** push button.

Viewing trace files

You can configure the Intelligent Miner to store trace information in files on the server. The Intelligent Miner then creates a file with an extension of TRC or trc each time you run a mining function. The trace files are stored in your idmres directory, which is located in your home directory on the server or in the directory identified by the IDM_RES_DIR environment variable. You can view the trace files with any text editor.

To enable the creation of trace files on the server, follow these instructions:

1. Click **Options → Preferences** on the menu bar in the Intelligent Miner main window.
2. Select the **On; trace level** radio button on the Diagnostics page.
3. Select a trace level between 1 and 10 by clicking the spin buttons right of the field that displays the trace level.

   Trace level 1 generates very basic trace information, whereas trace level 10 gives you the most detailed trace information.
4. Click the **Apply** push button to save your settings, or click the **OK** push button to save your settings and close the Preferences notebook.

Viewing supplementary information

After each mining run, the Intelligent Miner creates or updates a file with an extension of ERR or err on the server. This file contains all error messages, warnings, and some information messages that might also prove useful when you track down an error. Like the TRC files, this file is stored in your idmres directory.

Using help in the Intelligent Miner

The Intelligent Miner provides online help for all the tasks that you perform. On Windows, the online help is displayed in the default Web browser. To display the online help on AIX, use the Netscape Navigator. Your browser should be capable of displaying HTML 3.2 and frames. In the main window, click **Help → Help Contents** to display a table of contents that provides access to the available help content.

For help on the main window itself, click the appropriate push button on the task bar or **Help → Help for Main Window** on the menu bar.

To perform Intelligent Miner tasks, you use wizards that provide instructions on the left side for each page of a wizard. These instructions help you complete the task for that page, and also provide definitions of terms that appear on the interface. To see a term definition, click on any highlighted term on a wizard page. The first page of a wizard is a Welcome page that provides an overview of the task you are performing.

You can also display help for a particular page in a wizard or settings notebook by clicking the **Help** button on that page. For example, if you click the **Help** push button on the Parameters page of the mining wizard for the Associations mining function, you can see help about concepts and examples or valid values and actions.

When you receive an error message, you can get additional help from the Internet. Click the **Web help** push button in the error message window to start your default Web browser and load the corresponding Web site. If your problem occurred before, a detailed problem discussion is displayed, providing the most probable causes of your problem and a possible solution. If your problem did not occur before, you can fill in and submit an Internet form. The Intelligent Miner support team will then investigate your problem.

When the Intelligent Miner client is not running, you can view the help by opening the idmbmain.htm file in a frames-capable HTML browser. Depending on the client operating system and the system configuration that you use, this file resides in one of the directories that are listed in Table 30 on page 134.

Table 30. The location of help files

Operating system	Text file directory
AIX	/usr/lpp/IMiner/html (for online help in English) or /usr/lpp/IMiner/nls/$LANG/html (for online help in one of the supported languages, where $LANG is the directory of a particular language)
Windows	X:\im\html (where X is the drive letter) **Note:** The im directory is the default directory. During installation, you might have chosen another directory.

Exporting and importing mining bases

You can export mining bases from a server by writing them to files on a client workstation. When you import the mining base on a client workstation, these files are read and merged into the mining base that is currently open. By saving the current mining base, you add the contents of the imported mining base. You can repeat this process so that each time you import a mining base, it is appended to the current mining base. To import exactly the same mining base that you exported, merge the exported mining base into an empty mining base. Exported mining bases do not need to contain results from previous mining runs. You can exclude results from the export process.

To export a mining base, you must specify a package name. The package name is used to identify the group of files that is created by the export facility. The package name is the file-name stem. The export facility adds an extension to each file it creates, so that the file names consist of the package name and that extension. If, for example, you choose a package name sales, the following files are created during the export process:

sales.mnb	Contains the mining base.
sales.des	A description file for the export process. You can change the entries in this file to adjust the export specifications as required on the target system.
sales.1 to sales.n	The results of previous mining runs, if results were exported.

The description file contains specifications for objects that are part of the mining base to be exported. These objects can be:

- Data objects
- Name mapping objects
- Value mapping objects
- Discretization objects
- Taxonomy relation objects
- Preprocessing-function objects

Use the description file to specify the name of an input data object on the target system or to change the entry of its location. Field names and field specifications cannot be changed in the description file. If you change the name of input data in the description file, make sure that the new input data contains the same fields as the original. Problems can occur if the field names do not match.

You can change the following flat-file attributes in the description file:

- Record length

  The record length includes the trailing new-line character. You must include the trailing new-line character to exchange mining bases between different operating systems. On AIX and in the Solaris Operating Environment, the new-line character consists of one character. In AS/400 stream files, the new-line character also consists of one character. On OS/390, a new-line character is not available. On Windows, the new-line character consists of two characters.
- File names and paths
- Number of files

You can change the following attributes for database tables in the description file:

- Database server
- Schema
- Table name
- Table space (if specified)
- OS/390 database (if specified)

Figure 11 on page 136 shows a description file created during an export process. Entries with a # at the beginning are keywords that help you to identify parts in the description file. Do not change any keyword or its position. Not all possible keywords are shown in this example. For a complete list of keywords, see Appendix D, "Description file keywords" on page 355.

The numbers in parentheses refer to the items in the list of explanations immediately following the example. The square brackets on the left help you to identify groups that belong together.

You can edit the description file with an editor of your choice.

```
#idm-global-reclen-change                  -> (1)
0                                          -> (2)
#idm-name-mapping-objects                  -> (3)
#idm-name-mapping-object-name              -> (4)
Item gp names                              -> (5)
#idm-flat-file-table                       -> (6)
23                                         -> (7)
1                                          -> (8)
/home/idm/sample.gpt                       -> (9)
#idm-name-mapping-object-name
Item group names
#idm-flat-file-table
23
1
/home/idm/sample.pt
#idm-name-mapping-object-name
NmpObj
#idm-flat-file-table
24
1
/home/idm/sample.names
#idm-taxonomy-relation-objects             -> (10)
#idm-taxonomy-relation-object-categories   -> (11)
Item Groups                                -> (12)
Item Gp Groups                             -> (13)
#idm-flat-file-table
23
2
/home/idm/sample.tax
/home/idm/sample1.tax
#idm-taxonomy-relation-object-categories
Items
Item Groups
#idm-flat-file-table
23
2
/home/idm/sample.tax
/home/idm/sample1.tax
```

Figure 11. A description file

```
┌──#idm-data-objects                                  -> (14)
│┌─#idm-data-object-name                              -> (15)
││  DataObj                                           -> (16)
││┌#idm-db2-table                                     -> (17)
│││PESAMPLE                                           -> (18)
│││IMDB2PE                                            -> (19)
│││DATAWK12                                           -> (20)
│││#idm-db2-tablespace                                -> (21)
││└outspace                                           -> (22)
└┴─                                                   -> (23)
┌──#idm-aggregate-values-objects                      -> (24)
│┌─#idm-processing-object-name                        -> (25)
││  AggValues                                         -> (26)
││┌#idm-processing-object-values                      -> (27)
│││PESAMPLE                                           -> (28)
│││IMDB2PE                                            -> (29)
│││DATAWK12                                           -> (30)
│││IMDB2PE                                            -> (31)
│└┴DATAWK12WOP                                        -> (32)
│                                                     -> (33)
│  #imd-db2-mvs-database                              -> (34)
└──TestBase                                           -> (35)
┌──#idm-discard-records-with-missing-values-objects   -> (36)
│┌─#idm-processing-object-name
││  Drn
││┌#idm-processing-object-values
│││PESAMPLE
│││IMDB2PE
│││DATAWK12
│││IMDB2PE
└┴┴DATAWK12WOP
```

Figure 12. A description file, continued

Notes to figure:

(1) `#idm-global-reclen-change`

This keyword indicates that the next line applies to the record length of all flat files whose names occur in the description file.

(2) `0`

This value specifies the number of columns to be added or subtracted from the original record length.

- If the value is negative, the record length is reduced by the number of columns specified.
- If the value is positive, the record length increases by the number of columns specified.
- If the value is 0, as in the example, record length specifications remain unchanged during import.

(3) `#idm-name-mapping-objects`

This keyword indicates that the list of name mappings belonging to the mining base follows.

(4) `#idm-name-mapping-object-name`

The keyword indicates that the name of a particular name mapping follows. Each name mapping in the list of name mappings begins with this keyword.

(5) `Item gp names`

This line gives the name of the name mapping. Do not change the names of name mappings.

(6) `#idm-flat-file-table`

This keyword indicates that the name mapping is a flat file and that specifications for this flat file follow.

(7) `23`

This number specifies the record length of the flat file. You can change it if you have not specified a global value.

(8) `1`

This value specifies the number of files following. It must match the number of the following file name entries.

(9) `/home/idm/sample.gpt`

This file name entry gives you the path and the file name of the flat file.

(10) `#idm-taxonomy-relation-objects`

This keyword indicates that the list of taxonomy relations belonging to the mining base follows.

(11) `#idm-taxonomy-relation-categories`

This keyword indicates that the names of the item categories belonging to a taxonomy relation follow.

(12) `Item Groups`

The name of a child item category. Do not change the names of child item categories.

(13) `Item Gp Groups`

The name of a parent item category. Do not change the names of parent item categories.

(14) `#idm-data-objects`

This keyword indicates that the list of data objects belonging to the mining base follows.

(15) `#idm-data-object-name`

This keyword indicates that the name of a particular data object follows.

(16) `DataObj`

The name of a data object. Do not change the names of data objects.

(17) `#idm-db2-table`

This keyword indicates that the data object is a database table, and that specifications for that database table follow.

(18) `PESAMPLE`

The name of the database server on which the database table is located. If required, you can change it.

(19) `IMDB2PE`

The name of the schema of the database table. If required, you can change it.

On AS/400, the schema is the library where the table is located.

(20) `DATAWK12`

The name of the database table. If required, you can change it.

(21) `#idm-dB2-tablespace`

This keyword indicates that the name of the table space containing the database table follows.

(22) `outspace`

The name of the table space. If required, you can change it. If you do not need the table space specification after importing the mining base, delete the keyword and the name of the table space, and leave a blank line.

(23) Do not delete this blank line. It indicates that an OS/390 database was not specified.

(24) `#idm-aggregate-values-objects`

This keyword indicates that the list of objects containing settings for the Aggregate values preprocessing function follows.

(25) `#idm-processing-object-name`

This keyword indicates that the name of a particular object containing preprocessing-function settings follows.

(26) `AggValues`

The name of an output data object that contains settings for the Aggregate values function. Do not change it.

(27) `#idm-processing-object-values`

This keyword indicates that the following section applies to the database table specifications of the object that contains settings for the Aggregate values function. This section includes information about the output data and the input data related to it.

(28) `PESAMPLE`

The name of the database server on which the input data and the output data are located. If required, you can change it.

(29) `IMDB2PE`

The name of the schema of the input data.

(30) `DATAWK12`

The name of the input data that serves as the basis of the calculations.

(31) `IMDB2PE`

The name of the schema of the output data.

(32) `DATAWK12WOP`

The name of the output data.

(33) Do not delete this blank line. It indicates that a table space was not specified.

(34) `#idm-db2-mvs-database`

This keyword indicates that the next section applies to OS/390 databases.

(35) `TestBase`

The name of the OS/390 database. If required, you can change it. If you do not need the database specification after importing the mining base, delete the keyword and the name of the database, and leave a blank line.

(36) `#idm-discard-records-with-missing-values-objects`

This keyword indicates that the next section applies to objects that contain settings for the Discard records with missing values function.

To import a mining base, you must specify a package name and the directory containing the files that belong to the package. The import facility then imports all files for the package. When the mining base is read, the specifications in the description file are added to the specifications in the current mining base.

If you merge two mining bases, and both mining bases contain an object of the same type (for example, a data object) and with the same name, an extension `_1` is added to the name of the imported object. If an identical object with such an extension already exists, the extension is changed to `_2`. This process is repeated until the names of the objects are distinct. If an object name is changed, its reference entries in other objects are also changed accordingly. However, this is not true for results. If you referred to results in other objects, you must correct the reference entries manually after you have imported the mining base.

When you export mining bases without results, you must manually delete any references to result objects after importing the mining bases.

Chapter 9. The Associations mining function

The purpose of discovering associations is to find items in a transaction that imply the presence of other items in the same transaction.

Suppose that you have a database of purchase transactions. Let each transaction in this database consist of a set of items purchased by a customer. The Associations mining function discovers relationships between the items in a set. You might find that 60% of the customers who buy greeting cards also buy cosmetics.

The following sections describe the basic and advanced parameters of the Associations function. To find all parameter controls on the wizard or notebook pages, you must check the **Show the advanced pages and controls** box on the settings page.

Sample parameter settings

Table 31 shows sample settings summarizing the basic and advanced parameters of the Associations mining function. In addition, it indicates on which wizard page you must specify the required and optional parameters.

Table 31. Sample parameter settings for the Associations mining function

Wizard page	Parameter	Value
Settings	Name	Sample settings
	Comment	Data collected during 1st half of 1997
	Mining function	Associations
Input data	Input data	Sunset June transactions
	Optimize mining run for	Time
	Filter records	Conditions selected
	Power options	
Input fields	Transaction field	Customer number
	Item field	Article
	Sort the input data on the values in the Transaction ID field before running this function	False

Table 31. Sample parameter settings for the Associations mining function (continued)

Wizard page	Parameter	Value
Parameters	Minimum support	5
	Minimum confidence	75
	Maximum rule length	3
	Item constraints	5 selected
Parallel parameters	Run the parallel mode of this function	Use this number of parallel processes: 4
Taxonomy	Taxonomy name	Sunset non-food
Results	Results name	Sunset June associations
	Comment	Branch locations New York City
	If a result with this name exists, overwrite it	True

You can use the settings individually or as part of a sequence.

Specifying input data

To use the Associations mining function, select an input data object on the Input data page of a wizard or settings notebook. See "Selecting an input data object" on page 114 for more information.

Example: You might want to look for associations in the transactions of one of the Sunset retail branches in New York City in June.

A transaction might consist of several records. In this example, customer number 123 buys a T-shirt and tennis shoes. This is reflected by two records. However, both purchases belong to the same transaction, because the date and the transaction number are the same. In contrast, customer number 487 also purchased three items on the same day, but only the first two items belong to the same transaction, while the third item has a different transaction number.

Ensure that the data is sorted by the transaction identifier. In the example shown in Table 32 on page 145, this is the date and the transaction number.

Table 32. Sample data for the Associations mining function

Customer number	Item	Date	Transaction number	Type of sale	Customer age	Payment	Total purchase amount
123	T-shirt	06/02/97	2215	On sale	16	Cash	$10.50
123	Tennis shoes	06/02/97	2215	Regular	16	Cash	$98.00
148	Twin set	06/02/97	2437	On sale	21	Credit card	$35.50
168	Costume	06/02/97	2437	Regular	35	Credit card	$212.50
168	Jeans	06/03/97	2437	Regular	35	Credit card	$49.50
321	Suit	06/03/97	2820	Regular	45	Cash	$225.00
487	Swim suit	06/03/97	3046	On sale	22	Credit card	$49.80
487	Beach towel	06/03/97	3046	Regular	22	Credit card	$25.00
487	Sun glasses	06/03/97	4217	Regular	22	Credit card	$30.75

Affinities: A mining run for associations on these transactions returns affinities existing among the item sets. For example, if customers buy swimsuits and beach towels, in 90% of all cases they buy sun glasses, too.

The affinities are expressed in rules. An association rule X => Y consists of the following components:
- An item set X in the rule body
- An item set Y in the rule head

The transaction t supports the rule X => Y if all items occurring in the rule are found in the transaction.

Optimizing the mining run

You might want to optimize the mining run for time or disk space. See "Optimizing the mining run" on page 114 for more information.

Filtering records

You can restrict the mining run to records with certain field values. To do so, specify a filter records condition. See "Filtering records" on page 115 for more information.

Specifying input fields

To find items in a transaction that imply the presence of other items in the same transaction, you must specify the field names and the items of this transaction.

Transaction IDs, for example, represent the date or the time stamp, or the customer number. The data in the Transaction ID field must be sorted sequentially. If you are not sure whether this data is sorted, check the **Sort** option.

Restriction: This information applies to the IBM DB2 Intelligent Miner for Data for OS/390 only.

If your input data resides in a database table, the DB2 data type of the field that you select as the **Transaction ID** field must be *CHAR* or *VARCHAR*. In addition, the Intelligent Miner data type of this field must be *Categorical*. Otherwise you might get incorrect results because the data in this field cannot be sorted properly.

Item IDs represent the different articles in a transaction.

Example: To find associations in the June transactions of the Sunset retail stores in New York City, you must specify the fields shown in Table 33.

Table 33. Specifying fields for the transaction ID and the item ID

Transaction ID	Item ID
Customer number	Item

Support and confidence factor

You can control the results of a mining run by changing the values for the following parameters:

Minimum support
Indicates the relative occurrence of the detected association rules within the input data. It is determined by dividing the number of transactions supporting the association rule by the total number of transactions.

Minimum confidence
Indicates the relative strength or reliability of the detected association rules within the input data. It is determined by dividing the number of transactions supporting the association rule by the number of transactions supporting the rule body only.

Maximum rule length

Determines the number of items that occur in an association rule. If you specify 3 as maximum rule length, you receive association rules with at most two items in the rule body and one item in the rule head, as in this example:

[Swimsuit] AND [Beach towel] ==> [Sunglasses]

Only association rules with a support and confidence factor greater than or equal to the minimum value are included in the result.

Frequent item sets with a support factor greater than or equal to the minimum value are also included in the result. The minimum confidence factor, if specified, is irrelevant in this case. So no matter if a minimum confidence factor was specified or not, frequent item sets that meet the requirement for the minimum support are included in the result.

Rules based on a high support and confidence factor represent a higher degree of relevance than rules with a low support and confidence factor. If very few, or no association rules are discovered, the reason may be that the specified value for minimum support or minimum confidence is too high.

Default values for minimum support and minimum confidence:

By default, the mining function tries to determine a value for the minimum support. The value for minimum support is adjusted such that 1/4 of all single items are frequent.

The default value for minimum confidence is 25%.

Memory limits:

If the specified value for minimum support is too low, the system might run out of memory, and no results are generated. You can increase either the value for minimum support or the memory limit. To change the memory limit, click **Options → Preferences** on the menu bar in the Intelligent Miner main window.

Important: Do not attempt to increase the memory limit beyond the size available to the mining function.

By default, 32 MB of memory are used to generate frequent item sets. A large number of rules, typically more than ten thousand, are generated before this limit is reached.

Item constraints

The specification of the item constraints determines which rules are to be included in or excluded from the results. If you selected to include the specified items, only rules containing at least one of the specified items are generated. If you selected to exclude the specified items, rules containing one of the specified items are discarded from the results.

Example: To view only rules that include the items sunglasses, beach towels, shorts, swimsuits, and caps with a reliability of 75% and a relative occurrence of 5%, specify the values shown in Table 34.

Table 34. Specifying item constraints

Minimum support	Minimum confidence	Maximum rule length	Item constraints
5	75	3	Sunglasses; beach towels; shorts; swimsuits; caps

In this example, a semicolon is used as the delimiter for item constraints. To select a different delimiter:

1. Click **Options → Preferences** on the menu bar in the Intelligent Miner main window.
2. Select a different delimiter from the **List** drop-down list on the Delimiters page.

You can also use names at any level of a taxonomy hierarchy to specify item constraints. For example, if you created the taxonomy Beach wear, you can specify Beach wear in the **Item constraints** field to view association rules containing items such as sunglasses, beach towels, or caps.

Tips:

The minimum support value is a critical parameter. First, use the default system-determined minimum support value to get an idea of a reasonable value. You can see the system-determined value in the Statistics window when you view the result. The following list discusses factors that influence the number of rules the algorithm discovers.

- You will discover more rules that meet your criteria if you set the minimum support and confidence factors low.

  For example, if no frequent item sets are found for the given minimum support, decrease the value for minimum support.
- If no rules are found for the given support and confidence values, you might want to lower the minimum support value, lower the minimum confidence value, or lower both values.

- If you use a taxonomy, you usually get a large number of rules in contrast to not using a taxonomy. Using a taxonomy allows for a search for all levels in a hierarchy.
- If you restrict the rule length, you might find less rules that meet your criteria, however, processing time will be reduced.

Memory size effects whether the mining algorithm will run to completion. Depending on the factors in the previous list, the mining algorithm might find a large number of rules that meet your criteria. If the algorithm does not complete, increase your memory size.

Specifying parallel parameters

You can run the Associations mining function in serial or parallel mode. See "Specifying parallel parameters" on page 118 for more information.

Specifying taxonomies

A taxonomy is a hierarchy or lattice of relations between different categories of an item.

You can categorize items in several categories. Then you establish relations between these categories in terms of child item categories and parent item categories. Each child category can have several parent item categories. Each parent item category can have several grandparent item categories. The more item categories you define, the more detailed are the results of the mining functions.

Figure 13 on page 150 shows an example of a taxonomy of the Sunset retail store.

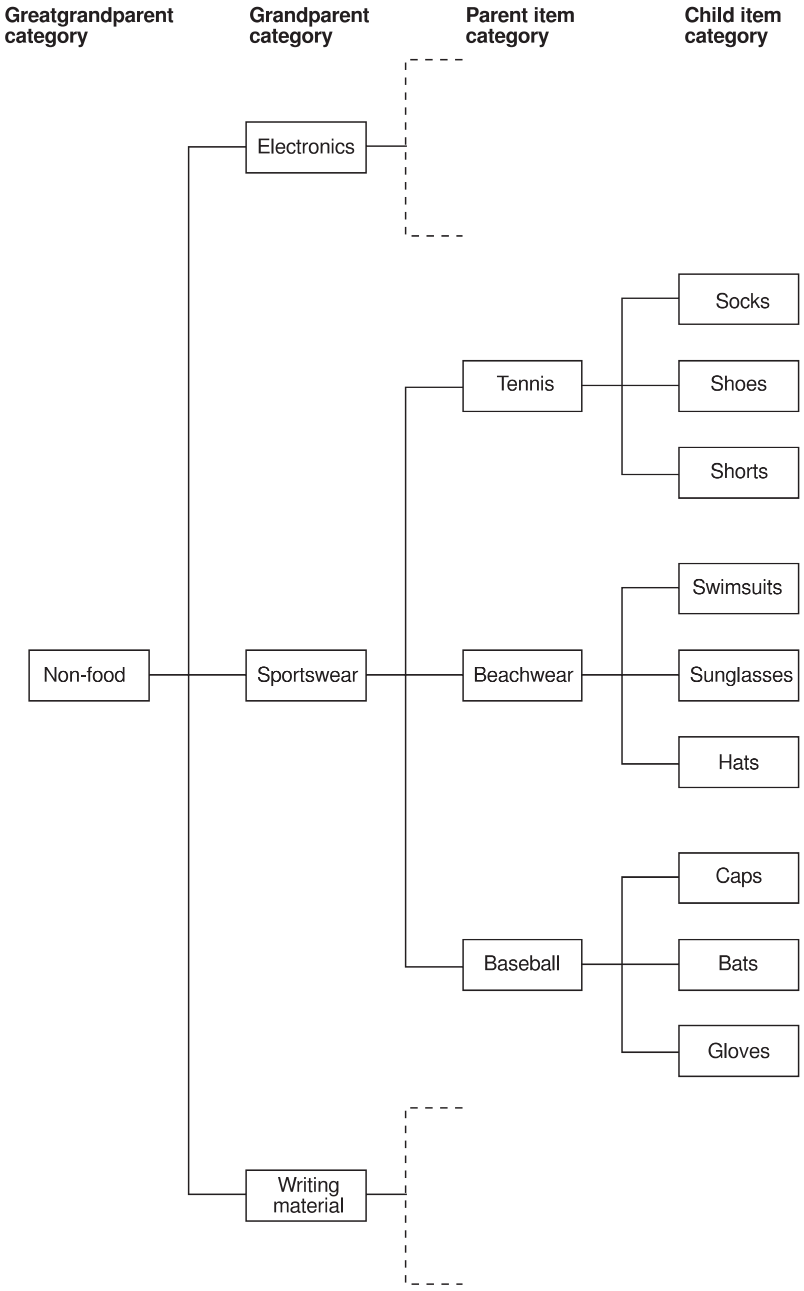


Figure 13. A taxonomy

Specifying result names

By default, a result object takes on the name of the settings object that created the results. Thus, you do not need to specify a result name when you run the settings object for the first time.

However, you might want to run the settings object again and keep the old results. In this case, you must change the original result name. See “Renaming result objects” on page 128 for more information.

Chapter 10. The Demographic Clustering mining function

The purpose of discovering clusters is to group records that have similar characteristics.

The Intelligent Miner searches the mining base for characteristics that most frequently occur in common, and groups the related records accordingly. The results of the clustering function contain the number of detected clusters and the characteristics that make up each cluster. In addition, the results show how these characteristics are distributed within the clusters.

Suppose that you have a database of a supermarket that includes customer identification and information about the date and time of the purchases. The clustering mining function clusters this data to enable the identification of different types of shoppers. For example, this might reveal that customers buy many articles on Fridays and usually pay by credit card.

Demographic Clustering provides fast and natural clustering of very large databases. It automatically determines the number of clusters to be generated.

Similarities between records are determined by comparing their field values. The clusters are then defined so that Condorcet's criterion is maximized.

Condorcet's criterion is the sum of all record similarities of pairs in the same cluster minus the sum of all record similarities of pairs in different clusters.

The following sections describe the basic and advanced parameters of the Demographic Clustering function. To find all parameter controls on the wizard or notebook pages, you must check the **Show the advanced pages and controls** box on the settings page.

Sample parameter settings

Table 35 shows sample settings summarizing the basic and advanced parameters of the Demographic Clustering mining function. In addition, it indicates on which wizard page you must specify the required and optional parameters.

Table 35. Sample parameter settings for the Demographic Clustering mining function

Wizard page	Parameter	Value
Mining function	Name	Sample settings

Table 35. Sample parameter settings for the Demographic Clustering mining function (continued)

Wizard page	Parameter	Value
	Comment	Data collected during 1st half of 1997
	Mining function	Clustering – Demographic
Input data	Input data	Quality food supermarket
	Optimize mining run for	Time
	Filter records	Conditions selected
	Power options	
Mode parameters	Use mode	Clustering mode
	Maximum passes	4
	Maximum clusters	9
	Accuracy improvement	1
	Similarity threshold	0.5
Input fields	Active fields	6 selected
	Supplementary fields	3 selected
Field parameters	Field parameters	5 selected
Additional field parameters	Additional field parameters	5 selected
Outlier treatment	Outlier treatment	Treat outliers as missing values
Similarity matrix		1 selected
Parallel parameters	Run the parallel mode of the function	Use this number of parallel processes: 4
Output fields	Output fields	1 selected
	Cluster ID field name	Cluster ID
	Record score field name	
	Cluster ID field name choice 2	
	Record score field name choice 2	
	Confidence field name	

Table 35. Sample parameter settings for the Demographic Clustering mining function (continued)

Wizard page	Parameter	Value
Output data	Output data	Quality food demographic clusters
Results	Results name	Quality food demographic clustering result
	Comment	May transactions
	If a result with this name exists, overwrite it	False

You can use the settings individually or as part of a sequence.

Specifying input data

To use the Demographic Clustering mining function, select an input data object on the Input data page of a wizard or settings notebook. See "Selecting an input data object" on page 114 for more information.

Example: The sample data in Table 36 is suitable for identifying various types of shoppers.

Table 36. Sample data for the Demographic Clustering mining function

Customer number	Number of articles	Day of the week	Customer age
123	2	Tuesday	65
196	24	Wednesday	65
234	4	Friday	72
308	19	Saturday	72
456	2	Monday	58

Gender	Marital status	Senior discount	Form of payment	Total purchase amount
Male	Married	Yes	Cash	9.67
Male	Single	Yes	Credit card	45.30
Female	Married	Yes	Cash	12.50
Male	Married	Yes	Check	67.09
Female	Single	No	Cash	2.48

Optimizing the mining run

You can optimize the mining run for time or disk space. See "Optimizing the mining run" on page 114 for more information.

If you optimize the mining run for time, the Demographic Clustering function stores information about the clusters to which records were assigned in the temporary file. This information is used during the mining run and might improve the quality of the model.

Filtering records

You can restrict the mining run to records with certain field values. To do so, you can specify a filter records condition. See "Filtering records" on page 115 for more information.

Using power options with demographic clustering

Starting with the version 6.1.1 of the Intelligent Miner, the Demographic Clustering mining function scores the records in a different way. Therefore you might receive different results if you use the same settings that you used in version 6.1. This can happen, for example, when you are building a model in training mode or when you are calculating clusters and scores in application mode. To maintain the compatibility with version 6.1, you can use the following power option in training mode and in application mode:

```
-scoreas6.1
```

Note: Use power options with great care. See "Using power options" on page 117 for a general introduction to power options.

Specifying input fields

To partition a database so that records that have similar characteristics are grouped together, you must specify active fields. You can also specify supplementary fields.

The active fields are used by the mining function for clustering. The supplementary fields are used to gain statistical information on the clusters that are found. They are not used for clustering, but in the clustering results viewer they appear as parts of the clusters. The fields in a cluster are ordered by importance. It is therefore possible that you see supplementary fields among the active fields in a cluster. This means that the supplementary fields would have influenced the creation of the cluster, perhaps more than the fields you specified as active fields.

Example: You might want to identify different shopper types and create clusters based on the number of articles purchased, the total purchase amount, the day of the purchase, and several characteristics of the customer. You might also want to have statistical information on the form of payment, gender, and marital status. Table 37 shows the required input fields, in accordance with the sample data.

Table 37. Selected input fields for the Demographic Clustering mining function

Active fields	Supplementary fields
Number of articles	Form of payment
Total purchase amount	Gender
Day of the week	Marital status
Customer age	
Senior discount	

Note: When selecting active fields, consider the following:

- In the **Available fields** list on the input fields page of the mining wizards, you can sort the fields in increasing or decreasing order. Click the right mouse button and move the pointer over the displayed selection field.
- Do not specify active fields that have the following characteristics:
  - Different values for almost every record
  - The same value in every record

  These fields do not contain any valuable information for building clusters. If fields have mostly identical values, they make any pair of records look more similar, and often only one cluster is created.
- If you have several fields in your input data containing very similar data, use the Principal Component Analysis or Factor Analyis functions before you run the clustering function. Principal Component Analysis and Factor Analysis amalgamate the similar fields to a single component or factor.

  When you run the clustering function, specify the components or factors as active fields, and the original fields as supplementary fields. This way, you can view the results of the clustering run plus the original field values.

Specifying field parameters

In the clustering process, all fields are treated equally. This might not always be useful. You can assign field weights and value weights to the active fields.

To specify a field weight, enter a value in the **Field weight** field of the appropriate input field on the Field parameters page of a Demographic Clustering wizard or settings notebook.

To specify a value weight, select the appropriate type of value weighting from the **Value weighting** drop-down list for the appropriate input field.

To access the Field parameters page, you must check **Show the advanced pages and controls** on the settings page.

Field weighting

Field weighting gives more or less weight to certain fields during the clustering process.

Example: You might not want to give too much weight to the strong correlation between the number of purchases and the total purchase amount. Therefore you might want to assign a smaller weight to these fields.

Table 38 shows lowered field weights for the **Number of purchases** and the **Total purchase amount** fields.

Table 38. Applying field weighting

Active fields	Field weighting	Value weighting	Compensate
Number of purchases	0.5	None	No
Day of purchase	1.0	None	No
Customer age	1.0	None	No
Senior discount	1.0	None	No
Total purchase amount	0.5	None	No

Value weighting

Value weighting deals with the fact that particular values in a field might be more common than other values in that field. The coincidence of rare values in a field adds more to the overall similarity than the coincidence of frequent values.

For example, most people do not have a Gold credit card. It is not very significant if two people do not hold such a card, however, if they do, it is significant. Therefore the coincidence of people not having a Gold credit card adds less to their overall similarity than the coincidence of people having such a card.

You can use one of the following types of value weighting:

Probability weighting
Probability weighting assigns a weight to each value according to its

probability in the input data. Rare values carry a large weight, while common values carry a small weight. This weight is used both for matching and disagreeing records.

Probability weighting uses a factor of $1/p$, where p is the probability of a value.

Information-theoretic weighting

Information-theoretic weighting measures the information content when judging the similarity score for a pair of records.

If a particular record has a field value that occurs very often in this field, it is not surprising if it encounters another record that has the same value for this field. This means that, for a common value, the information content of encountering another record with the same value is low.

For the same very common value, it is surprising to find a value in another record which is different. This means that the information content of finding a value different from a very common value is high. For a rare value the converse is true.

The Demographic Clustering mining function uses this information content to weight the score *for* similarity using the *agreement* information content value. The score *against* similarity is weighted using the *disagreement* information content value.

The different weighting of the score for similarity and the score against similarity means that records with rare values are more likely to produce positive scores. Consequently, they are more likely to produce a new cluster than a more common value.

Information-theoretic weighting assigns a $-\log(p)$ value to the agreement information content value and a $-\log(1-p)$ value to the disagreement information content value, where p is the probability of a value.

Both types of value weighting look at a problem from different sides. Depending on the value distribution, it might lead to very different results if you use one type or the other.

Example: If a supermarket is located in a retirement community, the Senior discount field has a high probability of having a value of Yes. You might want to use probabilistic value weighting to assign a weight to the values in the Senior discount field that is equal to its probability in the input data. See Table 39 on page 160.

Value weighting has the additional effect of emphasizing fields with many values because their values are less frequent than the values of fields with fewer possible values. By default, the mining function does not compensate for this additional effect.

You can select whether you want to compensate for the value weighting applied to each field. If you compensate for value weighting, the overall importance of the weighted field is equal to that of an unweighted field, regardless of the number of possible values. Compensated weighting affects only the relative importance of coincidences within the set of possible values.

To do so, select **Yes** from the **Compensate** drop-down list for the appropriate input fields. See Table 39.

Table 39. Applying field weighting and value weighting with compensation switched on

Active fields	Field weighting	Value weighting	Compensate
Number of purchases	0.5	None	No
Day of purchase	1.0	None	No
Customer age	1.0	None	No
Senior discount	1.0	Probabilistic	Yes
Total purchase amount	0.5	None	No

Specifying additional field parameters

You can define the unit and the distance factor for numeric data. When two records are compared, the absolute difference between the field values is compared with the value of the unit. Differences less than the value of the unit favor the inclusion of a record in a cluster, whereas differences greater than this value favor the exclusion of this record. The similarity measure varies from 0 to 1, where near to 0 indicates values far apart, and 1 indicates identical values. A similarity measure of 0.5 reflects values separated by one distance measure.

You can specify whether you want the distance factor to be interpreted as an absolute number, as a multiple of the standard deviation, or as a multiple of the field range.

Example: You might want to use the units and distance factors in Table 40 on page 161 for the selected active fields.

Table 40. Sample units and distance factors

Active fields	Units	Distance factor
Number of purchases	Absolute number	3.00
Customer age	Absolute number	6.00
Total purchase amount	Standard deviation	0.40

To keep your settings compatible with versions 1 and 2 of the Intelligent Miner, specify -1 if you want to use the default value of 0.5 standard deviations for the distance factor.

Tip: To obtain a larger number of clusters, decrease the mean similarity between pairs of clusters by smaller distance factors for numeric fields.

Treating outliers

The Demographic Clustering mining function detects outliers in continuous numeric fields. Outliers are values that lie beyond the scope of a field's value range.

The values of continuous numeric fields are assigned to buckets. Each bucket represents a range of values. All buckets together make up the value range of the field. The minimum of that value range is the lower limit of the bucket covering the lowest values; the maximum is the upper limit of the bucket covering the highest values. Outliers are the values that are either less than the minimum or greater than the maximum of that value range.

You can determine how outliers are treated by specifying one of the following options on the Outlier treatment page of a Demographic Clustering wizard or settings notebook:

Create lower and upper buckets until outlier is accommodated
Buckets are added to both ends of the value range until all outliers are contained in a bucket. The bucket size doubles each time another bucket is added.

Place outliers into lower and upper buckets
A single bucket is added to both ends of the value range to contain the outliers.

Replace outlier with MIN or MAX
Any outlier less than the minimum value is replaced with the minimum value, and any outlier greater than the maximum value is replaced with the maximum value.

Treat outliers as missing values
Outliers are disregarded during a clustering run. This is the default.

Specifying discrete similarities

You can use value mapping objects to define symmetric similarities for value pairs in discrete fields. In each line, the value mapping contains two values and their similarity. During the clustering process, the specified similarity is used for these pairs of values.

You can specify the similarity for each pair of possible values. One pair of possible values can be used only once. The similarity of the inverse pair is the same.

Example: If a pair occurs more than once in a value mapping, the similarity value of only one pair is used for the calculation. The similarity value must be between 0 and 1.

- 0 means completely different.
- 1 means identical.

Specifying mode parameters

The following sections discuss the parameters you can specify on the Mode parameters wizard page.

Clustering mode

In clustering mode, the function attempts to form an optimum clustering and creates clusters up to the specified number of clusters.

You can reduce run time if you select a results object in addition to a data object as input. The Intelligent Miner then uses the statistics of the selected results object, and does not have to calculate new statistics. If you specify a results object in clustering mode, the clustering is based only on the statistical information derived from this results object.

You can use previous results of the Demographic Clustering or Bivariate Statistics function, if these results contain the same fields.

When using previous results as input, you normally select a results object that was generated by the same input data. If you use results generated by different input data, the computation of similarities is based on the statistics of these results. It might be useful to use the statistics of a large input data source if you want to cluster a subset of that data.

By adjusting the values of the parameters discussed in the following sections, you can control the processing time and the accuracy improvement of a clustering run.

Maximum passes

You can limit the number of passes to reduce the processing time for clustering; however, this might reduce the overall accuracy. The Demographic Clustering mining function stops processing after the specified number of passes even if the desired accuracy improvement is not reached.

In contrast, specifying more passes through the data might improve the quality of the clustering model. Two or three passes are usually sufficient. More than three or four passes are rarely required. More passes might be needed if the order of the input records correlates with the clusters.

Maximum clusters

You can control the number of clusters to be created during a clustering run by specifying a value for the **Maximum clusters** field.

If the mining function cannot continue to create clusters because this limit is reached, the accuracy of the model cannot improve as much as without this limitation.

Limiting the number of clusters avoids that many small clusters are produced, and thus saves run time. You usually limit the number of clusters to obtain an overview.

Increasing the number of clusters improves the likelihood of finding niches.

Accuracy improvement

You can limit the number of passes and the processing time of the mining function by specifying a value for the accuracy improvement. For example, if you specify a value of 5, the iteration process ends when the quality improvement between two passes is less than 5%.

The percentage of improvement is measured at each pass over the data. The accuracy value is used as a stopping criterion. If the actual improvement is less than the value specified, then no more passes occur. The smaller the value, the more accurate is the clustering.

Similarity threshold

The similarity threshold limits the values accepted as best fit for a cluster. For example, if you set the similarity threshold to 0.25, records with 25% identical field values are likely to be assigned to the same cluster.

To obtain a larger number of clusters, increase the value for the similarity threshold.

Example

You might want to select clustering mode using the values shown in Table 41 on page 164.

Table 41. Specifying parameters for the clustering mode

Maximum passes	Maximum clusters	Accuracy improvement	Similarity threshold
2	9	10	0.5

Tip: To obtain a larger number of clusters, you can try value weighting, especially for fields with an unbalanced distribution of possible values.

Application mode

In application mode, the mining function assigns cluster IDs to data records. This applies a model which was built in clustering mode.

Important: The Intelligent Miner treats blank strings in database tables as valid values and blank strings in flat files as missing values.

Therefore, if you discovered clusters in a database table containing records with blank strings (clustering mode) and later want to see into which cluster each record in a flat file fits (application mode), you must preprocess your input data. Map blank strings in your database tables to `NULL` before you start the function in clustering mode. This ensures that blank strings are treated as missing values in both modes. Use the Map Values preprocessing function for this purpose. If you do not prepare your data in this way, you inevitably get wrong results.

Specifying parallel parameters

You can run the Demographic Clustering mining function in serial or parallel mode. See "Specifying parallel parameters" on page 118 for more information.

On the Parallel parameters page of a Demographic Clustering wizard or settings notebook, you can specify values for the following parameters:

Merge model after this many records
When running the Demographic Clustering function in parallel mode, each processing node manages a separate model. To determine the intervals after which these models merge together, enter a value in this field. The models merge every time the specified number of records was processed.

Records to use to build initial model
In this field, enter the number of records that you want the initial model to be based on.

Records per data stripe

In this field, you enter the value that was used for the parameter *nstripe* while the input data was partitioned. The default value is 1. Do not change the default value unless you specified a different value for *nstripe*.

Example: To change the default values for an additional parallel parameter on the Parallel parameters page:

1. Uncheck the **Use default** box if necessary.
2. Enter a value in the entry field for the parameters or select a value by using the spin buttons.

For example, you might use the values listed in Table 42.

Table 42. Sample values for the additional parallel parameters

Merge model after this many records	Records to use to build initial model	Records per data stripe
15000	2000	1

Specifying output fields

From the **Available fields** list on the Output fields page, you can select input fields to be included in the output data. See "Specifying output settings" on page 125 for more information.

In addition, you must specify a valid name in the **Cluster ID field name** entry field. The output data will contain a field of this name. The values in this field will be the identifier of the best fitting cluster for the corresponding input record.

If you specify valid names for the following entry fields, these fields are also included in the output data:

Record score field name
This field will contain values for the fitting quality of the corresponding input records with regard to the best fitting cluster.

Cluster ID field name: choice 2
This field will contain values that identify the second best fitting cluster for the input records.

Record score field name: choice 2
This field will contain values for the fitting quality of the corresponding input records with regard to the second best fitting cluster.

Confidence field name

This field will contain the confidence values that were calculated when the input records were assigned to clusters.

Tip: It is useful to select a key field that helps you identify the record in the table.

Example: You might want to include the data field Customer number in the output data and the optional **Record score field name** field in addition to the required **Cluster ID field name** field. Select **Customer number** from the **Available fields** list and type a name in the other fields as shown in the example in Table 43.

Table 43. Example of output field specification for the Demographic Clustering mining function

Output fields	Cluster ID field name	Record score field name
Customer number	Best fitting cluster	Score of best fit

Specifying output data

If you specified output fields, you must also specify a data settings object on the Output data page of the wizard or settings notebook. See "Specifying output settings" on page 125 for more information.

Specifying result names

By default, a result object takes on the name of the settings object that created the results. Thus, you do not need to specify a result name when you run the settings object for the first time.

However, you might want to run the settings object again and keep the old results. In this case, you must change the original result name. See "Renaming result objects" on page 128 for more information.

Chapter 11. The Neural Clustering mining function

The purpose of discovering clusters is to group records that have similar characteristics.

The Intelligent Miner searches the mining base for characteristics that most frequently occur in common, and groups the related records accordingly. The results of the clustering function show the number of detected clusters and the characteristics that make up each cluster. In addition, the results show how these characteristics are distributed within the clusters.

Suppose that you have a database of a supermarket that includes customer identification and information about the date and time of the purchases. The clustering mining function clusters this data to enable the identification of different types of shoppers. For example, this might reveal that customers buy many articles on Fridays and usually pay by credit card.

Neural Clustering employs a *Kohonen Feature Map neural network*. Kohonen Feature Maps use a process called self-organization to group similar input records together. The user specifies the number of clusters and the maximum number of passes through the data. These parameters control the processing time and the degree of granularity used when data records are assigned to clusters.

The main task of neural clustering is to find a center for each cluster. This center is also called the cluster prototype. For each record in the input data, the Neural Clustering mining function computes the cluster prototype that is closest to the record.

The score of each data record is represented by the Euclidean distance from the cluster prototype. Scores closer to zero have a higher degree of similarity to the cluster prototype. The higher the score, the more dissimilar the record is from the cluster prototype.

With each pass over the input data, the centers are adjusted so that a better quality of the overall clustering model is reached. The progress indicator shows the quality improvements at each pass while the mining function is running.

The following sections describe the basic and advanced parameters of the Neural Clustering function. To find all parameter controls on the wizard or notebook pages, you must check the **Show the advanced pages and controls** box on the settings page.

Sample parameter settings

Table 44 shows sample settings summarizing the basic and advanced parameters of the Neural Clustering mining function. In addition, it indicates on which wizard page you must specify the required and optional parameters.

Table 44. Sample parameter settings for the Neural Clustering mining function

Wizard page	Parameter	Value
Settings	Name	Sample settings
	Comment	Data collected during 1st half of 1997
	Mining function	Clustering – Neural
Input data	Input data	Quality Food supermarket
	Optimize mining run for	Time
	Filter records	Conditions selected
	Power options	
Mode parameters	Use mode	Clustering mode
	Maximum passes	7
	Maximum rows	5
	Maximum columns	1
	Regardless of the mode, normalize the input data	True
Input fields	Active fields	6 selected
	Supplementary fields	3 selected
Outlier treatment	Outlier treatment	Treat outliers as missing values
Parallel parameters	Run the parallel mode of the function	Use this number of parallel processes: 4
Output fields	Output fields	1 selected
	Cluster ID field name	Cluster ID
	Record score field name	
	Cluster ID field name choice 2	
	Record score field name choice 2	
	Confidence field name	

Table 44. Sample parameter settings for the Neural Clustering mining function (continued)

Wizard page	Parameter	Value
Output data	Output data	Clusters Quality Food supermarket
Results	Results name	Result Quality food supermarket
	Comment	May transactions
	If a result with this name exists, overwrite it	False

You can use the settings individually or as part of a sequence.

Specifying input data

To use the Neural Clustering mining function, select an input data object on the Input data page of a wizard or settings notebook. See "Selecting an input data object" on page 114 for more information.

Example: The sample data in Table 45 is suitable for identifying various types of shoppers.

Table 45. Sample data for the Neural Clustering mining function

Customer number	Number of articles	Day of the week	Customer age	Gender	Marital status	Senior discount	Form of payment	Total purchase amount
123	2	05/02/97	65	Male	Married	Yes	Cash	$9.67
196	24	05/03/97	65	Male	Single	Yes	Credit card	$45.30
234	4	05/05/97	72	Female	Married	Yes	Cash	$12.50
308	19	05/06/97	72	Male	Married	Yes	Check	$67.09
456	2	05/07/97	58	Female	Single	No	Cash	$2.48

Optimizing the mining run

Use the **Optimize for time** option when running the Neural Clustering function. This option reduces the processing time significantly because all neural functions require more passes over the training data than the other mining functions. See "Optimizing the mining run" on page 114 for more information.

Filtering records

You can restrict the mining run to records with certain field values. To do so, specify a filter records condition. See "Filtering records" on page 115 for more information.

Specifying input fields

To partition a database so that records that have similar characteristics are grouped together, you must specify active fields. You can also specify supplementary fields.

The active fields are used by the mining function for clustering. The supplementary fields are used to gain statistical information on the clusters that are found. They are not used for clustering, but in the clustering results viewer they appear as parts of the clusters. The fields in a cluster are ordered by importance. It is therefore possible that you see supplementary fields among the active fields in a cluster. This means that the supplementary fields would have influenced the creation of the cluster, perhaps more than the fields you specified as active fields.

Example: You might want to identify different shopper types and create clusters based on the number of articles purchased, the total purchase amount, the day of the purchase, and several characteristics of the customer. You might also want to have statistical information on the form of payment, gender, and marital status. Table 46 shows the required input fields, in accordance with the sample data.

Table 46. Selected input fields for the Neural Clustering mining function

Active fields	**Supplementary fields**
Number of articles	Form of payment
Total purchase amount	Gender
Day of the week	Marital status
Customer age	
Senior discount	

Tips: When selecting active fields, consider the following:

- In the **Available fields** list, you can sort the fields in increasing or decreasing order. Click the right mouse button and move the pointer over the displayed selection field.
- Do not specify active fields that have the following characteristics:
  - Categorical fields with different values for almost every record
  - The same value in every record

  These fields do not contain any valuable information for building clusters.
- If you have several fields in your input data containing very similar data, use the Principal Component Analysis or Factor Analyis functions before you run the clustering function. Principal Component Analysis and Factor Analysis amalgamate the similar fields to a single component or factor.

  When you run the clustering function, specify the components or factors as active fields, and the original fields as supplementary fields. This way, you can view the results of the clustering run plus the original field values.

Specifying mode parameters

The following sections discuss the parameters you can specify on the Mode parameters wizard page.

Clustering mode

The following sections describe the parameters that you can specify in clustering mode.

Maximum passes

Specifying multiple passes through the data improves the quality of the generated clusters. Limiting the number of passes reduces the processing time required to perform clustering, however, it also reduces the accuracy of the clustering. 5 to 10 maximum passes are usually sufficient.

You can reduce the number of passes through the data if you select a results object in addition to a data object as input. The Intelligent Miner then uses the clustering statistics of the selected results object, and does not have to calculate new statistics. If you specify a results object in clustering mode, the clustering is based only on the statistical information derived from this results object.

Maximum clusters

You can change the number of clusters to be generated by the Neural Clustering function by altering the value in the **Maximum clusters** field. When using the advanced pages and controls, this field is replaced with the **Maximum rows** and **Maximum columns** fields.

Maximum rows and maximum columns

In clustering mode, the mining function creates the specified number of clusters in the form of a rectangular grid. The number of *maximum rows* is one dimension of the grid, the number of *maximum columns* is the other dimension of the grid. The grid is equal to the largest number of rows and less than or equal to the maximum number of columns provided.

When using the advanced pages and controls, you can determine the maximum number of rows and the maximum number of columns instead of specifying the number of clusters. Thus you can influence the structure of the cluster grid.

Example

You might want to use the clusters of the February transactions of the Quality Food supermarket for the discretization of input data. Therefore you do not want a square grid. The specification looks like this:

Table 47. Specifying advanced parameters for the clustering mode

Result	Maximum passes	Maximum rows	Maximum columns
Quality food clusters February transactions	7	1	5

Application mode

In application mode, the function assigns new data to clusters. This assignment is based on the model created in clustering mode.

Important:

The Intelligent Miner treats blank strings in database tables as valid values and blank strings in flat files as missing values.

Therefore, if you discovered clusters in a database table containing records with blank strings (clustering mode) and later want to see whether records from a flat file fit into these clusters (application mode), you must preprocess your input data. Map blank strings in your database tables to `NULL` before you start the function in clustering mode. This ensures that blank strings are treated as missing values in both modes. Use the Map Values preprocessing function for this purpose. If you do not prepare your data in this way, you inevitably get wrong results.

Normalizing the input data

The input data must be normalized or scaled to a range of 0.0 to 1.0. In addition, categorical values must be converted into a numeric code for presentation to the neural network.

If you choose to normalize the input data, the Intelligent Miner scales continuous and discrete numeric fields to a range of 0.0 to 1.0 and converts categorical data into 1-of-*N* vectors. For a categorical field with *N* different string values, an input string is converted into a unique index *i*. In the internal input vector of the size *N*, a single 1-value is placed in the position of the string index value, and 0-values are placed in all other positions of the vector. This means that a categorical value with a large number of discrete values causes an expansion in the number of input units to the neural network.

Continuous and discrete numeric fields are converted to a 0.0 to 1.0 range using a stepwise linear algorithm. The mean of the value is mapped to 0.5. Input values from minimum to mean consequently range from 0.0 to 0.5 after scaling, while input values from mean to maximum range from 0.5 to 1.0. The scaling algorithm expands or normalizes the distribution of continuous variables which have non-normal distributions. For some types of input fields, this scaling results in a loss of information.

In such cases, it might be necessary to use a virtual computed field to perform special processing of the data. You also have the option of normalizing and scaling the data by yourself, and not to normalize the input data. In this case, the data is presented directly to the neural network.

Example: You might have input data with four input fields that are all continuous fields. The neural network would have four input units. However, if one of the four fields is discrete and represents the state the customer lives in, it has 50 possible values. This one field would expand into a 1-of-50 vector. The neural network (with four input fields) would have 53 input units (3 for the three continuous fields and 50 for the single discrete field).

If your data set has several categorical fields of this type, you could be unknowingly generating very large neural networks. This might negatively impact the training time. Possible solutions are to use the Intelligent Miner preprocessing functions to reduce the number of discrete values, to convert the field data type to discrete numeric, or to use computed fields with value mappings or computed field functions.

Tips

The following section contains useful background information about the Neural Clustering mining function. You can use this information to better understand the algorithm behind the function or to adjust your parameter settings.

Specifying the number of clusters

The Neural Clustering function uses a batch version of the Kohonen self-organizing map. When you select the number of clusters, you are defining the architecture of the output layer of the network. For example, selecting nine clusters results in an output architecture of a 3 by 3 square. In advanced mode, you can explicitly set the number of rows and columns to form any rectangular configuration.

Number of passes
Neural Clustering requires at least 5, preferably 10 or more passes in order to cluster a data set.

Number of clusters found
There is not necessarily a direct relationship between the number of clusters you specify for the network architecture and the number of clusters that are found in the data. For example, if you use the default settings, 9 output units (clusters) are allocated in the neural network. However, during training, the network might determine that there are only 3 significant data clusters. When you view the results in the clustering visualizer, you can see that these 3 clusters account for a large percentage of the population, while the remaining clusters make up a small percentage. This is normal behavior for neural clustering. If you want to get a finer granularity, you might want to create 16 outputs. Some of the records that fell into the same cluster in the first case are now distributed across adjacent clusters in the second case.

Serial mode compared with parallel mode
The parallel version of the neural clustering function uses the same batch algorithm as the serial version. Hence, results obtained with a single processor in parallel mode should agree with the serial results. However, results for problems with categorical input fields might differ when using different numbers of parallel processors because of differences in the precise 1-to-N mapping used to expand these fields. Parallel speedup should improve with increasingly larger input data sets.

Treating outliers

For the two types of numeric fields, the Neural Clustering function recognizes outliers, which are extreme values lying outside of a "normal" value range.

For discrete numeric fields, this normal value range is a set of values. By default, this set contains all valid values from the input data. You can override this default by defining value sets of your own for the fields in the input data.

The normal value range for continuous numeric fields consists of a set of contiguous bucket ranges. By default, these buckets are centered around the mean with a maximum distance of two standard deviations in both directions. Again, you can change the default by defining bucket ranges of your own.

For both types of numeric fields, the normal value range has a minimum and a maximum. Values less than the minimum and greater than the maximum are called outliers. Selecting one of the following options determines the way in which outliers are treated:

Treat outliers as missing values
: When you select this option, any outlier value is mapped to the scaled value of 0.5 for the corresponding neural input unit.

Replace outlier with MIN or MAX
: When you select this option, any outlier less than the minimum is replaced with the minimum value and any outlier greater than the maximum is replaced with the maximum value.

Treat outliers as valid values
: When you select this option, any outlier is treated as if it belongs to the normal value range, that is, outliers are not treated differently.

Tip: You can often transform an input field with a skewed distribution of values into a field with a normal distribution by defining a computed field. This helps you control the effects of outliers. If a field *f* has a distribution of an exponential kind, define a computed field as log(*f*). Select the computed field as an active input field and the original field *f* as a supplementary field.

Specifying parallel parameters

You can run the Neural Clustering mining function in serial or parallel mode. See “Specifying parallel parameters” on page 118 for more information.

Specifying output fields

From the **Available fields** list on the Output fields page, you can select input fields to be included in the output data. See “Specifying output settings” on page 125 for more information.

In addition, you must specify a valid name in the **Cluster ID field name** entry field. The output data will contain a field of this name. The values in this field will be the identifier of the best fitting cluster for the corresponding input record.

If you specify valid names for the following entry fields, these fields are also included in the output data:

Record score field name
: The values in this field will be the fitting quality of the corresponding input record to the best fitting cluster.

Cluster ID field name: choice 2
The values in this field will be the identifier of the second best fitting cluster for the corresponding input record.

Record score field name: choice 2
The values in this field will be the fitting quality of the corresponding input record to the second best fitting cluster.

Confidence field name
The values in this field will be the confidence of a record' assignment.

When you specify any of these optional fields, it is useful to select a key field that helps you identify the record in the table.

Example: You might want to include the data field Customer number in the output data and the optional **Record score field name** field in addition to the required **Cluster ID field name** field. Select **Customer number** from the **Available fields** list and type a name in the other fields as shown in the example in Table 48.

Table 48. Example of output field specification for the Neural Clustering mining function

Output fields	Cluster ID field name	Record score field name
Customer number	Best fitting cluster	Score of best fit

Specifying output data

If you specified output fields, you must also specify a data settings object on the Output data page of the wizard or settings notebook. See "Specifying output settings" on page 125 for more information.

Specifying result names

By default, a result object takes on the name of the settings object that created the results. Thus, you do not need to specify a result name when you run the settings object for the first time.

However, you might want to run the settings object again and keep the old results. In this case, you must change the original result name. See "Renaming result objects" on page 128 for more information.

Chapter 12. The Sequential Patterns mining function

The purpose of discovering sequential patterns is to find predictable patterns of behavior over a period of time. This means that a certain behavior at a given time is likely to produce another behavior or a sequence of behaviors within a certain timespan.

For example, you might find that 42% of new checking account customers who apply for an ATM card will also apply for a charge account within 90 days.

You might use this mining function to detect insurance fraud, to do item-placement planning, or to plan promotional sale.

The following sections describe the basic and advanced parameters of the Sequential Patterns function. To find all parameter controls on the wizard or notebook pages, you must check the **Show the advanced pages and controls** box on the settings page.

Sample parameter settings

Table 49 shows sample settings summarizing the basic and advanced parameters of the Sequential Patterns mining function. In addition, it indicates on which wizard page you must specify the required and optional parameters:

Table 49. Sample parameter settings for the Sequential Patterns mining function

Wizard page	Parameter	Value
Settings	Name	Sample settings
	Comment	Data collected during 1st half of 1997
	Mining function	Sequential Patterns
Input data	Input data	Sunset retail June transactions
	Optimize the mining run for	Time
	Filter records	Conditions selected
	Power options	
Input fields	Transaction group field	Customer ID

Table 49. Sample parameter settings for the Sequential Patterns mining function *(continued)*

Wizard page	Parameter	Value
	Transaction field	Date and transaction number
	Item field	Item ID
	Sort the input data on the value in the Transaction ID field before running this function	False
Parameters	Minimum support	5
	Maximum pattern length	3
	Item constraints	Selected
Parallel parameters	Run the parallel mode of this function	Use this number of parallel processes: 4
Taxonomy	Taxonomy name	Sunset non-food
Results	Results name	Sunset June sequential patterns
	Comment	Branch locations New York City
	If a result with this name exists, overwrite it	False

You can use the settings individually or as part of a sequence.

Specifying input data

To use the Sequential Patterns mining function, select an input data object on the Input data page of a wizard or settings notebook. See "Selecting an input data object" on page 114 for more information.

Example:

You might want to look for sequential patterns in the transactions of one of the Sunset retail branches in New York City in June.

A transaction might consist of several records. In this example, customer number 123 buys a T-shirt and tennis shoes. This is reflected by two records. However, both purchases belong to the same transaction because the date and the transaction number are the same. In contrast, customer number 487 also purchased three items on the same day, but only the first two items belong to the same transaction, while the third item belongs to a different transaction. This is reflected in the different transaction number.

Because the Sequential Patterns mining function searches for intertransaction patterns, transaction groups are used. A transaction group consists of several transactions. The customer number represents the transaction group. The mining function searches for sequences of items of different transactions per customer. In this example, customer number 487 supports the following sequence:

(Swimsuit)	(Sunglasses)
(Beach towel)	(Sunglasses)
(Swimsuit, beach towel)	(Sunglasses)

Ensure that the data is sorted like this:
1. By the transaction group identifier
2. By the transaction identifier

As shown in Table 50, the customer number represents the transaction group, and the date and the transaction number represent the transaction per customer.

Table 50. Customer transactions

Customer number	Item	Date	Transaction number	Type of sale	Customer age	Payment	Total purchase amount
123	T-shirt	06/02/97	2215	On sale	16	Cash	$10.50
123	Tennis shoes	06/02/97	2215	Regular	16	Cash	$98.00
148	Twin set	06/03/97	2437	On sale	21	Credit card	$35.50
168	Costume	06/04/97	2437	Regular	35	Credit card	$212.50
168	Jeans	06/05/97	2437	Regular	35	Credit card	$49.50
321	Suit	06/06/97	2820	Regular	45	Cash	$225.00
487	Swimsuit	06/07/97	3046	On sale	22	Credit card	$49.80
487	Beach towel	06/07/97	3046	Regular	22	Credit card	$25.00
487	Sun-glasses	06/09/97	4217	Regular	22	Credit card	$30.75

In this example, the item numbers in the **Item** field are mapped to the item names. You can also use name mappings for the customer IDs.

A mining run for sequential patterns on this data returns frequently occurring patterns of products bought over a period of time by the same customer. For example, if customers buy swimsuits and beach towels, in 70% of the cases, they buy sunglasses or other beachwear during their next visits to the store.

All purchase transactions of one customer are a sequence (customer sequence). Each transaction includes an item set. A customer supports a sequence *s* if *s* is contained in the customer sequence of this customer.

Optimizing the mining run

You might want to optimize the mining run for time or disk space. See "Optimizing the mining run" on page 114 for more information.

Filtering records

You can restrict the mining run to records with certain field values. To do so, specify a filter records condition. See "Filtering records" on page 115 for more information.

Specifying input fields

A mining run for sequential patterns returns the names or types of products that were bought by different customers over a given period of time. The names or types of products are displayed in the order in which they were bought. All transactions of a customer represent a customer sequence. Each transaction includes a set of items bought by a customer. A customer supports a sequence *s* if *s* is contained in the customer sequence of this customer.

You can select to perform a two-level sort on the input data, first by Transaction group fields, then by Transaction fields. The data in the Transaction group field must be sorted sequentially. If you are not sure that this data is sorted, select the **Sort** option.

Restriction:

This information applies to the IBM DB2 Intelligent Miner for Data for OS/390 only. If your input data resides in a database table, the DB2 data type of the fields you select as**Transaction group field** and **Transaction field** must be *CHAR* or *VARCHAR*. In addition, the Intelligent Miner data type of these fields must be *Categorical*. Otherwise you might get incorrect results because the data in these fields cannot be sorted properly.

To find customer sequences that indicate the presence of sequential patterns, specify the following fields:

Transaction group field
The **Transaction group field** represents any unit in a data table that comprises several transactions, for example, a customer number.

Transaction field
The **Transaction field** represents any unit in the data table that comprises several items, for example, the date or the time stamp of a purchase transaction.

Item field
The **Item field** represents the individual item of a transaction, for example, one of the articles of a purchase transaction.

Example: To discover sequential patterns in the transactions of the Sunset retail stores in New York City, you might specify data fields as shown in Table 51.

Table 51. Specifying input fields for the Sequential Patterns mining function

Transaction group field	Transaction field	Item field
Customer ID	Transaction number	Item ID

The **Available fields** list contains all data fields of the data tables you included in the input data. From this list, you can select the data fields to be mined.

Specifying parameters

You can specify parameters that control the results of the Sequential Patterns mining function.

Minimum support

The support factor indicates the relative occurrence of the detected sequential patterns within the input data. It is determined by dividing the number of customers supporting the sequence by the total number of customers.

Sequential patterns based on a high support factor represent a higher degree of relevance than sequential patterns with a low support factor. However, specifying high support factors may be the reason that no sequential patterns are discovered.

Messages: Depending on the specified support factor, one of the following messages might be displayed:
- No patterns found for given support and confidence

- Not enough memory

By default, the Intelligent Miner tries to determine a value for the minimum support. The minimum support is adjusted such that one quarter of all single items are frequent.

Memory limits: If the specified support value is too low, your system might run out of memory and no results are generated. You can increase either the value for minimum support or the memory limit. To change the memory limit, click **Options → Preferences** on the menu bar in the Intelligent Miner main window.

By default, 32 MB are used to generate sequential patterns. A large amount of patterns, for example, more than ten thousand, are generated before this limit is reached.

Maximum pattern length

The value for the maximum pattern length determines the number of items occurring in a sequential pattern. For example, if you specify 3 as maximum pattern length, you might receive the following pattern:

[Swimsuit]
[Beach towel]
[Sunglasses]

Item constraints

The specification of the item constraints determines which patterns are to be included in or excluded from the results. If you selected to include the specified items, only patterns containing at least one of the specified items are generated. If you selected to exclude the specified items, patterns containing one of the specified items are discarded from the results.

Example: You might want to view sequential patterns that include only the following items with a relative occurrence of 5% and a maximum pattern length of 3. Specify the values for the fields **Minimum support**, **Maximum pattern length**, and **Item constraints** as shown in Table 52.

Table 52. Specifying the minimum support factor, maximum pattern length, and item constraints

Minimum support	Maximum pattern length	Item constraints
5	3	Sunglasses; beach towels; shorts; swimsuits; caps

You can also use taxonomy names to specify item constraints. For example, if you created the taxonomy Beach wear, you can specify Beach wear in the **Item constraints** field to view sequential patterns containing items such as sunglasses, beach towels, or caps.

Tips:

The minimum support factor is a critical parameter. First, use the default system-determined minimum support factor to get an idea of a reasonable value. You can see the system-determined value in the Statistics window when you view the result.

If no patterns are found for the given support, you might want to decrease the value for minimum support.

If you set the minimum support value too low, you might get very large result files of more than 100 MB. In particular, there is a high probability to get large result files if you use a name mapping with very long names. You need sufficient disk space and memory to view the result on the client. Also, if you use a taxonomy, you usually get a large number of patterns in contrast to not using a taxonomy.

If you get the message Not enough memory, you might increase the minimum support value. You can also specify the memory limit beyond 32 MB in the Preferences notebook; however, this results in a large number of patterns. If you want to check what kind of patterns are created without increasing the minimum support value or the memory limit, you can limit the maximum pattern length.

Specifying parallel parameters

You can run the Sequential Patterns mining function in serial or parallel mode. See "Specifying parallel parameters" on page 118 for more information.

Specifying taxonomies

A taxonomy is a hierarchy or lattice of associations between different categories of an item.

You can categorize items in several categories. Then you establish relations between these categories in terms of child item categories and parent item categories. Each child category can have several parent item categories. Each parent item category can have several grandparent item categories. The more item categories you define, the more detailed are the results of the mining functions. See Figure 13 on page 150.

Specifying result names

By default, a result object takes on the name of the settings object that created the results. Thus, you do not need to specify a result name when you run the settings object for the first time.

However, you might want to run the settings object again and keep the old results. In this case, you must change the original result name. See “Renaming result objects” on page 128 for more information.

Chapter 13. The Similar Sequences mining function

The purpose of discovering similar sequences is to find all occurrences of similar subsequences in a database of sequences.

For example, take a database of a retailer who wants to optimize purchasing and storekeeping. A mining run on this database returns the names of sequence pairs with their degree of similarity and the number of subsequences. By interpreting this result, the retailer can find groups of products that have similar forecast seasonal sales for the next year. Based on this information, the retailer can combine purchases and inventory replenishment.

You can also use this technique to identify companies with similar patterns of growth, determine products with similar selling patterns, or determine stocks with similar price movements. Other uses include detecting seismic waves that are not similar, or spotting geological irregularities.

The following sections describe the basic and advanced parameters of the Similar Sequences function. To find all parameter controls on the wizard or notebook pages, you must check the **Show the advanced pages and controls** box on the settings page.

Sample parameter settings

Table 53 shows sample settings summarizing the basic and advanced parameters of the Similar Sequences mining function. In addition, it indicates on which wizard page you must specify the required and optional parameters.

Table 53. Sample parameter settings for the Similar Sequences mining function

Wizard page	Parameter	Value
Settings	Name	Sample settings
	Comment	Data collected during 1st half of 1997
	Mining function	Similar Sequences
Input data	Input data	Sunset retail store
	Optimize mining run for	Time
	Filter records	Conditions selected
	Power options	

Table 53. Sample parameter settings for the Similar Sequences mining function (continued)

Wizard page	Parameter	Value
Input fields	Sequence field	Beach wear
	Time field	Month
	Value field	Sales
Parameters	Epsilon	0.2
	Gap	8
	Window size	16
	Matching length	0.05
Results	Results name	Sunset sequences
	Comment	Branch location Chicago
	If a result with this name exists, overwrite it	False

You can use the settings individually or as part of a sequence.

Specifying input data

To use the Similar Sequences mining function, select an input data object on the Input data page of a wizard or settings notebook. See “Selecting an input data object” on page 114 for more information.

Example: You might want to find similar sequences in the data of the Sunset retail store in Chicago. The records of the input data must be of fixed length and arranged in the following columns, as shown in Table 54.

Table 54. Organization of input data for the Similar Sequences mining function

Sequence field	Time field	Value field
Beach wear female	1	9.7
Beach wear female	2	9.9
...	...	...
Beach wear female	12	8.5
Beach wear male	1	5.6
Beach wear male	2	5.8
...	...	...
Beach wear male	10	4.6

Other columns might be present in the data, but they are ignored.

A mining run on this database returns pairs of input sequences with similar subsequences of a minimal length (matching length) and the number of these subsequences. A graphical view of each sequence pair is also displayed.

Optimizing the mining run

You might want to optimize the mining run for time or disk space. See "Optimizing the mining run" on page 114 for more information.

Filtering records

You can restrict the mining run to records with certain field values. To do so, specify a filter records condition. See "Filtering records" on page 115 for more information.

Specifying input fields

Time-series data consists of values of one variable over a period of time. To find all occurrences of similar subsequences, you must specify a record with the sequence name for each value, the appropriate time unit, and the value.

Example: To find a subsequence of another sequence similar to the subsequence Beach wear during a year, you might specify the records shown in Table 55.

Table 55. Specifying input records for the Similar Sequences mining function

Sequence field	Time field	Value field
Beach wear	1	0.4
Beach wear	2	0.2
Beach wear	3	0.3
Beach wear	4	0.7
Beach wear	5	1.5
Beach wear	6	2.2
Beach wear	7	3.3
Beach wear	8	3.8
Beach wear	9	1.8
Beach wear	10	1.9
Beach wear	11	0.8
Beach wear	12	2.1

Requirement: The values in the **Time field** must be ordered and unique.

Restriction: If your input data resides in a database table, the DB2 data type of the field that you select as the **Sequence field** must be *CHAR* or *VARCHAR*. The Intelligent Miner data type of this field must be *Categorical*.

In addition, the DB2 data type of the field that you select as the **Time field** must be *FLOAT*. The Intelligent Miner data type of this field must be *Discrete numeric*.

If you use fields with different data types, you might get incorrect results because the data in this field cannot be sorted properly.

The **Available fields** list contains all data fields of the data tables you included in the input data. From this list, you can select the data fields to be mined.

Specifying parameters

Human beings are still better at gestalt-perception, for example, recognizing similar shapes or figures within pictures and drawings, than machines simulating this skill. To substitute the intuitive notion of similarity by simple mathematical notions is not easy. The Similar Sequences function approximates this goal for sequences with the parameters epsilon, window size, gap, and matching length.

These parameters consider the following facts:

- Finding similarities for two sets of data depends on how detailed the data is.
- You can find similarities only if irrelevant data such as errors or outliers are neglected.

Two subsequences of a pair of sequences are considered similar if they do not deviate from or exceed the values for the parameters that are illustrated in Figure 14 on page 189.

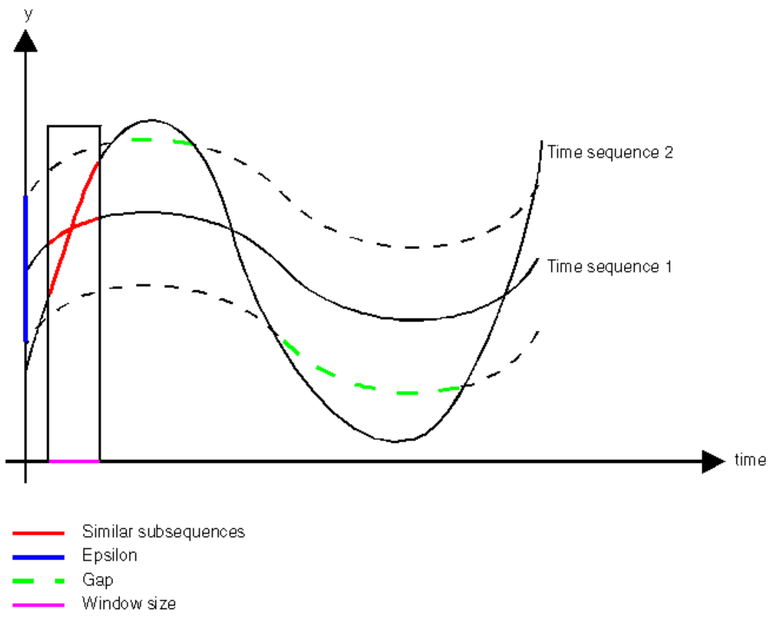


Figure 14. The parameters that define the rules for similarity

Epsilon
This value represents the relative tolerance range that one sequence allows another sequence to be different. One value is considered similar to the other if it is enclosed within an envelope of the width epsilon around the other plot. It is used after scaling the value ranges to intervals from -1.0 to 1.0. It must be a nonnegative floating-point number between 0.0 and 1.0.

If the epsilon value is too low, existing similar sequences may not be discovered. If the epsilon value is too high, you might discover sequences that do not look very similar if you compare their splines.

Gap This value represents the number of consecutive time units for which outliers are ignored. Outliers can fall outside the epsilon range of a sequence. They are ignored when similar sequences are determined. The gap value must be 0 or a positive integer.

If the gap value is too low, existing similar sequences may not be discovered because of noise or irregularities in the data. If the gap value is too high, you might discover sequences that do not look very similar if you compare their splines. In this case, errors and randomness of outliers have been overemphasized.

Window size
This value represents the atomic sequence for matching. Within this window size, outliers are not allowed. To find sequences longer than

the window size, the atomic sequences of a given window size are stitched together. A sequence is considered similar if it is positioned within the specified window size. The default value for **Window size** is 4.

Matching length

This value represents the minimal length of subsequences to be considered. It is the sum of the matching length of a subsequence divided by the total length of the complete sequence. The matching length value must be a positive integer between 0 and 1.

Two subsequences that are similar by the epsilon, the gap, and the window size are only displayed if their matching length is equal to or greater than the specified value.

If the matching length is too low, many similar sequences may be discovered. If the matching length is too high, less similar sequences may be discovered.

Example: To find similar sequences, you might specify the values shown in Table 56.

Table 56. Specifying parameters for the Similar Sequences mining function

Epsilon	Gap	Window size	Matching length
0.2	8	16	0.05

Tips:

To obtain good results, take two well-known and comprehensible input sequences, mark the similar parts, and adjust the parameters until you receive a comparable result. Save the settings object to use it for other data of the same type.

The parameters Epsilon, Gap, Window size, and Matching length are mutually dependent. For example, if you specify a small value for epsilon, the value for gap might be increased to get a comparable number of similar subsequences. Also, some combinations of parameter values might lead to very fast growing search space and might prevent a final result. Other combinations of parameter values might not be plausible, for example, an epsilon value of 0.0 means that you are looking only for identical input values (after scaling and perhaps accepting outliers).

You might want to specify high values for the following parameters:

Epsilon

To look at the rough structure of the spline

Gap If you are convinced that outliers are grouped together

Window size
If you think that subsequent data is connected inherently, for example, if data is rarely disturbed by outliers

Matching length
If you are interested only in very long similar subsequences compared to the input sequence

Example: You might want to use the Fund Sample Setting provided with the sample mining bases **imdemo** and modify only one parameter by considering the limitations shown in Table 57.

Table 57. Ranges for the parameters

Parameter	Range	Default
Epsilon	0.04 - 0.6	0.2
Gap	0 < 30	4
Window size	4 - 159	20
Matching length	0.01 - 0.8	0.01

Note that other data might require other limitations. Also, the concurrent change of two or more parameters might lead to different limitations.

An error message such as *Error writing to file <working path><tmp-file>* indicates that the file system is full and that it is not possible to write to the temporary result file. To solve this problem, you can choose one of the following options:

- Specify another work directory by clicking **Options → Preferences** on the menu bar in the Intelligent Miner main window.
- Modify the values for the parameters such that less similar subsequences are created. For example, epsilon values greater than, say, 0.7 create a huge number of similar subsequences in most of the cases. This leads to a large temporary result file. These files are about 1000 times larger in size than the final result. Note, however, that 0.7 is not an absolute limit. The number of similar subsequences strongly depends on the kind and the size of the input data, as well as on the values specified for the other parameters.

Specifying result names

By default, a result object takes on the name of the settings object that created the results. Thus, you do not need to specify a result name when you run the settings object for the first time.

However, you might want to run the settings object again and keep the old results. In this case, you must change the original result name. See “Renaming result objects” on page 128 for more information.

Chapter 14. The Tree Classification mining function

The purpose of predicting a classification is to create a model based on known data. You can use this model to analyze why a certain classification was made, or to calculate the classification of new data.

Historical data frequently consists of a set of values and a classification for these values. Analyzing data that was already classified reveals the characteristics that led to the previous classification. The resulting classification model can then be used to predict the classes of records containing new attribute values.

For example, an insurance firm has data about customers who allowed their insurance to lapse. Based on the common attributes of these customers, Intelligent Miner creates a risk group profile that is then used as a model for classifying new customers. Any new customer is checked against the model and classified as either belonging to the risk group or not belonging to the risk group.

- In training mode, a mining run on this database learns the attributes of each of the defined customer risk classes.
- In test mode, the insurer can test the accuracy of the model created in training mode by applying this model to test data with known customer risk classes.
- In application mode, the insurer can use the model created during the training mode to predict which customers let their insurance lapse in the future.

You can also apply this technique, for example, to approve or deny insurance claims, to detect credit-card fraud, to identify defects in images of manufactured parts, or to diagnose error conditions. Other applications are target marketing, medical diagnosis, medical treatment effectiveness, inventory replenishment, or store location planning.

The Tree-Induction Algorithm provides an easy-to-understand description of the underlying distribution of the data. This algorithm scales well regarding the number of training examples and the number of attributes in large databases. Use this technique to gain a deeper understanding of the structure of your database or to structure unclassified databases.

The following sections describe the basic and advanced parameters of the Tree Classification function. To find all parameter controls on the wizard or notebook pages, you must check the **Show the advanced pages and controls** box on the settings page.

Sample parameter settings

Table 58 shows sample settings summarizing the basic and advanced parameters of the Tree Classification mining function. In addition, it indicates on which wizard page you must specify the required and optional parameters.

Table 58. Sample parameter settings for the Tree Classification mining function

Wizard page	Parameter	Value
Settings	Name	Sample settings
	Comment	Data collected during 1st half of 1997
	Mining function	Classification – Tree
Input data	Input data	Insurance data Security First
	Optimize mining run for	Disk space
	Filter records	Conditions selected
	Power options	
Mode parameters	Use mode	Training mode
	Maximum tree depth	5
	Maximum purity per internal node	90
	Minimum records per internal node	7
	Classify result	None selected
Input fields	Input fields	13 selected
	Class labels	Risk class
Field parameters	Field weights	3 selected
Error matrix	Matrix	
Parallel parameters	Run the parallel mode of the function	Use 4 parallel processes
Output fields	Output fields	3 selected

Table 58. Sample parameter settings for the Tree Classification mining function (continued)

Wizard page	Parameter	Value
	Class ID field name	Predicted risk class
	Confidence field name	
Output data	Output data	Output Security First risk classes
Results	Results name	Result Security First risk classes
	Comment	Customers who allowed their insurance to lapse
	If a result with this name exists, overwrite it	False

You can use the settings individually or as part of a sequence.

Specifying input data

To use the Tree Classification mining function, select an input data object on the Input data page of a wizard or settings notebook. See "Selecting an input data object" on page 114 for more information.

Example: You might want to predict a classification using the input data of the insurance firm Security First. The input data has the attributes Age, Salary, Marital status, and a class field called Risk class. It is organized as in Table 59.

Table 59. Selected input data for the Tree Classification mining function

Age	Salary	Marital status	Risk class
30	65	Single	Low
23	115	Single	High
40	75	Married	Low
55	40	Married	High
55	100	Widowed	Safe
45	60	Divorced	Low

A mining run in training mode on this data discovers the relationships between the attributes and the **Risk class** field.

Optimizing the mining run

The Tree Classification mining function always optimizes a mining run for time. Optimization for disk space is not available. You must consider the amount of disk space available on your Intelligent Miner server. See the information about the **Optimize mining run for time** option in “Optimizing the mining run” on page 114 for more information.

Filtering records

You can restrict the mining run to records with certain field values. To do so, specify a filter records condition. See “Filtering records” on page 115 for more information.

Using power options with tree classification

You might want to specify a power option to include the confidence value in the output data.

This function creates output data with a layout corresponding to the appropriate input data. By default, it includes a column for the predicted class for a record. Additionally, you can include another column for the confidence value by specifying a field name for this column. The confidence value ranges from 0.0 to 1.0 . It indicates the probability of a correct classification.

Example: To include the probability of the class value *Safe* for all records, even if a different class value was predicted for some records, enter the following expression in the **Power options** field on the Input data page of a Tree Classification wizard or settings notebook:

```
-confidenceOnlyforClass Safe
```

If your field has a leading blank, enter `-confidenceOnlyforClass " Safe"`.

You can sort an output table like this by the confidence value and use it for gains charts or for selection as in mailing campaigns.

Attention: Use power options with great care. Make sure you read the general introduction in “Using power options” on page 117.

Specifying input fields

To discover a classification, you must specify the input fields to be classified and the class label. The class label represents the particular classification based on the attributes and values that contributed to this classification. The input fields are used by the mining function. The **Class labels** field contains the value to be classified by the mining functions.

Example: To identify customers who have allowed their insurance to lapse, you might specify the data fields shown in Table 60 on page 197.

Table 60. Selected input fields for the Tree Classification mining function

Input fields	Class label
Town districts Country Profession	Risk class

The **Available fields** list contains all data fields of the data tables you included in the input data. From this list you can select the data fields to be classified.

Tip: Fields whose values are constant or nearly constant do not add any value to your classification model. In fact, there is nothing to classify, so they are just sent down one of the branches. If you have constant fields among your input fields, the processing time is prolonged unnecessarily and the model becomes less precise.

To get rid of constant or almost constant fields in your input data, run the Factor Analysis function on your input data before you apply the Tree Classification mining function. The Factor Analysis function eliminates the constant fields; they do not become part of the factors.

You then use the generated factors as input fields for the Tree Classification mining function. This improves your classification model and reduces the processing time significantly. In fact, the processing time decreases linearly by the time it would have taken to process the number of fields that exceed the number of factors. For example, if you can reduce 100 input variables to 30 factors, processing takes only 30% of the time necessary to process all input variables.

Specifying field parameters

In the classification process, all fields have an equal weight. This might not always be desirable. You might want to specify different field weights for the input fields to give more or less weight to certain fields. The specified field weights determine whether these fields are included in the binary decision tree. When you specify field weights, the results are less accurate.

Taking the temperature of a patient can be considered as cheap medical treatment with low risk in contrast to taking an x-ray, which is expensive with high risk. Therefore you might want the binary decision tree to prefer taking the temperature instead of taking an x-ray.

Example: To determine the risk class of insurance customers, you might want the binary decision tree to use the attribute salary instead of marital status.

See Table 61.

Table 61. Specifying field weights

Input fields	Field weight
Age	1.0
Salary	1.0
Marital status	0.5

Error weighting

By default, all classifications have an equal weight. This might not always be useful. Asymmetric error weighting gives more or less weight to certain misclassifications.

You can assign different weights to misclassifications such as not to give too much weight to the underlying factor. A simple expedient gives a smaller weight to each of these classifications.

You can assign error weights to misclassifications by specifying a value mapping of the type 2-1. This means, a particular error weight is specified for the actual value and the predicted value.

Example: Table 62 shows error weights that you might want to assign if High risk is classified as Safe, if Low risk is classified as Safe, or if Safe is classified as Low risk.

Table 62. Specifying error weights

Arguments	Error weight
High risk;Safe	7.0
Low risk;Safe	3.0
Safe;High risk	0.3

To specify a value mapping for error weighting, select a value mapping in the **Available value mappings** list on the Error matrix page of the Tree Classification wizard or settings notebook.

The **Available value mappings** list contains only value mappings of the type 2-1. To create a new value mapping of this type, click the **Create value mapping** icon. This action takes you to the Value Mapping wizard.

If you have used a certain value mapping in training mode, select the same value mapping in application mode.

Tip:

You might have 99% of satisfied customers and 1% of customers not satisfied. If you try to build a model that predicts whether a customer is satisfied for a small training set of data, you might obtain a degenerated pruned tree. This tree might consist only of one node that predicts that all customers are satisfied. In terms of prediction power, this model is of high quality because the error rate is very low (1%). However, to understand which attribute values describe a customer who is not satisfied, a different behavior is required.

Example: You might want to enforce that a misclassified customer who is not satisfied is considered to be as expensive as 10 misclassified customers who are satisfied. To specify this, you might want to create a new value mapping manually and enter the following values in the **Arguments** field and the **Value** field on the Parameters page of a Value Mapping wizard or settings notebook:

Arguments	`dissatisfied;satisfied`
Value	`10`

If your data has leading blanks, enter the following expression in the **Arguments** entry field:

```
dissatisfied;' satisfied'
```

Use this value mapping in training and in application mode.

Specifying mode parameters

The following sections discuss the parameters you can specify on the Mode parameters wizard page.

Training mode

In training mode, the function builds a model based on the selected input data. This model is later used as a classifier.

You can customize the binary decision tree by specifying the following parameters:

Maximum tree depth
This parameter limits the number of node levels for the binary decision tree.

Maximum purity per internal node
This parameter stops further splitting of nodes that have reached the specified purity value.

Minimum records per internal node

This parameter determines how many records are included in an internal node before it splits to the next node.

Example: You might want to create a binary decision tree with 5 levels, at least 7 records per internal node, and stop further splitting of nodes when 90% correct classification is reached. Specify the values in Table 63 for the fields **Maximum tree depth**, **Maximum purity per internal node**, and **Minimum records per internal node**.

Table 63. Specifying training mode parameters for the Tree Classification mining function

Maximum tree depth	Maximum purity per internal node	Minimum records per internal node
5	90	7

Test mode

In test mode, the function uses new or the same data with known class values to verify that the model created in training mode produces results of satisfying precision.

Application mode

In application mode, the function uses a model created in training mode to predict the specified field for every record in the new input data. The data format must be identical to that used to generate the model.

Tips:

You can improve the performance of the mining runs at the expense of precision. You can try the following alternatives:

Limit the maximum tree depth

The default value for maximum tree depth is *Unlimited*. You can estimate the remaining time of the mining run by watching the current tree depth in the Progress Indicator window. However, you cannot estimate the amount of precision lost when limiting the tree depth.

Reduce the value for maximum purity per internal node

The default value for maximum purity per node is 100%. You can specify a lower value so the function stops the splitting of the nodes if it reached the specified value.

Increase the value for minimum records per internal node

The default value for this parameter is 5. If you increase this value, for example, to 50, only terminal nodes can have fewer records than 50.

Important:

The Intelligent Miner treats blank strings in database tables as valid values and blank strings in flat files as missing values.

Therefore, if you use database tables that contain records with blank strings to create a classification model (training mode) and later want to apply that model to data in flat-file format (application mode), you must preprocess your input data. Map blank strings in your database tables to `NULL` before you start the function in training mode. This ensures that blank strings are treated as missing values in both modes. Use the Map Values preprocessing function for this purpose. If you do not prepare your data in this way, you inevitably get wrong results.

Specifying parallel parameters

You can run the Tree Classification mining function in serial or parallel mode. See "Specifying parallel parameters" on page 118 for more information.

Specifying output fields

The layout of the output data corresponds to the appropriate input data, except for the field containing the predicted class IDs, which is part of the output data only. You must specify a name for the predicted class ID.

Additionally, you can include another column for the confidence value in the output data. The confidence value is a value between 0.0 and 1.0. It indicates the probability that the class is predicted correctly.

From the **Available** fields list, you can select additional output fields to be included in the output data. See "Specifying output settings" on page 125 for more information.

You might want to include additional output fields in the output data and name the predicted class ID field *predicted risk class*. See the specification in Table 64.

Table 64. Including additional fields and renaming existing fields in the output data

Output fields	Class ID field name	Confidence field name
Customer ID		
Town district	Predicted risk class	
Country		

Specifying output data

If you specified output fields, you must also specify a data settings object on the Output data page of the wizard or settings notebook. See “Specifying output settings” on page 125 for more information.

Specifying result names

By default, a result object takes on the name of the settings object that created the results. Thus, you do not need to specify a result name when you run the settings object for the first time.

However, you might want to run the settings object again and keep the old results. In this case, you must change the original result name. See “Renaming result objects” on page 128 for more information.

Chapter 15. The Neural Classification mining function

The purpose of predicting a classification is to create a model based on known data. You can use this model to analyze why a certain classification was made, or to calculate the classification of new data.

Historical data frequently consists of a set of values and a classification for these values. Analyzing data that was already classified reveals the characteristics that led to the previous classification. The resulting classification model can then be used to predict the classes of records containing new attribute values.

For example, an insurance firm has data about customers who allowed their insurance to lapse. Based on the common attributes of these customers, Intelligent Miner creates a risk group profile that is then used as a model for classifying new customers. Any new customer is checked against the model and classified as either belonging to the risk group or not belonging to the risk group.

- In training mode, a mining run on this database learns the attributes of each of the defined customer risk classes.
- In test mode, the insurer can analyze the results to understand the attributes that cause customers to let their insurance lapse.
- In application mode, the insurers can use the model created during the training mode to predict which customers let their insurance lapse in the future.

You can also apply this technique, for example, to approve or deny insurance claims, to detect credit-card fraud, to identify defects in images of manufactured parts, or to diagnose error conditions. Other applications are target marketing, medical diagnosis, medical treatment effectiveness, inventory replenishment, or store location planning.

The Neural Classification function employs a *back-propagation neural network* to classify data. The classification is based on the class value and the attribute relationships discovered by mining previously classified data. Developing a model to represent these relationships is called training the network. A trained network is one output of the mining run. Sensitivity analysis, the other output, is used to understand the relative contribution of attribute fields in the classification decision.

Back propagation is a general-purpose, supervised-learning algorithm. In supervised learning, the database contains a number of attribute fields and one or more fields that contain the desired results. In using back propagation for the neural classification application, the desired result is contained in a single field called the class field.

A trained neural network can generalize from its past experience and compute a reasonable classification even from combinations of attribute values it has never seen before.

The following sections describe the basic and advanced parameters of the Neural Classification function. To find all parameter controls on the wizard or notebook pages, you must check the **Show the advanced pages and controls** box on the settings page.

Sample parameter settings

Table 65 shows sample settings summarizing the basic and advanced parameters of the Neural Classification mining function. In addition, it indicates on which wizard page you must specify the required and optional parameters.

Table 65. Sample parameter settings for the Neural Classification mining function

Wizard page	Parameter	Value
Settings	Name	Sample settings
	Comment	Data collected during 1st half of 1997
	Mining function	Classification – Neural
Input data	Input data	Insurance data Security First
	Optimize mining run for	Time
	Filter records	Conditions selected
	Power options	
Mode parameters	Use mode	Training mode
	In-sample size	4
	Out-sample size	2
	Maximum number of passes	500
	Accuracy	80
	Error rate	20
	Regardless of the mode, normalize the input data	True
Input fields	Input fields	4 selected

Table 65. Sample parameter settings for the Neural Classification mining function *(continued)*

Wizard page	Parameter	Value
	Class labels	Risk class
Training parameters	Architecture determination	Manual
	Hidden units 1	2
	Hidden units 2	3
	Hidden units 3	2
	Parameter determination	Manual
	Learn rate	0.2
	Momentum	0.9
Parallel parameters	Run the parallel mode of the function	4 processes
Output fields	Output fields	3 selected
	Class ID field name	Risk class
	Confidence field name	
Output data	Output data	Output risk classes Security First
Results	Results name	Results Security First risk classes
	Comment	Customers who allowed their insurance to lapse
	If a result with this name exists, overwrite it	False

You can use the settings individually or as part of a sequence.

Specifying input data

To use the Neural Classification mining function, select an input data object on the Input data page of a wizard or settings notebook. See "Selecting an input data object" on page 114 for more information.

Example: You might want to predict a classification using the input data of the insurance firm Security First. The input data has the attributes Age, Salary, Marital status, and a class field called Risk class. It is organized as in Table 66 on page 206.

Table 66. Selected input data for the Neural Classification mining function

Age	Salary	Marital status	Risk class
30	65	Single	Good
23	115	Single	Bad
40	75	Married	Good
55	40	Married	Bad
55	100	Widowed	Good
45	60	Divorced	Good

A mining run in training mode on this data discovers the relationships between the attributes and the **Risk class** field.

Optimizing the mining run

You should use the **Optimize for time** option when running the neural mining functions, that is, Neural Clustering, Neural Classification, or Neural Prediction. Because the neural functions require more passes over the training data than the other mining functions, this option significantly reduces the amount of processing time. See "Optimizing the mining run" on page 114 for more information.

Filtering records

You can restrict the mining run to records with certain field values. To do so, specify a filter records condition. See "Filtering records" on page 115 for more information.

Using power options with neural classification

By using power options with the Neural Classification mining function, you can influence the treatment of outliers and the frequency of the weight updates. You can also apply simplified error weighting, which allows you to treat misclassifications differently.

Treating outliers

Entering one of the following expressions in the **Power options** field on the Input data page allows you to treat outliers as described in "Treating outliers" on page 174.

-outliers MissingValues
Employs the *Treat outliers as missing values* option.

-outliers MinMax
Employs the *Replace outlier with MIN or MAX* option.

-outliers ValidValues
Employs the *Treat outliers as valid values* option.

Setting the frequency of weight updates

You can set the frequency of the weight updates by entering a value in the **Power options** field. This value is the number of records processed between the weight updates. If the training process fails to converge, decrease this value until acceptable convergence is observed.

Example: To update the weights after each block of 900 training records, enter `900` in the **Power options** field on the Input data page of a Neural Classification wizard or settings notebook.

In general, using this power option represents a trade-off between improved training convergence and parallel speed-up. Smaller values improve convergence, while larger values improve the performance of parallel processing. Hence, it is desirable to set this value as high as possible while the convergence is still acceptable.

Applying simplified error weighting

Simplified error weighting allows you to weight misclassifications differently. A misclassification has occurred if a record has a certain class label, but the Neural Classification mining function predicted another class label. By default, all misclassifications are weighted with a factor of 1. In applying simplified error weighting, you change the default error weighting to the effect that you achieve a degree of error correction that would normally require more passes over the input data. For example, specifying a weight of 3 has the same effect with regard to the adjustment of the error weights as feeding the neural network with the same records for three times.

To change the error weighting, enter the following power option in the **Power options** field on the Input data page:

```
-errorWeight <SomeLabel> <Weight>
```

where <SomeLabel> is the class label of the field to which you want to apply a different error weighting, and <Weight> is a numeric value greater than or equal to 0. If <Weight> is greater than 1, the error weighting is stronger than the default error weighting. If <Weight> is lower than 1, the error weighting is weaker than the default error weighting.

Example: Suppose that you work for a company that produces machines. Compared with the American market, the company makes twice as much profit if it sells a machine in Asia. Now suppose that you use the Neural Classification mining function to detect potential customers for new machines in your customer database. If a customer in Asia is wrongly classified as not being a potential customer for the new machine, and therefore not contacted by a sales representative, the company possibly loses twice the profit it would lose if the same happened with an American customer. Therefore, the error weighting for the data records pertaining to Asian customers must be twice as strong.

Specifying mode parameters

The following sections discuss the parameters you can specify on the Mode parameters wizard page.

Training mode

In training mode, the function builds a model based on a subset of the selected input data. This model is later used as a classifier. The function also creates a sensitivity analysis report that ranks the input fields according to the degree of relevance to the classification application. The training consists of alternating learning and verification phases. The neural network algorithm automatically alternates between these phases.

In-sample size

Represents the number of consecutive records in the input data to be used in the learning phase. While the network is learning, it is calculating the weights it uses to represent the relationships in the data.

During the verification phase, the specified number of records is skipped.

Out-sample size

Represents the number of consecutive records in the input data used to determine if the desired accuracy and error limit objectives have been met when training shifts to the verification phase. When these goals are met, the training ends.

During the learning phase, the specified number of records is skipped.

Example: In the learning phase, you might want to use the first four records of the input data, skip the next two records, use the next four records, and continue this process until the last record of the input data is read. Specify the values in Table 67 for the **In-sample size** and **Out-sample size** fields on the Mode parameters page of a Neural Classification wizard or settings notebook.

Table 67. Specifying an in-sample size and out-sample size for training

In-sample size	Out-sample size
4	2

During the verification phase, this process is reversed: Alternately, four records are skipped and two records are used so that those records are used for verification that were omitted during the learning phase.

If you use one flat file or database table, the specification of the in-sample size and out-sample size splits the input data logically into two disjoint subsets of data.

If you use multiple flat files or database tables, these files are concatenated to one large data set. Then the four consecutive in-sample records might be used from separate adjacent data sets.

Classifying data is an iterative process. Specifying multiple passes through the data improves the quality of the classification. Limiting the number of passes reduces the processing time required to classify data, however, it also reduces the accuracy of the classification. At least 100 maximum passes are usually required.

The parameters *accuracy* and *error rate* also influence the number of iterations that are actually performed. The accuracy is the percentage of correct classifications; the error rate is the maximum percentage of incorrect classifications. Records that could not be classified are not considered during the calculation of the error rate.

When the network correctly classifies the percentage of records specified in the input data, it has obtained the accuracy objective. The network calculates this percentage using the records designated as out-sample. If the error limit objective is also met, the training completes and the mining function stops.

The network continues to train as long as it obtains the wrong classification from more than the specified percentage of records from the testing sample.

The network can also determine that the input fields represent an unknown class that is not considered to be either correct or incorrect. After the network reaches the error rate you specified, the training completes and the mining function stops if the accuracy objective is also met. In cases where the misclassification is worse than an unknown classification, you can set the error rate to trade off between misclassifications and unknown classes.

Example: You might want to use the values in Table 68 in training mode.

Table 68. Specifying training mode parameters

In-sample size	Out-sample size	Maximum number of passes	Accuracy	Error rate
4	2	500	80	20

Test mode

In test mode, the function is applied to new or the same data with known class values to test whether the trained network is producing correct results.

Application mode

In application mode, the function uses a model created in training mode to predict the specified field for every record in the new input data. The set of input fields must be identical to that used to generate the model.

Important:

The Intelligent Miner treats blank strings in database tables as valid values and blank strings in flat files as missing values.

Therefore, if you use database tables that contain records with blank strings to create a classification model (training mode) and later want to apply that model to data in flat-file format (application mode), you must preprocess your input data. Map blank strings in your database tables to `NULL` before you start the function in training mode. This ensures that blank strings are treated as missing values in both modes. Use the Map Values preprocessing function for this purpose. If you do not prepare your data in this way, you inevitably get wrong results.

Normalizing the input data

The input data must be normalized or scaled to a range of 0.0 to 1.0. In addition, categorical values must be converted into a numeric code for presentation to the neural network.

If you choose to normalize the input data, the Intelligent Miner scales continuous and discrete numeric fields to a range of 0.0 to 1.0 and converts categorical data into 1-of-N vectors. A single 1-value is converted at the string index value, and 0-values are converted at all other positions in the vector. This means that a categorical value with a large number of discrete values can cause an expansion in the number of input units to the neural network.

For example, you might have input data with four input fields that are all continuous fields. The neural network would have four input units. However, if one of the four fields is discrete and it represents the state the customer lives in, it has 50 possible values. This one field would expand into a 1-of-50 vector. The neural network (with four input fields) would have 53 input units (3 for the 3 continuous fields and 50 for the single discrete field). If your data set has several categorical fields of this type, you could be unknowingly generating very large neural networks. This might negatively impact the training time. Possible solutions are to use the Intelligent Miner preprocessing functions to reduce the number of discrete values, to convert the field data type to discrete numeric, or to use computed fields with value mappings or computed field functions.

Continuous and discrete numeric fields are converted to a 0.0 to 1.0 range using a piece-wise linear algorithm. The mean value is mapped to 0.5. Input values from the minimum to mean consequently range from 0.0 to 0.5 after scaling, while input values from the mean to the maximum scale to 0.5 to 1.0. The effect of this scaling algorithm is to expand or normalize the distribution of continuous variables that have non-normal distributions. For some types of input fields, this scaling would result in a loss of information.

In these cases, it might be necessary to use a virtual computed field to perform special processing of the data. You also have the option of normalizing and scaling the data yourself, and choosing not to normalize the input data. In this case, the data is presented directly to the neural network.

Tips:

1. The value for maximum number of passes can have a significant impact on the training time for a neural model. In general, you should start with a smaller number of passes, for example, 50 to 100, to find out whether the neural network will converge. Depending on the problem and the amount of data, it might take thousands of passes to train the neural network. If you have a large amount of data, you might want to sample the data to reduce the training time.

   In some cases the selected neural network is not able to converge or learn to distinguish between the classes. Typically, the output confusion matrix shows all records belonging to a single class. This indicates that either you did not specify enough passes over the data or your data does not contain enough information for the neural network to build a classification model.
2. To get more detailed results than shown in the confusion matrix, you can use the Bivariate Statistics function. Follow these steps:
   a. Specify the settings necessary to let the Neural Classification mining function produce an output data object. Select as many output fields as you want to appear in the results. Remember the name of the Class ID field.
   b. Create a Bivariate Statistics settings object and make sure to include the following settings:
      1) Specify the output data object produced by the Classification – Neural mining function as input.
      2) Select **Compute Statistics** when you reach the Statistics page of the wizard.
      3) Select the Class ID field and move it to the selection field headed **Compute Chi-square and bivariate statistics using this field**.

4) Move the other fields to the **Compute univariate statistics for these fields** list.

c. Run the Bivariate Statistics object. The results are displayed in the Bivariate Statistics viewer, showing detailed information about each class field.

d. Optionally, save these steps in a Sequence object. This allows you to repeat the foregoing steps without having to re-create the settings objects.

Specifying input fields

The Neural Classification mining function uses a set of input fields and a distinguished class field. You must specify input fields that are likely to contain information that is relevant to the items represented by the records.

The values of the class field serve as class labels determining the class to which the records belong. The class field must be a categorical or discrete numeric field. During the training phase, a model is built that predicts a class label for each record. The classification is based on the specified input fields only.

Example: To identify customers who have allowed their insurance to lapse, you might specify the data fields shown in Table 69.

Table 69. Selected input fields for the Neural Classification mining function

Input fields	Class label
Town districts Country Profession	Risk class

The **Available fields** list contains all data fields of the data tables you included in the input data. From this list you can select the data fields to be classified.

Training parameters

You can let the system determine the architecture and the training parameters or specify your own values.

Architecture determination and parameter determination are independent from each other.

Architecture determination

The architecture of a neural network includes the number of the following components:

- Processing units in the input and output layers
- Hidden layers used
- Processing units in each hidden layer

The number of input and output units directly depends on the input fields and the class field, respectively.

One or more processing units in the input layer represent a numeric or categorical value of the input data. Every processing unit is connected to every processing unit in the next layer by a weighted value expressing the strength of the relationship. These weights are called adaptable connections because their values are adjusted during training so that the network output approaches the class values present in the data.

Using automatic architecture determination, the mining function evaluates different neural network architectures with different numbers of hidden layers including the processing units in these layers. These alternative models are trained for a fixed number of passes, and then the best network architecture is selected for further training.

Automatic architecture determination requires more processing time than manual architecture determination. The function constructs 5 different neural network architectures with 0, 1, and 2 hidden layers. These network architectures are trained for a fixed number of passes over the data. The passes depend on the input size. The maximum number of passes is 50. After the architecture-training passes are completed, the best performing network is selected to undergo additional training.

Using *manual* architecture determination requires knowledge of the selection criteria for the neural network architecture, because the exact architecture must be specified.

Setting the architecture manually, requires less processing time than automatic architecture determination. Only one network with the specified architecture is trained by the function.

Example: You might want to specify the values in Table 70 on page 214 for the manual architecture determination.

Table 70. Specifying values for the manual architecture determination

Hidden unit 1	Hidden unit 2	Hidden unit 3
2	3	2

Parameter determination

You can let the system determine values for the *learn rate* and the *momentum* or specify your own values.

The learn rate controls to which extent the weights are dynamically adjusted during the training process to improve the model convergence. Larger values indicate higher change so the network trains more quickly with higher values; however, the accuracy might be lower.

The momentum adjusts the change applied to a weight by factoring in previous weight updates. It acts as a smoothing parameter that reduces oscillation and helps attain convergence. Lower values mean more training time and more influence of the outlier data on the weights.

Example: You might want to specify the values in Table 71 for the learn rate and the momentum.

Table 71. Sample values for the learn rate and the momentum

Learn rate	Momentum
0.2	0.9

You can define the delimiters for floating-point numbers in the Preferences notebook by selecting **Options** from the Main window.

Tip: If you use the default value for the learn rate, which is 0.0, the Neural Classification mining function uses an adaptive learn rate algorithm. This algorithm monitors the rate of convergence of the neural network and increases or decreases the learn rate as required. For the majority of cases, the default learn rate works best.

However, if you want to control the learn rate, you can select **Manual** on the Training Parameters page and set this parameter to a constant. Typical values range from 0.01 to 0.9.

Specifying parallel parameters

You can run the Neural Classification mining function in serial or parallel mode. See “Specifying parallel parameters” on page 118 for more information.

Specifying output fields

The layout of the output data corresponds to the appropriate input data. By default, it includes a column for the predicted Class ID.

Additionally, you can include another column for the Confidence value in the output data. The Confidence value is a value between 0.0 and 1.0. It indicates the estimated probability that the class is predicted correctly.

From the **Available** fields list, you can select additional output fields to be included in the output data. See "Specifying output settings" on page 125 for more information. To filter records in the **Available** fields list, see "Filtering records" on page 206.

You might want to include additional output fields in the output data and replace the output field name Class ID with Risk class. See the specification in Table 72.

Table 72. Including additional fields and renaming existing fields in the output data

Output fields	Class ID field name	Confidence field name
Customer ID		
Town district	Risk class	
Country		

Specifying output data

If you specified output fields, you must also specify a data settings object on the Output data page of the wizard or settings notebook. See "Specifying output settings" on page 125 for more information.

Specifying result names

By default, a result object takes on the name of the settings object that created the results. Thus, you do not need to specify a result name when you run the settings object for the first time.

However, you might want to run the settings object again and keep the old results. In this case, you must change the original result name. See "Renaming result objects" on page 128 for more information.

Chapter 16. The RBF-Prediction mining function

The purpose of predicting values is to discover the dependency and the variation of one field's value upon the values of the other fields within the same record. A model is generated that can predict a value for that particular field in a new record of the same form, based on other field values.

For example, a retailer wants to use historical data to estimate the sales revenue for a new customer. A mining run on this historical data creates a model. This model can be used to predict the expected sales revenue for a new customer, based on the new customer's data. The model might also show that for some customers, incentive campaigns improve sales. In addition, it might reveal that frequent visits by sales representatives lead to a lower revenue if the customer is young.

On a practical level, the algorithms process a table of data in which every record has an identical format. A single field within the table must be designated as containing the value to be fitted, while the coordinates are selected from the other fields in the table.

You can use the Radial-Basis Function (RBF) method to fit data that is a function of many variables. The basic algorithm can form a model that predicts the value of a particular field from the other attribute values.

A Radial-Basis Function fitting requires a number of fitting centers. A fitting center is a vector in the attribute space. At each of these centers, a basis function is defined. The basis function is a nonlinear function of distance from the fitting center. This is why the basis functions are called Radial-Basis Functions; they have the same value for any point with the same distance or radius from the fitting center.

The prediction given by the radial-basis fit for a particular set of attributes (called a *point*) is a weighted sum of these basis functions at that point. During the fitting process, the weight values producing the best fits are determined at each fitting center. In addition, it is decided where the fitting centers are placed.

The following sections describe the basic and advanced parameters of the RBF-Prediction function. To find all parameter controls on the wizard or notebook pages, you must check the **Show the advanced pages and controls** box on the settings page.

Sample parameter settings

Table 73 shows sample settings summarizing the basic and advanced parameters of the RBF-Prediction mining function. In addition, it indicates on which wizard page you must specify the required and optional parameters.

Table 73. Sample parameter settings for the RBF-Prediction mining function

Wizard page	Parameter	Value
Settings	Name	Sample settings
	Comment	Data collected during 1st half of 1997
	Mining function	Prediction – Radial Basis Function
Input data	Input data	Insurance data Security First
	Optimize mining run for	Time
	Filter records	<Conditions selected>
	Power options	
Mode parameters	Use mode	Training mode
	In-sample size	12
	Out-sample size	4
	Maximum number of passes	12
	Maximum centers	500
	Minimum region size	20
	Minimum passes	7
Input fields	Active fields	20 selected
	Supplementary fields	1 selected
	Prediction field	Revenue
Categorical field parameters	List of values to predict	<None selected>
Quantiles	Generate quantiles	True
	List of quantile limits	2, 10, 25, 50, 75, 90, 98
Output fields	Output fields	1 selected
	Predicted value field name	Predicted revenue
	Region ID field name	Region ID
	Lower quantile field name	Lower quantile
	Upper quantile field name	Upper quantile

Table 73. Sample parameter settings for the RBF-Prediction mining function (continued)

Wizard page	Parameter	Value
Output data	Output data	Output sales revenue
Results	Results name	Prediction model for sales revenue
	Comment	
	If a result with this name exists, overwrite it	True

You can use the settings individually or as part of a sequence.

Specifying input data

To use the RBF-Prediction mining function, select an input data object on the Input data page of a wizard or settings notebook. See "Selecting an input data object" on page 114 for more information.

Optimizing the mining run

The RBF-Prediction mining function expects the same order of records at each pass. If this is not the case, records of the in-sample and the out-sample might be mixed, which sometimes results in an error when you have rare categorical values in your data. For more information on the terms in-sample and out-sample, see "Training mode" on page 220.

To ensure that the order of records remains the same when you use DB2 input data, optimize the mining run for time. See "Optimizing the mining run" on page 114 for more information.

Filtering records

You can restrict the mining run to records with certain field values. To do so, specify a filter records condition. See "Filtering records" on page 115 for more information.

Using power options with RBF prediction

You might want to specify a power option to change the default value for the normalized error.

Example: By default, a mining run in training mode stops when a model was found with a global normalized error of less than 5. The normalized error is 25 root mean squared errors divided by the standard deviation in the predicted field. You can change this default value by using the Power options parameter.

To stop the mining run in training mode when the normalized error gets below 8.25, enter the following expression in the **Power options** field on the Input data page of an RBF-Prediction wizard or settings notebook:

```
NORMALIZED_ERROR=8.25;
```

Attention: Use power options with great care. Make sure you read the general introduction in "Using power options" on page 117.

Specifying mode parameters

The following sections discuss the parameters you can specify on the Mode parameters wizard page.

Training mode

In training mode, the function builds a model based on a subset of the selected input data. The remainder of the input data is used to check for the overtraining of the model.

The model predicts a value for the specified prediction field for each record in the input data. This field is called the dependent field, because the prediction process considers it to be dependent on the other input fields. The other fields are called attribute fields, because their values can be considered as attributes that contribute to the definition of the dependent field.

The goal of the training is to find patterns in the attribute fields that give particular results. Mathematically, training the data produces a nonlinear function that best predicts the dependent field value based on the values for the attribute fields.

In-sample size and out-sample size

The *in-sample size* represents the number of consecutive records to select from the input data to be used for training.

During the verification phase, the specified number of records is skipped.

The *out-sample size* represents the number of consecutive records to select from the input data. It is used when training shifts to the verification phase to determine if the desired accuracy and error limit objectives have been met. It is also used to detect overtraining of the model in time. When these goals are met, the training mode ends.

During training, the specified number of records is skipped.

If you use one flat file or database table, the specification of the in-sample size and out-sample size splits the input data logically into 2 disjoint subsets of data.

If you use multiple flat files or database tables, these files are concatenated to one large data set. Then the 4 consecutive in-sample records might be used from separate adjacent data sets.

You might use input data including 100 records and specify the values in Table 74 for the **In-sample size** and **Out-sample size** fields on the Mode parameters page of an RBF-Prediction wizard or settings notebook.

Table 74. Specifying an in-sample size and out-sample size for training

In-sample size	Out-sample size
4	2

This has the effect that during the learning phase, alternately four records are used and the next two records are skipped. This process continues until all records of the input data are read.

You can use separate data sets for training and verification. You might want to use a data set with 800 records for training and a data set with 200 records for verification. Specify the values in Table 75 for the **In-sample size** and **Out-sample size** fields.

Table 75. Specifying an in-sample size and out-sample size for verification

In-sample size	Out-sample-size
800	200

Add the training data set before you are adding the verification data set to the input data. Ensure that the value for the in-sample size is equal to the number of records in the training data set, and that the value for the out-sample size is equal to the number of records in the test data set. Otherwise the two data sets can be mixed.

Maximum number of passes

The value for *maximum number of passes* limits the number of times the mining function goes through the data. If the model reaches the desired prediction accuracy before this limit is reached, training will stop.

Maximum centers

The value for *maximum centers* limits the number of centers built by the mining function at each pass through the input data. The actual number of centers might be higher than the number you specified because the number of centers can increase up to twice the initial number during a pass.

Maximum region size
The value for *minimum region size* determines the minimum number of records assigned to a region. If at the end of a pass a region has a value less than the specified value, the region is deleted.

Minimum passes
The value for *minimum passes* determines the minimum number of passes through the input data. During these passes, no checking for overtraining or accuracy is done. This saves time, however, you should only specify a large value for minimum passes if you have enough training data and if you are sure that a good model exists.

Example
You might want to specify the values in Table 76 in training mode:

Table 76. Specifying training-mode parameters for the RBF-Prediction mining function

In-sample size	Out-sample size	Maximum passes	Maximum centers	Minimum region size	Minimum passes
12	4	12	500	20	7

Test mode
In test mode, the function uses existing data with known predicted field values to determine the accuracy of the model created in training mode.

Application mode
In application mode, the function uses a model created in training mode to predict the specified field for every record in the new input data. The data format must be identical to that used to generate the model.

Important:

The Intelligent Miner treats blank strings in database tables as valid values and blank strings in flat files as missing values.

Therefore, if you use database tables that contain records with blank strings to create a prediction model (training mode) and later want to apply that model to data in flat-file format (application mode), you must preprocess your input data. Map blank strings in your database tables to `NULL` before you start the function in training mode. This ensures that blank strings are treated as missing values in both modes. Use the Map Values preprocessing function for this purpose. If you do not prepare your data in this way, you inevitably get wrong results.

Specifying input fields

To discover the dependency and the variation of one field's value within a record upon the values of the other fields within the same record, you must specify the active fields and the prediction field. You can also specify supplementary fields. Then a model is generated that can predict a value for that particular field in a new record of the same form, based on the values of the active field.

The active fields are used by the mining function. The supplementary fields are used to gain statistical information on their distributions in the regions or quantile ranges. The prediction field holds the value to be predicted by the mining function.

Example: To predict the sales revenue for a new customer, you might specify the fields in Table 77.

Table 77. Selected input fields for the RBF-Prediction mining function

Active fields	Supplementary fields	Prediction field
Age	Year 1st policy	Revenue
Sex	Commute distance	
Income		
Visits by sales representative		

The **Available fields** list contains all data fields of the data tables you included in the input data. From this list, you can select the data fields to be mined.

Tips:

- Do not specify fields as active or supplementary fields that have different values for almost every record. These fields might be useful as key fields in the output data.
- Do not specify fields as active fields that have the same value in almost every record.

Such fields do not contain any valuable information for building the prediction model.

Categorical field parameters

For categorical prediction fields, you can provide a list of values to be predicted. The chosen field values are mapped to 1, and all other possible values are mapped to 0. You can then interpret the prediction as the probability of a record to have one of the listed values.

Example: Suppose that you have a categorical field for revenue with possible values of *Low*, *Medium*, and *High*. To predict the probability of a new customer to produce medium or high revenue, enter the following expression in the **List of values to predict** field on the Categorical field parameters page:

```
"Medium"; "High"
```

If a field value has leading blanks, also type the blanks within the quotes, for example *" High"* instead of *"High"*. Trailing blanks are ignored, so left alignment in flat-file columns is favorable.

Restriction: The Categorical field parameters page is displayed only if you select a categorical field as **Prediction field**.

You can specify the delimiter you want to use by clicking **Options → Preferences** on the menu bar in the Intelligent Miner main window.

Quantile ranges

Quantile ranges are subsets of data within a specific range of the predicted value. The quantile limits represent the lower and the upper percentage of the predicted value. For example, the quantile range Q[50,75] contains records whose predicted values are between the quantiles Q(50) and Q(75) of the multiset of all predicted values.

You can classify the value in the predicted field according to the quantile in which it falls.

If you select to generate quantiles, you can include the **Lower quantile** field and the **Upper quantile** field in the output data.

Specifying parallel parameters

Using the RBF-Prediction mining function in the training and test modes, you can only perform serial mining runs.

However, in application mode, you can also run the RBF-Prediction mining function on parallel processing nodes. See “Specifying parallel parameters” on page 118 for more information.

Specifying output fields

The layout of the output data corresponds to the appropriate input data. The output data includes the output fields that you specify. In addition, you must specify the **Predicted value field name**. Specifying the **Region ID** field is optional.

If you choose to generate quantiles, you must specify the names of the following output fields:

- **Lower quantile field name**
- **Upper quantile field name**

Before you can proceed to the next wizard page, you must at least enter the **Predicted value field name**.

From the **Available** fields list, you can select additional output fields to be included in the output data. See "Specifying output settings" on page 125 for more information.

Example: To include the data field *Customer number* and all optional and required fields in the output data, specify the field names in Table 78.

Table 78. Specifying output field names for the RBF-Prediction mining function

Output fields	Predicted value field name	Region ID field name	Lower quantile field name	Upper quantile field name
Customer number	Predicted revenue	Region ID	Lower quantile	Upper quantile

Specifying output data

If you specified output fields, you must also specify a data settings object on the Output data page of the wizard or settings notebook. See "Specifying output settings" on page 125 for more information.

Specifying result names

By default, a result object takes on the name of the settings object that created the results. Thus, you do not need to specify a result name when you run the settings object for the first time.

However, you might want to run the settings object again and keep the old results. In this case, you must change the original result name. See "Renaming result objects" on page 128 for more information.

Chapter 17. The Neural Prediction mining function

The purpose of predicting values is to discover the dependency and the variation of one field's value upon the values of the other fields within the same record. A model is generated that can predict a value for that particular field in a new record of the same form, based on other field values.

For example, a retailer wants to use historical data to estimate the sales revenue for a new customer. A mining run on this historical data creates a model. This model can be used to predict the expected sales revenue for a new customer, based on the new customer's data. The model might also show that for some customers, incentive campaigns improve sales. In addition, it might reveal that frequent visits by sales representatives lead to a lower revenue if the customer is young.

On a practical level, the algorithms process a table of data in which every record has an identical format. A single field within the table must be designated as containing the value to be fitted, while the coordinates are selected from the other fields in the table.

The Neural Prediction mining function creates a model to use for predicting new values for regression and time-series forecasting.

The Neural Prediction mining function employs a back-propagation neural network to predict values. The prediction is based on the prediction value and the attribute relationships discovered by mining a set of training data containing the independent as well as the dependent variables. Developing a model to represent these relationships is called training the neural network.

In addition to standard value prediction (also known as regression), the Neural Prediction function provides support for time-series prediction by allowing the user to specify a forecast horizon and an input window size. These two parameters are used to internally format the training records so that the neural network takes a set of *m* consecutive records (the window size) and predicts the dependent value of *n* records (the horizon) into the future. Note that the default values, a horizon of 0, and a window size of 1, define a standard value prediction model. Any horizon value greater than 1 defines a time-series prediction model.

The following sections describe the basic and advanced parameters of the Neural Prediction function. To find all parameter controls on the wizard or notebook pages, you must check the **Show the advanced pages and controls** box on the settings page.

Sample parameter settings

Table 79 shows sample settings summarizing the basic and advanced parameters of the Neural Prediction mining function. In addition, it indicates on which wizard page you must specify the required and optional parameters:

Table 79. Sample parameter settings for the Neural Prediction mining function

Wizard page	Parameter	Value
Settings	Name	Sample settings
	Comment	Data collected during 1st half of 1997
	Mining function	Prediction – Neural
Input data	Input data	Insurance data Security First
	Optimize mining run for	Time
	Filter records	<Conditions selected>
	Power options	
Mode parameters	Use mode	Training mode
	In-sample size	4
	Out-sample size	2
	Maximum number of passes	500
	Forecast horizon	0
	Window size	1
	Average error	0.1
	Regardless of mode, normalize the input data	True
Input fields	Active fields	2 selected
	Supplementary fields	1 selected
	Prediction fields	Revenue
Training parameters	Architecture determination	Manual
	Hidden units 1	2
	Hidden units 2	3
	Hidden units 3	2
	Parameter determination	Manual
	Learn rate	0.2
	Momentum	0.9
Quantiles	Generate quantiles	True

Table 79. Sample parameter settings for the Neural Prediction mining function (continued)

Wizard page	Parameter	Value
	List of quantile limits	2, 10, 25, 50, 75, 90, 98
Parallel parameters	Run the parallel mode of the function	On 4 processor nodes
Output fields	Output fields	1 selected
	Predicted value field name	Predicted value
Output data	Output data	Predicted sales revenue
Results	Results name	Results insurance data Security First
	Comment	Branch office Cleveland only
	If a result with this name exists, overwrite it	True

You can use the settings individually or as part of a sequence.

Specifying input data

To use the Neural Prediction mining function, select an input data object on the Input data page of a wizard or settings notebook. See "Selecting an input data object" on page 114 for more information.

Optimizing the mining run

You should use the **Optimize for time** option when running the neural mining functions, that is, Neural Classification, Neural Clustering, or Neural Prediction. Because the neural functions require more passes over the training data than the other mining functions, this option can significantly reduce the processing time. See "Optimizing the mining run" on page 114 for more information.

Filtering records

You can restrict the mining run to records with certain field values. To do so, specify a filter records condition. See "Filtering records" on page 115 for more information.

Using power options with neural prediction

Entering one of the following expressions in the **Power options** field on the Input data page allows you to treat outliers as described in "Treating outliers" on page 174.

-outliers MissingValues
Employs the *Treat outliers as missing values* option.

-outliers MinMax
Employs the *Replace outlier with MIN or MAX* option.

-outliers ValidValues
Employs the *Treat outliers as valid values* option.

Specifying mode parameters

The following sections discuss the parameters you can specify on the Mode parameters wizard page.

Training mode

In training mode, the function builds a model based on a subset of the selected input data. The remainder of the input data is used to check for the overtraining of the model. The model predicts a value for the specified prediction field for each record in the input data. The training consists of alternating learning and verification phases. The neural network algorithm automatically alternates between these phases.

In-sample size and out-sample size

The *in-sample size* represents the number of consecutive records to select from the input data to be used in the learning phase. While the network is learning, it is calculating the weights it uses to represent the relationships in the data. During the verification phase, the specified number of records is skipped.

The *out-sample size* represents the number of consecutive records to select from the input data used when training shifts to the verification phase to determine if the desired accuracy and error limit objectives have been met. When these goals are met, the training mode ends. During the learning phase, the specified number of records is skipped.

In the learning phase, you might want to use the values in Table 80 for the in-sample size and the out-sample size.

Table 80. Specifying an in-sample size and out-sample size for the learning phase

In-sample size	Out-sample size
4	2

This has the effect that during the learning phase, alternately four records are used and the next two records are skipped. This process continues until all records of the input data are read.

During the verification phase, this process is reversed: Alternately, four records are skipped and two records are used so that those records are used for verification that were omitted during the learning phase.

If you use one flat file or database table, the specification of the in-sample size and out-sample size splits the input data logically into two disjoint subsets of data.

If you use multiple flat files or database tables, these files are concatenated to one large data set. Then the 4 consecutive in-sample records might be used from separate adjacent data sets.

Add the training data set before you are adding the test data set to the input data. Ensure that the value for the in-sample size is equal to the number of records in the training data set, and that the value for the out-sample size is equal to the number of records in the test data set. Otherwise the two data sets can be mixed.

Maximum number of passes

The *maximum number of passes* limits the number of times the mining function goes through the data. If the network reaches the desired prediction accuracy before this limit is reached, training stops.

Forecast horizon

The *forecast horizon* represents the relationship between the input data and the prediction field. If the forecast horizon is set to 0, input fields and prediction fields are based on the same record. To predict values in the future, specify the number of periods to predict.

Window size

The *window size* Represents the number of records in the input data used by the mining function to predict a value. By default, one record is used to predict a value. For time-series forecasting, three or more records are required to accurately predict a value.

The values for forecast horizon and window size are combined to create the size of a logical input record.

Average error

The *average error* represents the percentage of records in the testing sample used to determine if the specified error limit has been met. You can specify the value for the average Root Mean Square (RMS) error. This error value is computed on the verification records in the out-sample. During training, the

RMS error normally decreases because the quality of the predictions improves with the ongoing learning process. When the algorithm reaches the specified error limit, that is, when the RMS error is less than or equal to the specified value, the training stops. Values between 0.25 and 0.01 indicate a perfect prediction. Note that the RMS error is computed on the normalized input.

The network continues to train while the average RMS error is greater than the average error you specified. Once the average error has been reached or if the average error starts to increase, the network training stops.

Example

To run monthly sales data to predict the sales in 6 months, you might want to specify the values in Table 81.

Table 81. Specifying mode parameters for the Neural Prediction mining function

In-sample size	Out-sample size	Max number of passes	Forecast horizon	Window size	Average error
4	2	500	6	3	0.1

Test mode

In test mode, the function uses new or existing data with known results to verify that the trained network is producing correct results.

Application mode

In application mode, the function uses a model created in training mode to predict the specified field for every record in the new input data. The data format must be identical to that used to generate the model.

Important:

The Intelligent Miner treats blank strings in database tables as valid values and blank strings in flat files as missing values.

Therefore, if you use database tables that contain records with blank strings to create a prediction model (training mode) and later want to apply that model to data in flat-file format (application mode), you must preprocess your input data. Map blank strings in your database tables to `NULL` before you start the function in training mode. This ensures that blank strings are treated as missing values in both modes. Use the Map Values preprocessing function for this purpose. If you do not prepare your data in this way, you inevitably get wrong results.

Normalizing the input data

The input data must be normalized or scaled to a range of 0.0 to 1.0. In addition, categorical values must be converted into a numeric code for presentation to the neural network. The normalization technique is the same for all neural mining functions.

If you choose to normalize the input data, the Intelligent Miner scales continuous and discrete numeric fields to a range of 0.0 to 1.0 and converts categorical data into 1-of-N vectors. A single 1-value is converted at the string index value, and 0-values are converted at all other positions in the vector. This means that a categorical value with a large number of discrete values can cause an expansion in the number of input units to the neural network.

For example, you might have input data with four input fields that are all continuous fields. The neural network would have four input units. However, if one of the four fields is categorical and it represents the state the customer lives in, it has 50 possible values. This one field would expand into a 1-of-50 vector. The neural network (with four input fields) would have 53 input units (3 for the three continuous fields and 50 for the single discrete field). If your data set has several categorical fields of this type, you could be generating very large neural networks. This might negatively impact the training time. Possible solutions are to use the Intelligent Miner preprocessing functions to reduce the number of discrete values, to convert the field data type to discrete numeric, or to use computed fields with value mappings or computed field functions.

Continuous and discrete numeric fields are converted to a 0.0 to 1.0 range using a piece-wise linear algorithm. The mean value of the value is mapped to 0.5. Input values from the minimum to mean consequently range from 0.0 to 0.5 after scaling, while input values from the mean to the maximum scale to 0.5 to 1.0. The effect of this scaling algorithm is to expand or normalize the distribution of continuous variables that have non-normal distributions. For some types of input fields, this scaling results in a loss of information.

In these cases, it might be necessary to use a virtual computed field to perform special processing of the data. You also have the option of normalizing and scaling the data yourself, and choosing not to normalize the input data. In this case, the data is presented directly to the neural network.

Tips

The following section contains useful background information about the Neural Prediction mining function. You can use this information to better understand the algorithm behind the function or to adjust your parameter settings.

Determination of the average error

The Neural Prediction function uses back propagation to perform regression and time-series prediction. The average error rate is represented in terms of the average root-mean squared (RMS) error of the desired and actual network output objects. Note that this is after the data is normalized.

Standard regression (default values for window size and forecast horizon)

With the default values 1 for window size and 0 for forecast horizon, the Neural Prediction function performs standard regression. Only continuous fields can be designated for the predicted field. For problems with non-linearities, the Neural Prediction function can usually provide much better fits than standard linear regression. By setting the number of hidden units on the **Training parameters** page to 0,0,0, in advanced mode, you can perform logistic regression.

Time-series prediction (forecast horizon > 1)

When you set the value for forecast horizon greater than 1, you are performing time-series prediction. The neural network takes a set of m consecutive records (the window size) and predicts the dependent value of n records (the horizon) in the future. Typically, you would also need to increase the window size to give the neural network enough context information to build a model that is able to predict the future. However, one consideration is that the Window size parameter results in an increase of the number of input units to the neural network and a corresponding growth in the number of network weights. For example, if an input record with four variables expands to 10 input units when the data is normalized, and you set the value for Window size to 5, the network input layer would have 5 times 10 or 50 input units.

The Neural Prediction function automatically buffers the data to compensate for the forecast horizon values greater than 1. For example, if you specify a window size of 5, then five records have to be read to construct a single input record for the neural network. If the forecast horizon was set to 3, three records have to be buffered internally to be able to present the prediction value from the record 3 time steps into the future as the target value.

Sizes of in-sample and out-sample

If the size of the in-sample and the out-sample is rather small, the sequence of records is split into small subsequences. Use large samples for time-series prediction.

Time-series prediction in parallel mode

When you are running time-series prediction in parallel mode, the input data must be partitioned into contiguous blocks for each parallel process to preserve the time-sequence order in the original input data.

Specify a high value for *nstripe* when you use the **idmdpart** command. See "Running the partitioning program" on page 122 for more information.

Categorical fields

In most cases, to predict the values of a categorical field, you better use one of the classification mining functions. However, neither the Tree Classification nor the Neural Clustering mining function assumes that the input values are ordered. The prediction mining functions, on the other hand, do assume that the range of input values is ordered. So if you want the order of the input values to be considered in the results, use a prediction mining function.

Trying to put this into practice, you might have the problem that the categorical field whose future values you want to predict contains string values. The prediction mining functions cannot handle such values. To overcome this problem, you can define a value mapping object for your input data object.

Example: You can map the string values *Yes*,*Almost sure*, and *No* of a categorical field to 1, 0.8, and 0, respectively. Using the converted values as input, the Neural Prediction mining function is able to process the input data.

For more information on how to define a value mapping object, see the following sections in this book:

- "Name mapping and value mapping objects" on page 103
- "Value mapping objects" on page 104
- "Creating name mapping or value mapping objects" on page 105
- "Computed fields" on page 90

Specifying input fields

To discover the dependency and the variation of one field's value within a record upon the values of the other fields within the same record, you must specify the active fields, the supplementary fields, and the prediction field. Then a model is generated that can predict a value for that particular field in a new record of the same form, based on its other field values.

The active fields are used by the mining function. The supplementary fields are used to gain statistical information on their distributions in the regions or quantile ranges. The prediction field is the value to be predicted by the mining function.

Example: To predict the sales revenue for a new customer, you might specify the fields shown in Table 82 on page 236.

Table 82. Selected input fields for the Neural Prediction mining function

Active fields	Supplementary fields	Prediction field
Age	Year 1st policy	Revenue
Sex	Commute distance	
Income		
Visits by sales representative		

The **Available fields** list contains all data fields of the data tables you included in the input data. From this list, you can select the data fields to be mined.

Tips:

- Do not specify categorical fields as active fields that have different values for almost every record.
- Do not specify fields as active fields that have the same value in almost every record.

Such fields do not contain any valuable information for building the prediction model.

Training parameters

You can let the system determine the architecture and the training parameters or specify your own values.

Architecture determination and parameter determination are independent from each other.

Architecture determination

The architecture of a neural network includes the number of the following components:

- Processing units in the input and output layers
- Hidden layers
- Processing units in each hidden layer

One or more processing units in the input layer represent a numeric value of an input record. Every processing unit is connected to every processing unit in the next layer by a weighted value expressing the strength of the relationship. These weights are called *adaptable connections* because their values are adjusted during training so that the network output approaches the class values present in the data.

Using automatic architecture determination, the mining function evaluates different neural network architectures with different numbers of hidden layers including the processing units in these layers. These alternative models are trained for a fixed number of passes and then the best network architecture is selected for further training.

Automatic architecture determination requires more processing time than manual architecture determination. The function constructs 5 different neural network architectures with 0, 1, and 2 hidden layers. These network architectures are trained for a fixed number of passes over the data. The passes depend on the input size. The maximum number of passes is 50. After the architecture-training passes are completed, the best performing network is selected to undergo additional training.

Using manual architecture determination requires knowledge of the selection criteria for the neural network architecture, because the exact architecture must be specified.

Example: You might want to specify the values in Table 83 for the manual architecture determination.

Table 83. Specifying values for the manual architecture determination

Hidden unit 1	Hidden unit 2	Hidden unit 3
2	3	2

Parameter determination

You can let the system determine values for the learn rate and the momentum or specify your own values.

The learn rate controls to which extent the weights are dynamically adjusted during the training process to improve the model convergence. Larger values indicate higher change so the network trains more quickly with higher values; however, the accuracy is lower.

The momentum adjusts the change applied to a weight by factoring in previous weight updates. It acts as a smoothing parameter that reduces oscillation and helps attain convergence. Lower values mean more training time and more influence of the outlier data on the weights.

Example: You might want to specify the values in Table 84 for the learn rate and the momentum.

Table 84. Sample values for the learn rate and the momentum

Learn rate	Momentum
0.2	0.9

Quantiles

The Neural Prediction mining function allows you to generate quantile ranges.

A quantile is a generalization of a median. The median is the 50%-quantile. For a collection of values, at most 50% of the values are less than the median, and at most 50% of the values are greater than the median. Similarly, if *N* represents a number between 0 and 100, at most *N*% of the values are less than the the *N*%-quantile or greater than the *N*%-quantile, respectively. The 0%-quantile represents the minimum value, and the 100%-quantile the maximum value. Quantiles are particularly useful to define ranges of values.

When quantiles are generated, the values for the predicted field are classified according to the quantile range in which they fall.

Specifying parallel parameters

You can run the Neural Prediction mining function in serial or parallel mode. See “Specifying parallel parameters” on page 118 for more information.

Specifying output fields

The layout of the output data corresponds to the appropriate input data. By default, the output data includes the **Predicted value** field. If you choose to generate quantiles, the output data also includes the following output fields:

- Lower quantile
- Upper quantile

To proceed to the next wizard page, you must enter a field name for the **Predicted value** entry field.

From the **Available** fields list, you can select additional output fields to be included in the output data. See “Specifying output settings” on page 125 for more information.

Example: To generate quantile information, and additionally include the data field Customer number in the output data, specify the values shown in Table 85 on page 239.

Table 85. Specifying output fields for the Neural Prediction mining function

Output fields	Predicted value field name	Lower quantile field name	Upper quantile field name
Customer number	Predicted value	Lower quantile	Upper quantile

Specifying output data

If you specified output fields, you must also specify a data settings object on the Output data page of the wizard or settings notebook. See “Specifying output settings” on page 125 for more information.

Specifying result names

By default, a result object takes on the name of the settings object that created the results. Thus, you do not need to specify a result name when you run the settings object for the first time.

However, you might want to run the settings object again and keep the old results. In this case, you must change the original result name. See “Renaming result objects” on page 128 for more information.

Chapter 18. The statistical functions

The statistical functions of the Intelligent Miner provide various statistical and forecasting methods to support your business decisions. You can use the statistical functions to learn more about your data. This information will help you make better decisions when mining the data. Statistics functions operate on *input data* and produce *output data* and results.

The Intelligent Miner's statistical functions apply various statistical theories and calculations to your input data to discover hidden patterns in your data. You can use the statistical functions in the transformation step and the mining step of the data mining process. Here are some examples of how you might use the statistical functions:

- You might use the Linear Regression statistical function to predict values using a linear fitting model. You might also use Principle Component Analysis to see the more dominant attributes in your data.
- You might also use the statistical functions to learn more about your data. This information helps you make better decisions when mining the data. For example, you might use the Factor Analysis statistical function to reduce the number of variables in your input data.

Considerations when using statistical functions

With statistical functions, you can easily analyze your data using several different statistical methods. You might be tempted to try several different methods without considering the nature of the data. If you use several methods on a set of data, you might get some statistically significant findings by chance. However, you should choose a statistical method based on the nature of the data and the information you can use.

There are many complex statistical methods available. However, you should use and interpret them correctly. Also, you should pay careful attention to the assumptions and limits of each method.

As a rule, you should not attempt to use a statistical method unless you know that your data meets the assumptions for that method and how to interpret the results. If you want to use a method that you are not familiar with, you should first refer to statistics books for more information about that method.

Categorizing your data

Before you can select the appropriate statistical function, you must state your business requirement and know the type of data to analyze.

You might want to test a hypothesis, search for possible trends, gather preliminary information, or refine a research question. For example, to sell car insurances, you might need to decide whether to continue a discount for people who have taken an advanced driving course. You want to know if the cost of claims from drivers who have taken an advanced driving course differs significantly from the cost of claims from drivers who did not take such a course.

You can categorize your data by observing the number of variables, the scale of measurement, and the number of cases. Depending on these criteria, you can select the appropriate statistical function.

Number of variables

A variable is a measureable characteristic in a population that interests you. Each of the statistical functions requires a certain number of variables.

Scale of measurement

Statistical methods require allowable scales of measurement for the data. After you know the allowable scales of measurement for your data, you can narrow down the methods that you can use.

You can measure data using one of the following scales of measurement: Nominal (lowest) measurement makes no assumptions on the values assigned to the data. Each value serves as a label or name for that distinct category. It does not imply anything about the order or distance between numbers. For example, you might use a numerical code to designate cities in your data: 1 is assigned to Paris, 2 to London, and 3 to New York.

Ordinal measurement assumes that you can assign the order to the data, though you cannot measure the distance between numbers. For example, a teacher might rank exams of students: 1 is the best, 2 is intermediate, 3 is bad. This scale tells you that 1 is better than 2, but it does not tell how much better it is.

Interval measurement measures the distance between values and the ranks of the data. However, you cannot make a statement about proportional relationships on the scale, because there is no true zero point. For example, comparing the scales of Celsius and Fahrenheit it becomes clear, that 20 degrees Celsius are not twice as warm as 10 degrees Celsius. 10 degrees Celsius correspond to 50 degrees Fahrenheit. 20 degrees Celsius correspond to 86 degrees Fahrenheit. 86

degrees Fahrenheit are not twice as much as 50 degrees. The ratios are not the same, because both scales have an arbitrary zero point.

Ratio (highest) measurement has a true zero point. All other properties correspond to interval measurement. For example, because time and distance are ratio measures, it is true that a car travelling 50 miles per hour is twice as fast as a car travelling 25 miles per hour. You can use the methods developed for a lower scale of measurement with data on a higher scale of measurement.

Number of observations

Some statistical functions require a minimum number of observations to perform valid calculations. Use the information in the following table to select the appropriate statistical function. In Table 86, I represents an interval scale, R represents a ratio scale, and O represents an ordinal scale.

Table 86. Scales of measurement and number of variables required by the statistical functions

Statistical function	Scale of measurement	Number of variables
Regression	I R	2 or more
Univariate Curve Fitting	I R	1
Principal Component Analysis	I R	2 or more
Factor Analysis	I R	2 or more

Factor Analysis

Factor Analysis discovers the relationships among many variables in terms of a few underlying, but unobservable, random quantities called factors.

Determining the number of factors

You can determine the number of factors by using one of the following methods:

- You can let the system determine the number of factors.
- You can specify the percentage of variance to be explained by the calculated factors.
- You can specify a certain number of factors. This number must be less than or equal to the number of input variables.

Interpreting the factors

To interpret the factors found during a factor analysis, view the Factor Loadings window. The Factor Loadings window offers a graphical representation of the factors, which in many cases makes it easy to identify the most important factors and the underlying variables. See “Viewing factor loadings” on page 263 for more information.

If viewing the Factor Loadings window does not allow a clear interpretation of the factors, you can use a factor rotation to simplify the factor structure. This helps you to better identify the meanings of the calculated factors. The result of a factor rotation is a new factor solution, which produces the same correlation matrix. In the factor space, the reference axes of the factor solution are rotated. You can use either of the following types of rotation:

Quartimax rotation
: The quartimax rotation tends to produce one factor which is mainly correlated with a given variable. The other factors produced tend to be lower correlated than for the varimax rotation.

Varimax rotation
: The varimax rotation maximizes the variance of the factor loadings for each input variable. The rotated factors have a high correlation with one smaller set of input variables and little or no correlation with another set of input variables.

Handling of missing values: When the Factor Analysis function builds a model in training mode, it takes all records into account that contain at least one correlating pair of values. The correlation coefficients are written to a design matrix, which serves as the basis for the calculation of the factors. The design matrix is also called transformation model. The Factor Analysis function includes as much information as possible in the transformation model.

During a phase called scoring, the Factor Analysis function calculates the factors on the basis of the correlation coefficients in the transformation model. Only those records that contain at least one valid value influence the calculation of the factors. Again, all records are checked. If a valid record contains missing values, these values are “replaced” with the mean value of the field or variable in question.

Tip: If your data contains many missing values, it is useful to do a factor analysis as a preparatory step and then run a mining function using the generated factors as input fields. The missing value handling of the Factor Analysis function ensures that the information in records with missing values is retained.

Linear Regression

Use Linear Regression to determine the best linear relationship between the dependent variable and one or more independent variables. The dependent variable is the one that you want to predict, whereas the independent variables are those that you base your prediction on.

"Best" means that you commit the smallest error possible if you base your prediction on the linear function generated by the program.

The general regression formula is:

$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \dots \beta_n x_n + \epsilon$

where *y* denotes the dependent variable, β_0 to β_n denote the unknown coefficients, x_1 to x_n denote the independent variables, and ϵ denotes the error term.

Handling of missing values

When building a prediction model in training mode, the Linear Regression function takes all records into account, except those to which one of the following exception applies:

- The field that represents the *dependent* variable does not contain a valid value.
- None of the fields that represent the *independent* variables contains a valid value.

During the model-building phase, a regression equation is computed, which is then used to predict the values of the dependent variable for each record. The prediction phase is called scoring. During scoring, the Linear Regression function checks all records. If the function scores after a model-building phase, that is, in training mode, the check also includes variables that were previously removed. Records that do not contain a valid value in at least one of the fields representing independent variables are marked as *missing* in the output table.

Using power options with linear regression

You can determine a significance threshold so that independent variables whose probability value is above that threshold are not considered when the regression model is built. Thus only those independent variables that contribute most significantly to the result become part of the model.

A probability value close to 0 marks a highly significant variable; a probability value close to 1 marks an insignificant variable. That is why variables *above* the threshold are not considered.

To use a significance threshold, enter the following expression in the **Power options:** field on the Input data page of a Linear Regression wizard or settings notebook:

```
-stepwise [sig level]
```

Specify a value greater than 0.0 and less than 1.0 for *sig level*. The value that you choose determines the significance threshold. For example, if you specify

-stepwise 0.2, only variables with a probability value of 0.2 or less are considered during the model-building phase. If you do not specify a value for *sig level*, the default value of 0.5 is used.

When you use this power option, the Linear Regression function first selects the most significant independent variable and builds an initial regression model. The function then checks whether this selection diminished the significance of other variables. If this is the case, the variable is removed from the model. After that, the function selects the independent variable with the second highest significance. This process is repeated until all variables whose significance is equal or higher than the specified threshold are discarded or added to the model.

Attention: Use power options with great care. Make sure you read the general introduction in "Using power options" on page 117.

Principal Component Analysis

Principal Component Analysis seeks the standardized linear combination of the original variables. You can use this type of analysis to summarize data and identify linear relationships among variables. With Principal Component Analysis, you can reduce the number of variables in regression, clustering, and other analysis methods for multivariate data.

Handling of missing values: When the Principal Component Analysis function builds a model in training mode, it takes all records into account that contain at least one covariant or correlating pair of values. The covariance values or correlation coefficients are written to a design matrix, which serves as the basis for the calculation of the components. The design matrix is also called transformation model. The Principal Component Analysis function includes as much information as possible in the transformation model.

During a phase called scoring, the Principal Component Analysis function calculates the components on the basis of the covariance values or correlation coefficients in the transformation model. Records that contain one or more valid values influence the calculation of the components. Only records that are completely empty do not influence the calculation. Again, all records are checked. If a valid record contains missing values, these values are replaced with the mean value of the field or variable in question.

Tip: If your data contains many missing values, it is useful to do a principal component analysis as a preparatory step and then run a mining function using the generated components as input fields. In general, the function generates as many components as it processed input variables. The components do not contain any missing values.

Univariate Curve Fitting

Univariate Curve Fitting finds a mathematical function that closely describes the distribution of your data over time. You can select the following curve types:

- Best Fit
- Exponential
- Hyperbola
- Linear
- Power
- Rational
- Reciprocal

Handling of missing values: Missing values are not allowed in the time-series data.

Using power options with univariate curve fitting

Sometimes, you are interested in the continuous development of your time-series data. Here seasonal highs and lows can distort the overall picture. You can use a power option to get a curve without seasonal trends. Enter the following expression in the **Power options** field on the Input data page of a Univariate Curve Fitting wizard or settings notebook:

```
-removetrend
```

Attention: Use power options with great care. Make sure you read the general introduction in "Using power options" on page 117.

Bivariate Statistics

You can use the Bivariate Statistics function to do the following tasks:

- Calculate basic statistics for numeric fields, such as the maximum, minimum, mean, variance, and frequencies.

  Frequencies for continuous numeric fields are calculated for values within bucket limits. If you do not specify your own bucket limits, the Intelligent Miner determines them automatically.
- Calculate frequencies for categorical fields and discrete numeric fields.
- Perform the Chi-square test for a selected field. If you select a field, the Chi-square value is calculated for all existing combinations of this field with other fields.
- Perform the F-test. The F-test is applied to all pairs of numeric fields.
- Compute quantiles for selected numeric fields.
- Take samples out of the input data.
- Copy input data to an output table, possibly as a sample.

Using power options with bivariate statistics

You can compute the complement of a sample by using a power option. By computing the complement of a sample, you can divide the input data into sets of training and test data. However, this only works if the order of the input records does not change. If you run the Intelligent Miner in parallel mode on DB2 data, you might have to create a flat-file copy of your data and run the Bivariate Statistics function on the flat file. Use the Copy Records to File preprocessing function to create the flat file. To specify the power option, enter the following expression in the **Power options** field on the Input data page of a Bivariate Statistics wizard or settings notebook:

```
-complementOfSample
```

Chapter 19. The preprocessing functions

The preprocessing functions are used to transform data before, between, and after mining runs. You run each function on *input data*, and each function produces *output data*, except for the Run SQL and the Clean Up Data Sources functions.

Input data consists of database tables or database views on a server. Preprocessing functions never modify input data. Output data can be written to database tables or views, except for the Copy Records to File function, which produces files only. To avoid duplication of data, output data usually consists of views. Tables can be used if data replication is desired.

The preprocessing functions might not support every type of output data. For example, the discretization functions only produce tables, while the Copy Records to File function only produces files.

In the transformation step of the data mining process, you can use the Intelligent Miner's preprocessing functions to prepare the data for mining. You might exclude fields or records from the input data that are irrelevant to your data mining purposes, or perform mathematical operations on fields in the input data before mining the data. Here are some examples of how you might use preprocessing functions:

- You might have a large flat file of customer information, including name, address, age, gender, marital status, and income. The data in the flat file might contain blanks or NULL values for some fields, maybe because the customers did not supply all the information. Before using this flat file for mining, you might want to remove or replace missing values so that they do not affect the results of your mining run. You can do this using the Discard Missing Values preprocessing function or the Encode Missing Values preprocessing function.
- If you want to create output data containing the results of a mathematical expression, you can use the Aggregate Values, the Calculate Values, or the Group Records preprocessing function. For example, you might be using a database table containing the monthly salary and monthly commision earned by each of your employees. If you are mining for data related to annual earnings, you might use the Group Records preprocessing function to add the monthly income and monthly commision values for each employee, and produce an output data with new fields containing these values.

The following list provides an overview of each function. See the wizard for more information about these functions.

Aggregate Values

Use this function to produce output data that contains *aggregate* values from the input data. For each aggregation expression you specify, you must also specify a new field name that holds the data created by the aggregation expression. You can provide multiple aggregation expressions and multiple new field names.

The output data consists of a single record, containing the fields you named. Each field contains the value resulting from its corresponding aggregation expression.

This function supports all SQL column functions supported by the database server you are using.

Calculate Values

Use this function to create output data that contains new fields by using SQL expressions. The output data contains all the fields in the input data, plus an additional field for each SQL expression you specify. The output data consists of one record per input record.

This function supports SQL date functions, SQL arithmetic functions, and SQL string manipulation functions. It does not support SQL column functions, which are supported by the Aggregate Values or Group Records functions instead.

Clean Up Data Sources

Use this function to delete database tables or database views from input or output data.

Convert to Lowercase or Uppercase

Use this function to convert one or more fields in the input data to lowercase or uppercase in the output data. The output data contains all the fields in the input data, plus an additional field for each conversion you specify. The output data consists of one record per input record.

Restriction: On OS/390 and OS/400 operating systems, the output data is always in the form of a table. You cannot create views.

Copy Records to File

Use this function to copy records from a database table or database view to a flat file.

You can also use this function to sort the input data records based on the contents of fields you specify. The number of sort fields cannot exceed the maximum for the ORDER BY clause of SQL. You can specify the sort sequence separately for each field as ascending or descending. The first field you specify has the highest sort precedence.

Restriction for OS/390: You can only sort fields of the DB2 data types FLOAT, CHAR, and VARCHAR because the Copy Records to File function uses an ORDER BY clause. To change the DB2 data types in your input table, you can use the Calculate Values preprocessing function. You can also define a mathematical expression in connection with a computed field so that the data types are converted. See "Calculate Values" on page 250 and "Computed fields" on page 90 for more information.

The output data contains all the fields in the input data. The output data consists of one record per input record, sorted by the fields you specify.

If you want to copy only a few of the fields in the input data to a file, proceed as follows:

1. Run the Filter Fields preprocessing function first to create a table or view with just those columns.
2. Run Copy Records to File on the output table or view created by the Filter Fields preprocessing function.

If the input data resides in a DB2 Parallel Edition database, you can use this function to create partitioned output files by running it in the parallel mode.

Important:

1. If you use the Intelligent Miner for iSeries, this functions writes records to stream files in the Integrated File System (IFS).
2. The Intelligent Miner treats blank strings in database tables as valid values and blank strings in flat files as missing values. This might give inconsistent results when you process your data after you used the Copy Records to File function.

   For blank strings to be still treated as valid values after the conversion, you have to map them to valid flat-file values before you use the Copy Records to File function. Use the Map Values preprocessing function for this purpose.

Discard Records with Missing Values

Use this function to remove input data records containing a missing (NULL) value in any of the fields you specify. The output data includes all fields from the input data.

Discretization into Quantiles

Use this function to assign input data records to the number of quantiles you specify. The function assigns the input data records, as evenly as possible, to quantiles in the output data.

The output data is always in the form of a table.

The output data consists of one record per input record. An attempt is made to limit the output data to a set of fields that guarantee uniqueness with respect to their values. Precedence is given to a primary key defined on the input data; if necessary, other unique indexes defined on the input data are checked.

Discretization Using Ranges

Use this function to assign the input data records by splitting the value range of a continuous field into intervals and then mapping each interval to a discrete value.

The output data is always in the form of a table.

The output data consists of one record per input record. An attempt is made to limit the output data to a set of fields that guarantee uniqueness with respect to their values. Precedence is given to a primary key defined on the input data; if necessary, other unique indexes defined on the input data are checked.

Encode Missing Values

Use this function to encode missing (NULL) values in the input data by specifying one or more fields to search for missing values. For each of the fields being searched, you must specify an encoding value and a new field name. The output data contains all the fields in the input data, plus a new field for each field being searched.

The output data consists of one record per input record. Each new field in an output record contains the corresponding encoding value if the corresponding field in the input record is NULL. If the value of a field in the input record is not NULL, this value is transferred to the corresponding new field in the output record.

Encode Nonvalid Values

Use this function to encode values found in the first input data if they do not match valid values from the second input data. The output data contains all the fields in the first input data, plus an additional field containing the encoding value.

The output data consists of one record per input record. The new field in the output record contains the value of the corresponding field in the input record if that value matches one of the valid values. Otherwise, it contains the encoding value.

Filter Fields

Use this function to create output data that contains only the fields you specify, or that contains all the fields except those you specify.

The output data consists of one record per input record.

Filter Records

Use this function to create output data that contains only the records that meet the filtering conditions you specify. You can use this function to discard the records that contain invalid values or values that fall outside of an acceptable range.

The output data contains all the fields from the input data. The output data consists of one record per input record that satisfies the specified filtering condition.

Filter Records Using a Value Set

Use this function to compare field values in a first input data with values in a value set specified for a second input data. The function creates output data that contains either of the following record types:

- Records for which the value of a specified field is found in the value set (matching records)
- Records for which the value of a specified field lies outside the value set (nonmatching records)

The matching records option determines whether matching records are included or excluded from the output data.

The value set used to determine if an input record is a matching record is comprised of the values in the valid values field in the second input data.

The output data contains all the fields from the input data.

Get Random Sample

Use this function to reduce an input data to a smaller sample by specifying the size of the sample as a percentage of the input data. The output data contains the same fields as the input data, but a smaller number of records.

Restriction: On OS/390 and OS/400 operating systems, the output data is always in the form of a table. You cannot create views.

Group Records

Use this function to summarize groups of records into a single record that contains aggregated values for the group. The output data contains the grouping fields you specified and a field for each aggregation expression and new field name you specified.

The output data consists of one record per unique combination of grouping field values found in the input data.

Join Data Sources

Use this function to join two database tables or database views based on one or more pairs of join fields from the input data. The data types of the two fields in each pair of join fields must be compatible.

Map Values

Use this function to map values found in the first input data to values found in the second input data. If the values in the specified fields match, this function copies a mapping value you specify to the output data. If the fields do not match, this function either copies the original field values or NULL values to the output data.

The output data consists of one record per input record.

Pivot Fields to Records

Use this function to split each record of the input data into multiple records.

The fields you specify to repeat are repeated in each record, and each field to pivot becomes part of a single, new record. If there are NULL values in the fields being pivoted, this function preserves the NULL values in the output data. You can then use another preprocessing function to discard or encode the records with NULL values in the output data.

Restriction: On OS/390 and OS/400 operating systems, the output data is always in the form of a table. You cannot create views.

Run SQL

Use this function to enter and submit SQL statements to the database server for immediate processing. Do not use this function to run SELECT statements. This function supports all SQL column functions supported by the database server you are using.

Creating indexes for output data

If the input data contained indexes, some preprocessing functions attempt to create the same *indexes* for the output data. An index for the output data is created when the following criteria are met:

- The input data consists of a database table (views are not suitable).
- **Table** is specified as the output data format.
- All index fields in the input data are included in the output data.

Important: With the Intelligent Miner for AIX, the creation of an output index fails if any of the index field names in the input data contains a **+** or a **–** symbol. These symbols cause ambiguity when creating an index.

If the preprocessing function does not create an index, a warning is given, but the function continues processing.

Chapter 20. Viewing results

The Intelligent Miner V8.1 provides new Java visualizers for the following mining functions:

- Associations mining function
- Classification mining function
- Clustering mining function

You can use the new Java visualizers or the visualizers that were provided with the Intelligent Miner V6.1.1. By default, the new Java visualizers are displayed when you are viewing results. For more information about the new visualizers, see *IBM DB2 Intelligent Miner Visualization: Using the Intelligent Miner Visualizers*.

The visualizers that were provided with the Intelligent Miner V6.1.1 are described in this chapter. If you want to use these visualizers, you must specify them in the Preferences notebook.

For example, if you want to use the Associations visualizer Intelligent Miner V6.1.1, follow these steps:

1. On the Intelligent Miner main window, select **Options** → **Preferences ...** → **Visualizers**.
2. On the Visualizers page of the Preferences notebook, specify the following options:
   a. Select Result - Association from the **Result type** list.
   b. Select Browse Associations from the **Result format and visualizer** list.
3. Click **OK** to close the Preferences notebook.

When the visualizers of the Intelligent Miner V6.1.1 are specified in the Preferences notebook, and you want to use the visualizers of the Intelligent Miner V8.1, you must select *IM Visualization V8.1 from the **Result format and visualizer** list on the Visualizers page of the Preferences notebook. The * in front of the IM Visualization V8.1 entry denotes that this visualizer is the default settings.

When a mining run is completed, you can open a result window that gives you a general overview first. Changing the representation of the results allows you to go into detail or focus on specific aspects.

Most result viewers offer the possibility to print results. In general, the standard print panel of your client operating system comes up when you select the print option. Table 87 shows the functions that produce printable results.

Table 87. Functions that produce printable results

Mining	Statistics
Tree Classification	Bivariate Statistics
Demographic Clustering	Factor Analysis
Neural Clustering	Linear Regression
RBF-Prediction	Principal Component Analysis
Neural Prediction	Univariate Curve Fitting
Similar Sequences	

The same functions, except for Similar Sequences, allow you to copy textual results to the clipboard. On a Windows client, you can also copy graphical results to the clipboard.

In addition, some result viewers are equipped with a *Print report* facility. Reports allow you to select the parts of the results that you want to print.

With the Associations and Sequential Patterns mining functions, you can print the result statistics.

Viewing statistics results

You can view the statistics you calculated with the statistics result viewer.

There are several ways to start the result viewer:

- You can click the **View Result** button in the Progress window when a statistics function has finished processing.
- You can double-click the icon of an existing results object in the Contents container or the Workarea.
- You can customize your Intelligent Miner user interface so that the result viewer is started automatically after you ran a statistics settings object. To do so, follow these instructions:
  1. Click **Options → Preferences** on the menu bar in the Intelligent Miner main window. The Preferences notebook opens.
  2. Check **After a settings object is run, automatically visualize its results if it produces a result** on the Miscellaneous page.
  3. Click **OK** to save this setting and close the Preferences notebook.
- You can enter the following command on the command line to use the statistics result viewer alone, without the Intelligent Miner:

  ```
  idmgvchv <filename>
  ```

  where <filename> stands for a result file.

If you do not specify a filename, a file selection box is displayed before the result viewer to let you choose a statistics result file. You can select all kinds of statistics result files.

You find the idmgvchv program in the directory identified by the $IDM_BIN_DIR environment variable on the client.

The menu bar of the statistics result viewer contains the following menus:

File Select **Save As** to store the results in a file on your local client workstation.

Select **Print** to print the visible part of the current view. For windows containing graphics, only the visible parts are printed.

Select **Export** to store the contents of the current view in a comma-separated file (CSV) for processing with applications other than the Intelligent Miner.

Select **Exit** to close the statistics result viewer.

Edit Select **Copy** to copy the contents of the current view to the clipboard. You can copy text data, but not graphical data. The data is copied in CSV format. Column entries are separated by tabs.

View You can change the mode in which the results are presented by selecting a different view from this menu. The range of available choices on the menu varies with the statistics calculated.

Wherever you see a table or matrix, you can sort the column entries by clicking on the column header. The sorting direction is reversed when you click the column header again. Note that the columns are sorted alphabetically. Thus a sort might lead to unexpected results.

Using the zoom function

Many statistical functions produce charts as results, and the results of some functions allow you to enlarge the charts or certain areas on those charts. This is helpful if the coordinates on a chart, represented by dots, concentrate in one place or overlap each other. The enlarged view allows you to clearly identify the position of the coordinates. You can use the following enlargement methods:

- To enlarge the whole chart gradually:
  1. Place the mouse pointer somewhere on the chart.
  2. Click the right mouse button. A pop-up menu opens.

3. Click **Zoom in**. The area around the center of the chart becomes larger, while areas close to the edge of the circle become invisible.

   You can repeat these steps to further enlarge the chart. To get back to the initial view, repeat steps 1 and 2 and click **Zoom out** on the pop-up menu.

- To enlarge a marked area on the chart:
  1. Place the mouse pointer close to the area that you wish to enlarge.
  2. Hold down the Shift key and the left mouse button.
  3. Drag the mouse over the area that you want to enlarge. The marked area shows as a rectangle on the chart.
  4. Release the left mouse button. The marked area expands to the full size of the chart.

     You can repeat these steps to enlarge areas in your enlargements. To get back to the initial view:

     a. Place the mouse pointer somewhere on the chart.
     b. Click the right mouse button. A pop-up menu opens
     c. Click **Zoom out** on the pop-up menu.

Viewing Factor Analysis results

Factor analysis is a dimension reduction technique based on the estimated correlation matrix of the input variables. The factor analysis viewer displays charts representing the percentage of factors. See Figure 15 on page 261.

The bar chart in the upper part of the window shows to what degree each factor contributed to the variance in the overall data. The pie chart shows you the part of the variance that can be ascribed to all the factors. The slice that sticks out of the pie chart represents the residual variance.

When you click the name of a factor in the Labels list, the number of that factor appears next to the corresponding slice in the pie chart. This way, you can identify the part of the variance that can be ascribed to a particular factor.

As an OS/2 or Windows user, you can also click slices in the pie chart. This has the same effect as clicking the names of factors in the Labels list.

When you click the **Select all** button, all items in the Labels list are highlighted, and the numbers of the factors are displayed for all the slices. Clicking **Deselect all** restores the initial view.

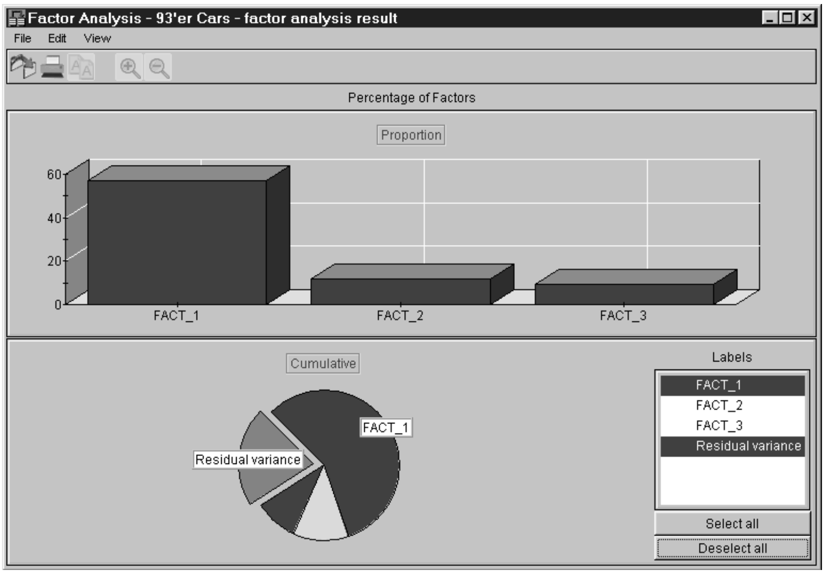


Figure 15. The Percentage of Factors window

Viewing correlation coefficients of input variables

The correlation matrix of input variables shows the Pearson product correlation coefficients for each pair of variables. The calculated values describe the strength of linear association between pairs of variables. The Pearson coefficient has a value between -1 and 1.

The Pearson coefficient for the correlation of a variable with itself is always 1. If you select **No change** for the leading diagonal on the Additional parameters page of a Factor Analysis wizard or settings notebook before you run the function, the leading diagonal contains the value 1 in all positions. You can select between two further options for the leading diagonal, so that the 1s are replaced with the maximum absolute row value or the squared multiple correlation coefficient.

To view the correlation coefficients of the input variables as a table, click **View → Input Variables → Correlation Coefficients of Variables → Table** on the menu bar of the statistics result viewer.

To view the correlation coefficients of the input variables as charts, click **View → Input Variables → Correlation Coefficients of Variables → Chart**.

Note: The charts only display the eight most important correlation coefficients.

To return to the Percentage of Factors window, click **View → Factors → Percentage of Factors → Chart**.

Viewing eigenvalues of input variables

This table shows the eigenvalues of the estimated correlation matrix. Eigenvalues are coefficients of a diagonalized correlation matrix, meaning that the original matrix was converted into a new matrix that has values only in the leading diagonal. As a result, the coefficients in the leading diagonal change because after the conversion they include the values that were in the other cells.

To view the eigenvalues of the input variables, click **View → Input Variables → Eigenvalues of Variables** on the menu bar of the statistics result viewer.

To return to the Percentage of Factors window, click **View → Factors → Percentage of Factors → Chart**.

Viewing the Percentage of Factors matrix

You can also display the percentage of factors in the matrix form. The matrix shows how many factors you need to explain a certain percentage of the variance in the input data. The Percentage of Factors matrix has the following columns:

Factor	This column contains the names of the calculated factors.
Percentage	The percentages in this column show to what degree each factor contributed to the variance in the overall data.
Cumulative	This column contains the sums of previous and the current factors' percentages. The cumulative value for the last factor shows in percent the total variance that can be explained by the discovered factors. A higher value indicates a more precise model.

To display the Percentage of Factors matrix, click **View → Factors → Percentage of Factors → Table** on the menu bar of the statistics result viewer.

To return to the Percentage of Factors window, click **View → Factors → Percentage of Factors → Chart**.

Viewing eigenvectors

This matrix shows the eigenvectors corresponding to the eigenvalues. The eigenvectors generate a subspace of the variable space.

To view the eigenvectors of the factors, click **View → Factors → Eigenvectors of Factors in the Variable Space** on the menu bar of the statistics result viewer.

To return to the Percentage of Factors window, click **View → Factors → Percentage of Factors → Chart**.

Viewing factor loadings

Factor loadings describe the linear relationship between the input variable and the determined factors. The values of factor loadings are between -1 and 1 because their calculation is based on the correlation matrix of the input variables. If the factor loading is 1, there is a perfect positive relationship between the variable and the factor. If it is -1, there is a perfect negative relationship. If the factor loading is 0, there is no relationship between the input variable and the factor. The Factor Loadings window is shown in Figure 16.
When the Factor Loadings window opens, the Y-axis represents factor 1, and

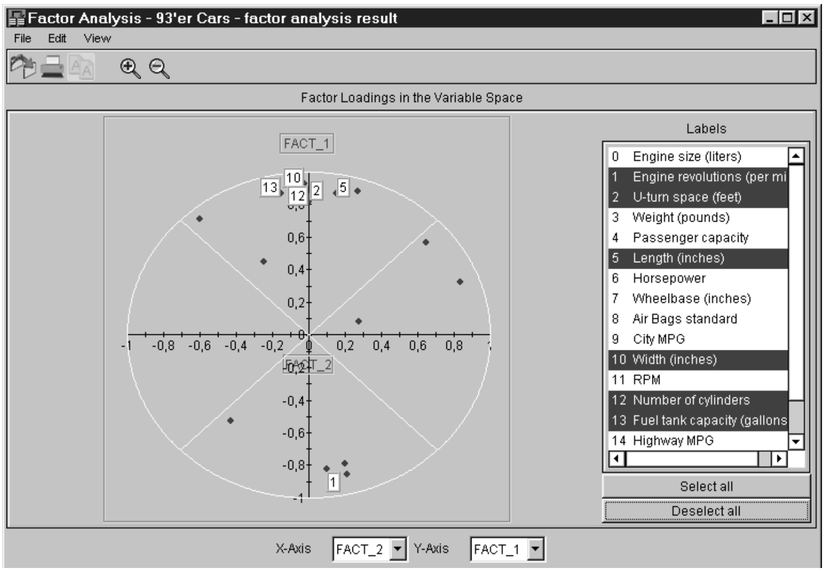


Figure 16. The Factor Loadings window

the X-axis factor 2. The dots depict the factor loadings. The labels next to the dots show the numbers of the input variables. By looking up a number in the Labels list on the right, you can identify the name of the variable belonging to a particular factor loading. If a dot has a high coordinate value on one of the axes and lies in close proximity to it, there is a distinct relationship between one of the two factors and a certain variable. Clicking the button next to X-Axis opens a drop-down list from which you can select a different factor for the X-axis. Analogously, the same applies when you click the button next to Y-Axis. The names of the selected factors are displayed on the buttons.

Clicking the **Deselect all** button in the lower right corner removes all the labels from the graph. By clicking individual variable names in the Labels list, you can display the numbers of only a few factor loadings. Alternatively, you can click individual dots in the graph, which displays the numbers of the selected dots and highlights the variable names in the Labels list. Clicking the **Select all** button restores the initial view.

To view the factor loadings, click **View → Factors → Factor Loadings in the Variable Space → Chart** on the menu bar of the statistics result viewer.

Note: The zoom function is available for this this chart. See "Using the zoom function" on page 259 for more information.

You can also display the numeric values of the factor loadings. To do so, click **View → Factors → Factor Loadings in the Variable Space → Table**.

To return to the Percentage of Factors window, click **View → Factors → Percentage of Factors → Chart**.

Viewing communalities (cos square)

This matrix gives you the communalities of the input variables. The communality has a range between 0 and 1. The value describes to what extent all detected factors contribute to the variance of a single variable.

The communality is defined as the squared cosine of the angle between the vector of the input variable and the projected vector.

To view communalities, click **View → Factors → Communalities (cos square) of Variables** on the menu bar of the statistics result viewer.

To return to the Percentage of Factors window, click **View → Factors → Percentage of Factors → Chart**.

Viewing rotations

You can view rotations only if you select **Use a varimax rotation** or **Use a quartimax rotation** on the Additional parameters page of a Factor Analysis wizard or settings notebook before you run the function. Rotation means that you rotate the reference axes of the factor solution, so that a variable correlates high with only one factor, and low or not at all with the other factors. Rotation is often necessary to make the results of a factor analysis interpretable. The result of a rotation produces a new factor solution, but is based on the same correlation matrix as the original solution.

To view rotations, click **View → Rotation** on the menu bar of the statistics result viewer. From the submenu that is displayed, select one of the following choices:

Iterations of Rotation

When you select this menu choice, a table displays the number of iterations, and for each iteration gives you the overall variance attributable to the factors.

Rotated Factor Loadings in the Variable Space

When you select this menu choice, a matrix shows the changed factor loadings for each factor/variable combination. You can display a table or a chart.

Percentage of Variance for Each Factor

When you select this menu choice, a matrix displays the variance attributable to each single factor in percent.

Communalities (cos square) of Variables

When you select this menu choice, you can view a table that shows the communalities of the original and the rotated factors in the first and second column respectively. The third column gives you the differences between the communalities.

To return to the Percentage of Factors window, click **View → Factors → Percentage of Factors → Chart** on the menu bar of the statistics result viewer.

Viewing regression coefficients

This matrix displays the regression coefficients calculated for each variable/factor combination. The regression coefficients are calculated in training mode. They are used to determine factor scores (in application mode) when the model created in training mode is applied to new input data.

To view the regression coefficients, click **View → Regression coefficients variable to factor** on the menu bar of the statistics result viewer.

To return to the Percentage of Factors window, click **View → Factors → Percentage of Factors → Chart**.

Viewing the factor structure

You can only view the factor structure if you select **Show factor structure** on the Additional parameters page of a Factor Analysis wizard or settings notebook before you run the function. To view the factor structure, select **View → Structure of Factors** on the menu bar of the statistics result viewer. From the submenu that is displayed, select one of the following choices:

Sorted Factor Loadings

When you select this menu choice, a table shows the factor loadings in descending order.

Rearranged Correlation Matrix

When you select this menu choice, the correlation matrix shows the correlation coefficient for each pair of variables in percent. The leading diagonal shows the most frequent correlated factor.

To return to the Percentage of Factors window, click **View → Factors → Percentage of Factors → Chart**.

Exportable views

Table 88 lists the views that you can export or copy to the clipboard in comma-separated variables format (CSV). Note that you can only export or copy tables, but not charts.

Table 88. Exportable views of Factor Analysis results

Input Variables	Factors	Rotation	Structure of Factors
Correlation Coefficients of Variables	Percentage of Factors	Iterations of Rotation	Sorted Factor Loadings
Eigenvalues of the Correlation Matrix	Eigenvectors of Factors in the Variable Space	Rotated Factor Loadings in the Variable Space	Rearranged Correlation Matrix
	Factor Loadings in the Variable Space	Percentage of Variance for Each Factor	
	Communalities (cos square) of Variables	Communalities (cos square) of Variables	
		Regression Coefficients Variable to Factor	

Viewing Principal Component Analysis results

Principal component analysis is also a dimension-reduction technique. Contrary to factor analysis, principal component analysis tries to transform the vector describing the original variables linearly into a lower-dimensional subspace. Depending on the selections on the Parameters page of a Principal Component Analysis wizard or settings notebook, the principal components are calculated on the basis of:

- The correlation matrix of the input fields

  If you select this method, the principal components take on values between -1 and 1.
- The covariance matrix of the input fields

  If you select this method, the range of values for the principal components varies.

The principal components result viewer, as shown in Figure 17, displays charts representing the principal attributes.
The bar chart in the upper part of the window shows the proportion of each

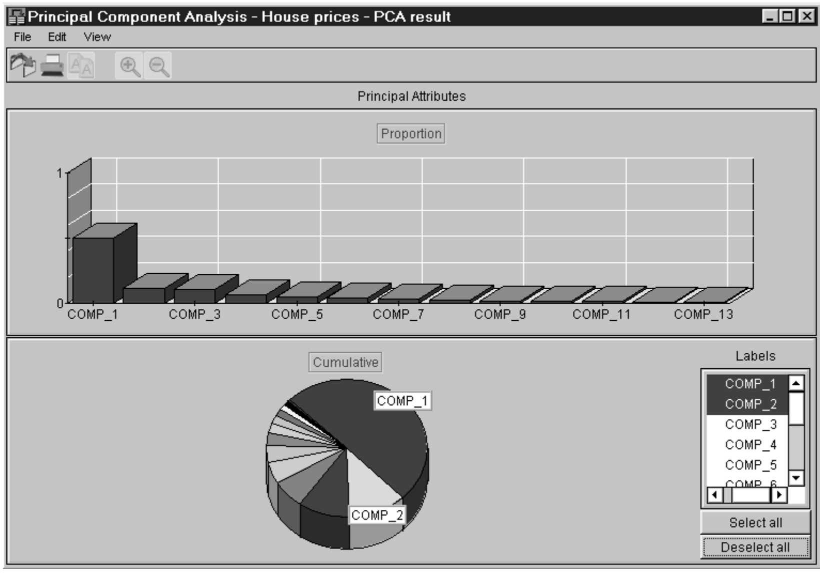


Figure 17. The Principal Attributes window

principal component. The pie chart represents the cumulative values of the principal components.

When you click the name of a component in the Labels list, this name also appears next to the corresponding slice in the pie chart. This way, you can identify a principal component in the pie chart.

As an OS/2 or Windows user, you can also click slices in the pie chart. This has the same effect as clicking component names in the Labels list.

When you click the **Select all** button, all items in the Labels list are highlighted, and the numbers of the components are displayed for all the slices. Clicking **Deselect all** restores the initial view.

Viewing correlation coefficients of input variables

The correlation matrix of input variables shows the Pearson product correlation coefficients for each pair of variables.

To view the correlation coefficients as a table, click **View → Correlation Coefficients of Variables → Table** on the menu bar of the statistics result viewer.

To view the correlation coefficients as charts, click **View → Correlation Coefficients of Variables → Chart**.

Note: The charts only display the eight most important correlation coefficients.

To return to the Principal Attributes window, click **View → Principal Attributes → Chart**.

Viewing covariances of input variables

The covariance matrix of input variables shows the covariances for each pair of variables in the same manner as the correlation matrix.

To view the covariances as a table, click **View → Covariance of Variables → Table** on the menu bar of the statistics result viewer.

To view the covariances as charts, click **View → Covariance of Variables → Chart**.

To return to the Principal Attributes window, click **View → Principal Attributes → Chart**.

Viewing the Principal Attributes matrix

You can display the principal attributes in the matrix form. The Principal Attributes matrix has the following columns:

Eigenvalue	This column contains the eigenvalues for each calculated principal component. Eigenvalues are coefficients of a diagonalized correlation or covariance matrix.
Difference	This column contains the differences between two successive eigenvalues, which are sorted in descending order.
Proportion	This column contains values between 0 and 1 used to measure the degree of information that can be explained by the components.
Cumulative	This column contains the cumulative sums of the proportions. Each row thus contains values calculated by adding the proportion in the current row to the proportions in the rows before.

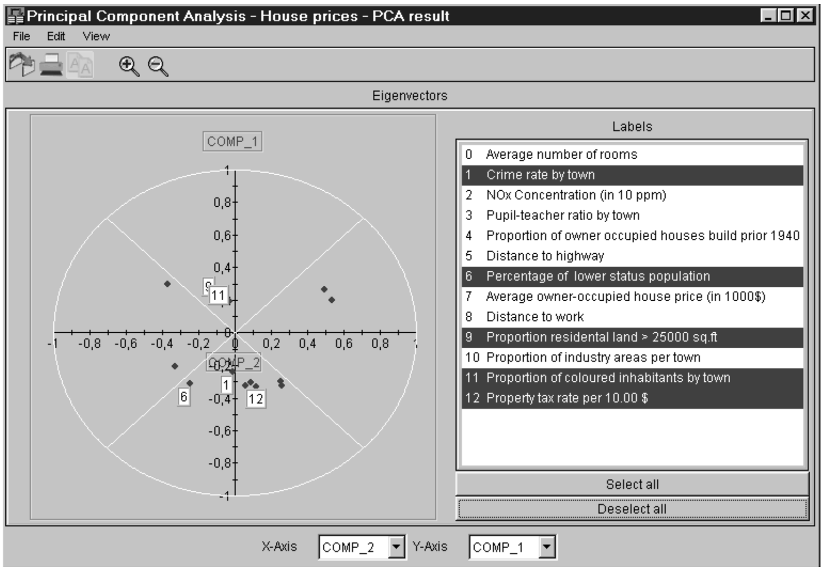
To view the Principal Attributes matrix, click **View → Principal Attributes → Table** on the menu bar of the statistics result viewer.

To return to the Principal Attributes window, click **View → Principal Attributes → Chart**.

Viewing eigenvectors

This chart shows the relationship between principal components and input variables. For this purpose, the eigenvectors are used. See the Eigenvectors window in Figure 18.

When the Eigenvectors window opens, the Y-axis represents principal



0 Average number of rooms
1 Crime rate by town
2 NOx Concentration (in 10 ppm)
3 Pupil-teacher ratio by town
4 Proportion of owner occupied houses build prior 1940
5 Distance to highway
6 Percentage of lower status population
7 Average owner-occupied house price (in 1000$)
8 Distance to work
9 Proportion residental land > 25000 sq.ft
10 Proportion of industry areas per town
11 Proportion of coloured inhabitants by town
12 Property tax rate per 10.00 $

Figure 18. The Eigenvectors window

component 1, and the X-axis principal component 2. The dots depict the eigenvectors. The labels next to the dots show the numbers of the input variables. By looking up a number in the Labels list on the right, you can identify the name of the variable belonging to a particular eigenvector. If a dot has a high coordinate value on one of the axes and lies in close proximity to it, there is a distinct relationship between one of the two components and a certain variable. Clicking the button next to X-Axis opens a drop-down list from which you can select a different component for the X-axis. Analogously, the same applies when you click the button next to Y-Axis. The names of the selected components are displayed on the buttons.

Clicking the **Deselect all** button in the lower right corner removes all the labels from the graph. By clicking individual variable names in the Labels list, you can display the numbers of only a few eigenvectors. Alternatively, you can click individual dots in the graph, which displays the labels of the selected dots and highlights the variable names in the Labels list. Clicking the **Select all** button restores the initial view.

To view the eigenvectors of the components, click **View → Eigenvectors → Chart** on the menu bar of the statistics result viewer.

Note: The zoom function is available for this this chart. See "Using the zoom function" on page 259 for more information.

You can also display the numeric values of the eigenvectors. To do so, click **View → Eigenvectors → Table**.

To return to the Principal Attributes window, click **View → Principal Attributes → Chart**.

Exportable views

Table 89 lists the views that you can export or copy to the clipboard in comma-separated variables format (CSV).

Table 89. Exportable views of Principal Component Analysis results

Correlation Coefficients of Variables	Covariance Coefficients of Variables	Principal Attributes	Eigenvectors
Table	Table	Table	Table

Viewing Linear Regression results

The result viewer for Linear Regression shows the Fitting Chart window. The Fitting Chart window consists of two charts representing the fitted values and the observed values in different orders. See Figure 19.

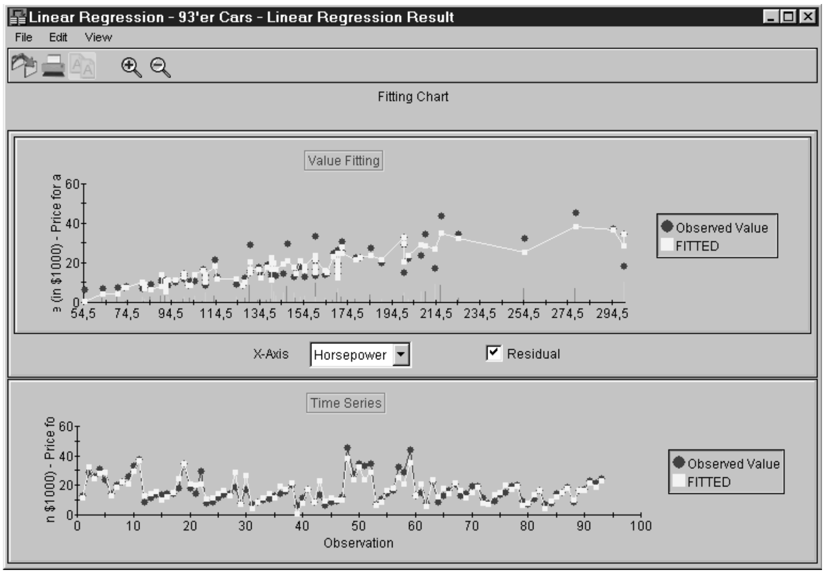


Figure 19. The Fitting Chart window

You can view the following charts:

Value Fitting	By default, this chart shows the fitted values and the observed values in the order of the values for the first independent variable. To change the first independent variable, select another variable from the **Choice** drop-down list under the chart. Your choice is limited to one of the first 50 independent variables because this is the maximum that the result file covers.
Time Series	This chart shows the fitted values and the observed values in the order of observations.

Note: The zoom function is available for this these charts. See "Using the zoom function" on page 259 for more information.

Viewing the Linear Regression Table

The Linear Regression Table shows you the estimates for β_0 to β_n in the linear regression model.

The general statistical model is:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \ldots \beta_n x_n + \epsilon$$

where y denotes the dependent variable, β_0 to β_n denote the unknown coefficients, x_1 to x_n denote the independent variables, and ϵ denotes the error term. The first row in the table shows the results for β_0. It is not associated with any specific variable, and thus marked as a constant.

The Linear Regression Table shows the following columns:

Variable	This column contains the names of the independent variables.
Coefficient	This column contains the estimates for the β-values belonging to the independent variables.
Std Error	This column gives you the standard error for each variable.
Beta Coefficient	This column gives you the inversed β-values.
F Value	The value of the test statistic used to test the hypothesis that all coefficients in the linear regression model (except the constant term β_0) are zero. You can use this value to determine the level of significance.
Probability > F	The probability of obtaining a larger F value than the value shown.

	If the probability of obtaining a larger F value is less than 0.05, the regression is conventionally considered to be significant; that is, there is statistical evidence for rejecting the null hypothesis. When the probability is less than 0.01, the regression is highly significant, and there is strong evidence for rejecting the null hypothesis.

Note: The F-test whose results are shown in the Anova matrix is used to check whether there is a relationship between dependent and independent variables. Here, it is used to find out the independent variables that contributed mostly to the validity of the regression model.

To view the Linear Regression Table, click **View → Linear Regression Table** on the menu bar of the statistics result viewer.

To return to the Fitting Chart window, click **View → Fitting Chart**.

Viewing the Linear Regression Anova window

The Linear Regression Anova window consists of two parts. The upper part shows the variable name and two key values, and the lower part the Analysis of Variance (Anova) matrix.

The following fields are displayed in the box in the upper half of the window:

Dependent Variable Name
Contains the name of the dependent variable used for the mining run.

R-Squared
R-Squared is the square of the estimated multiple correlation coefficient. It measures how good the linear regression model is. R-Squared has a value between 0 and 1. Its closeness to unity indicates a good fitting.

Standard Error
This is the standard deviation of the error term in the linear regression model.

The Analysis of Variance (Anova) matrix in the lower half of the window shows the significance of the regression in the following columns:

Source of Variation	The name of each component contributing; that is, Regression, Residual, and Total.
Degrees of Freedom	The value of the degrees of freedom for each

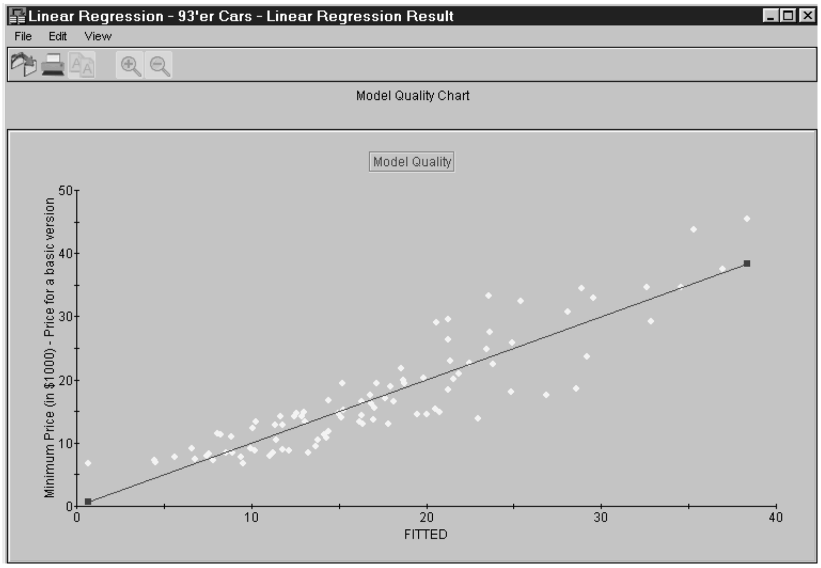
	component in the table. Degrees of freedom are given for Regression, Residual, and Total sum of squares.
Sum of Squares	The sum of squares attributable to each component. A correction factor has been subtracted from each crude sum of squares to give values corrected due to the mean.
Mean Square	The value found by dividing the Sum of squares by the Degrees of freedom. This is sometimes called the estimated variance.
F Value	The value of the test statistic used to test the hypothesis that all coefficients in the linear regression model (except the constant term β_0) are zero. You can use this value to determine the level of significance.
Probability > F	The probability of obtaining a larger F value than the value shown. If the probability of obtaining a larger F value is less than 0.05, the regression is conventionally considered to be significant; that is, there is statistical evidence for rejecting the null hypothesis. When the probability is less than 0.01, the regression is highly significant, and there is strong evidence for rejecting the null hypothesis.

To view this window, click **View → Linear Regression Anova** on the menu bar of the statistics result viewer.

To return to the Fitting Chart window, click **View → Fitting Chart**.

Viewing the Model Quality Chart

This chart shows the deviation of the fitted values from the observed values. The diagonal represents the observed values. The closer the points are to the diagonal, the better is the fitting. See Figure 20 on page 274.



Linear Regression - 93'er Cars - Linear Regression Result
File
Edit
View
Model Quality Chart
Model Quality
Minimum Price (in $1000) - Price for a basic version
FITTED
0
10
20
30
40
50

Figure 20. The Model Quality Chart window

To view this window, click **View → Model Quality Chart** on the menu bar of the statistics result viewer.

To return to the Fitting Chart window, click **View → Fitting Chart**.

Note: The zoom function is available for this this chart. See "Using the zoom function" on page 259 for more information.

Viewing the Residual Chart

The Residual Chart shows two charts representing the residuals (differences between the fitted and the observed values) in different orders. See Figure 21 on page 275.

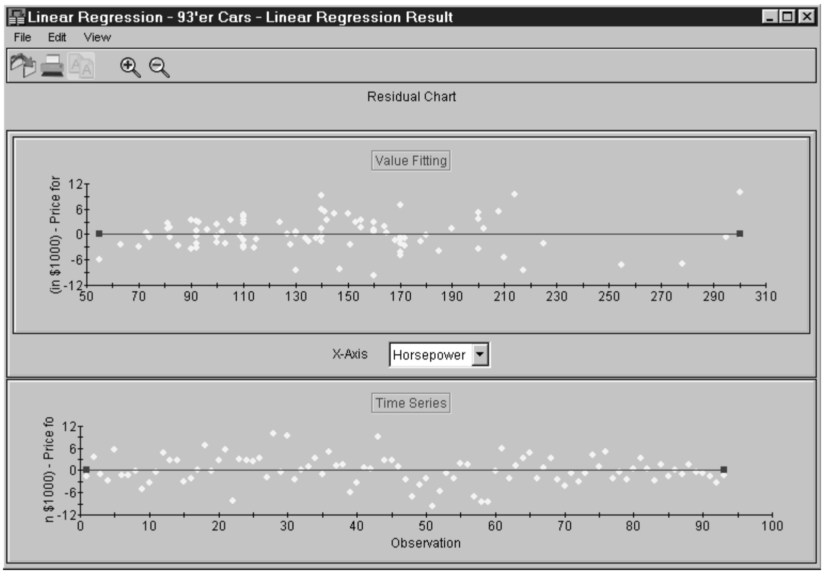


Figure 21. The Residual Chart window

You can view the following charts:

Value Fitting By default, this chart shows the residuals in the order of the values for the first independent variable. To change the first independent variable, select another variable from the **Choice** drop-down list under the chart. Your choice is limited to one of the first 50 independent variables because this is the maximum that the result file covers.

Time Series This chart shows the residuals in the order of observations.

Note: The zoom function is available for this these charts. See "Using the zoom function" on page 259 for more information.

To view the Residual Chart, click **View → Residual Chart** on the menu bar of the statistics result viewer.

To return to the Fitting Chart window, click **View → Fitting Chart**.

Viewing the Fitting Table

The Fitting Table shows the values on which the charts are based. The table contains the following columns:

1. The first column shows the numbers of the observations on which the charts are based.
2. The second column shows the observed values.
3. The third column shows the fitted values.

4. The fourth column shows the residuals. The residuals are the differences between the observed values and the fitted values.
5. The fifth and any further columns show the first 50 independent variables.

To view the Fitting Table, click **View → Fitting Table** on the menu bar of the statistics result viewer.

To return to the Fitting Chart window, click **View → Fitting Chart**.

Exportable views

The following list shows the views that you can export or copy to the clipboard in comma-separated variables format (CSV):

- Linear Regression Table
- Fitting Table

Viewing Univariate Curve Fitting results

The result viewer for Univariate Curve Fitting shows the Fitting Chart or the Fitting & Forecast Chart.

The Fitting Chart graphically displays the fitted values and the observed values in the order of observations. It is similar to the Time Series Fitting Chart in Linear Regression. See "Viewing Linear Regression results" on page 270.

If you entered a value other than `no` in the **Forecast model** field, this chart also shows the forecast period, that is, the values that were extrapolated. You find this field on the Additional parameters page of a Univariate Curve Fitting wizard or settings notebook. In this case, the name of the menu choice on the **View** menu is **Fitting & Forecast Chart**. When you select this menu choice, the window shown in Figure 22 on page 277 opens.

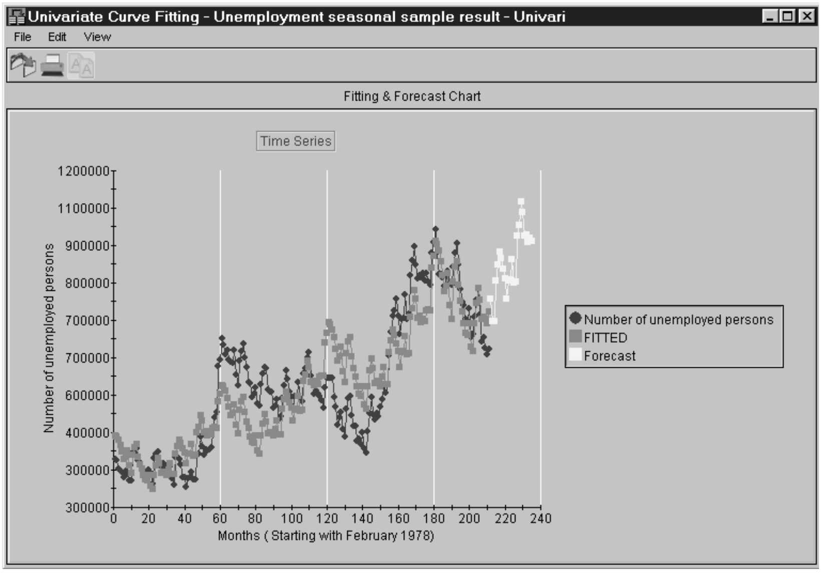


Figure 22. The Fitting & Forecast Chart window

The part of the curve representing the forecast period is highlighted. The vertical lines mark the beginning and the end of the seasonal periods you selected.

Viewing the Equation window

The Equation window displays the equation for the calculated curve on top of the result window. In addition, the window contains the following fields:

Variable
This field gives you the name of the dependent variable that you correlated with time data.

Correlation
This field gives you the correlation coefficient.

Residual standard deviation
This field gives you the unexplained part of the overall standard deviation.

Seasonal periods
This field gives you the number of seasonal periods on which the calculation of the correlation coefficient was based.

Number of forecast periods
This field gives you the number of forecast time periods.

Selected curve type
This field gives you the curve type you selected for the fitting.

To view the Equation window, click **View → Equation** on the menu bar of the statistics result viewer.

To return to the first view, click **View → Fitting Chart** or **View → Fitting & Forecast Chart**.

Viewing the Autocorrelation

Autocorrelation shows how the values in your input field, observed for different time periods, correlate with each other. Thus Autocorrelation allows you to perceive seasonal trends in your data. The Autocorrelation chart shows the value of the Autocorrelation for various lags. A lag is the time interval between the starting points of two time periods. Figure 23 shows the Autocorrelation window.

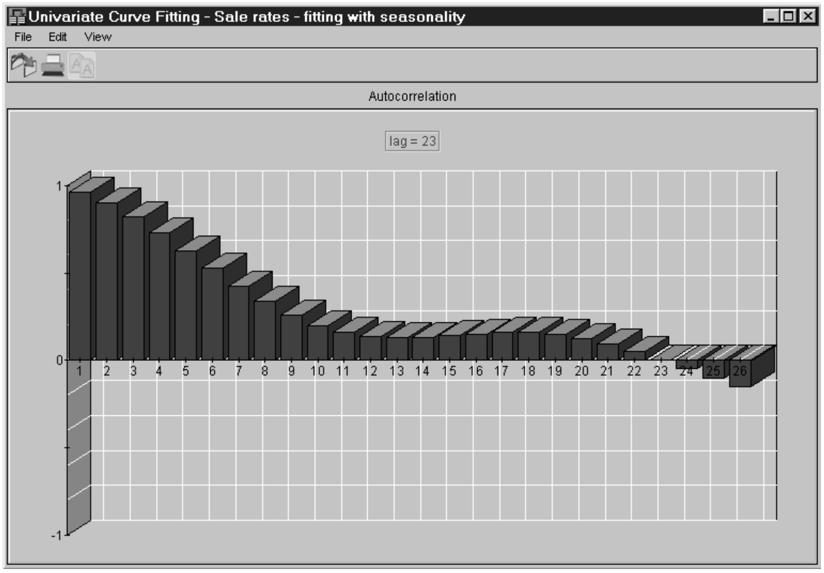


Figure 23. The Autocorrelation window

You can enter the value of the lag in the **Seasonal model** field of your Univariate Curve Fitting settings object to obtain a much better fitting after a repeated run of the function. The result is better because the seasonal trends are considered during the fitting.

In addition, the lag value is required when you specify the *-removetrend* power option to remove the seasonal variation from your fitting model.

To view the Autocorrelation, click **View → Autocorrelation** on the menu bar of the statistics result viewer.

To return to the first view, click **View → Fitting Chart** or **View → Fitting & Forecast Chart**.

Viewing the Model Quality Chart

The Model Quality Chart shows the deviation of the fitted values from the observed values. The diagonal represents the observed values. The closer the points are to the diagonal, the better is the fitting. The Model Quality Chart for Univariate Curve Fitting is identical to the Model Quality Chart for Linear Regression. See "Viewing the Model Quality Chart" on page 273.

To view the Model Quality Chart, click **View → Model Quality Chart** on the menu bar of the statistics result viewer.

To return to the first view, click **View → Fitting Chart** or **View → Fitting & Forecast Chart**.

Viewing the Residual Chart

The Residual Chart shows the residuals (differences between the fitted and the observed values) in the order of observations. It is identical to the Time Series Residual Chart for Linear Regression (the second chart in the window). See "Viewing the Residual Chart" on page 274. To view the Residual Chart, click **View → Residual Chart** on the menu bar of the statistics result viewer.

To return to the first view, click **View → Fitting Chart** or **View → Fitting & Forecast Chart**.

Viewing the Fitting Table

The Fitting Table for Univariate Curve Fitting is very similar to the Fitting Table in Linear Regression (see "Viewing the Fitting Table" on page 275). However, the column for the first independent variable (fifth column) is missing because there is only one variable under consideration. Instead, you find a new column in the second position. Here is a description of the columns in the Fitting Table:

1. The first column shows the numbers of the observations on which the charts are based.
2. The second column shows you in which seasonal period the values were observed.
3. The third column shows the observed values.
4. The fourth column shows the fitted values.
5. The fifth column shows the residuals. The residuals are the differences between the observed values and the fitted values.

To view the Fitting Table, click **View → Fitting Table** on the menu bar of the statistics result viewer.

To return to the first view, click **View → Fitting Chart** or **View → Fitting & Forecast Chart**.

Viewing the Forecast Table

Except for the fact that the Forecast Table deals with the forecast period rather than historical time-series data, it is identical to the Fitting Table. However, there are no columns for the observed values and the residuals because this kind of data does not exist for forecast periods.

The Forecast Table is available only if you entered a value other than `no` in the **Forecast model** field. You find this field on the Additional parameters page of a Univariate Curve Fitting wizard or settings notebook.

To view the Forecast Table, click **View → Forecast Table** on the menu bar of the statistics result viewer.

To return to the first view, click **View → Fitting Chart** or **View → Fitting & Forecast Chart**.

Exportable views

The following list shows the views that you can export or copy to the clipboard in comma-separated variables format (CSV):

- Fitting Table
- Fitting & Forecast Table (if available)

Viewing Bivariate Statistics results

Calculating bivariate statistics produces results similar to clustering results. The result viewer thus resembles the clustering result viewer. See “Viewing bivariate statistics, clusters, quantile ranges, or regions” on page 292 and “Bivariate Statistics” on page 247 for more information.

Printing statistics results

To print statistics results, follow these steps:

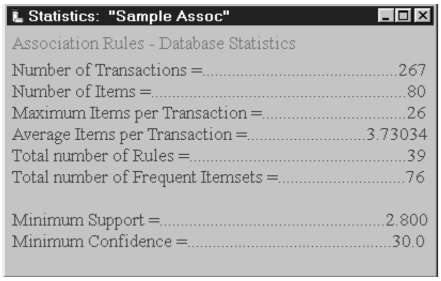
1. Click **File → Print** in the result viewer window. The standard print window of your client operating system opens.
2. Click **OK**. The visible part of the active window is printed.

Viewing association rules

This section describes how to view association rules using the textual visualizer on client operating systems other than Windows. The Windows client software provides a graphical visualizer. For more information about this visualizer, see *IBM DB2 Intelligent Miner Visualization: Using the Intelligent Miner Visualizers*.

When a mining run is completed, the following windows are opened simultaneously, displaying the results:
- Statistics window
- Association Rules - File window

The Statistics window displays statistical information of the mining run. See Figure 24.



Association Rules - Database Statistics

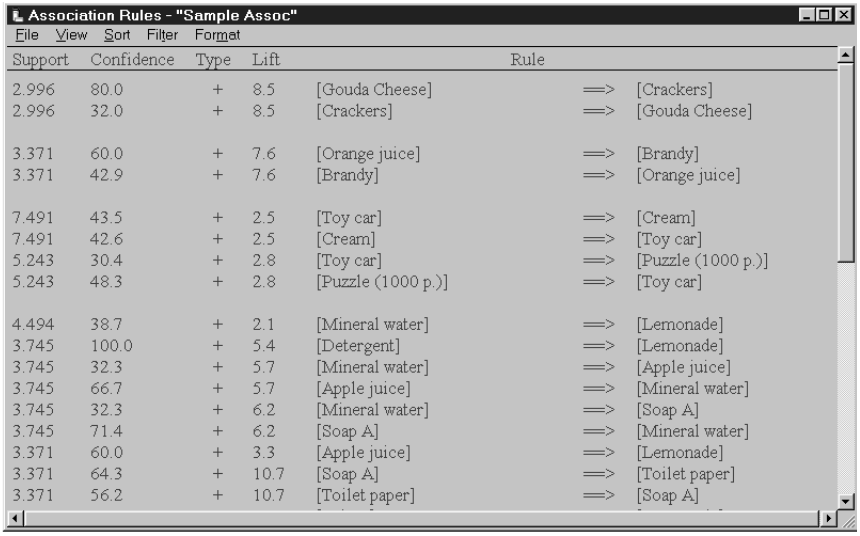
```
Number of Transactions =..........................267
Number of Items =.................................80
Maximum Items per Transaction =...................26
Average Items per Transaction =...................3.73034
Total number of Rules =...........................39
Total number of Frequent Itemsets =...............76

Minimum Support =.................................2.800
Minimum Confidence =..............................30.0
```

Figure 24. The Statistics of Association Rules window

You can only browse this window.

The Association Rules window displays the detected association rules in condensed format. See Figure 25.



Support	Confidence	Type	Lift	Rule		
2.996	80.0	+	8.5	[Gouda Cheese]	==>	[Crackers]
2.996	32.0	+	8.5	[Crackers]	==>	[Gouda Cheese]
3.371	60.0	+	7.6	[Orange juice]	==>	[Brandy]
3.371	42.9	+	7.6	[Brandy]	==>	[Orange juice]
7.491	43.5	+	2.5	[Toy car]	==>	[Cream]
7.491	42.6	+	2.5	[Cream]	==>	[Toy car]
5.243	30.4	+	2.8	[Toy car]	==>	[Puzzle (1000 p.)]
5.243	48.3	+	2.8	[Puzzle (1000 p.)]	==>	[Toy car]
4.494	38.7	+	2.1	[Mineral water]	==>	[Lemonade]
3.745	100.0	+	5.4	[Detergent]	==>	[Lemonade]
3.745	32.3	+	5.7	[Mineral water]	==>	[Apple juice]
3.745	66.7	+	5.7	[Apple juice]	==>	[Mineral water]
3.745	32.3	+	6.2	[Mineral water]	==>	[Soap A]
3.745	71.4	+	6.2	[Soap A]	==>	[Mineral water]
3.371	60.0	+	3.3	[Apple juice]	==>	[Lemonade]
3.371	64.3	+	10.7	[Soap A]	==>	[Toilet paper]
3.371	56.2	+	10.7	[Toilet paper]	==>	[Soap A]

Figure 25. The Association Rules window

You can also display the rules in textual format:

```
When a customer buys Orange juice, then the customer
also buys Brandy in 60.0% of cases.
This pattern is present in 3.371% of transactions.
```

The support factor means that this rule applies to 3.371% of the total number of transactions.

The confidence factor means that the rule body (orange juice) applies to 60.0% of the rule head (brandy).

Expected Confidence

Given the rule A→B, where A and B are item sets, the confidence of this rule is determined by the following formula:

$$\text{Confidence}(A \rightarrow B) = \frac{\text{Support}(A \,\&\, B)}{\text{Support}(A)}$$

where the support of (A & B) is the percentage of transactions containing both item sets A and B.

Assuming that A and B are statistically independent of each other, the following formula applies:

$$\text{Support}(A \,\&\, B) = \text{Support}(A) \times \text{Support}(B)$$

This results in the following value for the expected confidence of this rule:

$$\text{Expected Confidence}(A \rightarrow B) = \frac{\text{Support}(A) \times \text{Support}(B)}{\text{Support}(A)} = \text{Support}(B)$$

Lift

The lift of a rule indicates the factor by which the actual confidence exceeds the expected confidence:

$$\text{Lift}(A \rightarrow B) = \frac{\text{Confidence}(A \rightarrow B)}{\text{Exp.Conf.}(A \rightarrow B)} = \frac{\text{Confidence}(A \rightarrow B)}{\text{Support}(B)}$$

Type

The type of a rule is neutral if A and B are statistically independent. The type is positive if A has a positive influence on the occurrences of B. The type is negative if this influence is negative. The chi-square test of statistical independence is used for determining the rule type.

Environment variables

You can change the TEXTRULENOUN and TEXTRULEVERB variables at the Intelligent Miner client. The values you set are displayed in the Association Rules window. In the example:

```
When a customer buys Orange juice, then the customer
also buys Brandy in 60.0% of cases.
```

the values `customer` and `buys` are default values for the TEXTRULENOUN and TEXTRULEVERB environment variables.

The following example shows how you would set the variables in an AIX environment:

```
export TEXTRULENOUN = "doctor"
export TEXTRULEVERB = "prescribes"
```

The new environment variable values would display as:

```
When a doctor prescribes Orange juice, then the doctor
also prescribes Brandy in 60.0% of cases.
```

Changing the representation of association rules

You can change the representation of the results:

- To view the textual explanation of the rules, click **Format → Textual**.
- To view the association rules condensed, click **Format → Condensed**. **Condensed** is the default selection.
- To view the rules corresponding to the selected support and confidence values, click **View → Rules**. **Rules** is the default selection.
- To view large item sets corresponding to the selected support value, click **View → Large Item Sets**.
- To view the Statistics window, click **File → Statistics**.

You can sort the association rules corresponding to different criteria.

To do this, select one of the following choices from the **Sort** menu:

Support
Sorts the support values from highest to lowest value.

Confidence
Sorts the confidence values from highest to lowest value.

Rule Head
Sorts the derived items of the rules alphabetically.

Lift Sorts the lift values from highest to lowest.

Support*Confidence
Sorts the product of support and confidence from highest to lowest.

Group => Support
Sorts the support values within the rule groups.

Group => Confidence
Sorts the confidence values within the rule groups.

Group => Rule head
Sorts the derived items of the rules alphabetically within the rule groups.

Group => Support*Confidence
Sorts the product of support and confidence within the rule groups.

You can save or print this view of the association rules, or close the Association Rules - File window:

- To save the results, click **File** → **Save As**.
- To print the results, click **File** → **Print**. The Printer details window opens. Enter the name of the printer and the amount of required copies in the corresponding entry fields, and click **OK**.
- To close the Association Rules - File window, click **File** → **Close**.

Filtering association rules

When the detected association rules are displayed in the Association Rules - File window, you might want to view individual association rules separately. You can specify several items from the association rules to display the association rules containing the specified items in a separate window.

To filter association rules:

1. If you are viewing association rules, that is, when the **Rules** menu choice from the **View** menu of the Association Rules - File window is active, click **Filter** → **Rules**. See "Changing the representation of association rules" on page 283 for more information.
2. If you are viewing frequent item sets, that is, when the **Frequent Item Sets** menu choice is active, click **Filter** → **Frequent Item Sets**. See "Changing the representation of association rules" on page 283 for more information.

   Depending on the menu choice you selected, the Filter Association Rules window or the Filter Frequent Item Sets window opens. Except for their window titles, these windows are identical.

   When you open the Filter Association Rules window or the Filter Frequent Item Sets window for the first time, the entry fields are empty. When you open one of these windows again after you have defined filtering rules, the entry fields reflect the current definitions. Figure 26 on page 285 shows the Filter Association Rules window.

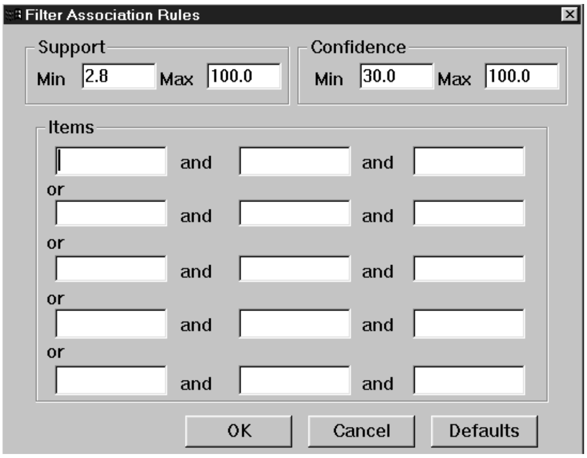


Figure 26. The Filter Association Rules window

3. To view rules containing all of the the items you specify, enter the items you want to view in a horizontal row of the **Items** entry fields, so the specified items are combined with the **And** operator.
4. To view rules containing any of the items you specify, enter the items you want to view in a vertical row of the **Items** entry fields, so the specified items are combined with the **Or** operator.

Optional: You can use the default values for specifying the support and the confidence factors displayed in the corresponding entry fields, or you can specify your own values.

5. Enter the minimum and the maximum support factor you want to use in the corresponding entry fields, or use the following default values:

Support Factor	Default Value
Minimum	2.8
Maximum	100

6. Enter the minimum and the maximum confidence factor you want to use in the corresponding entry fields, or use the following default values:

Confidence Factor	Default Value
Minimum	30.0
Maximum	100

7. Click **OK**. The current window closes.

Filtering rules and item sets are displayed in a separate window. This window provides the same menu-bar choices as the Association Rules Results window. See “Changing the representation of association rules” on page 283 for a detailed description.

Viewing classifications

This section describes how to view classification results by using the Intelligent Miner Version 6.1.1 visualizers on client operating systems other than Windows. The Windows client software provides a graphical visualizer. For more information about this visualizer, see*IBM DB2 Intelligent Miner Visualization: Using the Intelligent Miner Visualizers*.

Depending on the mining function you used, the results are displayed as a binary decision tree or as a neural confusion matrix.

Viewing the binary decision tree

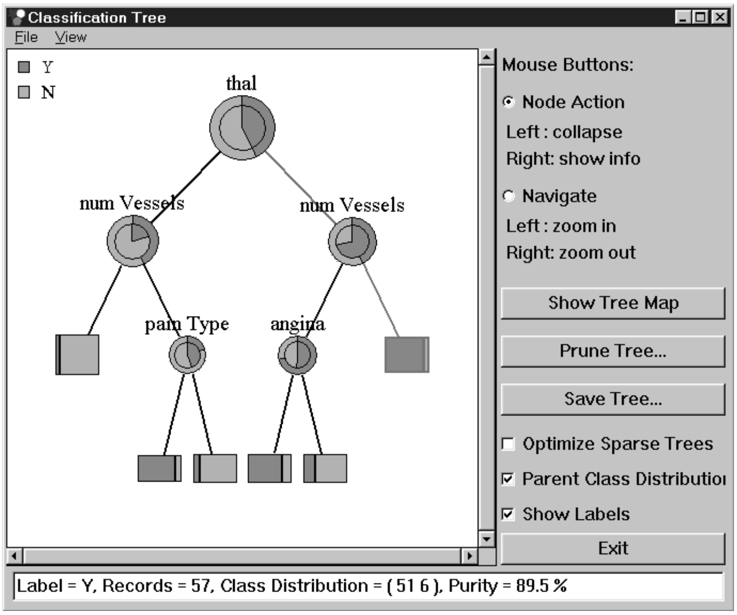
The Tree Classification mining function builds a classification model as a binary decision tree. Each interior node of the binary decision tree tests an attribute of a record. If the attribute value satisfies the test, the record is sent down the left branch of the node. If the attribute value does not meet the requirements, the record is sent down the right branch of the node.

The classification visualizer displays the best split criterion for each node. The binary decision tree is pruned, but the visualizer allows you to view both pruned and unpruned versions of the tree.

When the classification visualizer is launched, the Statistics - Classification Results window opens. This window displays statistical information and the confusion matrix of the mining run.

To open a separate window displaying the pruned or unpruned decision tree, click the appropriate push button.

The Classification Tree window in Figure 27 on page 287 shows an example of a pruned binary decision tree:



Label = Y, Records = 57, Class Distribution = (51 6), Purity = 89.5 %

Figure 27. The Classification Tree window

The binary decision tree consists of the root node on top, followed by non-leaf nodes and leaf nodes. Branches connect a node to 2 other nodes. Root and non-leaf nodes, are represented as pie charts. Leaf nodes are represented as rectangles.

Data assigned to different classes of the classification is represented in different colors.

You can view detailed information about the tree:

- To view information about a node, place the cursor over the desired node. The selected node is surrounded by a red frame. Textual information about the selected node is displayed in the information area at the bottom of the Classification Tree window. This information includes:

Label	The pre-dominant class label of the selected node.
Test	The split criterion for this node. This applies only to non-leaf nodes and specifies a simple selection.
Records	The number of records contained in each of the subnodes the selected node.
Distributions	The number of records corresponding to each of the possible class labels.

The classification is most meaningful if all records belong to one leaf node only. However, by pruning the binary decision tree, records of other nodes can be assigned to the selected node.

Purity	The percentage of correctly classified records assigned to a node.

- To collapse a node:
  - Click the **Node Action** radio button.
  - Click the node.

    All subnodes and leaves below this node are folded into the collapsed node. A collapsed node is represented by a triangle. The collapsed node represents the sum of all values of the subnodes.
  - To expand the subnodes and leaves again, click the collapsed node.
- To view the decision path of a node:
  - Click the **Node Action** radio button.
  - Click the node using mouse button 2. The Decision Path window opens, displaying information about the test of this node and the series of decisions to arrive at this node. The decision is represented in the following form:

    ```
    if (dec1) and (dec2) and (dec3) ... then node = nodename
    ```
  - To view information for another node, click that node. The contents of the Decision Path window are updated with information for the current node.

    If you close the window and want to open it again for a specific node, click that node.

You can zoom in and out on the view of the binary decision tree:
- To zoom in, select the **Navigate** radio button and click the desired node.
- To zoom out, click the node again with mouse button 2.

To view the number of records distributed to each node:
1. Click the **Show Tree Map** push button. The Node Weight window opens. The relative distribution sizes of the nodes are represented as rectangles. Each rectangle has the same color assigned to the class label that is assigned to the node.
2. Move the mouse pointer into the area of a rectangle to highlight a rectangle. The rectangle is surrounded by a red frame and the corresponding node in the Classification Tree window is highlighted in red.

To view the distribution of the parent class of a non-leaf node, click the **Parent Class Distribution** check box. The distribution is shown in the outer circle of a non-leaf node's pie chart.

To make better use of the window space in which the decision tree is shown, click the **Optimize Sparse Trees** check box. This action enlarges the tree and sometimes rearranges its branches so that the visibility is enhanced. When you view an unpruned decision tree, checking this option brings up the distribution chart at each node.

To prune the binary decision tree:

1. Click the **Prune Tree** push button. The Prune Tree window opens.
2. Enter the minimum number of records a node should represent in the **Minimum Node Size** entry field. The default value is 0.
3. Enter the maximum number of decision levels or branch levels the tree can have in the **Maximum Tree Depth** entry field. By default, all levels are displayed.
4. Click the **Prune Tree** push button to save the current specifications, or click the **Reset Parameters** push button to reset the default values.
5. Click the **Close** push button to close the Prune Tree window.

To save the tree in its current representation, click the **Save Tree** push button.

Printing Tree Classification results

You can print the confusion matrix and the classification tree. To print the confusion matrix of Tree Classification results, follow these steps:

1. Click **File → Print** in the Classification Results window. The standard print window of your client operating system opens.
2. Click **OK**. The whole contents of the active window is printed, including the invisible parts.

When printing the classification tree, you can choose between the following output formats:

- Print as Graphic. Selecting this option prints the currently displayed classification tree. To print the classification tree, Click **File → Print as Graphic** in the Classification Tree window. The Print window opens, displaying the name of the currently selected printer. Perform one of the following steps, or, if you are printing from an AIX client, perform step 2:
  1. Click the **OK** push button to print the classification tree on the selected printer.
  2. Click the **Print setup** push button to change the printer. The standard printer selection window of your client operating system opens. Follow these steps:

   a. Select a printer from the list. On AIX, enter the name of the printer in the appropriate field.
   b. Click the **OK** push button to return to the Print window.
   c. Click the **OK** push button to print the classification tree.

- Print as Text. Selecting this option prints the classification tree in a textual format. The printout is organized in chunks giving information about the tree nodes. This information includes:
  - The number assigned to the tree node
  - The number of records that arrived at the tree node
  - The number of records correctly classified. Both, the absolute value and the percentage, are given.
  - The number of records incorrectly classified. Both, the absolute value and the percentage, are given.
  - The type of the node, that is, whether it is a leaf node or an internal node
  - The decision path of records that arrived at the node.

  To print the classification tree in textual format, click **File → Print as Text** in the Classification Tree window. The Print window opens, displaying the name of the currently selected printer. Perform one of the following actions, or, if you are printing from an AIX client, perform step 2:
  1. Click the **OK** push button to print the classification tree on the selected printer.
  2. Click the **Print setup** push button to change the printer. The standard printer selection window of your client operating system opens. Follow these steps:
     a. Select a printer from the list. On AIX, enter the name of the printer in the appropriate field.
     b. Click the **OK** push button to return to the Print window.
     c. Click the **OK** push button to print the classification tree in textual format.

You can also change the selected printer directly from the Classification Results window or the Classification Tree window. To change the currently selected printer, follow these steps:
1. Click **File → Print setup**. The standard printer selection window of your client operating system opens.
2. Select a printer from the list.
3. Click the **OK** push button.

Restriction: The confusion matrix is not scaled to fit on the page.

Viewing Neural Classification results

The result of a neural classification is a trained neural network. You can use it to classify records from other data sources.

When the Neural Classification - Results window opens, the results are displayed as a confusion matrix of the predicted values versus the actual values.

The confusion matrix shown in Figure 28 shows the percentage of correct and incorrect classifications for each output category. You can use the confusion matrix to inspect the accuracy of a trained neural network classifier.

```
          Predicted
Actual   A    B    C   Unknown
     A  94    3    2   1
     B   6   80   10   4
     C  39    5   48   8
```

Figure 28. Example of a confusion matrix

The results in Figure 28 show that in 94% of the cases an item A pattern is presented to the network and classified as item A. In 5% of the cases the network incorrectly classified the item A pattern as item B pattern (3%) or as item C pattern (2%). In 1% of the cases the network was unable to classify the item.

These results show that the network has not learned enough at this time to differentiate item C from item A. The results imply that there are probably relatively few records containing item C in the data source.

To improve the accuracy in classifying item C and to help differentiate item C from item A, the network needs more information. You can:

- Restart a mining run using more records in training mode.
- Specify additional input fields to be mined.

You can print, copy, or change the representation of the results:

- To print the results, click **File → Print**. You can also press Ctrl+P.

  If you want to change the printer name or the orientation of the printout, click **File → Print setup** and select the printer name and the orientation you want to use.
- To copy the results, click **Edit → Copy**. You can also press Ctrl+Insert.
- To change the view of the results, select **View → Text** or **View → Bar Chart**. The **Text** menu choice is the default selection.

- To change the representation of the records, select one of the following menu choices from the **Data** menu:

Raw	View the raw number of the records. You can also press Ctrl+R.
Percentage	View the percentage of the records. You can also press Ctrl+E.
Sensitivity	View the sensitivity analysis in either text or bar chart form. You can also press Ctrl+S. The sensitivity analysis report is represented as a list of input fields ranked according to their respective importance to the classification function. The results are normalized so that they total 100%. A parameter that is listed having 20% performance is twice as important in making the desired classification as a parameter with a 10% score.

Viewing bivariate statistics, clusters, quantile ranges, or regions

You can view bivariate statistics, clusters, quantile ranges, or regions to see groups of records with similarities. These groups, known as partitions, represent clusters, bivariate statistics, regions, or quantiles. Each partition contains pie charts and histograms that represent fields in the result. The following terms apply to the Intelligent Miner windows that display partitions.

Clusters	A cluster is a group of records with similar characteristics in their active field values.
Bivariate statistics	Bivariate statistics are created as the result of an analysis of two variables for the purpose of studying the relationship between the variables. The Bivariate Statistics function generates univariate statistics, including the mean, minimum, maximum, variance, frequency and standard deviation of each variable in the data. In addition, it gives you the Chi-square and F-test measures for each bivariate pair.
Regions	A group of records with similar characteristics in the active field values, similar to clusters. While clusters are discovered by the clustering algorithms based on a similarity relation, regions provide qualitative information about a value prediction model.
Quantile ranges	A group of records whose predicted field

values fall within the same range, out of a finite number of non-overlapping ranges.

You can change the result view or look at details of the result to focus on specific areas that you might use to make business decisions. For example, you might find that the largest group of buyers for boating equipment have similar demographic characteristics, such as gender, age, and income.

You can view individual fields, clusters, regions, quantiles, or bivariate statistics in various levels of detail. For more information on viewing your results, click **Help** on the menu bar.

Clusters

This section describes how to view clusters by using the Intelligent Miner Version 6.1.1 visualizers on client operating systems other than Windows. The Windows client software provides a graphical visualizer. For more information about this visualizer, see *IBM DB2 Intelligent Miner Visualization: Using the Intelligent Miner Visualizers*.

The Demographic Clustering and Neural Clustering mining functions produce clusters as their results.

The Intelligent Miner All Clusters View window displays all clusters generated by the clustering function. See Figure 29.

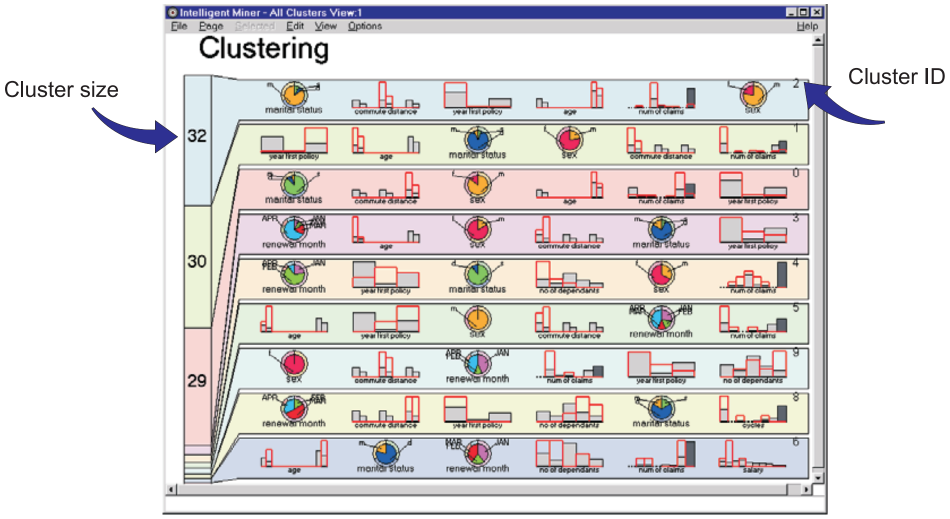


Figure 29. The Intelligent Miner - All Clusters View window

You can view clusters using the All Clusters View, Cluster View, and Cluster Field View.

Bivariate Statistics

The All Clusters View can also display partitions consisting of bivariate statistics. See Figure 30. Partitions that contain bivariate statistics have identifier labels consisting of the bivariate statistics field value and its name. The label is displayed in the upper right hand corner of a partition. If the bivariate statistics field is a continuous numeric field, the identifier represents a value range with a lower bucket limit and an upper bucket limit enclosed in brackets. The value of the field, for all records in the partition, falls inside the range shown. If the bivariate statistics field is a categorical or discrete numeric field, the identifier represents the value of that field. Each value or value range represents one partition in the view.

When the field is discrete numeric or categorical, then the label consists of a single field value. This is the value of the field for all records in the partition. The value is not enclosed in brackets.

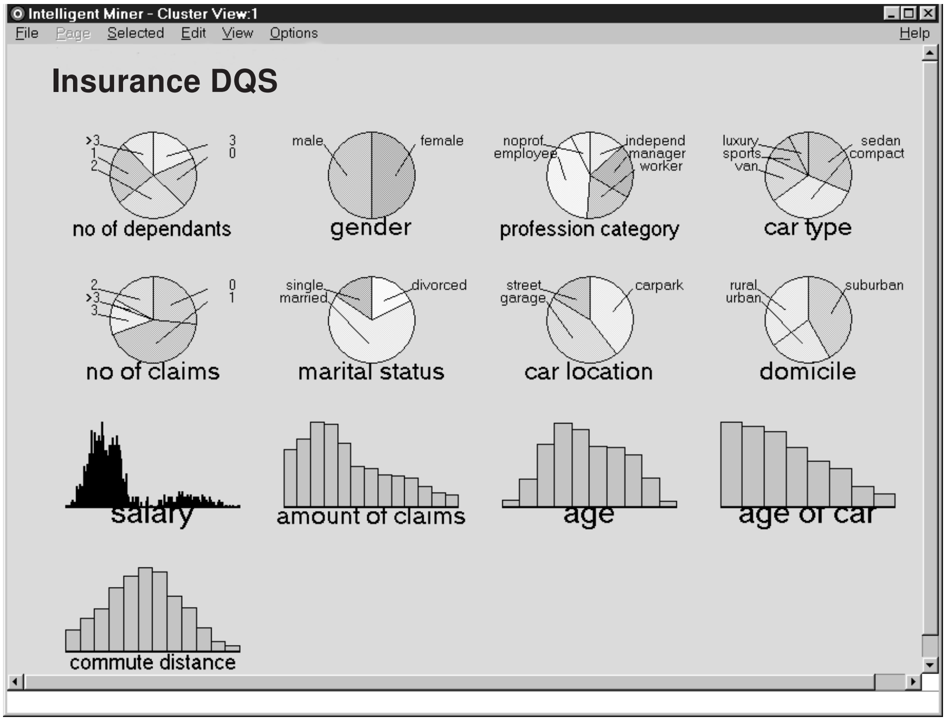


Figure 30. The Intelligent Miner - All Clusters View window

Regions and quantile ranges

The RBF-Prediction mining function produces results with regions and, optionally, quantile ranges. The Neural Prediction mining function only produces results with quantile ranges.

Both Prediction mining functions produce results that contain groups of records with similar field values or similar attribute values.

The All Regions View window displays all regions in a result generated by the RBF-Prediction mining function. See Figure 31. The All Quantiles View window displays all quantile ranges in a result generated by the RBF-Prediction mining function or the Neural Prediction mining function.

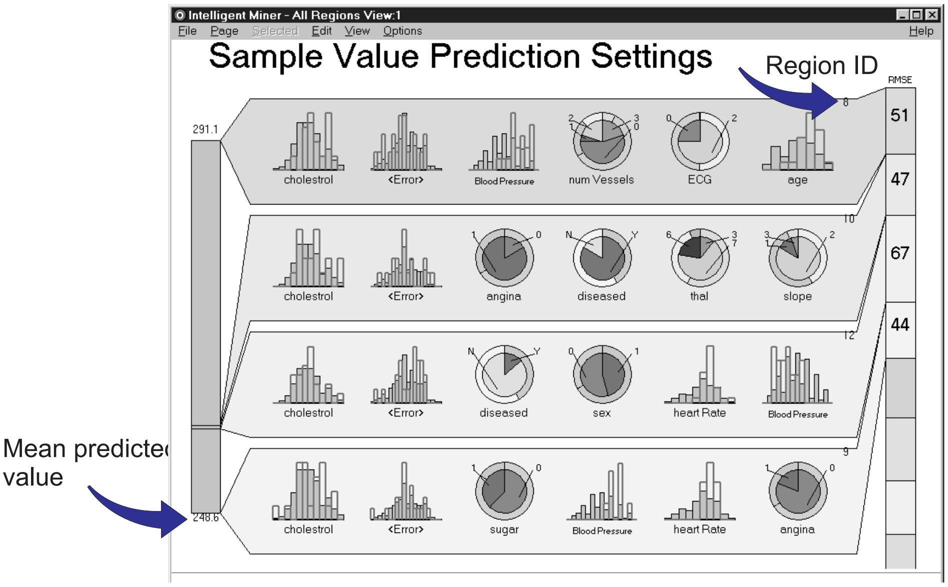


Figure 31. The Intelligent Miner - All Regions View window

The Prediction mining functions do not produce clusters.

Printing clusters, regions, or quantile ranges

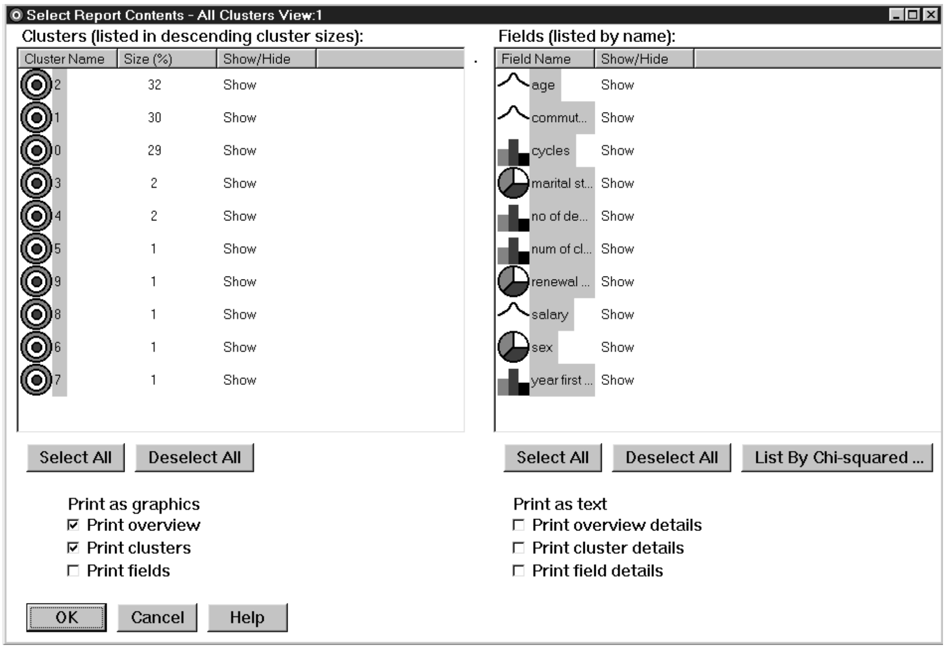
You can print the results produced by the Clustering, RBF-Prediction, or Bivariate Statistics functions. To print the results of one of these functions, follow these steps:

1. Click **File** → **Print** in the result viewer window. The standard print window of your client operating system opens.
2. Click the **OK** push button. The whole contents of the active window is printed.

On Windows client workstations, the result viewer for the Clustering, RBF-Prediction, and Bivariate Statistics functions also allows you to print reports. To print a report, follow these steps:

1. Click **File** → **Print Report**. The Select Report Content window opens.

2. Select fields and clusters. Only results pertaining to the selected items are printed. See Figure 32.



Clusters (listed in descending cluster sizes):

Cluster Name	Size (%)	Show/Hide
2	32	Show
1	30	Show
0	29	Show
3	2	Show
4	2	Show
5	1	Show
9	1	Show
8	1	Show
6	1	Show
7	1	Show

Fields (listed by name):

Field Name	Show/Hide
age	Show
commut...	Show
cycles	Show
marital st...	Show
no of de...	Show
num of cl...	Show
renewal ...	Show
salary	Show
sex	Show
year first ...	Show

Print as graphics
- [x] Print overview
- [x] Print clusters
- [ ] Print fields

Print as text
- [ ] Print overview details
- [ ] Print cluster details
- [ ] Print field details

Figure 32. The Select Report Content window

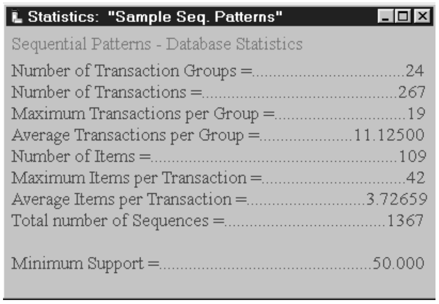
3. Select one or more options by clicking the appropriate checkboxes.
4. Click the **OK** push button. The standard print window of your client operating system opens. Click the appropriate button to print the results. Note that the output might consist of numerous pages.

In addition, you can save results of the Clustering, RBF-Prediction, and Bivariate Statistics functions to a Postscript file, which you can use with other applications. To save the whole contents of the active window to a Postscript file, click **File → Save as PostScript** in the results viewer window.

Viewing sequential patterns

When a mining run is completed, the following windows open to display the results:
- Statistics window
- View window

The Statistics window displays statistical information of the mining run. See Figure 33 on page 297.



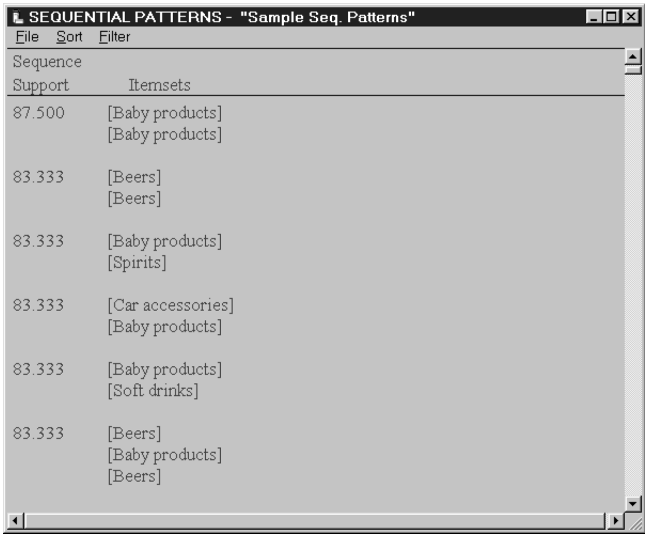
```
Sequential Patterns - Database Statistics
Number of Transaction Groups =...................................24
Number of Transactions =.............................................267
Maximum Transactions per Group =..............................19
Average Transactions per Group =.....................11.12500
Number of Items =.........................................................109
Maximum Items per Transaction =................................42
Average Items per Transaction =...........................3.72659
Total number of Sequences =.....................................1367

Minimum Support =.................................................50.000
```

Figure 33. The Statistics of Sequential Patterns window

You can only browse this window.

The Sequential Patterns window displays the detected sequential patterns for all transactions within the transaction group. See Figure 34.



```
Sequence
Support       Itemsets
87.500      [Baby products]
            [Baby products]

83.333      [Beers]
            [Beers]

83.333      [Baby products]
            [Spirits]

83.333      [Car accessories]
            [Baby products]

83.333      [Baby products]
            [Soft drinks]

83.333      [Beers]
            [Baby products]
            [Beers]
```

Figure 34. The Sequential Patterns window

The first pattern displayed in the Sequential Patterns window consists of a transaction containing **Miscel Toys** followed by a transaction containing **Baby products**. This pattern is found in 87.5 % of the transaction groups.

Changing the representation of sequential patterns

You can sort sequential patterns corresponding to different criteria.

To sort the sequential patterns, select one of the following menu choices from the **Sort** menu:

Support
To sort the support factor from the highest to the lowest value.

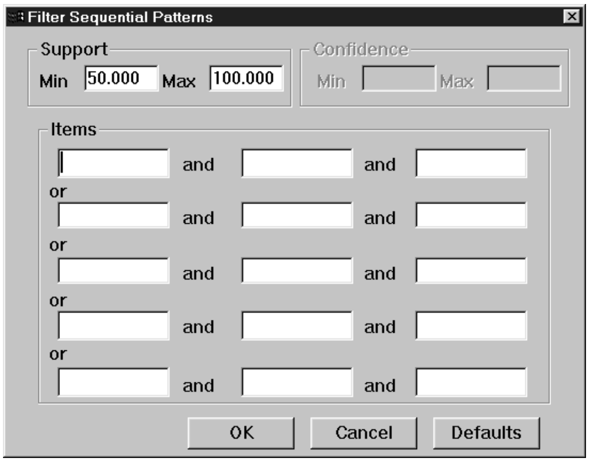
Group → Support
To sort the support factor from the highest to the lowest value within a group.

Filtering sequential patterns

You can filter sequential patterns. You can also select several items from the results and display these in a separate window.

To filter sequential patterns:

1. Start a mining run.
2. Click **Filter → Sequences** on the menu bar of the Sequential Patterns - File window. Figure 35 shows the Filter Sequential Patterns window.


Filter Sequential Patterns

Support: Min 50.000 Max 100.000

Confidence: Min Max

Items: and, and, or

OK Cancel Defaults

Figure 35. The Filter Sequential Patterns window

When you open the Filter Sequential Patterns window for the first time, the entry fields are empty. When you open the window after you have defined filtering rules, the entry fields reflect the current definitions.

3. To view sequential patterns containing all the items you specify, enter the item names in a horizontal row of the **Items** entry fields, so that the **And** operator is used.
4. To view sequential patterns containing one or more of the items you specify, enter the item names in a vertical row of the **Items** entry fields, so that the **Or** operator is used.

Optional: You can use the default values for the support factor displayed in the corresponding entry fields, or you can specify your own values.

5. Enter the minimum and the maximum support factor you want to use:

Support Factor	Default Value
Minimum	2.8
Maximum	100

6. Click **OK**. The Sequential Patterns - File window closes.

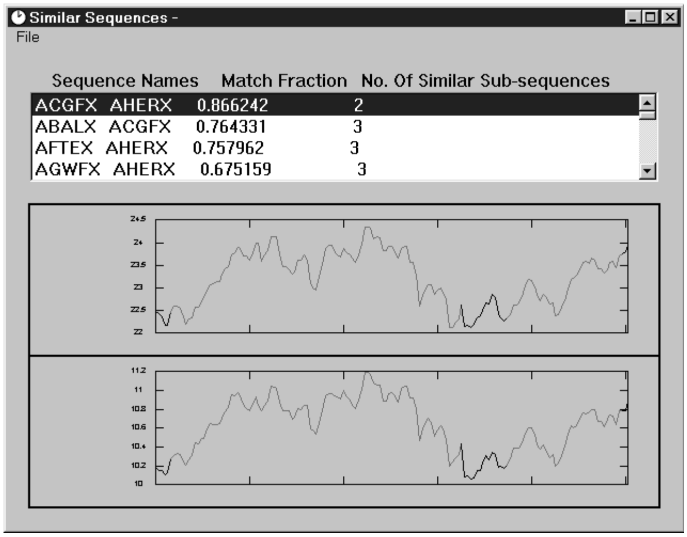
The sequential patterns containing the specified items are displayed in a separate window. This window provides the same menu-bar choices as the Sequential Patterns - File window. See "Viewing sequential patterns" on page 296 for more information.

You can view the statistics, save or print the result, or close the Sequential Patterns - File window:

- To view the statistics of the mining run, click **File → Statistics**.
- To save the result, click **File → Save As**.
- To print the result, click **File → Print**. The Printer details window opens. Enter the name of the printer and the number of copies. Click **OK**.
- To close the Sequential Patterns - File window, click **File → Close**.

Viewing similar sequences

When you view Similar Sequences results, the window shown in Figure 36 on page 300 opens.



Sequence Names		Match Fraction	No. Of Similar Sub-sequences
ACGFX	AHERX	0.866242	2
ABALX	ACGFX	0.764331	3
AFTEX	AHERX	0.757962	3
AGWFX	AHERX	0.675159	3

Figure 36. The Similar Sequences window

A scrollable list box shows the names of data sequences that contain similar subsequences. The names are grouped together in pairs. The pair at the top of the list is highlighted. The first two columns in the list box contain the names of the sequences in each pair. The third column gives you the match fraction for each pair of sequences. The match fraction indicates the degree to which the similar sequences that make up the pair resemble each other. A match fraction of 1 would indicate total resemblance according to the tolerance range defined by the parameters Epsilon, Gap, Window size, and Matching length. The pairs are listed in the order of their match fractions. The fourth column indicates the number of similar subsequences in each sequence pair.

The two graphs below show the data sequences of the selected pair. Similar subsequences are highlighted in red. On the horizontal axis, time is plotted in discrete, user-defined units, such as years, months, days, periods, or hours. On the vertical axis, values of the variable to be described are plotted, for example, prices of mutual funds or stocks, temperature at a given point, or the strength of seismic waves.

You can work with the results in the following ways:

- To browse the graphical views of another sequence name, click another sequence name in the list box.
- To close the Similar Sequences window, click **File → Exit**.

Printing similar sequences

To print results produced by the Similar Sequences function, click **File → Print** in the result viewer window.

Exception: On AIX, you must first click **File → Print setup** to select a printer, and then click **File → Print**.

The Print window opens, showing the name of the currently selected printer. Perform one of the following actions:

- Click the **OK** push button to print the whole contents of the active window on the selected printer.
- Click the **Print setup** push button to change the printer. The standard printer selection window of your client operating system opens. Follow these steps:
  1. Select a printer from the list.
  2. Click the **OK** push button to return to the Print window.
  3. Click the **OK** push button to print the Similar Sequences results.

The Similar Sequences result viewer also allows you to print reports. To print a report, follow these steps:

1. Click **File → Print report**. The Select Report Content window opens.
2. Select one or more of the following print options:

   Parameter settings
   Prints the names of the sequences and the values of the following fields:
   - Time
   - Value
   - Epsilon
   - Gap
   - Window size
   - Matching length

   List of similar subsequences
   Prints the contents of the list in the result viewer window.

   Selected splines
   Prints the selected splines that are shown in the result window.

   Complete data set
   Prints all sequences and thus all records stored in the result file, whether they contain similar subsequences or not. Note that the output might consist of numerous pages.

   See Figure 37 on page 302 for more information.

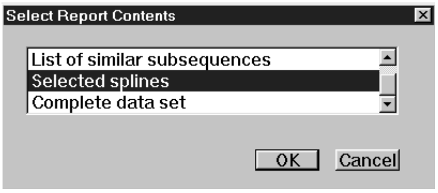


Figure 37. The Select Report Content window

3. Click the **OK** push button. The Print window opens, displaying the name of the currently selected printer. Perform one of the following steps, or, if you are printing from an AIX client, perform step 302:
   - Click the **OK** push button to print the Similar Sequences results on the selected printer.
   - Click the **Print setup** push button to change the printer. The standard printer selection window of your client operating system opens. Follow these steps:
     a. Select a printer from the list.
     b. Click the **OK** push button to return to the Print window.
     c. Click the **OK** push button to print the Similar Sequences results.

You can also change the selected printer directly from the Similar Sequences window. Note that on AIX, you must select a printer before you can click any of the **Print** menu choices. To change the currently selected printer, follow these steps:

1. Click **File** → **Print setup**. The standard printer selection window of your client operating system opens.
2. Select a printer from the list.
3. Click the **OK** push button.

Important: On OS/2, you cannot print to a file.

Chapter 21. The Gains chart

You can compare the quality of output tables that are created in application mode or in test mode to the average curve and the optimum curve by using the Gains chart browser.

This chapter describes how to specify the Gains chart browser in the Intelligent Miner. It also explains how to create output tables in application mode or in test mode, and how to explore these output tables with the Gains chart browser.

The Gains chart browser is available on AIX and Windows operating systems.

Understanding Gains chart

You can use Gains charts with the following mining functions:

- RBF Prediction
- Neural Prediction
- Linear Regression

You can use the following data types to create a Gains chart:

Numerical data
A numerical target variable is used when training a model.

Applying this model predicts a numerical value.

Categorical data
A categorical target variable is used when training a model.

Applying this model predicts both a categorical value and a numerical value that represents the confidence for this prediction.

After training a model, you can use this model on test data by assigning a predicted numerical value for each record in application mode or in test mode.

- If you are using numerical data, this is the predicted numerical value for that record.
- If you are using categorical data, this is the confidence with which a certain fixed user-selected value is assigned to that record.

The records of the output table are sorted by the predicted numerical value. After sorting, the accumulated sum of the actual values is computed stepwise.

For categorical values, the accumulated sum of the actual values is the number of occurrences of the desired categorical value.

The values accumulated up to record number i, ∑(i), can be considered as points in a graph. If there are n records, you will get n+1 points, where the x-coordinate and the y-coordinate are defined like this:

The first point
: {x=0;y=0}

The following points
: {x=i; y=∑(i)}

To allow easy comparisons, consider {x=100*i/n; y=100*∑(i)/∑(n)} to create a percentage scale.

Comparing different graphs in the same diagram is a powerful way to see the quality of different models at first glance, especially when graphs are computed for test data that is not used to train the model.

Instead of records, you can use a different granularity for producing graphs with smaller resolution. For example, you can aggregate records into quantiles. The Intelligent Miner automatically aggregates records to 100 quantiles, producing a chart of 101 points.

Specifying the Gains chart browser

Before you can view a Gains chart, you must specify the Gains chart browser for the result type Result - Exploration Sample in your Preferences notebook.

1. On the Intelligent Miner main window, select **Options ▸ Preferences ▸ Visualizers**.
2. From the **Result type** drop-down list, select Result - Exploration Sample.
3. From the **Result format and visualizer** drop-down list, select Gains chart browser.
4. Click **Add**.
5. Click **OK**.

Creating output tables

You can create output tables for the following predictive functions:

- Tree Classification
- RBF Prediction
- Linear Regression

Creating a model in training mode

For information about creating a model in training mode for the following predictive functions, see the appropriate section:

Tree Classification
“Training mode” on page 199.

RBF Prediction
“Training mode” on page 230.

Linear Regression
“Linear Regression” on page 244.

Creating an output table in test mode or application mode

To create an output table using test mode or application mode, you must define a mining run using an input table that contains similar data as your training data. The input table must also contain actual values in the field to predict. Depending on the function that you want to use, you must specify different parameters.

Creating output tables for the Tree Classification mining function

To create an output table from the Tree Classification settings `clf heart training data`, follow these steps:

1. On the Intelligent Miner main window, open the Tree Classification notebook `clf heart training data`.
2. On the **Settings** page of this notebook, select **Show the advanced pages and controls**.
3. On the **Input data** page, type `-gains Y` in the **Power options** entry field to specify the value of your particular interest. In this example, this value predicts how many patients diseased under the specified circumstances.
4. On the **Mode Parameter** page, specify the following parameters:
   - **Test mode** or **Application mode**
   - One of the existing results.
5. On the **Output fields** page, specify the following parameters:
   a. Select the **Create output data** radio button.
   b. Select the field to be predicted, for example, Diseased, from the **Available fields** list and move it to the **Output fields** list by clicking the > button.
   c. Type `class id` in the **Class ID field name** entry field.
   d. Type `confidence` in the **Confidence field name** entry field.
6. On the **Output data** page, select an existing data table by clicking on one of the available output data tables, for example, insur.out. You can also create a new data table by clicking **Create data**. The name of the data table is used as label for the graph you are going to create.

7. Start this mining run to rewrite or to create the data table that you specified on the **Output data** page.

   The progress indicator shows you the progress of the mining run.

Creating output tables for the RBF Prediction mining function

If you want to use the RBF Prediction mining function on categorical values, add a computed field to your data object that uses the function “StringEqual()” and compares the field that you want to predict against a value that you define as new constant. Then, use the computed field as predicted field.

To create an output table using the RBF Prediction mining function, follow these steps in the RBF Prediction notebook:

1. On the **Mode parameters** page, select **Test mode** or **Application mode**.
2. On the **Output fields** page, specify the following parameters:
   a. Select the field to be predicted from the **Available fields** list and move it to the **Output fields** list by clicking →.
   b. Type `predicted` in the **Predicted Value field name** entry field.
   c. Type `region` in the **Region ID field name** entry field.
3. On the **Output data** page, select the **Create output data** radio button. Select an existing data table by clicking on one of the available data tables, or create a new data table by clicking **Create data**. The name of this data table is used as label for the graph that you are going to create.
4. Start the mining run to create or to rewrite the data table that you specified on the **Output data** page.

The progress indicator shows the progress of the mining run.

Creating output tables for the Linear Regression statistical function

If you want to use the Linear Regression statistical function on categorical values, add a computed field to your data object that uses the function “StringEqual()” and compares the field you want to predict against a value that you define as new constant. Then, use the computed field as predicted dependent variable.

To create an output table using the Linear Regression statistical function, follow these steps in the Linear Regression notebook:

1. On the **Mode parameters** page, select **Application mode**.
2. On the **Output fields** page, specify the following parameters:
   a. Select the **Create output data** radio button.
   b. Select the field to be predicted from the **Available fields** list and move it to the **Selected fields** list by clicking →.
   c. Keep the default value `FITTED` in the **Fitted value field** entry field.

3. On the **Output data** page, select an existing data table by clicking on one of the available data tables, or create a new data table by clicking **Create data**. The name of this data table is used as label for the graph you are going to create.
4. Start the mining run to create or to rewrite the data table that you specified on the **Output data** page.

The progress indicator shows the progress of the mining run.

Exploring output tables created in application or test mode

To explore the new output table, right-click the output table on the Intelligent Miner main window and select **Explore** from the pop-up menu. The progress indicator shows the progress of the exploration of the output table. Depending on your Preferences settings, the results are launched automatically, or you can view the results by clicking **View results** on the Progress window.

The Gains chart browser displays the optimum curve, the average curve, and the curve of the output table. You can explore as many output tables as you like by repeating the following steps:

1. "Creating a model in training mode" on page 305.
2. "Creating an output table in test mode or application mode" on page 305.
3. "Exploring output tables created in application or test mode".

For each output table that you are exploring, an additional curve is added to the Gains chart browser. You can view all these curves in the Gains chart browser, or you can deselect the curves that you want to exclude from the view.

Appendix A. Intelligent Miner tutorial

This Intelligent Miner mining tutorial consists of several mining tasks. The tutorial starts with data in a flat file, details the process of defining Intelligent Miner data objects, running Intelligent Miner functions, and viewing results using the Intelligent Miner's visualizers.

This Intelligent Miner tutorial consists of an abbreviated data mining scenario with five phases: Defining data, building a model, applying the model, automating the process, and analyzing the results. By following the steps in this tutorial, you will learn how to use the Intelligent Miner wizards to define data objects, run mining functions, and view results in the Intelligent Miner.

This tutorial and the sample data used in this tutorial are designed to support the learning objectives. As such, they do not represent actual or recommended methods for using the Intelligent Miner. To shorten the time it takes to complete the tutorial, the data file is small and can be processed quickly. Additionally, the five phases represent an important subset of the activities at the core of many mining projects. Finally, the tutorial uses the Demographic Clustering function to accomplish its goals. There are other functions within the Intelligent Miner that can be used to accomplish the same end. Typical mining investigations would compare the results of more than one function.

Before you start

Before you work with this tutorial, read the following information::

- "Understanding basic concepts" on page 77 to understand how to use the Intelligent Miner main window.
- "Using help in the Intelligent Miner" on page 133 for information on how to display the online help of the Intelligent Miner.
- The readme.txt file provided with the Intelligent Miner client installation.

To use this tutorial, the following prerequisites are required:

- The Intelligent Miner server is installed on an AIX, iSeries, OS/390, Solaris Operating Environment, or Windows server.
- The Intelligent Miner client is installed on an AIX or a Windows client.

The business problem

Imagine that you work for a bank that sells several products, including Regular Checking, Premier Checking, and Exclusive Checking accounts and option packages for each account. The bank already knows that Premier Checking is their most profitable product, and wants to increase the number of customers who have this type of checking account. The marketing department wants to identify different groups based on demographic data, such as age and income, within the Premier Checking customers so that the department can prepare different ad campaigns for each of the groups. Additionally, the department wants to identify customers who are not currently Premier Checking customers who have similar demographics to the customers who are Premier Checking customers.

You have obtained some customer data from corporate headquarters to solve this business problem. This data is named banking.txt. It contains information about customers from all branches of the bank. You can use the Intelligent Miner to mine this data and provide demographic information to the marketing department. Your customer data includes information about customers who already have the Premier Checking account, so you can use the Demographic Clustering mining function to identify different groups, based on demographic data, among customers who already have Premier Checking.

To learn more about demographic clustering, see a description of the function in Chapter 10, “The Demographic Clustering mining function” on page 153.

The mining run tasks

This tutorial will demonstrate five phases of data mining tasks:

Defining the data

Define a data object that points to a flat file containing your customer data file banking.txt. The data object will be named **Customers**.

You must specify which properties of your customers are contained in the data, their data types, and the columns in the flat file that they occupy.

The Intelligent Miner data objects simply point to the location of your data, so that the Intelligent Miner can process this data. You will not actually be changing the contents of the banking.txt file. See “Defining a data object” on page 317 for instructions on how to complete this step.

Building the model

Define a Demographic Clustering settings object named **Build model**. This settings object uses the Customers data object as the input data. It runs in clustering mode, and produces a results object named **Model**. This model contains information that describes the clusters identified during the mining run. See "Building a model" on page 321 for instructions on how to complete this step.

Applying the model

Define a Demographic Clustering settings object named **Apply model**. This settings object uses the Customers data object as the input data. It runs in application mode using the Model results object and produces an output data object named **Scored customers** and a flat file named **scored.txt**. This output file identifies the subgroup associated with a customer record. See "Applying the model" on page 328 for instructions on how to complete this step.

Automating the process

To automate the process you create a sequence object **Target Marketing** containing the Build model settings object and the Apply model settings object. A sequence is an object containing several other objects in a specific sequential order. You can run a sequence, which runs each of the objects within the sequence in the order that you specified. This allows you to combine several mining tasks into one step. See "Creating a sequence" on page 333 for instructions on how to complete this step.

Analyzing the results

Define a Bivariate Statistics function named **Analyze**. This statistical function analyzes the data object **Scored customers** and an produces an output data object **Target customers**, a flat file **target.txt**, and a result object **Target customer demographics**. See "Creating a statistics function" on page 334 and "Interpreting the results" on page 338 for instructions on how to complete this step.

Starting the Intelligent Miner in demo mode

The Intelligent Miner includes a sample flat file named banking.txt that contains fictitious customer data. To access this sample data, you must start the Intelligent Miner in demo mode.

Running the Intelligent Miner in demo mode provides all the same functionality as running Intelligent Miner in regular mode, but also includes sample mining bases and sample data. The interface that you use to create objects, define data, and perform other data mining tasks for this tutorial is the same interface that you use in regular mode.

Using the demonstration data on AIX servers

To start the Intelligent Miner in demo mode, first determine if you are running in AIX local mode or if you are using remote access to an AIX server.

If you are running in AIX local mode: Start the Intelligent Miner server in demo mode using these commands:

1. `cd /usr/lpp/IMiner/bin`
2. `./idmstart`
3. `./imdemo`

If you are using remote access to an AIX server running the Intelligent Miner in demo mode: Start the Intelligent Miner server using these commands:

1. `cd /usr/lpp/IMiner/bin`
2. `./idmstartdemo`

Start the Intelligent Miner client as usual. Clients connecting to a server running the Intelligent Miner in demo mode share the sample mining bases. The mining bases are opened in read-only mode because many users might access the same sample mining base at the same time. Each client must sign on to the Intelligent Miner server using the appropriate user ID and password on the Server Logon page of the Preferences notebook.

Restarting the Intelligent Miner in demo mode: To remove all additional mining bases and reset them to their initial state, issue these commands:

1. `idmstop`
2. `idmstartdemo` to start the Intelligent Miner in demo mode, or `idmstart` to start the Intelligent Miner in regular mode

Note: In demo mode, the IDM_MNB_DIR environment variable points to a temporary directory rather than to the home directory on the server. This temporary directory is defined as /tmp/dmtksample.xxxxx, where *xxxxx* is a five-digit number.

Using the demonstration data on OS/390 servers

To make the demonstration data available for particular client users, you must install this data in each user's home directory on the server. To do so, perform the following steps:

1. Run the IDMDEMO job. The IDMDEMO job is provided with the installation sample library (SIDMSAM1).
2. Adjust the job to your needs before you submit it by following the instructions.

You can now open the sample mining bases at the client side and use them to get familiar with the product.

3. Repeat this action for any user who needs access to the demonstration data.

Using the demonstration data on iSeries servers

To make the demonstration data available for particular client users, you must install this data in each user's home directory on the server. Thus you must repeat the following steps for each user who needs access to the demonstration data. The description below uses a home directory profile named *MYPROF* as an example. Replace this name with an actual profile name when you perform these steps:

1. Enter `DSPUSRPRF MYPROF` and record the value of the HOMEDIR parameter for that directory.
2. Make sure that the directory exists by entering `WRKLNK '/HOME/MYPROF'`. If the directory does not exist, create it by entering `CRTDIR '/HOME/MYPROF'`.
3. You might want to create a subdirectory in the home directory so that the demonstration data is kept separate from the "real" data. This example uses a subdirectory named *IMDEMO*.
4. Copy the demonstration data to this directory. Enter:

   ```
   CALL QIDM/QYDMDEMOC '/HOME/MYPROF/IMDEMO'
   ```

   This command creates the following directories as subdirectories of the

   HOME/MYPROF/IMDEMO directory:

DATA	Contains the data stream files
IDMMNB	Contains mining bases
IDMRES	Contains results

   Files in the IDMMNB and IDMRES directory contain explicit links to this directory and do not work if you copy them to another directory without using the `QIDM/QYDMDEMOC` command.
5. Change the profile of the home directory so that you can use the demonstration data. Enter the following command:

   ```
   CHGPRF HOMEDIR('/HOME/MYPROF/IMDEMO') CCSID(37)
   ```

   Set the Coded Character Set Identity (CCSID) to 37 because the demonstration data is available in English only.
6. Grant access to this user profile unless the user has *ALLOBJ authority. To do so, enter the following commands and specify a valid client user ID for *USERID*:

a.
```
CHGAUT OBJ('/HOME/MYPROF/IMDEMO')
    USER(USERID)
    DTAAUT(*RWX OBJAUT(*ALL)
```
b.
```
CHGAUT OBJ('/HOME/MYPROF/IMDEMO/*')
     USER(USERID)
     DTAAUT(*RWX OBJAUT(*ALL)
```

Repeat this step for the DATA, IDMMNB, and IDMRES subdirectories.

7. Undo the changes to a home directory profile if the user does not need the demonstration data anymore.

Using the demonstration data on Solaris Operating Environment servers

When you install the Intelligent Miner, demonstration data in English is copied to your server. To replace the English demonstration data with demonstration data in one of the supported languages, enter the appropriate command from the command line:

pkgadd -a ./admin -d . IMdemoES
To install the demonstration data in Spanish.

pkgadd -a ./admin -d . IMdemoFR
To install the demonstration data in French.

pkgadd -a ./admin -d . IMdemoGE
To install the demonstration data in German.

pkgadd -a ./admin -d . IMdemoHU
To install the demonstration data in Hungarian.

pkgadd -a ./admin -d . IMdemoIT
To install the demonstration data in Italian.

pkgadd -a ./admin -d . IMdemoJP
To install the demonstration data in Japanese.

pkgadd -a ./admin -d . IMdemoKR
To install the demonstration data in Korean.

pkgadd -a ./admin -d . IMdemoBR
To install the demonstration data in Portuguese.

pkgadd -a ./admin -d . IMdemoRU
To install the demonstration data in Russian.

pkgadd -a ./admin -d . IMdemoTW
To install the demonstration data in Traditional Chinese.

pkgadd -a ./admin -d . IMdemoCN
To install the demonstration data in Simplified Chinese.

During the installation process, several files must be overwritten. Confirm the overwriting of files when prompted to do so.

To reinstall the English demonstration data, enter the following command from the command line:

pkgadd -a ./admin -d . IMdemoEN

To start the Intelligent Miner in demo mode on the server, enter **idmstartdemo**.

A set of sample mining bases with predefined mining and statistics settings is created in a temporary directory. All clients connecting to this server share these mining bases.

To protect the sample mining bases against accidental deletion or overwriting, they are locked when a user opens them. Thus to save any changes or modifications, select **Save as** from the **File** menu to save the mining base in question under a different name.

To remove all additional mining bases and reset the original sample mining bases to their initial state, enter these commands:

1. **idmstop**
2. **idmstartdemo**

To return to regular (non-demo) mode, enter the following commands:

1. **idmstop**
2. **idmstart**

Note: In demo mode, the IDM_MNB_DIR environment variable points to a temporary directory rather than to the home directory on the server. This temporary directory is defined as /tmp/dmtksample.xxxxx, where *xxxxx* is a five-digit number.

Using the demonstration data on Windows servers

Before you start the Intelligent Miner in demo mode, check whether you use the Intelligent Miner in local mode, in stand-alone mode, or in client/server mode. Then follow the instructions in the appropriate section.

Local mode and stand-alone mode

To start the Intelligent Miner in demo mode:

1. Make sure that you started the Intelligent Miner server.
2. Enter **imdemo** from an MS-DOS window.

   This creates temporary sample mining bases and starts the graphical user interface (GUI) with predefined mining and statistics settings. All functions available with the regular program are also supported in demonstration mode. When you close the GUI, the temporary mining bases are deleted.

The demonstration mode does not affect any Intelligent Miner service started in regular mode on the server. Other users who are connected to the same server can work on their own mining bases while a stand-alone user is running the Intelligent Miner in demonstration mode.

Client/server mode

To make the temporary sample mining bases accessible to a remote client, enter **idmstartdemo** in an MS-DOS window.

A set of sample mining bases with predefined mining and statistics settings is created in a temporary directory. All clients connecting to this server share these mining bases. The mining bases are locked to prevent accidental overwriting or deletion. Use **Save as** from the **File** menu to save any modifications to the sample mining bases.

To remove all additional mining bases and reset the original sample mining bases to their initial state, follow these steps:

1. Enter **idmstopdemo** to run the idmstopdemo batch file. This file switches the server back to regular mode and removes the sample mining bases.
2. Enter **idmstartdemo** again.

To return to regular (non-demo) mode, enter **idmstopdemo**.

In demo mode, the IDM_MNB_DIR environment variable points to a temporary directory rather than to the home directory on the server. This temporary directory is defined as %TEMP%\dmtkdemo*X*\idmmnb, where *X* is a number from zero to nine.

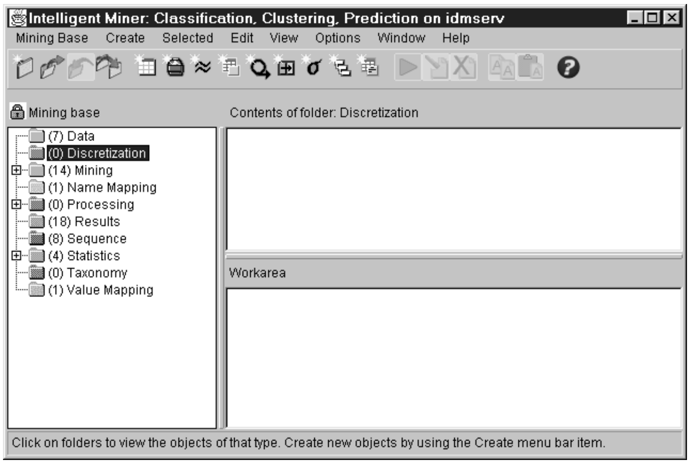
Attention: While the server is in demonstration mode, users cannot access mining bases created in regular mode. Before you start the Intelligent Miner in demonstration mode, make sure that no other client user is connected to the server. Otherwise, the other client users might lose data or results. This applies similarly to the **idmstopdemo** command: Verify that no client user is connected to the server before you stop the demonstration mode. If other users also work in demonstration mode, they cannot access the sample mining bases any longer.

The Intelligent Miner main window

The Intelligent Miner main window helps you manage mining bases and perform data mining tasks. A mining base is a collection of the mining objects needed to conduct a mining run.

After starting the Intelligent Miner client on AIX or Windows:

1. Select **Options → Preferences**. The Preferences notebook opens.
2. Click the Miscellaneous tab to display the Miscellaneous page.
3. Check the **After mining function is run, automatically visualize a result** box.
4. Under Autosave Mining Base, click the **Save after creating** radio button.
5. Click **OK** to save your preferences.



Click on folders to view the objects of that type. Create new objects by using the Create menu bar item.

Figure 38. The Intelligent Miner main window

Figure 38 shows the Intelligent Miner main window. The components of the main window are described in "Getting familiar with the Intelligent Miner main window" on page 77.

Defining a data object

The first step in this tutorial is to define an Intelligent Miner data object that points to the raw customer data you want to mine. For this tutorial, you are using data from a flat file banking.txt that resides on the Intelligent Miner server. To define a data object:

1. Click the **Create Data** icon on the toolbar to start the Data wizard from the Intelligent Miner main window.
2. From the Welcome page of the Data wizard, click **Next** to continue.

Specifying the data format and object name

On the **Data format and settings** page of the Data wizard, you must specify what kind of data you are using, and the name of the Intelligent Miner data object.

1. Select **Flat files** from the list, if it is not already selected.

2. Enter the name of the data object in the Settings name field: `Customers`. Optionally, you can also specify a comment associated with this data object, for example, you might type the comment: `Data about banking customers`.
3. Ensure that the **Show the advanced pages and controls check box** is not checked.
4. Click **Next** to continue.

Specifying the location of the data

On the Flat files page of the Data wizard, you specify the name and location of the data for this Intelligent Miner data object.

1. Double-click on the folder **dmtksample.n** in the list on the left, where *n* is a number. The contents of this folder are displayed in the list.
2. Double-click on the **Data** folder. The contents of this folder are displayed in the list on the right.
3. Scroll through the list and select **banking.txt**.

   If you do not see the banking.txt file in the list of available files, make sure that you provided the correct user ID and password for the Intelligent Miner server on the Server Logon page of the Preferences notebook.
4. Click **Add file**.
5. Make sure that the use mode **Read only** is selected.
6. Click **Next** to continue.

Specifying the field parameters

On the Field Parameters page, you specify the begin and end positions, field names, field types, and optional name mappings for fields in this data object. The flat file display shows you a few lines of the flat file data, to help you enter the begin and end positions for each field. You do not have to specify the field parameters for all fields in the flat file, only the ones that you want to use for the mining run.

Table 90 shows the field parameters for the flat file banking.txt. The begin and end positions are the numerical positions of the columns representing each field in the flat file. For example, the Flat file display field in Figure 39 on page 319 shows columns 1–6 contain values for the field gender, which is of the categorical field type.

Table 90. Field parameters for the Customers data object

Begin and end position	Field name	Field type
1-6	gender	Categorical
10-16	age	Continuous
24-25	siblings	Continuous
30-36	income	Continuous

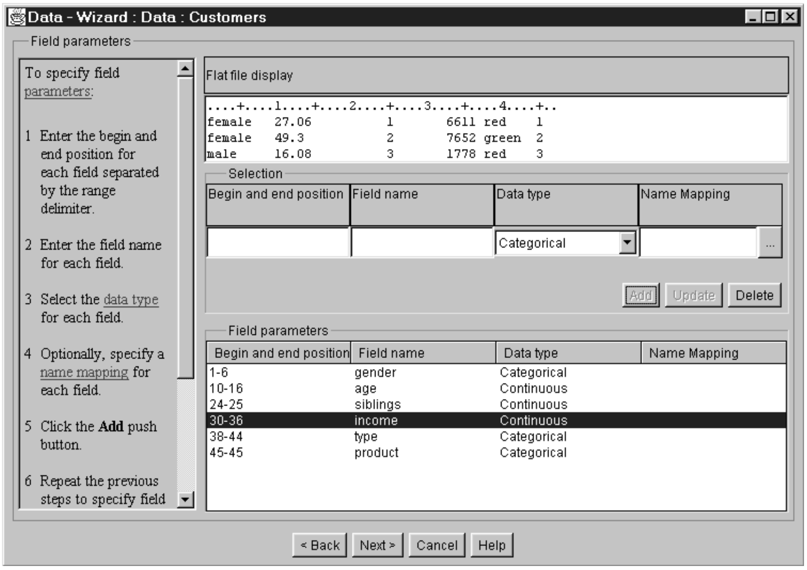
Table 90. Field parameters for the Customers data object (continued)

Begin and end position	Field name	Field type
38-44	type	Categorical
45-45	product	Categorical

To enter the field parameters:

1. Type 1–6 in the **Begin and end position** entry field.
2. Type `gender` in the **Field name** entry field.
3. Select **Categorical** in the **Field type** entry field.
4. Click **Add**.
5. Repeat the previous steps define the field parameters for this data object, as shown in Table 90 on page 318.

   When you finish defining the field parameters, the window should look like Figure 39.



To specify field parameters:

1 Enter the begin and end position for each field separated by the range delimiter.

2 Enter the field name for each field.

3 Select the data type for each field.

4 Optionally, specify a name mapping for each field.

5 Click the **Add** push button.

6 Repeat the previous steps to specify field

```
....+....1....+....2....+....3....+....4....+..
female   27.06         1       6611 red    1
female   49.3          2       7652 green  2
male     16.08         3       1778 red    3
```

Begin and end position	Field name	Data type	Name Mapping
		Categorical	

Begin and end position	Field name	Data type	Name Mapping
1-6	gender	Categorical	
10-16	age	Continuous	
24-25	siblings	Continuous	
30-36	income	Continuous	
38-44	type	Categorical	
45-45	product	Categorical	

Figure 39. Field parameters page

Tip: You can use a name mapping to substitute a character string for a numeric code. For example, you might find it convenient to use a name mapping for the product field, which has values ranging from 1–8. You can map each numeric value to the name of the product, for example, product 1 is the Premier Checking account, product 2 is the Regular Checking account, and so on. This tutorial does not use name mappings for any fields in this data object.

6. Click **Next** to continue.

Defining computed fields

Because there is no need for a computed field in this tutorial, you do not have to define any computed fields for this data object.

Click **Next** to continue.

Saving the data object

The next page of the Data wizard is the Summary page. The Summary page provides a summary of the parameters you defined for the current object you are creating.

Click **Finish** to complete the wizard. If you receive an error message after clicking **Finish**, you can use the **Back** button to return to any page and make corrections based on the message. At this point, you have defined the data object for your mining run. By selecting the Data folder in the mining base container, you can see the icon representing this data object.

Now that you have defined the first object for your mining run, you should save the mining base.

1. To save the mining base, click on the **Save mining base as** icon from the main window toolbar.
2. Enter `Target Marketing` as the mining base field. Optionally, enter a descriptive comment about the purpose of that mining base.
3. Click **Save** to save the mining base.

   Note that, in demonstration mode, the sample mining bases are deleted when you stop the Intelligent Miner server. Therefore, to save the results of this tutorial permanently, export the sample mining base by following these steps:

   a. Click **Mining Base → Export Mining Base** in the Intelligent Miner main window.
   b. To specify an export path on your current drive, navigate through the tree view and select a suitable folder.

      To export the sample mining base to another drive, enter the full path, including the drive letter, in the **Mining base and supporting files filename stem** field.
   c. Enter a name for the mining base in the **Mining base and supporting files filename stem** field or append a mining-base name to the path you specified.
   d. Click **OK** to export the mining base.

Building a model

You can use the Demographic Clustering mining function to obtain information about customers who already have the Premier Checking account. You run this function in clustering mode to produce a model as a result object.

This function generates clusters from your input data. Information about these clusters is stored in a result object that you can view with the clustering visualizer.

To create the settings object for building the model:

1. Click on the **Create mining** button from the Intelligent Miner main window tool bar. The Intelligent Miner displays the Welcome page of the Mining wizard.
2. Click **Next** to continue.

Specifying the mining function and name

On the Mining functions and settings page, you select the type of settings object that you want to create and specify the name and an optional comment for this settings object:

1. Select **Clustering – Demographic** from the list of mining functions.
2. Type the settings name `Build Model`. Optionally, you can specify a more descriptive comment associated with this settings object.
3. Check the **Show the advanced pages and controls** check box.

   The advanced pages and controls of this wizard allow you to use additional options when defining your settings object. For example, using the advanced pages and controls, you can specify a filter condition for the records of the input data. The Intelligent Miner filters the records based on the condition you specify while it is running the mining function.
4. Click **Next** to continue.

Specifying the input data

On the input data page of the wizard, you specify the input data for this mining function:

1. From the list of available input data, select the **Customers** data object that you created in a previous step.
2. Select **Disk space** under **Optimize mining run for**.
3. Under the Advanced options, click the **...** button that is next to **Filter records condition**. You can filter the records of the input data for this settings object.

For filtering records, the Intelligent Miner displays the Expression Builder, which is shown in Figure 9 on page 116.

For this tutorial, you want to include all input data records for customers who have the Premier Checking account. Because the code for the Premier Checking account is 1, the filter condition you want to specify is product=1.

1. Click the **AND** push button. The expression builder creates a template for the expression, which displays as `((Arg1 = Arg2))`.
2. In the **Category** list, click on **Field Names**. The **Value** list displays all the available fields that you can include in this expression.
3. From the **Value** list, select the field **product**.
4. Click on the **Arg1** button. This sets the field product as the first argument in the expression.
5. Select **Constants** from the **Category** list.
6. Double-click on **<new constant>** in the **Value** list.
7. Type in the new constant value 1.
8. Press Enter. The new constant is added to the list of constants.
9. Select the constant **1** from the **Value** list.
10. Click the **Arg2** button. This sets the constant value of 1 as the second argument in the expression.
11. Click **OK** to return to the Input data page of the Mining wizard.
12. Click **Next** to continue.

Setting the mode parameters

On the mode parameters page of the wizard, you specify the mode parameters for the mining function. You can run the Demographic Clustering mining function in two modes: clustering mode or application mode. For more information on the mode parameters, see the online help for the Demographic Clustering mining function.

For this tutorial, you will run this mining function in clustering mode.

Clustering mode: In clustering mode, the function identifies groups of similar records called clusters. The function has parameters that you can use to control the results, including the maximum number of clusters, maximum number of passes, accuracy, and similarity threshold.

Use the default values for these parameters for this tutorial.

Maximum number of passes
The default value is 2.

Maximum number of clusters
The default value is 9.

Accuracy
The default value is 2.

Similarity threshold
The default value is 0.5.

To set the mode parameters, ensure that the **Clustering mode** radio button is selected.

Click **Next** to continue.

Specifying the input fields

On the Input fields page of the Mining wizard, you specify the active fields and supplementary fields for this settings object.

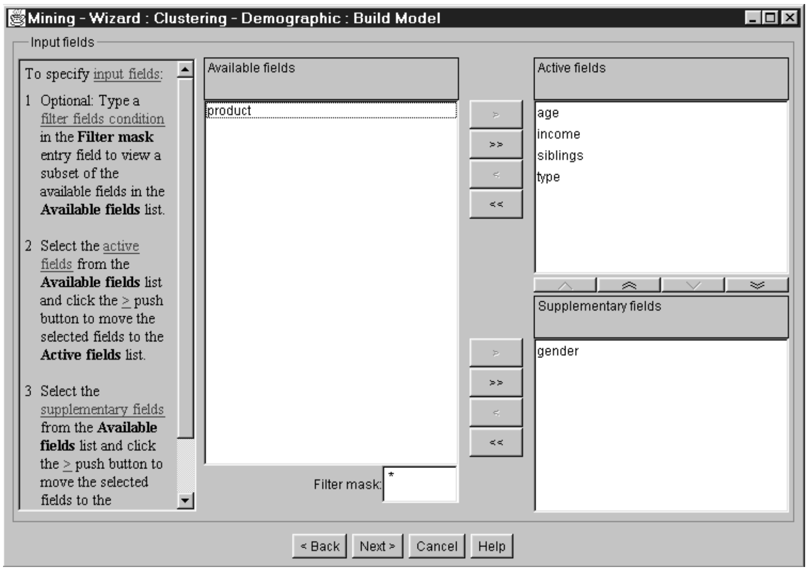
The Demographic Clustering mining function searches the input data for records with similarities, and places similar records into clusters. The active fields you specify are used to determine whether or not the input data records are similar. Statistics about the supplementary fields you specify are included in the result, but are not used to determine similarities.

For example, one of the fields in the customer data is gender. If you specify gender as an active field, the Demographic Clustering mining function uses this as a criterion in determining whether or not two customers are similar. You do not use gender as an active field because of the bank's policy not to include gender information in marketing decisions. Instead, you use demographic information like income, age, and siblings. To see how gender is distributed within the clusters, you add it as a supplementary field.

Because you have filtered out all customers who do not have the Premier Checking account, you do not want to use product as an active field in the clustering process.

1. Select **age**, **income**, **siblings**, and **type** from the list of **Available fields**. Click the **>** push button to add them to the list of **Active fields**.
2. Select **gender** from the list of **Available fields**. Click the **>** push button to add it to the list of **Supplementary fields**.

   After you have specified the active fields and supplementary fields, the input fields page should look like Figure 40 on page 324.


Mining - Wizard : Clustering - Demographic : Build Model

Input fields

To specify input fields:

1 Optional: Type a filter fields condition in the **Filter mask** entry field to view a subset of the available fields in the **Available fields** list.

2 Select the active fields from the **Available fields** list and click the > push button to move the selected fields to the **Active fields** list.

3 Select the supplementary fields from the **Available fields** list and click the > push button to move the selected fields to the

Available fields: product

Active fields: age, income, siblings, type

Supplementary fields: gender

Filter mask: *

Figure 40. Input fields page

3. Click **Next** to continue.

Specifying advanced parameters

The next few pages of the wizard are advanced pages. For this tutorial, you accept the default values.

1. On the Field parameters page of the wizard, click **Next** to continue.
2. On the Additional field parameters page of the wizard, click **Next** to continue.
3. On the Outlier treatment page of the wizard, click **Next** to continue.
4. On the Similarity matrix page of the wizard, click **Next** to continue.

Specifying other parameters

You should see the Output fields page now. However, if you are connected to a server on which the parallel version is installed, you see the Parallel parameters page. In this case, proceed as follows:

1. Make sure that the **Run the serial mode of the function** radio button is selected.
2. Click **Next** to proceed to the Output fields page.

When you reach the Output fields page, proceed as follows:

1. Make sure that the **Create output data** radio button is not selected.
2. Click **Next** to continue.

Specifying the result object name

Each time that you run the Demographic Clustering mining function in clustering mode, it creates a result object and saves it with the name you that specify on this page. Because the mining process is an iterative one, you will probably run a settings object more than once. You can choose to allow this settings object to replace a result object with the same name, because an existing result object with the same name may have been generated by a previous iteration of this settings object.

To specify the results object name:

1. Type `Model` as the name of the result object. Optionally, specify a comment associated with this result object.
2. Check the **If a result with this name exists, overwrite it** check box.
3. Click **Next**.
4. On the Summary page of the Mining wizard, review the parameters for the settings object that you are defining. Select the **Run this settings immediately** check box.
5. Click **Finish** to complete this task.

After you completed each step in the Mining wizard, the Intelligent Miner runs the settings object and displays a progress indicator that allows you to monitor the status of the mining function. After successfully running the mining function, the Intelligent Miner displays the result object generated by this settings object.

This settings object generates a result object named Model, which describes clusters of customers who have the Premier Checking account. Each cluster contains customers with similar characteristics for the fields income, age, type, and siblings.

Whether the Intelligent Miner displays the result object immediately after generating it depends on a preference setting on the Miscellaneous page of the Preferences notebook. You can specify whether you want to visualize results immediately after they are generated. If the Intelligent Miner does not display the result object immediately, you can view the result object by double-clicking on the result object in the Results folder from the Intelligent Miner main window or by pressing **View Results** in the Progress Indicator window.

Interpreting the results generated

The results generated by the mining function are shown in Figure 41 on page 326. The multiple rows of graphs are designed to give you an understanding of the clusters described in the result.

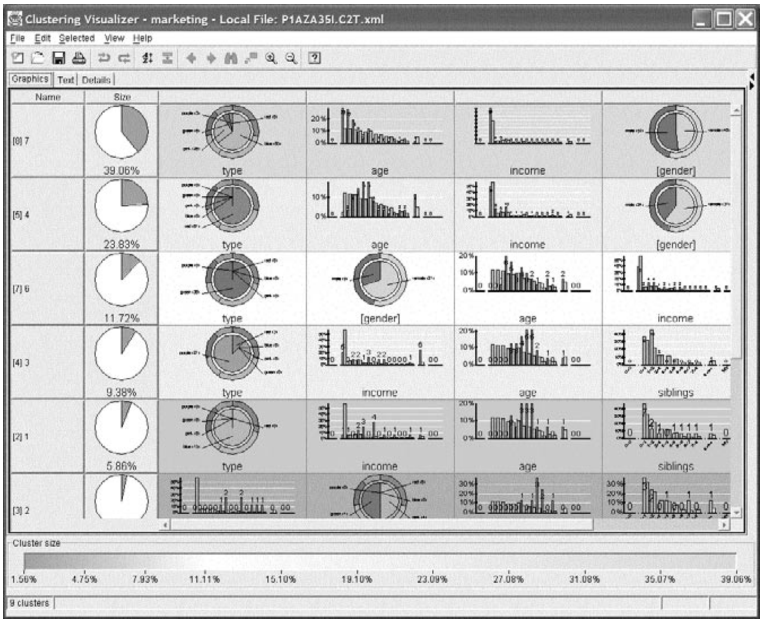


Figure 41. Results of the build model settings object

The display shows nine rows, each representing one of the nine clusters identified by the mining run. Within each cluster, the pie charts and the bar charts represent active and supplemental fields used in the cluster. In this case, fields that have the greatest influence on forming the cluster are displayed on the left, while fields with the least influence are displayed on the right. The numbers in the left side column identify the cluster ID (in brackets) and the cluster name. The next column represents the cluster size as a percentage; for example, the top cluster represents 39% of the data, the next lower cluster represents 23%, and so on. The numbers down the right side identify the cluster ID.

The top row is the cluster with the largest number of customers, accounting for 39% of the customers. Each bar chart or pie chart shows the distribution of the field for the cluster and for the entire set of Premier Checking (product 1) customers. Supplementary fields are indicated with square brackets around the field names.

You can display more detail about a cluster by double-clicking on a chart in the row. Figure 42 on page 327 shows the display of the top cluster, the largest cluster. This cluster contains the account type, age, gender, and siblings fields. Assume that account type indicates the options package the customer purchased, specified at the bank by colors. Double-clicking on any of the graphs shows a single graph.

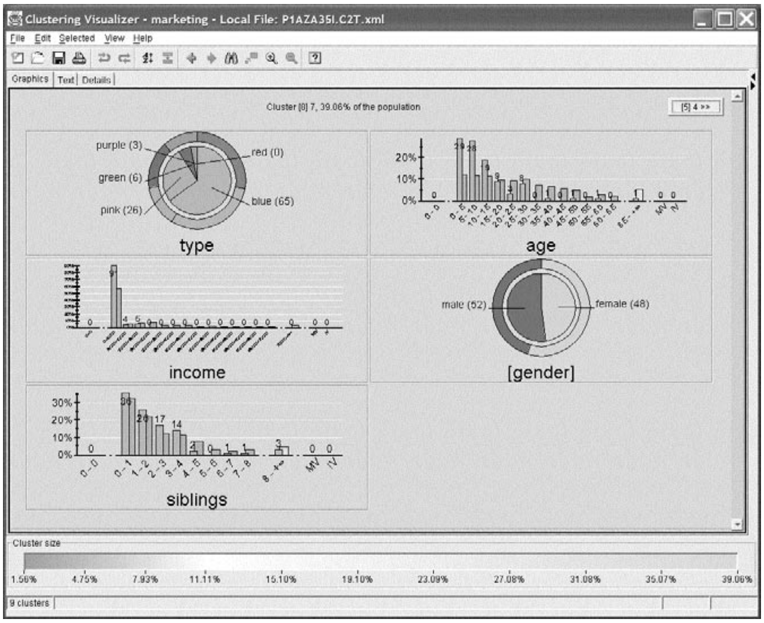


Figure 42. Top cluster of result object

Figure 42 includes a pie chart for gender. Each pie chart produced by the Intelligent Miner shows two distributions. The outside ring shows the distribution for the entire sample. The inside ring shows the distribution for the associated cluster. For example, Figure 42 shows a pie chart in which the outside ring represents the distribution of male and female customers for all Premier Checking customers; the inside ring represents the distribution of male and female customers in this cluster. The chart indicates that this cluster has a slightly greater percentage of males in it than the group of Premier Checking customers as a whole. Looking back at Figure 41 on page 326 you can quickly see that cluster 2 is mostly male and cluster 7 is mostly female.

Figure 43 on page 328 shows the distribution of age for the first cluster. The full bar represents the percentage of people in each age group for all the data and the grey bars in the background represent the distribution in cluster 6. As shown in the graph, a higher percentage of the customers represented by cluster 6 are below the age of 15 when compared to the entire population of Premier Checking customers.

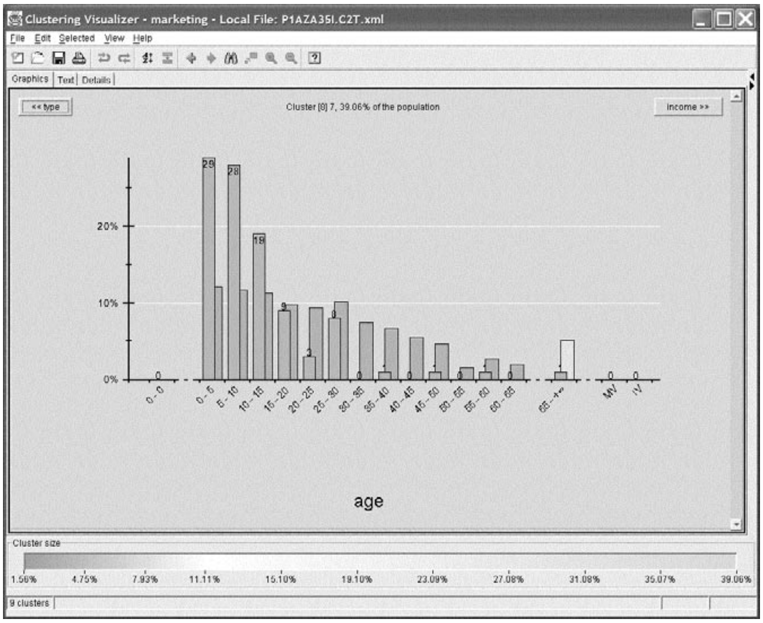

Clustering Visualizer - marketing - Local File: P1AZA35I.C2T.xml
File Edit Selected View Help
Graphics Text Details
<< type
Cluster [8] 7, 39.06% of the population
income >>
age
Cluster size
9 clusters

Figure 43. Age information from the top cluster

Viewing the clustering results can provide insights about the characteristics of a subgroup of Premier Checking customers. The result object contains detailed statistical information, which you use as a model to apply to a new set of data in a subsequent step of this tutorial.

Applying the model

The next step in the tutorial is to apply the clustering model created in the previous step. The result object named Model contains descriptions of nine clusters of customers who already have the Premier Checking account.

To create a settings object for applying the model:

1. Click on the **Create mining** button from the Intelligent Miner main window toolbar. The Intelligent Miner displays the Welcome page of the Mining wizard.
2. Click **Next** to continue.

Specifying the settings object and name

For this step in the tutorial, you again run the Demographic Clustering mining function to apply the model created previously.

1. Select **Clustering – Demographic** from the list of mining functions.
2. Type the settings name `Apply Model`. Optionally, specify a more descriptive comment associated with this settings object.

3. Check the **Show the advanced pages and controls** check box.
4. Click **Next** to continue.

Specifying the input data

For this step in the tutorial, you use the same customer data used by the previous settings object. In most situations, you will build a model using one set of data, and apply the model to a different set of data. The flat file used in this tutorial contains customers with Premier Checking accounts and customers without Premier Checking accounts. In this case, you use the Intelligent Miner filtering feature to separate the customer records into two sets.

1. From the list of available input data, select the **Customers** data object.
2. Ensure that you are optimizing the mining run for disk space.
3. Under **Advanced parameters**, click the **...** button next to **Filter records condition**. You will be filtering the records of the input data for this statistics function.

   The Intelligent Miner displays the Expression Builder for filtering records.

   In this step, you use the filter to select only the customers who do not have premier checking. The expression looks like this: `((product<>1))`. To create this expression:

   a. Click the **AND** push button. The expression builder creates a template for the expression, which displays as `((Arg1 = Arg2))`.
   b. Click on **Field Names** in the **Category** list. The **Value** list displays all the available fields that you can include in this expression.
   c. From the **Value** list, select the field **product**.
   d. Click the **Arg1** button. This sets the field product as the first argument in the expression.
   e. Click the **<>** button. This sets the operand as ″not equal to.″
   f. Select **Constants** from the **Category** list.
   g. Select the constant 1 from the **Value** list.
   h. Click on the **Arg2** button. The constant value of 1 is set as the second argument in the expression. The expression you defined looks like this: `((product<>1))`.
4. Click **OK** to return to the Input data page of the Mining wizard.
5. Click **Next** to continue.

Setting the mode parameters

In a previous step, you ran the Demographic Clustering mining function in clustering mode to create a model. In this step, you apply the model to the customer data by running this mining function in application mode.

In application mode, the mining function scores how similar each customer is to the two most similar clusters of customers with Premier Checking.

For more information on the mode parameters, see the online help for the Demographic Clustering mining function.

To set the mode parameters:

1. Click the **Application mode** radio button.
2. Select the result object **Model** from the application mode container under the **Application mode** group box.
3. Click **Next** to continue.

Specifying the input fields

In this step, you score customers using the same fields that were used to define the clusters in the Build model.

1. Select **age**, **income**, **siblings**, and **type** from the list of **Available fields**. Click the > push button to add them to the list of **Active fields**.
2. Click **Next** to continue.

Specifying advanced parameters

The next few pages of the wizard are advanced pages. For this tutorial, you accept the default values by clicking the **Next** button on the the next four pages.

1. On the Field parameters page of the wizard, click **Next** to continue.
2. On the Additional field parameters page of the wizard, click **Next** to continue.
3. On the Outlier treatment page of the wizard, click **Next** to continue.
4. On the Similarity matrix page of the wizard, click **Next** to continue.

Specifying parallel parameters

You should see the Output fields page now. However, if you are connected to a server on which the parallel version is installed, you see the Parallel parameters page. In this case, proceed as follows:

1. Make sure that the **Run the serial mode of the function** radio button is selected.
2. Click **Next** to proceed to the Output fields page.

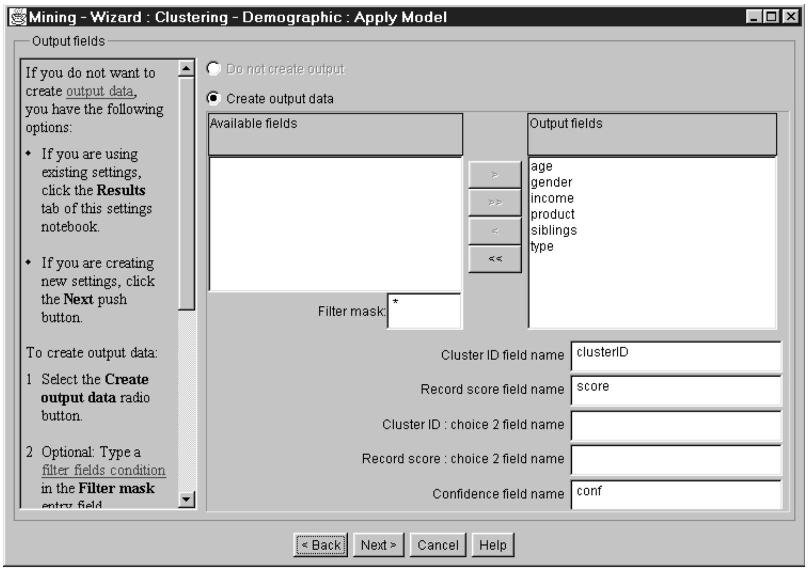
Specifying output fields

On the output fields page of the Mining wizard, you must select the fields that the output data contains. The output data will also contain the cluster ID, the record score, and the confidence values, which are generated by the mining function. In this tutorial, the record score value is a measure of how similar the customers are to the clusters that they are part of.

1. Select >> to add all the available fields to the list of output fields.

2. Type `clusterID` in the **Cluster ID field name** entry field.
3. Type `score` in the **Record score field name** entry field.
4. Type `conf` in the **Confidence field name** entry field.

   The output fields page now looks like this Figure 44.


Mining - Wizard : Clustering - Demographic : Apply Model

Output fields

If you do not want to create output data, you have the following options:

- If you are using existing settings, click the **Results** tab of this settings notebook.
- If you are creating new settings, click the **Next** push button.

To create output data:

1 Select the **Create output data** radio button.

2 Optional: Type a filter fields condition in the **Filter mask** entry field

( ) Do not create output
(●) Create output data

Available fields:

Output fields: age, gender, income, product, siblings, type

Filter mask: *

Cluster ID field name: clusterID
Record score field name: score
Cluster ID : choice 2 field name:
Record score : choice 2 field name:
Confidence field name: conf

Figure 44. Output fields page

5. Click **Next** to continue.

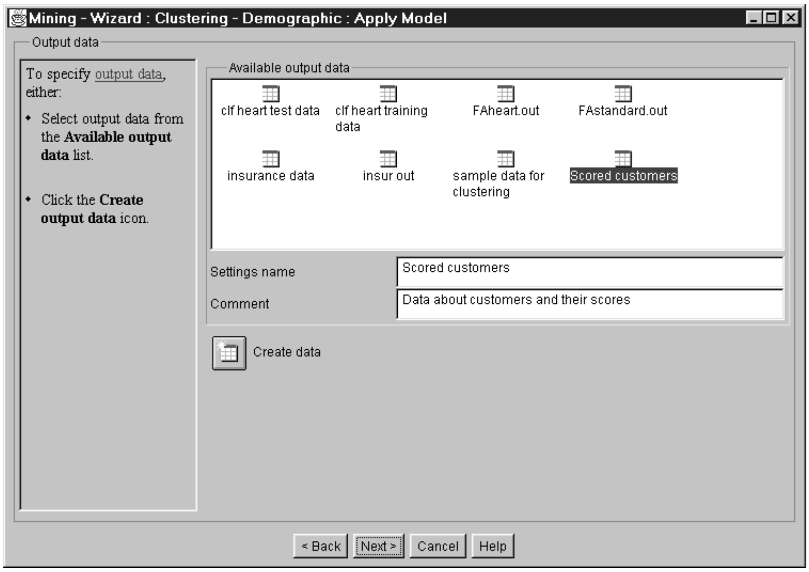
Specifying the output data object name

On the output data page of the wizard, you specify the name of the output data object for this settings object. This page shows the existing data objects in the current mining base. Because you want to preserve the Customers data object, you need to create a data object that contains the output data named Scored customers. To do this, you open the Data wizard from this page of the Mining wizard, define the data object, and then return to this page of the Mining wizard.

1. Click on **Create data**. The Data wizard's Welcome page opens.
2. Click **Next** to continue.
3. Select **Flat files**.
4. Type `Scored customers` in the settings name field. Optionally, type a comment to describe this data object.
5. Click **Next**.
6. On the Flat files page, change to the directory that contains the file banking.txt.

7. In the **Path and file name** entry field append `scored.txt` to the path.
8. Click on **Add file**.
9. Select the **The specified flat file does not yet exist** check box.
10. Click **Next** to continue.
11. On the Summary page of the Data wizard, click **Finish** to continue.

    After defining the output data object, return to the Output data page of the mining wizard to continue the process of defining the mining object. You see the data object in the **Available output data** container, as shown in Figure 45.



To specify output data, either:

- Select output data from the **Available output data** list.
- Click the **Create output data** icon.

Settings name: Scored customers

Comment: Data about customers and their scores

Figure 45. Output data page

1. Select the data object **Scored customers**.
2. Click **Next**. The Summary page opens.
3. Select the **Run this settings immediately** check box.
4. Click **Finish** to continue. The Intelligent Miner will run the mining function and display a progress indicator that allows you to monitor the status of the mining function.
5. Click **OK** in the progress indicator window after the mining function has stopped running.

You now have a flat file that contains a list of customers and scores of how similar these customers are to the Premier Checking customers.

Creating a sequence

Now that we have created a few of the functions for our mining run, we can create a sequence that runs each of the functions in the order we specify.

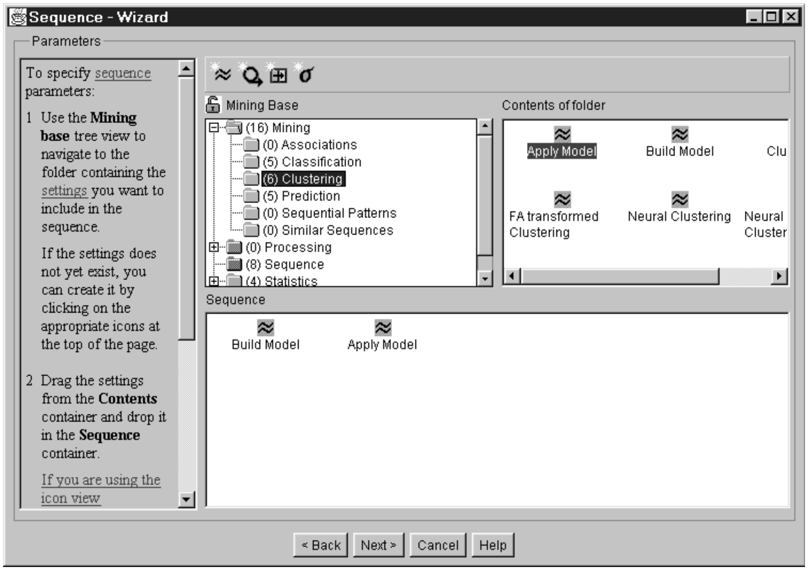
The benefit of using a sequence object is being able to combine several steps into one step. If you combine several functions into a sequence object, you need to run only the sequence object, which then runs each of the objects within it.

To create the sequence for this tutorial:

1. Click the **Create sequence** button on the Intelligent Miner main window toolbar.
2. On the **Welcome** page of the Sequence wizard, click **Next**.
3. In the **Setting name** field type `Target Marketing` as the name of this sequence object.
4. Click **Next**.
5. On the Parameters page of the Sequence wizard, you can use the mining base tree view to navigate to the objects you want to include in the sequence. The Parameters page is shown in Figure 46 on page 334. Because the first object you want to run is the Build model settings object, click on the **+** next to the **Mining** folder.
6. Click the **Clustering** folder. The contents of the folder are displayed.
7. Select the settings object **Build model** and drag and drop it to the **Sequence** work area.

   The settings object build model is added to the sequence as the first object to run.
8. In the **Contents of folder** area, select the mining object **Apply model** and drag and drop the object to the **Sequence** work area.

   This adds the settings object **Apply model** to the sequence as the second object to run, after **Build model**.



To specify sequence parameters:

1 Use the **Mining base** tree view to navigate to the folder containing the settings you want to include in the sequence.

If the settings does not yet exist, you can create it by clicking on the appropriate icons at the top of the page.

2 Drag the settings from the **Contents** container and drop it in the **Sequence** container.

If you are using the icon view

(16) Mining
(0) Associations
(5) Classification
(6) Clustering
(5) Prediction
(0) Sequential Patterns
(0) Similar Sequences
(0) Processing
(8) Sequence
(4) Statistics

Contents of folder: Apply Model, Build Model, Clu, FA transformed Clustering, Neural Clustering, Neural Cluster

Sequence: Build Model, Apply Model

Figure 46. Parameters page of Sequence wizard

9. Click **Next**.
10. On the Additional parameters page of the Sequence wizard, check the **If a settings object in the sequence fails, continue running the sequence** check box.
11. Click **Next**.
12. On the Summary page of the Statistics wizard, select the **Run this settings immediately** check box.
13. Click **Finish**.

Creating a statistics function

In this step of the mining tutorial, you will create a Bivariate Statistics function. This function will produce descriptive statistics about the fields in for scored data.

To create a statistical function:

1. Click the **Create statistics** button from the toolbar on the Intelligent Miner main window. The Statistics wizard opens.
2. Click **Next** to continue.

Specifying the statistics function and name

On the Statistics functions and settings page, you select the type of statistics function that you want to create, and specify the name and comment for this statistics function:

1. Ensure that **Bivariate Statistics** in the list of statistics functions is selected.
2. Type the settings name `Analyze`. Optionally, type a descriptive comment.
3. Check the **Show the advanced pages and controls** check box.
4. Click **Next** to continue.

Specifying the input data for statistics function

To specify the input data for this statistics function:

1. From the list of available input data, select the **Scored customers** data object that you created in a previous step.
2. Under the Advanced options, click the **...** button next to **Filter records condition**. You will be filtering the records of the input data for this statistics function.

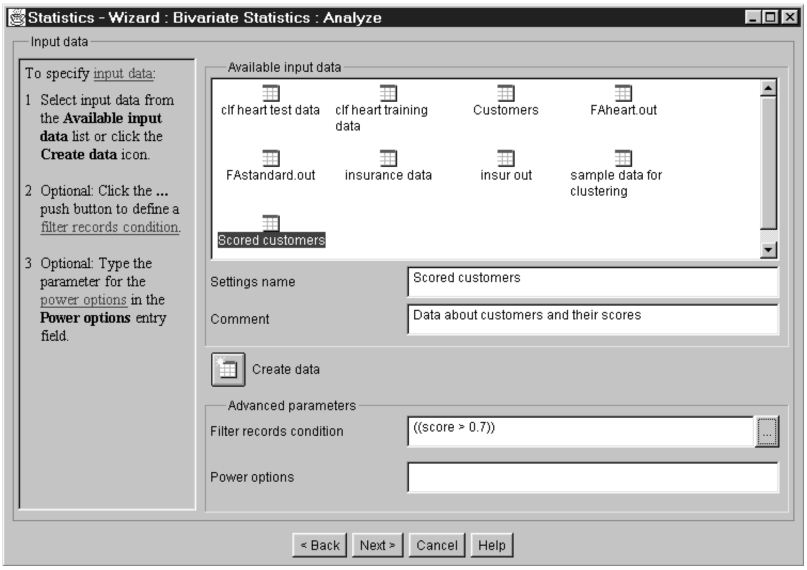
   The Intelligent Miner displays the Expression Builder for filtering records.

   In a previous step of this tutorial, you created a score for each record in the input data. The score ranges from 0 to 1, and a higher score means a greater similarity. For this tutorial, you include all input data records for customers whose score is greater than 0.7.

   The expression you want to create looks like this: `((score > 0.7))`. To create this expression:

   a. Click the **AND** push button. The expression builder creates a template for the expression, which appears as `((Arg1 = Arg2))` on the first line.
   b. Click on **Field Names** in the **Category** list.
   c. Scroll through the list of field names and select the field **score**.
   d. Click the **Arg1** button. The field score is set as the first argument in the expression.
   e. Click the **>** button.
   f. Click **Constants** in the **Category** list.
   g. Double-click **<new constants>** from the **Value** list.
   h. Type the constant value `0.7`.
   i. Press **Enter**.
   j. Select **0.7** from the list of constants.
   k. Click on the **Arg2** button.

   The expression looks like this: `((score > 0.7))`.
3. Click **OK** to return to the Input Data page of the Statistics wizard, which now looks like Figure 47 on page 336.



To specify input data:

1 Select input data from the **Available input data** list or click the **Create data** icon.

2 Optional: Click the **...** push button to define a filter records condition.

3 Optional: Type the parameter for the power options in the **Power options** entry field.

Settings name: Scored customers

Comment: Data about customers and their scores

Filter records condition: ((score > 0.7))

Power options:

Figure 47. Input data page of the Statistics wizard

4. Click **Next**.
5. On the Parallel parameters page of the Statistics wizard, make sure that you are running this function in serial mode, on the Intelligent Miner server node.
6. Click **Next**.

Computing statistics, quantiles, or a sample

On the Statistics page of this wizard, you specify whether you want to compute statistics. For this tutorial, you will compute univariate statistics for the selected input data fields:

1. Select the **Compute statistics** radio button.
2. Select **age**, **clusterID**, **conf**, **gender**, **income**, **product**, **score**, and **siblings**, and click the **>** button to compute the univariate statistics for these fields.
3. Click **Next** to continue.
4. You will not compute quantiles in this tutorial. Click **Next**.
5. You will not create a sample in this tutorial. Click **Next**.

Specifying output fields

On this page of the Statistics wizard, you can choose whether to create output data, and what fields to include in the output data. Only input data records that meet the filter condition that you specified will be included in the output data. In this case, the output data will contain customers whose scores are greater than 0.7.

1. Click **Create an output table**.

2. Click >> to add all the available fields to the list of output fields.
3. Click **Next** to continue.

Specifying the output data object name

On this page of the wizard, you specify the name of the output data object for this statistics function. This output data object must exist before you can proceed. Because you have not yet defined the output data object, you will need to define the output data object from this page of the Statistics wizard:

1. Click on **Create data**. The Data wizard opens.
2. Click **Next** to continue.
3. Select **Flat files**.
4. Enter `Target customers` as the name of the data object. Optionally, type a comment that describes this data object.
5. Click **Next** to continue.
6. On the Flat files page, change to the directory that contains the file banking.txt.
7. In the **Path and file name** entry field, append `target.txt` to the path.
8. Click **Add** file.
9. Click **The specified flat file does not yet exist**.
10. Click **Next**.
11. On the Summary page of the Data wizard, click **Finish**. You now return to the Statistics wizard to continue the process of defining the statistics object.
12. On the output data page of the Statistics wizard, select the data object **Target customers** from the **Available output data** field.
13. Click **Next**.

Specifying the result object name

On the Results page of the Statistics wizard, enter the name of the result object that is generated by this statistical function:

1. Type `Target customer demographics` in the **Results name** field.
2. Check the **If a result with this name exists, overwrite it** check box.
3. Click **Next** to continue to the Summary page of the wizard.

Running the statistical function

To run the statistical function, check the **Run this settings immediately** check box.

Click **Finish** to continue.

The Intelligent Miner now runs this statistical function. A progress indicator shows the status of the function. After completion, the Intelligent Miner automatically displays the results of the statistical function.

Save the mining base from the Intelligent Miner main window. Click on the Save Mining Base icon.

You have produced an output data file named Target customer demographics. This file contains the customers identified as having high scores of similarity with typical customers of the Premier Checking account. You can now analyze the results.

Interpreting the results

The result generated by the Analyze statistical function is shown in Figure 48 on page 339. The multiple graphs show the distribution of the fields that you selected for statistics. The visualizer allows you to display more detail by double-clicking on any of the graphs. The Bivariate Statistics function provides you with statistics about the customers that have been targeted for your ad campaign. A quick look at the product graph shows that the customers similar to typical Premier Checking account customers are distributed fairly evenly over several other products.

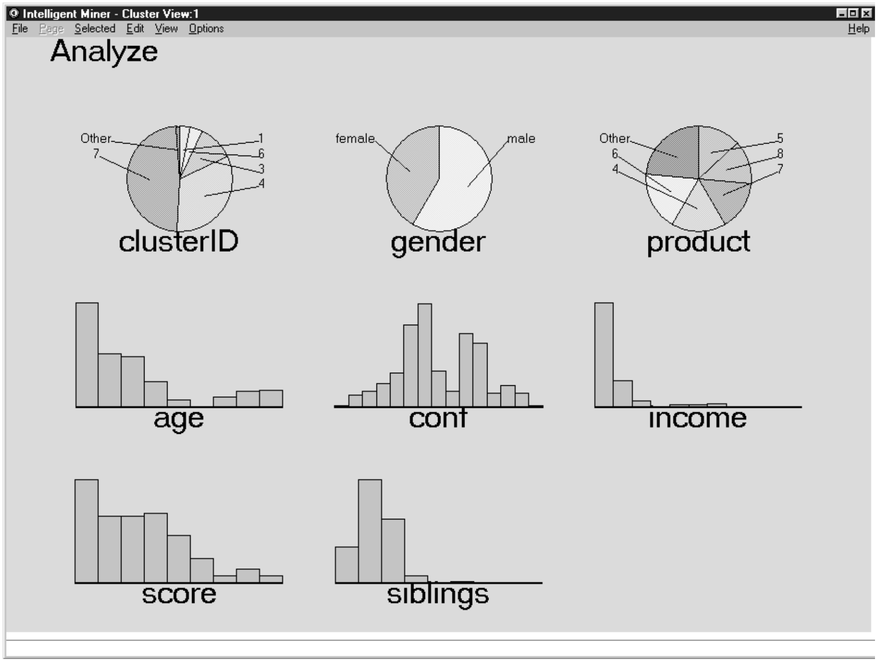
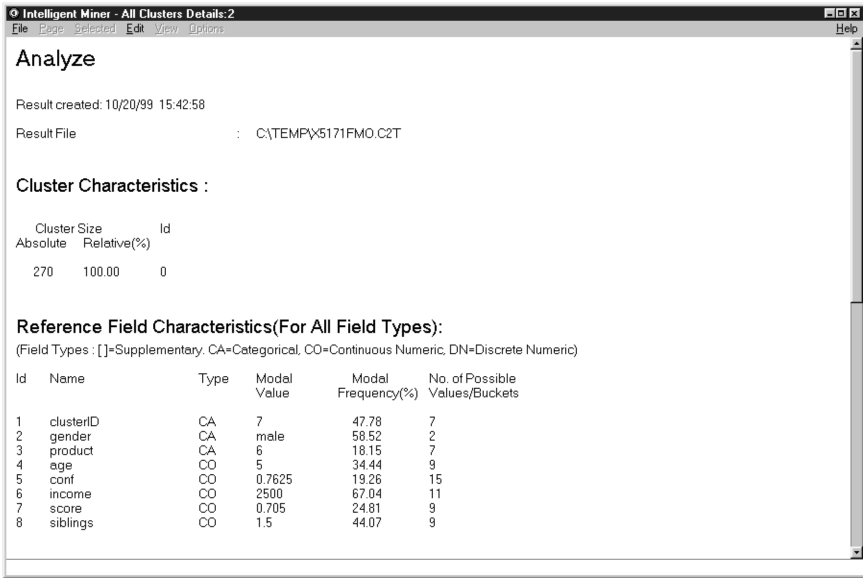


Figure 48. Results of the Analyze statistical function

The detailed statistics computed by mining and statistics object are shown on the details page. The top portion of the details page for the results generated by the Analyze statistical function is shown in Figure 49 on page 340.


Intelligent Miner - All Clusters Details:2
File Page Selected Edit View Options Help

Analyze

Result created: 10/20/99 15:42:58

Result File : C:\TEMP\X5171FMO.C2T

Cluster Characteristics :

```
    Cluster Size          Id
Absolute    Relative(%)

   270       100.00        0
```

Reference Field Characteristics(For All Field Types):

(Field Types : [ ]=Supplementary. CA=Categorical, CO=Continuous Numeric, DN=Discrete Numeric)

Id	Name	Type	Modal Value	Modal Frequency(%)	No. of Possible Values/Buckets
1	clusterID	CA	7	47.78	7
2	gender	CA	male	58.52	2
3	product	CA	6	18.15	7
4	age	CO	5	34.44	9
5	conf	CO	0.7625	19.26	15
6	income	CO	2500	67.04	11
7	score	CO	0.705	24.81	9
8	siblings	CO	1.5	44.07	9

Figure 49. Details for results generated by the Analyze statistical function

To access this page, use the **Details for all partitions** menu item under the View menu on the menu bar. In Figure 49, you can see that of the possible 1792 customers who do not have Premier Checking, 315 had a score of 0.7 or greater. These are the customers who have similar demographics to those who have purchased Premier Checking. You could adjust the filter to 0.6 to include more customers in the analysis.

Appendix B. Command-line tools

The Intelligent Miner is equipped with a number of useful tools, which you can start from the command line or the command prompt. In general, these tools carry out actions that you are likely to perform in isolation so that there is no need to start the graphical user interface of the Intelligent Miner.

Converting CSV files into Intelligent Miner flat files

The idmcsv program converts files in comma-separated variables format (CSV) to flat files with a fixed record length, which you can process with the Intelligent Miner. This allows you to mine data that you create with external applications, such as SPSS. Optionally, the idmcsv program also generates a structure file that you can use with the loadmnb program. See "Creating flat-file data objects with a single command" on page 345 for more information.

The idmcsv program is available on all Intelligent Miner servers. To make it easy to use, the idmcsv program automatically determines separating characters and string delimiters in your CSV file. However, if this automated process does not produce the desired results, you can customize the idmcsv program by setting several environment variables.

The following diagram shows the syntax of the **idmcsv** command:

idmcsv

```
                                (1)
►►──idmcsv──CsvFile──FixedFile──────┬──────────────────┬──────────────────────────────────────►◄
                                    │             (2)  │
                                    └─LoadmnbFile──────┘
```

Notes:

1 Include the absolute path to the file to make sure that you can later access the file from various locations, regardless of a specific directory structure.

2 To use it, you must copy the loadmnb file to a client workstation.

Parameters

CsvFile
Name of the input file in CSV format.

FixedFile
Name of the Intelligent Miner flat file that you want to create.

LoadmnbFile
Name of the structure file that you want to create for use with the loadmnb program.

The idmcsv program determines the separating character in the CSV file by inference. For a character to be selected as the separating character, the following conditions must be met:

- The character must occur at least once per line.
- There must be an equal number of this character in each line, disregarding any occurrences within string delimiters.

The first character that meets all the conditions is regarded as the separating character. During the selection process, the idmcsv program checks the following characters in the order shown, and then chooses one of them:

1. Tab stop (\t)
2. Vertical bar (|)
3. Semicolon (;)
4. Comma (,)
5. Blank or space character ( )

String delimiters are considered during the selection process. If a character occurs within a string, it does not affect the determination of the separating character. The idmcsv program performs two checks. By default, double quotes and single quotes are considered to be the string delimiters. The idmcsv program assumes that double quotes are the delimiters during the first check. If the program does not find any instances supporting this assumption, it performs a second check, suggesting that the string delimiters are single quotes. The string delimiters themselves do not appear in the output.

The escape symbol policy is heeded; if two string delimiters follow one another immediately, they do not mark the end of a string; instead, one of these string delimiters is ignored when the output file is written.

In the output file, all occurrences of the separating character are replaced with blank characters by default. Blank characters are added to column entries that are shorter than the longest entry in the column so that each entry has the same length.

The first line of the CSV file is supposed to contain column headers. It is not written to the output file, unless one of the following conditions applies:

- The values of different column entries in the first line are identical.
- One or more columns in the first line do not contain an entry.

- One or more columns in the first line contain nonintegral float numbers.

The idmcsv program also determines the Intelligent Miner data type. If a column contains one or more values that are not numerical, the data in that column is considered to be categorical. If one or more values in a column containing exclusively numerical data are float numbers, the data is considered to be continuous. If the data in a column consists of integers, it is considered to be discrete-numeric.

Eventually, a character that marks the end of the line is added to each line in the output file. By default, this is a period (.).

If the idmcsv program does not determine the proper characters, or if you simply want to use other characters, you can customize the idmcsv program by setting the following environment variables:

Table 91. Environment variables for the idmcsv program

Environment variable	**Valid values**	**Function/Effect**
IDM_CSV_SEPCHAR	Any character.	Specifies the character to be taken as the separating character in the CSV file. If you set this variable, you bypass the selection process, which is based on inference.
IDM_CSV_SEPCHAR_OUT	Any character string. The default is a blank character.	Replaces the separating character in the output file (blank character) with a character string of your choice.
IDM_CSV_TERMINATOR_STRING	Any character string. The default is a period.	Replaces the end-of-line character in the output file (period) with a character string of your choice and adds a terminating sequence at the end of each line.
IDM_CSV_NEWLINE_BYTES	Any positive integer including 0. On AIX, the default is 1. On OS/2 and Windows, the default is 2.	Specifies a number to be added to the *reclen* value in the last line of a loadmnb structure file. The *reclen* value specifies the record length. For a discussion of the record length, see “Exporting and importing mining bases” on page 134.

Table 91. Environment variables for the idmcsv program (continued)

Environment variable	Valid values	Function/Effect
IDM_CSV_HEADER_LINES	Any positive integer including 0. The default is 1.	Specifies the number of lines from the top of your CSV file to be regarded as column headers. By default, this is only the first line. Set this variable to 0 if your file does not contain column headers. Set it to a value > 1 if the column headers occupy more than 1 line.
IDM_CSV_STRING_DELIM	Any character string. The default are double quotes.	Replaces the default string delimiter in your CSV file (double quotes) with a character string of your choice and bypasses the string delimiter selection process.
IDM_CSV_MISSING_VAL	Any character string.	Specifies the character string in your CSV file considered to represent missing values. In the output file, this value is replaced with the empty string.

Notes:

- The length of each column is restricted to 4096 bytes. Longer columns are truncated.
- The period and the comma are accepted as decimal symbols, regardless of any locale setting.
- In the CSV file, the following character strings that indicate a new line are correctly interpreted:
  - Just line-feed (UNIX)
  - Just carriage return (Macintosh)
  - Line-feed and carriage return (DOS)
- For the lines that are written to the output file, the type of the new lines conforms to the standards of the operating system on which you run the idmcsv program.

Creating flat-file data objects with a single command

Normally, when you want to mine flat files, you must create a data object. In this data object, you define the fields to be mined by specifying column ranges. If your flat file contains a large number of “fields”, this can be a wearisome task. Using the loadmnb program, you can define all the necessary fields with a single command. Depending on whether the mining base already exists or not, the Intelligent Miner creates a new data object including the field definitions and a new mining base, or adds the new data object to an existing mining base.

The loadmnb program is available for AIX and Windows clients. You find the loadmnb program in the bin directory, which is a subdirectory of the directory in which you installed the Intelligent Miner. To run the loadmnb program, enter **loadmnb** from the command line or command prompt. An interactive program starts, which requests you to make further specifications.

You are asked to specify the name of the structure file. The structure file is a file containing field definitions. The samples directory, which is a subdirectory of the directory in which you installed the Intelligent Miner, contains a sample structure file called loadmnb.dat. Open this file with an editor of your choice to view its contents. Line entries in the structure file contain blank-separated information for field definitions, in the order shown:

```
startcol stopcol type dataFieldName
```

startcol
The beginning column of the field. The first column is column 1.

stopcol
The end column of the field.

type
The Intelligent Miner data type, defined by an integer. Integers correspond to Intelligent Miner data types as shown in Table 92.

Table 92. Integers defining the Intelligent Miner data type

Integer	Intelligent Miner data type
0	categorical
1	continuous
2	discrete numeric
4	numeric
5	binary

dataFieldName
The name of the field defined by the line entry. You can choose a name containing blanks.

When prompted, change the default name of the *CurrentSource/TargetFile*. Click **Compute** at the end of the dialog to adjust the record length.

Using the idmcsv program, you can create the structure file automatically from an input file in CSV format. See “Converting CSV files into Intelligent Miner flat files” on page 341 for more information.

Running settings objects

By using the idmeruns program, you can perform a single data mining operation without having to start an Intelligent Miner client session. The idmeruns program opens the mining base and runs the settings object that you specify. Enter the **idmeruns** command from the command line or the command prompt. The following diagram shows the syntax of this command:

idmeruns

```
►►──idmeruns──┬──────────────────┬──┬────────────────────┬──MiningbaseName──SettingsName──────────────►◄
              └─-result Filename─┘  ├─-server────────────┤
                                    ├─-s─────────────────┤
                                    └────────────────────┘
                                      ┌─-userid──────────┐
                                      ├─-u───────────────┤
                                      ├──────────────────┤
                                      │ ┌─-password──┐   │
                                      │ └─-p─────────┘   │
                                      └─-p-──────────────┘
```

Parameters

MiningbaseName
: The name of the mining base that you want to open. If the name of the mining base contains blanks, enclose the name in single or double quotes. See the following example:

```
idmeruns "My Project" "Fraud Classification"
```

-password *Password* or **-p** *Password*
: Your password on the server that you specified.

  You can also set the IDM_PASSWORD environment variable to specify the password. If you neither specify *-password*, nor IDM_PASSWORD, the idmeruns program tries to use the password of the user who started the program.

-p-
: Prompts you to enter your password for the specified server. Use this parameter as an alternative to the *-password* parameter.

-result Filename
: The name of the file that results are written to. When you specify this parameter, any results that are created at the end of the mining run are written to the file that you specify.

Example: When you specify *-result EndRes99.dat*, the results are written to a file named EndRes99.dat.

-server *ServerName* or **-s** *ServerName*
The name of the server on which your mining base resides.

You can also set the IDM_SERVER environment variable to specify the server. If you neither specify the *-server* option, nor the IDM_SERVER environment variable, the idmeruns program tries to locate the mining base on the local workstation.

SettingsName
The name of the settings object that you want to run. The settings object must be included in the mining base that you specify. If the settings name contains blanks, enclose the name in single or double quotes. See the following example:

```
idmeruns "My Project" "Fraud Classification"
```

-userid *UserName* or **-u** *UserName*
Your user ID on the server that you specified.

You can also set the IDM_USERID environment variable to specify the user ID. If you neither specify the *-userid* option, nor the IDM_USERID environment variable, the idmeruns program tries to use the user ID of the user who started the program.

Important:

- If the mining run is successfully completed, the idmeruns program returns a 0. A return code other than 0 indicates an error.
- When you enter an interrupt command such as CTRL + C to stop the idmeruns program, you do not necessarily stop the server process, too.

Appendix C. Environment variables

This chapter describes the environment variables that you can use with the Intelligent Miner.

Server variables

Table 93 lists the environment variables that you can set on the server.

Table 93. Server environment variables

Name of variable	Possible values	Comments
IDM_BIN_DIR		Contains the executables. This variable is set by the shell script that starts the Intelligent Miner. *OS/390*: This variable points to the HFS directory that contains all Intelligent Miner load modules. These are empty files having the sticky bit set. The variable must be set.
IDM_DB_COMMIT_COUNT	Any positive integer, unset.	Specifies the number of database table inserts after which a commit is performed. If the variable is unset, a commit is performed after the last insert.
IDM_DB_FILE	Any path, unset.	*OS/390 only*. Sets the path to an HFS file that contains a list of DB2 subsystem IDs (SSIDs) valid for your installation. This file must contain one line for each SSID starting at column 1. (Required only if IDM_CLI_USED is not set on the client.)
IDM_HOME_DIR	Any directory, unset.	*Windows NT only.* Specifies the location of mining bases and result files if IDM_MNB_DIR or IDM_RES_DIR are not set.
IDM_LOCAL_DB2_SSID	Any valid DB2 subsystem ID (SSID)	*OS/390 only.* Name of the DB2 subsystem used for preprocessing functions. The variable must be set.

Table 93. Server environment variables (continued)

Name of variable	Possible values	Comments
IDM_MNB_DIR	Any directory, unset.	Specifies the location of the mining bases. If the variable is unset, the HOME directory is used. On Windows NT, the directory specified by %IDM_HOME_DIR% is used. Note that the mining bases are located in a subdirectory called idmmnb.
	Note: In C/S mode, you can set this variable before starting the server using the **idmstart** command. If you do this, all connecting clients share the same set of mining bases.	
IDM_RES_DIR	Any directory, unset.	Result file location. If this variable is unset, IDM_MNB_DIR is used. On Windows NT, the directory specified by %IDM_HOME_DIR% is used. Note that the result files are located in a subdirectory called idmres.
IDM_SERVER_PORT	Any server port, unset.	Forces RPC communication on the specified port, bypassing the RPC portmapper. On AS/400, this variable has no meaning and is therefore ignored. If the variable is unset, the port to be used is assigned by the RPC portmapper.
	Note: This variable must be set to the same port on the client.	
IDM_STOGROUP_PREFIX	Any valid value for a storage group name or storage group prefix.	*OS/390 only*. This variable is used when tablespaces or indexes are created on a DB2 for OS/390 database server.

Notes:

1. If the variable IDM_MNB_DIR is set, all clients share a common directory and thus the same set of mining bases. For OS/390, this means that before performing any mining runs, you must:

- Create the IDM_MNB_DIR/idmmnb directory.
- Grant all clients READ/WRITE/EXECUTE access to the directory IDM_MNB_DIR/idmmnb (using GROUP or OTHER).
- After mining bases are created, grant all clients READ/WRITE access to the file fileids.dat and READ access to all other files.
- Grant all clients EXECUTE access to all subdirectories in the IDM_MNB_DIR directory.

2. If the variable IDM_RES_DIR is set, all clients write results to the same directory. For OS/390, this means that before performing any mining runs, you must:
   - Create the IDM_RES_DIR/idmres directory.
   - Grant all clients READ/WRITE/EXECUTE access to the IDM_RES_DIR/idmres directory (using GROUP or OTHER).
   - Grant all clients EXECUTE access to all subdirectories in the IDM_RES_DIR directory.
3. On the Windows NT server, the location of mining bases and result files is usually determined by the value of the IDM_HOME_DIR environment variable. Under the directory that this variable is set to, the Intelligent Miner creates a subdirectory for each client user. The clients' user IDs determine the names of these subdirectories. These subdirectories are the parent directories of the users' idmmnb and idmres directories.

   Example: If IDM_HOME_DIR is set to the directory E:\imhome, the mining bases of user Ted would be located in the E:\imhome\Ted\idmmnb directory. Ted's result files would be located in a directory called E:\imhome\Ted\idmres.
4. Use the variable IDM_DB_COMMIT_COUNT to control the COMMIT frequency when inserting rows into an output table. If this number is low, performance can be impaired. If it is high, more disk space is needed to buffer the output data for rollbacks. The recommended value for this variable is 10000.

Client variables

Table 94 on page 352 lists the environment variables that you can set on the clients.

Table 94. Client environment variables

Name of variable	Possible values	Comments
IDM_DEBUG	Unset or `on`.	Show submitted kernel/viewer calls. If the variable is unset, the kernel/viewer call information is suppressed.
IDMG_BLINK_RATE	Any integer, unset.	Animation in milliseconds per picture. If the variable is unset, the animation is 250 milliseconds per picture.
IDM_CS_TIMEOUT	Integer (indicating seconds).	Set to avoid timeouts on the client side while waiting for a server reply. If this variable is unset, the system default for TCP/IP is used.
IDM_CLI_USED	Any value, unset.	If set, the client accesses DB2 directly. Note that DB2 CAE or DB2 Connect must be installed and configured. If unset (default), the client accesses DB2 through the Intelligent Miner server.
IDM_MAX_BROWSE_SIZE	Any positive integer, unset.	Specifies a limit in bytes for the maximum size of the HTML file generated when you apply the Browse function. If the variable is unset, the maximum size of the HTML file is one MB.
IDM_MAX_EXP_RESULT_SIZE	Any positive integer, unset.	Specifies a limit in bytes for the maximum size of a result file to be exported. If the variable is unset, the size of the exported result file is unlimited.

Table 94. Client environment variables (continued)

Name of variable	Possible values	Comments
IDM_SERVER_PORT	Any server port, unset (default).	Forces RPC communication on the specified port, bypassing the RPC portmapper. If the variable is unset, the port to be used is assigned by the RPC portmapper.
	Note: This variable must be set to the same port on the server.	
IDM_TREEVIEW_OPTIMIZE	`1`, `0` or unset.	If this variable is set to 1, the Optimize Sparse Trees check box is checked when the Classification Tree window opens.
TEXTRULENOUN	Any text string.	Determines the subject in the sentences used to display association rules in textual format. The default is "customer".
TEXTRULEVERB	Any text string.	Determines the verb in the sentences used to display association rules in textual format. The default is "buys".

Restriction: If you set the IDM_MAX_EXP_RESULT_SIZE environment variable, the system considers it only if no limit is set in the client-side tool registration file.

Appendix D. Description file keywords

The following keywords indicate object groups in the description file. This file is created when you export a mining base. Each object contains the values of settings you specified for the various functions.

Keywords related to mining base objects

The following keywords might exist in the description file, depending on the objects you have created:

```
#idm-name-mapping-objects
#idm-data-objects
#idm-taxonomy-relation-objects
#idm-value-mapping-objects
#idm-discretization-objects
```

If one of these keywords exists in the description file, it is always followed by another keyword that introduces the list of object names belonging to the object group, for example, `#idm-name-mapping-object-name`. Any of these object name keywords in turn involves one of the following keywords because each object is related to a specific data format:

```
#idm-flat-file-table
#idm-db2-table
#idm-pipe-table
```

These keywords indicate that specifications for the described data format follow.

Keywords related to settings for preprocessing functions

The following keywords might exist in the description file, depending on the preprocessing function settings you specified:

```
#idm-copy-records-to-file-objects
#idm-aggregate-values-objects
#idm-filter-fields-objects
#idm-cleanup-data-sources-objects
#idm-calculate-values-objects
#idm-discretization-into-quantiles-objects
#idm-discard-records-with-missing-values-objects
#idm-discretization-using-ranges-objects
#idm-encode-missing-values-objects
#idm-encode-nonvalid-values-objects
#idm-filter-records-using-a-value-set-objects
#idm-group-records-objects
#idm-run-sql-objects
```

```
#idm-join-data-sources-objects
#idm-convert-to-lowercase-or-uppercase-objects
#idm-pivot-fields-to-records-objects
#idm-filter-records-objects
#idm-get-random-sample-objects
#idm-map-values-objects
```

If one of these keywords exists in the description file, it is always followed by the keyword `#idm-processing-object-name`, which introduces the list of object names belonging to the object group. Any occurrence of `#idm-processing-object-name` is in turn followed by `#idm-processing-object-values`, which indicates that the values for a specific object follow.

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 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 the

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

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.

Trademarks

AIX	AFS
Database 2	DataJoiner
DB2	DB2 Universal Database
IBM	Intelligent Miner
iSeries	MVS
MVS/ESA	OpenEdition
OS/390	OS/400
pSeries	@server
RACF	RS/6000
SP	System/390

Glossary

This glossary defines terms as they are used in this book. If you do not find the term you are looking for, see the *IBM Dictionary of Computing*.

A

adaptive connection. A numeric weight used to describe the strength of the connection between two processing units in a neural network. The connection is called adaptive because it is adjusted during training. Values typically range from zero to one, or -0.5 to +0.5.

AFS. Andrew File System. A distributed file system developed by IBM and Carnegie-Mellon University.

aggregate. To summarize data in a field.

application program interface (API). A functional interface supplied by the operating system or a separately orderable licensed program that allows an application program written in a high-level language to use specific data or functions of the operating system or the licensed program.

architecture. The number of processing units in the input, output, and hidden layers of a neural network. The number of units in the input and output layers is calculated from the mining data and input parameters.

associations. The relationship of items in a transaction in such a way that items imply the presence of other items in the same transaction.

attribute. Characteristics or properties that can be controlled, usually to obtain a required appearance. For example, the color is an attribute of a line. In object-oriented programming, a data element defined within a class.

B

back propagation. A general-purpose neural network named for the method used to adjust its weights while learning data patterns. The Neural Classification mining function uses such a network.

boundary field. The upper limit of an interval as used for the Discretization using ranges processing function.

bucket. One of the bars in a bar chart representing the frequency distribution of a continuous field. A bucket shows how many values lie within a specific frequency range.

C

Central Processing Complex (CPC). A group of up to 10 interconnected processors in a parallel sysplex.

chi-square test. A test to check whether two variables are statistically dependent or not. Chi-square is calculated by subtracting the expected frequencies (imaginary values) from the observed frequencies (actual values). The expected frequencies represent the values that were to be expected if the variables in question were statistically independent.

classification. The assignment of objects into groups or categories based on their characteristics.

cluster. A group of records with similar characteristics.

cluster prototype. The attribute values that are typical of all records in a given cluster. Used to compare the input records to determine if a record should be assigned to the cluster represented by these values.

clustering. A mining function that creates groups of data records within the input data on the basis of similar characteristics. Each group is called a *cluster*.

confidence factor. Indicates the strength or the reliability of the associations detected.

comma-separated variables format (CSV). A file format used by spreadsheet, database, and statistical applications.

CPC. See *Central Processing Complex*.

CSV. See *comma-separated variables file format*.

D

Database 2 (DB2). An IBM relational database management system.

database table. A table residing in a database.

database view. An alternative representation of data from one or more database tables. A view can include all or some of the columns contained in the database table or tables on which it is defined.

data field. In a database table, the intersection from table description and table column where the corresponding data is entered.

data format. There are different kinds of data formats, for example, database tables, database views, pipes, or flat files.

data table. A data table, regardless of the data format it contains.

data type. There are different kinds of Intelligent Miner data types, for example, categorical, continuous, or discrete-numeric.

delimiter. A character used to indicate the beginning and end of a character string.

discrete. Pertaining to data that consists of distinct elements such as characters, or to physical quantities having a finite number of distinctly recognizable values.

discretization. The act of assigning continuous values to intervals.

distributed file system. A file system composed of files or directories that physically reside on more than one computer in a communication network.

dotted decimal. A common notation for Internet host addresses that divides the 32-bit address into four 8-bit fields. The value of each field is specified as a decimal number and the fields are separated by periods, for example, 010.002.000.052 or 10.2.0.52.

double-byte character set (DBCS). A set of characters in which each character is represented by two bytes.

E

envelope. The area between two curves that are parallel to a curve of time-sequence data. The first curve runs above the curve of time-sequence data, the second one below. Both curves have the same distance to the curve of time-sequence data. The width of the envelope, that is, the distance from the first parallel curve to the second, is defined by epsilon.

epsilon. The maximum width of an envelope that encloses a sequence. Another sequence is epsilon-similar if it fits in this envelope.

epsilon-similar. Two sequences are epsilon-similar if one sequence does not go beyond the envelope that encloses the other sequence.

equality compatible. Pertaining to different data types that can be operands for the = logical operator.

Euclidean distance. The square root of the sum of the squared differences between two numeric vectors. The Euclidean distance is used to calculate the error between the calculated network output and the target output in Neural Classification, and to calculate the difference between a record and a prototype cluster value in Neural Clustering. A zero value indicates an exact match; larger numbers indicate greater differences.

F

field. A set of one or more related data items grouped for processing. In this document, with regard to database tables and views, *field* is synonymous to *column*.

file. A collection of related data that is stored and retrieved by an assigned name.

file name. (1) A name assigned or declared for a file. (2) The name used by a program to identify a file.

file-selection box. A box that enables the user to choose a file to work with by selecting a file name from the ones listed or by typing a file name into the space provided.

file specification. The name and location of a file.

file system. The collection of files and file management structures on a physical or logical mass storage device, such as a diskette or minidisk. See distributed file system, virtual file system.

flat file. (1) A one-dimensional or two-dimensional array: a list or table of items. (2) A file that has no hierarchical structure.

formatted information. An arrangement of information into discrete units and structures in a manner that facilitates its access and processing. Contrast with *narrative information*.

frequent item sets. The total volume of items above the specified support factor returned by the Associations mining function.

F-test. A statistical test that checks whether two estimates of the variances of two independent samples are the same. In addition, the F-test checks whether the null hypothesis is true or false.

function. Any instruction or set of related instructions that perform a specific operation.

H

hidden layer. A set of processing units in a neural network used to calculate its outputs. Hidden layer processing units take their inputs from the preceding hidden layer units, or from the input layer. Their outputs are passed to either a succeeding hidden layer or the network's output layer. The number of hidden layers and the number of processing units in each hidden layer is part of the network architecture.

host. Pertaining to a computer controlling all or part of a network, and providing an access method to that network.

I

index. In SQL, pointers that are logically arranged by the values of a key. Indexes provide quick access and can enforce uniqueness on the rows in a table.

input data. The metadata of the database table, database view, or flat file containing the data you specified to be mined.

input layer. A set of processing units in a neural network which present the numeric values derived from user data to the network. The number of fields and type of data in those fields is used to calculate the number of processing units in the input layer.

interval. A set of real numbers between two numbers either including or excluding both of them.

interval boundaries. Values that represent the upper and lower limits of an interval.

item category. A categorization of an item. For example, a room in a hotel can have the following categories: Standard, Comfort, Superior, Luxury. The lowest category is called child item category. Each child item category can have several parent item categories. Each parent item category can have several grandparent item categories.

item description. The descriptive name of a character string in a data table.

item ID. The identifier for an item.

item set. A collection of items. For example, all items bought by one customer during one visit to a department store.

K

key. In SQL, a column or an ordered collection of columns identified in the description of an index.

Kohonen Feature Map. A neural network model comprised of processing units arranged in an input layer and output layer. All processors in the input layer are connected to each processor in the output layer by an adaptive connection. The learning algorithm used involves competition between units for each input pattern and the declaration of a winning unit. Used in neural clustering to partition data into similar record groups.

L

learning algorithm. The set of well-defined rules used during the training process to adjust the connection weights of a neural network. The criteria and methods used to adjust the weights define the different learning algorithms.

learning parameters. The variables used by each neural network model to control the training of a neural network which is accomplished by modifying network weights.

lift. Confidence factor divided by expected confidence.

M

metadata. In databases, data that describes data objects.

mining. Synonym for analyzing or searching.

mining base. A repository where all the information about the mining data, the mining run settings, and the corresponding results is stored.

model. A specific type of neural network and its associated learning algorithm. Examples include the Kohonen Feature Map and back propagation.

mount. (1) To place a data medium in a position to operate. (2) To make recording media accessible.

N

name mapping. A table containing descriptive names or translations of other languages mapped to the numerals or the character strings of a data table.

named pipe. A named buffer that provides client-to-server, server-to-client, or full duplex communication between unrelated processes.

narrative information. Information that is presented according to the syntax of a natural language. Contrast with formatted information.

neural network. A collection of processing units and adaptive connections that is designed to perform a specific processing function.

Neural Network Utility (NNU). A family of IBM application development products for creating neural network and fuzzy rule system applications.

nonsupervised learning. A learning algorithm that requires only input data to be present in the data source during the training process. No target output is provided; instead, the desired output is discovered during the mining run. A Kohonen Feature Map, for example, uses nonsupervised learning.

O

offset. (1) The number of measuring units from an arbitrary starting point in a record, area, or control block, to some other point. (2) The distance from the beginning of an object to the beginning of a particular field.

operator. (1) A symbol that represents an operation to be done. (2) In a language statement, the lexical entity that indicates the action to be performed on operands.

output data object. The metadata of the database table, database view, or flat file containing the data being produced or to be produced by a function.

output layer. A set of processing units in a neural network which contain the output calculated by the network. The number of outputs depends on the number of classification categories or maximum clusters value in Neural Classification and Neural Clustering, respectively.

P

Parallel Operating Environment (POE or poe). POE is a program (poe in AIX) required for parallel processing. It assigns the tasks to the different parallel processing nodes or CPCs.

pass. One cycle of processing a body of data. During a pass, each record is read once.

path. The route used to locate files; the storage location of a file. A fully qualified path lists the drive identifier, directory name, subdirectory name (if any), and file name with the associated extension.

pipe. A named or unnamed buffer used to pass data between processes.

POE. See *Parallel Operating Environment*.

prediction model. A model of the dependency and the variation of one field's value within a record on the other fields within the same record. A profile is then generated that can predict a value for the particular field in a new record of the same form, based on its other field values.

processing unit. A processing unit in a neural network is used to calculate an output value by summing all incoming values multiplied by their respective adaptive connection weights.

Q

quantile range. One of a finite number of nonoverlapping subranges or intervals, each of which is represented by an assigned value.

Q is an *N%-quantile* of a value set *S* when:

- Approximately *N* percent of the values in *S* are lower than or equal to *Q*.
- Approximately (100-*N*) percent of the values are greater than or equal to *Q*.

The approximation is less exact when there are many values equal to *Q*. *N* is called the quantile label or quantile limit. The 50%-quantile represents the median.

R

Radial Basis Function (RBF). The individual Radial Basis Functions are functions of the distance or the radius from a particular point. They are used to build up approximations to more complicated functions. The RBF-Prediction mining function uses Radial Basis Functions to predict values.

record. A set of one or more related data items grouped for processing. In reference to a database table, *record* is synonymous to *row*.

region. (Sub)set of records with similar characteristics in their active fields. Regions are used to visualize a prediction result.

root. In the AIX operating system, the user name for the system user with the highest authority.

round-robin method. A method by which items are sequentially assigned to units. When an item has been assigned to the last unit in the series, the next item is assigned to the first again. This process is repeated until the last item has been assigned. The Intelligent Miner uses this method, for example, to store records in output files during a partitioning job.

rule. A clause in the form head ⇐ body. It specifies that the head is true if the body is true.

rule body. Represents the specified input data for a mining function.

rule group. Covers all rules containing the same items in different variations.

rule head. Represents the derived items detected by the Associations mining function.

S

scale. A system of mathematical notation: fixed-point or floating-point scale of an arithmetic value.

scaling. To adjust the representation of a quantity by a factor in order to bring its range within prescribed limits.

scale factor. A number used as a multiplier in scaling. For example, a scale factor of 1/1000 would be suitable to scale the values 856, 432, -95, and /182 to lie in the range from -1 to +1, inclusive.

schema. A logical grouping for database objects. When a database object is created, it is assigned to one schema, which is determined by the name of the object. For example, the following command creates table X in schema C:

```
CREATE TABLE C.X
```

self-organizing feature map. See *Kohonen Feature Map.*

sensitivity analysis. An output from the Neural Classification mining function that shows which input fields are relevant to the classification decision.

sequential patterns. Intertransaction patterns such that the presence of one set of items is followed by another set of items in a database of transactions over a period of time.

similar sequences. Occurrences of similar sequences in a database of sequences.

Structured Query Language (SQL). An established set of statements used to manage information stored in a database. By using these statements, users can add, delete, or update information in a table, request information through a query, and display the results in a report.

server node. The processing node of the Intelligent Miner server that you logged on to. If you run the Intelligent Miner in stand-alone mode, this is the node of your local workstation.

supervised learning. A learning algorithm that requires input and resulting output pairs to be presented to the network during the training process. Back propagation, for example, uses supervised learning and makes adjustments during training so that the value computed by the neural network will approach the actual value as the network learns from the data presented. Supervised learning is used in the techniques provided for classification as well as value prediction.

support factor. Indicates the occurrence of the detected association rules and sequential patterns based on the input data.

swapping. A process that interchanges the contents of an area of real storage with the contents of an area in auxiliary storage.

symbolic name. In a programming language, a unique name used to represent an entity such as a field, file, data structure, or label. In the Intelligent Miner you specify symbolic names, for example, for input data, name mappings, or taxonomies.

T

taxonomy. Represents a hierarchy or a lattice of associations between the item categories of an item. These associations are called taxonomy relations.

taxonomy relation. The hierarchical associations between the item categories you defined for an item. A taxonomy relation consists of a child item category and a parent item category.

trained network. A neural network containing connection weights that have been adjusted by a learning algorithm. A trained network can be considered a virtual processor; it transforms inputs to outputs.

training. The process of developing a model which understands the input data. In neural networks, the model is created by reading the records of the input data and modifying the network weights until the network calculates the desired output data.

translation process. Converting the data provided in the database to scaled numeric values in the appropriate range for a mining kernel using neural networks. Different techniques are used depending on whether the data is numeric or symbolic. Also, converting neural network output back to the units used in the database.

transaction. A set of items or events that are linked by a common key value, for example, the articles (items) bought by a customer (customer number) on a particular date (transaction identifier). In this example, the customer number represents the key value.

transaction ID. The identifier for a transaction, for example, the date of a transaction.

transaction group. The identifier for a set of transactions. For example, a customer number can represent a transaction group that includes all purchases of a particular customer during the month of May.

V

vector. A quantity usually characterized by an ordered set of numbers.

virtual file system. In the AIX operating system, a remote file system that has been mounted so that it is accessible to the local user.

W

weight. The numeric value of an adaptive connection representing the strength of the connection between two processing units in a neural network.

winner. The index of the cluster which has the minimum Euclidean distance from the input record. Used in the Kohonen Feature Map to determine which output units will have their weights adjusted.

Bibliography

This bibliography contains lists of publications and related information that are relevant to the contents of this book.

Where appropriate, IBM publication numbers are given after the document title. This will assist you in finding the document online at the IBM Publications Center. It is available on the Web at

`http://www.elink.ibmlink.ibm.com/public/applications/publications/cgibin/pbi.cgi`

IBM Intelligent Miner for Data publications

- *IBM DB2 Intelligent Miner for Data: Using the Intelligent Miner for Data*, SH12-6750
- *IBM DB2 Intelligent Miner for Data: Application Programming Interface and Utility Reference*, SH12-6751
- *IBM DB2 Intelligent Miner Scoring: Administration and Programming for DB2*, SH12-6745
- *IBM DB2 Intelligent Miner Scoring: Administration and Programming for Oracle*, SH12-6746
- *IBM DB2 Intelligent Miner Modeling: Administration and Programming*, SH12-6736
- *IBM DB2 Intelligent Miner Visualization: Using the Intelligent Miner Visualizers*, SH12-6737

IBM DB2 UDB publications

- *IBM DB2 Universal Database: Quick Beginnings for DB2 Servers*, GC09-4836
- *IBM DB2 Universal Database: Quick Beginnings for DB2 Clients*, GC09-4832
- *IBM DB2 Universal Database: Quick Beginnings for DB2 Personal Edition*, GC09-4838

IBM Red books

- *Mining your own Business in Health Care using DB2 Intelligent Miner for Data*, SG24-6274
- *Mining your own Business in Retail using DB2 Intelligent Miner for Data*, SG24-6271
- *Mining your own Business in Banking using DB2 Intelligent Miner for Data*, SG24-6272
- *Mining your own Business in Telecoms using DB2 Intelligent Miner for Data*, SG24-6273

Related Information

- The IBM Software Support Handbook is available on the following Web site:

  `http://techsupport.services.ibm.com/guides/handbook.html`
- Information relating to PMML is available on the following Web site of the Data Mining Group (DMG) at: http://www.dmg.org

Index

Readers' Comments — We'd Like to Hear from You

IBM DB2 Intelligent Miner for Data
Using the Intelligent Miner for Data
Version 8 Release 1

Publication No. SH12-6750-00

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name	Address
Company or Organization	
Phone No.	

Readers' Comments — We'd Like to Hear from You
SH12-6750-00

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape

PLACE
POSTAGE
STAMP
HERE

IBM Deutschland Entwicklung GmbH
Information Development, Dept. 0446
Schoenaicher Strasse 220
71032 Boeblingen
Germany

Fold and Tape | **Please do not staple** | Fold and Tape

SH12-6750-00

Cut or Fold
Along Line

Part Number: CT1K0IE
Program Number: 5724-B93 IBM DB2 Intelligent Miner for Data
5733-IM3 IBM DB2 Intelligent Miner for Data for iSeries
5655-IM3 IBM DB2 Intelligent Miner for Data for OS/390

Printed in Denmark by IBM Denmark A/S

SH12-6750-00

(1P) P/N: CT1K0IE

Spine information:

IBM DB2 Intelligent Miner for Data

Using the Intelligent Miner for Data

Version 8
Release 1