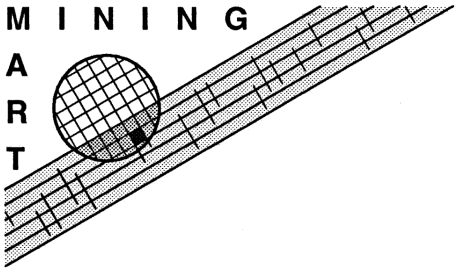

Enabling End-User Datawarehouse Mining
Contract No. IST-1999-11993
MiningMart Installation Guide

Installing the MiningMart system

Timm Euler, Martin Scholz,
Detlef Geppert, Olaf Rem

Dortmund, February 2003

1 General issues

This chapter contains all installation procedures for the different parts of the MiningMart system. Soon an InstallWizard will be available to ease the installation procedure.

MiningMart consists of several modules, which have to be installed separately. One part is an Oracle database which this chapter assumes to be already installed. Into this database, a *metadata schema* must be installed; see section 2. Another central part is the so-called *compiler*, which runs as a Java-written server under Unix and whose installation is described in section 3. The fourth part is the graphical interface to the user, the so-called *HCI* (human-computer interface), which runs as a Java-written client and has so far been tested under Windows, Linux and Unix; section 4 deals with this module. Finally, there is a Java-written interface to the database, which runs as a module in a JBoss server; see section 5.
The system can be downloaded from:
http://www-ai.cs.uni-dortmund.de/MMWEB/downloads/
For downloading JBOSS please visit:
http://www.jboss.org/downloads.jsp

2 Metadata schema

MiningMart makes use of a metamodel to describe the data that the system deals with. This metamodel is called *M4* (MiningMart MetaModel). It is stored in the database in the form of relational tables.

Please note that the MiningMart system generally handles two schemata. The first one is called the *business data schema.* It holds the data you want to analyse and preprocess with the MiningMart system. The second schema, the so called *M4 schema* holds meta-data information about your business data and your preprocessing chains. You should not only reserve sufficient space on disk for your source business data, but account some extra space for materializing some of the views. For the M4 schema, on the other hand, 100 MByte should be sufficient for normal usage. In principle it should be no problem to split the schemata to two different Oracle databases or to use just one schema, referenced for both purposes. Please note, that this has never been tested! The standard installation foresees a separate schema for M4 and business data in the same database.

After creating the two database schemata, the tables of M4, as well as other database-related parts of the MiningMart system, can be created by running an installation script. The scripts can be downloaded from the MiningMart webpages.

1. Please download the file InstallingM4.zip.

2. Unpack it in a new directory, on Unix or Linux you may use the command unzip <filename>.

3. Edit the start script, this is install.sh on Unix and Linux and install.bat on Windows. The database connection information must be entered for the M4 schema. Please adjust the variables M4USER (database user of M4 schema), M4PASS (password), and M4SID (database server). Then you should change the according variables for the business schema, namely BDUSER, BDPASS, and BDSID.

4. If your system does not recognize the commands sqlplus and loadjava, then please set the variable to the absolute path to these ORACLE tools. You should find them in a subdirectory of your ORACLE software.

5. If you have never installed the metamodel before, you can now type ./install.sh on a Unix or Linux machine, or ./install.bat on Windows to have it installed. Otherwise, before running the installation, make sure that no data you might need is still in your previous metamodel, because such data will be lost during installation. If there are compilation errors during installation, please try to run the script for a second time.

3 Compiler

Although the compiler was implemented in Java, it is recommended to install it on a Unix system[1], because some of the external algorithms used by the compiler run only on Unix. The compiler itself was tested for Unix, Linux and Windows2000. Only tested external operators are provided in the runtime packages for each platform. A list of operators using external algorithms can be found in the appendix (section 6).

To run the compiler as a server, please download two files:

- First of all you need the file M4CompilerServer.zip, no matter which operating system you are using. When you unpack it, a directory compiler/ is created with the subdirectories runtime/ and classes/.

- The second file contains the platform specific runtime environment, namely binaries of the external algorithms and some configuration files. It currently is one of the files M4CompilerRuntime_SunOS.zip, M4CompilerRuntime_Linux.zip or M4CompilerRuntime_Windows.zip. Please unpack the file for your operating system into the runtime runtime/ directory previously created when unpacking M4CompilerServer.zip.

There are two things to do:

1. Set up a file compiler/runtime/etc/db.config. You may do so by editing the file db.config.template in the same directory, or by creating a file with a similar content. This file contains the connection information for the two database schemas that are used in MiningMart, that is, the business

[1] For this reason we are going to use the separator character '/' between subdirectories, as common on Unix systems.

data schema and the metadata schema (see section 2). In the file, the two information sets are separated by a blank line; each set contains the name of the database, the user name, the password, the JDBC driver and the database location in one line respectively (so there are five lines for each set). See the template file compiler/runtime/etc/db.config.template and the example file db.config.example. The resulting file must be consistent with the one used by the HCI (section 4).

2. On Unix/Linux, please edit the file compiler/runtime/etc/properties. On Windows there is a similar file with the same functionality, namely compiler\runtime\etc\properties.bat. These property files contain all paths and settings for the start and stop scripts/batch files of the compiler server.

   The following variables must be adjusted to your own environment. In this file, for variables defining directories, please do *not* end the definition by a "/" ("\") !

   M4C_HOME : The location of the compiler up to the top level directory compiler/ created when unpacking the file M4CompilerServer.zip.

   JDBC_ZIP: The complete path and file name of the Oracle JDBC classes zip file which should be used. This file is part of your Oracle installation.

   VERBOSITY: A number between 0 and 20 which gives the **default** verbosity for logging. This verbosity may be overridden by the HCI. 0 means most verbose, 20 is least verbose.

   RMIREGISTRY:

   - On Windows: The complete path to the rmiregistry command of your JDK.
   - On Unix/Linux: The command to start an rmiregistry. This variable is not only used to run this service, but also to find a running instance by grep, looking at the process table. So please note that specifying the *complete* path to the binary might not work. The processes are listed by using the following definition. Please note that if you should have to change any of these two variable definitions, then you should have a closer look at the file ${M4C_HOME}/start.sh as well!

   PROCESSES: Only for Unix/Linux users: The command line to list all processes of the current user.

   NOHUP: Only for Unix/Linux users: The location of the nohup command. It might be necessary to protect the server process from being terminated together with its creating shell.

   Usually you should not have to change the default settings of the following variables, because they are defined relatively to the variable M4C_HOME.

ML_HOME: The complete path to (and including) compiler/runtime. This is needed to find the algorithms for the external operators.

DB_CONFIG: The complete path and file name of the file described above under 1.

COMPILER_JAR: The complete path to the file M4Compiler.jar.

TEMP_DIR: A directory in which temporary files can be written for handling the server status.

LOGFILE: The complete path and file name of a log file for standard output messages of the compiler server. For each compilation another logfile is created, appending the ID of the corresponding case to the filename. On Unix/Linux the log files can be viewed using the script compiler/showlog.sh. Without a parameter it shows the standard log file, e.g. for case unrelated messages. The log messages for a specific case are shown, if you specify the case id as a parameter.

JAVA_POLICY: The complete path and file name of a file that contains the Java security policy for the server process. All possible rights are granted to this process. An example file is included as compiler/classes/java.policy.

PID_FILE: On Unix and Linux machines, only: the complete path and file name of a file that will contain the process ID (PID) of the server process. This is used for handling the server status.

Once all the information about configurations is entered, you can start the compiler. Please note, that you should not run more than one compiler *server* for a single M4 schema, because this leads to deadlocks. Usually this should not be necessary, because you can compile different cases stored in the same M4 schema using a single compiler server. If you cannot ship around accessing data in another than a single business data schema, however, then you will also have to set up another separate installation of the M4 schema, of the JBOSS, the HCI, and of the M4 compiler.

On Unix/Linux the compiler is started by typing ./start.sh in the directory compiler/classes/. Now the server is running as a background process and can answer requests from the HCI client (section 4). Compiler output will be found in the specified log files. To shut down the server, type ./stop.sh in the same directory (on the same computer). Note that when the server is started, a rmiregistry process is also started; if you wish to stop it after shutting down the server, you need to do so by hand.

On Windows first of all you need to run the batch file rmiregistry.bat in the directory compiler\. If you see the message RMIregistry is running. you can run the batch file start.bat in the same directory. After a short delay you should see two messages of successfully set up JDBC driver connections in the corresponding window. The server is running and waiting for connections, unless you close the window. Please do not close the rmiregistry window in the meantime.

The operators using a Support Vector Machine can use an implementation of the SVM inside the database. To install this software (called mySVM/db), go to the website http://www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVMDB/ and follow the instructions.

Thanks to Bart Goethals (see http://www.cs.helsinki.fi/u/goethals/) for making available his Apriori implementation!

4 HCI

The human-computer interface (HCI) comes in the file hci.zip. The file should be unpacked in a new directory. Two additional files GraphView.jar and hotdraw.jar, which are graphics packages distributed under the GNU license; please download it separately and place it into the subdirectory lib/ of your HCI directory (this subdirectory will be created when unpacking the first file, hci.zip). been tested on Windows, Linux and Unix. There are different start scripts for the systems—file start_hci.bat for Windows, and start_hci.sh for Linux and Unix. Before the HCI can be used, the start script files must be edited. Three lines have to be adjusted to your own environment:

1. Set the variable mypath to the directory where you unpacked the HCI.
   Example for Windows:
   set mypath=C:\HCI
   Example for Linux/Unix:
   MYPATH=/home/myusername/hci

2. Set the variable compilerServer to the machine where the compiler server is running (see section 3).
   Example for Windows: set compilerServer=mycomp.cs.uni-do.de
   Example for Linux/Unix: COMPILERSERVER=mycomp.cs.uni-do.de

3. Set the JBoss server name and port (see section 5.1) for the JNDI3 variable.
   Example for Windows: set jndi3=java.naming.provider.url=jnp://
   yourcomp.cs.uni-do.de:1099
   Example for Linux/Unix: JNDI3="java.naming.provider.url=jnp://
   yourcomp.cs.uni-do.de:1099"

Further, a file db.config must be set up as described in section 3 under 1.; for this, the files db.config.example and db.config.template are provided in the HCI directory. The two files (for the HCI and for the compiler) must be kept consistent.

Finally, parts of the HCI will need your Oracle JDBC drivers. Please copy the file classes12.zip from your Oracle libraries (for example, oracle/jdbc/lib/ might be the name of the directory on a Linux or Unix system) to the subdirectory lib of your HCI directory.

5 Database interface

The interface to the database, which the HCI uses, runs on a JBoss application server. The JBoss software must be installed first.

5.1 Installing JBoss

Download JBoss 3.0.0 from http://www.jboss.org/downloads.jsp. Both versions, Jetty or Tomcat web engine, should do. Unzip the software to the desired location; it will be placed in a directory named jboss-3.0.0.

Next, the JBoss software must be configured. The jboss server/ directory contains different types of JBoss server installations: *all*, *default* and *minimal*. The *all* configuration contains all JBoss features whereas the *minimal* version only contains the minimally needed set of features.

It makes sense to make a separate server configuration for MiningMart. Simply copy the contents of the directory jboss-3.0.0/server/default/ to a new directory jboss-3.0.0/server/mm/.

The connection to the Oracle database that contains the metamodel (M4) needs to be configured in two steps:

- Configure the file oracle-service.xml (see below) and place it in the directory jboss-3.0.0/server/mm/deploy.
- Copy the Oracly JDBC library file classes12.zip, which is mentioned in section 4, to the directory jboss-3.0.0/server/mm/lib. This file is part of your Oracle libraries; for example, oracle/jdbc/lib/ might be the name of its directory on a Linux or Unix system. Another copy of it is needed by the HCI (section 4).

An example of the file oracle-service.xml can be found in jboss-3.0.0/ docs/examples/jca. The only section in the file that has to be changed is the following:

```
<depends optional-attribute-name="ManagedConnectionFactoryName">
   <!--embedded mbean-->
   <mbean code="org.jboss.resource.connectionmanager.RARDeployment"
   name="jboss.jca:service=LocalTxDS,name=OracleDS">

     <attribute name="JndiName">MiningMartDB</attribute>

     <attribute name="ManagedConnectionFactoryProperties">
       <properties>
         <config-property name="ConnectionURL" type="java.lang.String">
            jdbc:oracle:thin:@servername:1521:SID
         </config-property>
         <config-property name="DriverClass" type="java.lang.String">
               oracle.jdbc.driver.OracleDriver
```

```
      </config-property>

<!--set these only if you want only default logins,
    not through JAAS -->

      <config-property name="UserName" type="java.lang.String">
            user
      </config-property>
      <config-property name="Password" type="java.lang.String">
            passwd
      </config-property>
   </properties>
 </attribute>
```

The JndiName must be MiningMartDB and further the ConnectionURL (servername:1521:SID above), UserName (this is the schema name) and Password should be specified. Note that the user name and password of the metadata schema must be used, rather than the business data schema.

You may want to run the compiler server and JBoss on the same machine. In this case, you must make sure that they use different ports. The compiler server uses the RMI port 1099 which cannot be changed. The port number that JBoss uses can be changed in the file .../conf/jboss-service.xml.

The server can be started and stopped using the scripts in the directory jboss-3.0.0/bin. Using ./run.sh or run.bat will start the default server configuration.

To run the newly created mm server configuration use the command ./run.sh -c mm (Unix) or run.bat -c mm (Windows).

5.2 Deploying the M4 interface

The HCI client uses the M4 interface to get access to the M4 metadata schema. Part of the M4 interface is stored on the client (it is part of the HCI client software) and another part resides on the JBOSS server. Currently this is the only part of the MiningMart system that uses the JBOSS server. The server part for the M4 interface is contained in the file M4InterfaceServer.jar. This file should be placed in the directory jboss-3.0.0/server/mm/deploy.

6 Appendix: List of operators that use external algorithms

- Apriori
- FeatureSelectionWithSVM
- GeneticFeatureSelection
- StatisticalFeatureSelection
- SGFeatureSelection
- MissingValuesWithRegressionSVM
- MissingValueWithDecisionTree
- MissingValueWithDecisionRules
- PredictionWithDecisionTree
- PredictionWithDecisionRules
- DecisionTreeForRegression
- SupportVectorMachineForRegression
- SegmentationWithKMean