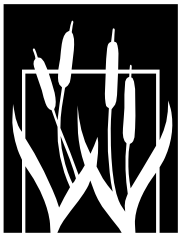

Whitemarsh
Information Systems Corporation

Whitemarsh Metabase
Getting Started Users Guide

April 2015

Whitemarsh Information Systems Corporation
2008 Althea Lane
Bowie, Maryland 20716
Tele: 301-249-1142
Email: Whitemarsh@wiscorp.com
Web: www.wiscorp.com

Table of Contents

List of Figures

1.0 Whitemarsh Metabase Getting Started Guide

This Getting Started Guide is a quick-start reference to users of the Whitemarsh Metabase System. This guide is not a replacement of the Metabase User Guides, however. This guide also provides information on how to use the various SQL DBMS engines to which the Metabase System is interfaced.

The Whitemarsh Metabase system is a 32 bit database client that uses the an SQL Engine as a DBMS back-end, via ODBC. The SQL engine can be either 32 or 64 bit. This is addressed in a later section of this guide. This guide briefly explains:

- Whitemarsh Metabase System Installation (Section 2)
- Using the SQL Engine (Section 3.0).
- Installing the Metabase System production version (Section 4)
- Migrating a Metabase database to a different SQL Engine (Section 5)
- Using the Metabase System for Real Work (Section 6)
- Other SQL Engine backends (Section 7)

2.0 Whitemarsh Metabase Installation

There are two versions of the Metabase system: Free and Production. The installation process for the Free version is contained in Section 3.0. The installation process for the Production system is contained in Section 4.0.

Regardless of the SQL DBMS, the overall architecture is the same and is presented in Figure 1. On the left side of the figure are the metabase databases. There are two classes: mbsysdb (Metabase System Database), and metabase database instances. The mbsysdb contains the information used by the metabase system functional modules to properly control and manage the environment. The metabase databases are where the actual functional metadata is stored.

The SQL DBMS can be any one that has been interfaced via ODBC with the metabase system. There is only one place in the overall system where a specific DBMS-SQL command is employed. Hence, the metabase system is largely independent of the SQL DBMS. Because the databases are accessed through ODBC, any ODBC compliant tool and/or report writer can access the metabase data.

From this diagram it should become apparent that there is no server module. All the capability for ODBC access, user management, security, etc., is all controlled by the functional client modules.

2.1 Installing the Metabase Free System

The Metabase Free version runs indefinitely after it is installed. There are only two restrictions on the Free version. The first is concurrent user count and it is restricted to 1. You can download and install 50 instances of the Metabase System software or download one instance and copy and distribute it 50 times and then do 50 installations. Given that they all point to the same single set of metabase databases, that is OK. Just know that the Metabase System has the ability to restrict the quantity of concurrent users to just one. As far as reporting is concerned, the are no restrictions at all. That is because if you use Crystal Reports or some other ODBC access report writer then you are not using the Metabase System software at all, hence it is not possible for there to be any concurrent users restriction.

Second, only two metadata databases are allowed, and their names are: Movies and Metabase. If you wish to increase the quantity of concurrent users and/or the quantity of metabase databases that can be known by the metabase system please contact Whitemarsh.

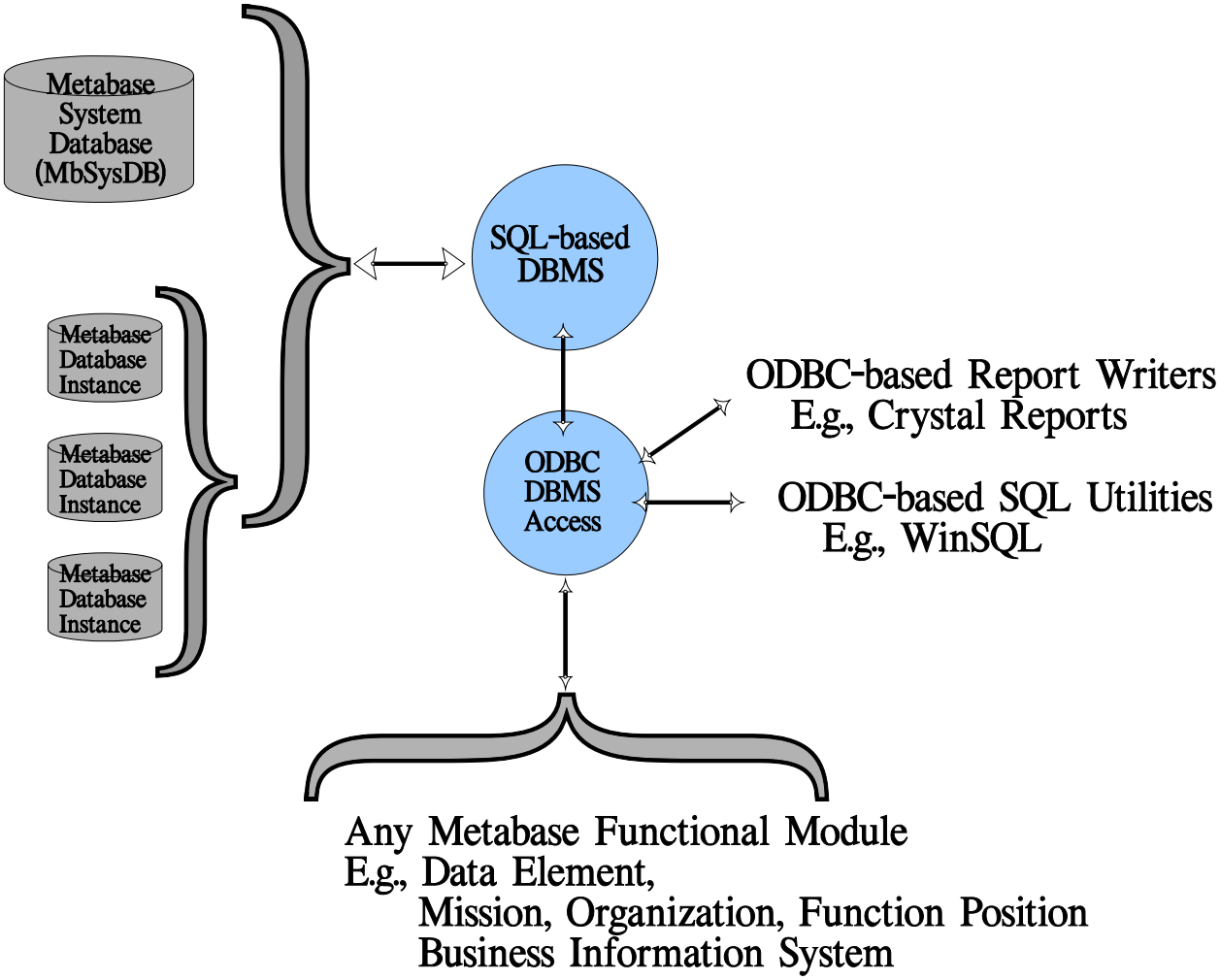

Figure 1. Metabase System Architecture.

If you are installing the Whitemarsh Metabase Free system the steps are:

- Install Whitemarsh Metabase System Software
- Manually Create ODBC Connections to the SQL DBMS

2.1.1 Install Whitemarsh Metabase System Software

Run the downloaded Metabase System install file and execute it to begin the Whitemarsh Metabase System software installation process. The installation is quite typical, during which the user will be prompted to supply a directory into which the installation process will install the Metabase system files. The main directory, Wiscorp will then contain:

1. Program files and DLL files
2. A set of TXT files that are the default files for various textual outputs
3. A "TPS" file that contains default ODBC SQL Connect Strings for Mimer and MS/SQL
4. The following subdirectories:
   a. The "Aspell" directory that stores the spelling correction files
   b. Crystal Reports, and 10 contained subdirectories for "rpt" files for each metabase module.
   c. DOCS, a directory that includes:
      (1) Abbreviation files for banking, finance, and health care
      (2) Getting Started Guide
      (3) Metabase Administrators Guide
      (4) Metabase System User Guides
      (5) Data Modeler Architecture and Concept of Operations book
      (6) Reverse and Forward Engineering Book
      (7) SQL DDL for the mbsysdb and metabase databases for each interfaced SQL DBMS
   d. Metabase Databases
      i. For the Mimer DBMS
         (1) mbsysdb, a directory for the Metabase's System Administration database
         (2) Metabase, a directory for a default-only and extended reference data populated metabase
         (3) Movies, a directory for the "demo" metabase
      ii. For the Microsoft MS/ SQL DBMS
         (1) mbsysdb, a directory for the Metabase's System Administration database
         (2) Metabase, a directory for a default-only and extended reference data populated metabase
         (3) Movies, a directory for the "demo" metabase
      iii. Readme file
   e. Reference Data
      i. A referenced data file that contains extended reference data for Mimer
      ii. A referenced data file that contains extended reference data for MS SQL
      iii. Data Types data

The Whitemarsh Metabase install process automatically installs metabase databases for Mimer and for MS/SQL. The databases for the Mimer DBMS are mbsysdb, movies, and metabase. The databases for Microsoft's MS SQL are msmbsysdb, msmovies, and msmetabase

The install process preloads the mbsysdb or msmbsysdb database with one users:

- FN: Metabase; LN: Administrator with a user name of sysadm and a password of sysadm
- FN: MyFirstName; LN: MyLastName with a user name of MyName and a password of MyPassword

You SHOULD, of course, change the user names and passwords as you wish via the Metabase system's Administrator module. Metabase administrator based user names and passwords have no effect on the actual accessing of database records.

The **mbsysdb** is the metabase system's administration database. This database enables the metabase administrator to create new users and to allocate users to metabase modules in support of access security. This database also contains "connect strings" for ODBC access to SQL DBMSs, that, in turn, access the metadata databases (e.g., movies).

The Metabase's Administrators User Guide for MbAdmin module describes the mbsysdb and Administrator functions necessary to keep these connect strings accurate.

The **movies** metabase database is a complete sample metabase. This metabase was modeled after the largest Movie Rental corporation in the United States, which, at the time it was created was still in business. Most of the "metadata" for this organization was obtained directly from their website.

In contrast to the movies database, the ***metabase*** metabase only contains:

- Default data
- Data type data
- Commonly used reference data

The purpose of the ***metabase*** database is to provide metabase system users a database into which they can put their own metadata. The default data that is contained in the ***metabase*** database is only one or two rows of data for most tables. These rows represent "unknown" values.

The second pre-populated set of reference ***metabase*** database reference data are data types (e.g., Integer, Character, etc.). This exists in three tables:

- Data type pictures
- DBMS data type
- SQL data type
- Value domain data type

These tables are related to each other with Value Domain data types as the parent of SQL Data Types, which, in turn are the parents of the DBMS data types.

The third type of pre-populated reference data within the ***metadata*** database is commonly used reference data mainly for the following functional modules:

- Administrative module
- Business Information Systems module
- Database Objects
- Data Elements
- Information Needs Analysis
- Operational Data Model
- View Data Model

The tables within the Administrative module that employ commonly used reference data are:

- Abbreviations

The Admin module has an ASCII import process for importing files that contain abbreviations. The install files that contain abbreviations contained in the Docs directory are:

- AbbreviationsBankingFinance.txt
- AbbreviationsGeneral.txt
- AbbreviationsHealthCare.txt

The tables within the Business Information System module that employ commonly used reference data are:

- Application type
- Construction method
- DBMS Environment
- Environment Type
- Level
- Production User Class
- Programming Language
- Status

The tables within the Database Objects module that employ commonly used reference data are:

- Rationale

The tables within the Data Element module that employ commonly used reference data are:

- Meta Category Values
- Meta Category Value Types
- Meta Category Value Class Type

The tables within the Information Needs Analysis module that employ commonly used reference data are:

- Characteristic Type
- Characteristic
- [mb]Rank
- Management

The tables within the Operational Data Model module that employ commonly used reference data are:

- Data Architecture Class
- Database Management Systems (DBMS)
- Nature

The tables within the View Data Model module that employ commonly used reference data are:

- Business Information System View Role

The first three kinds of reference data is loaded into empty metabase databases through the Administrator Module (mbAdmin). The fourth type of reference data, extended reference data, is loaded through the use of an ODBC SQL utility such as WinSQL. This fourth kind of reference data provides a large collection of commonly employed reference data values that have been engineered and agreed-to by a large number of users over the past 10 or so years. Because of the wide acceptability of this reference data, it is provided as an assist to "getting started."

In the case of the fourth type of reference data, the foreign keys must NOT be active during the load. Hence, the best approach is to first create a metabase database through the application of the Create Tables script file (file #1), then the indexes and keys script file (file #3). At that point, execute the Insert Values scripts for the extended reference data. Once that is complete, run the Foreign Keys script (file #4).

All of these reference data loading processes are detailed in the Administrator user guide. Once these reference data are loaded, users are able to update these as may deem appropriate.

3.0 Using the SQL Engine

As of the version of the Metabase System, it operates under two different SQL DBMS: Mimer and MS/SQL.

3.1 The Mimer SQL Engine

Employing Mimer with the Metabase System involves:

- Downloading the Mimer DBMS
- Creating Mimer ODBC Connections

3.1.1 Downloading the Mimer DBMS

The Mimer SQL engine is obtained from the Mimer website, www.mimer.com. The only requirement for downloading a "developers version" is filling out a registration screen.

Run the Mimer SQL Engine install. The installation is quite typical, during which the user will be prompted to supply a directory into which the installation process will install the Mimer SQL Engine system files.

Mimer Version 10 or later is required for this release of the Metabase System. Whitemarsh is using the 64 bit version.

3.1.2 Creating Mimer ODBC Connections

The Whitemarsh Metabase Software System installation process automatically installs three Mimer datasets: mbsysdb, movies, and metabase. These databases are installed in the subdirectory, Mimer within the directory Metabase Databases.

Each of these has to be connected to Mimer's ODBC through the Mimer Admin software. These Mimer Metabase databases are contained in the appropriate subdirectory within the Metabase Database directory. The 64 bit ODBC driver is the version Whitemarsh uses.

Locate the software, mimadmin.exe, within the Mimer DBMS directory and execute it. When you do, Figure 2 is presented. Perform the following steps:

- Under the Local Tab, press the Add button. A dialogue as shown in Figure 2 is presented.

- Fill this screen with data as presented in Figure 3. You, of course, need to find the home directory for the mbsysdb. The metabase system automatically places it in the C:\Program

Files\Wiscorp\Metabase directory tree. Leave the "Create System Data Source" box checked.. Then press apply. When this is accomplished, Figure is presented.

- Finally, "right mouse click" and select, "Start Server" from the list of choices presented.

Repeat this process for Movies, and for Metabase. Note, the correct "CaSe" for mbsysdb is ***mbsysdb***. For movies is has to be ***Movies***, and Metabase has to be ***metabase***.

At this point, with all three databases connected, and all three Mimer Database Servers started you can then start using the Metabase software. During the install process a program group, Whitemarsh Metabase Demo was created. All the functional modules are listed within this program group.

Figure 2. ODBC Mimer Connection Screen.

Once all the Mimer databases are defined to the server, Figure 4 shows the result. In this screen all three database are defined and all three servers are running.

The ODBC connection presentation in Figure 4 presumes that the computer has both the DBMS Mimer and also the Mimer databases. If the DBMS Mimer is on a server and the databases are on that same server then a different set of Mimer processes need occur. Figure 5 presents the Mimer ODBC database definitions screen, but from the client computer's vantage point.

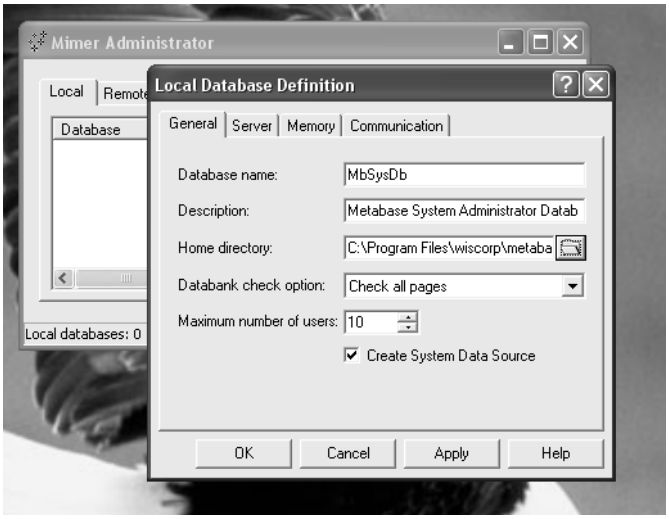

Figure 3. Completed Mimer ODBC dataset update screen.

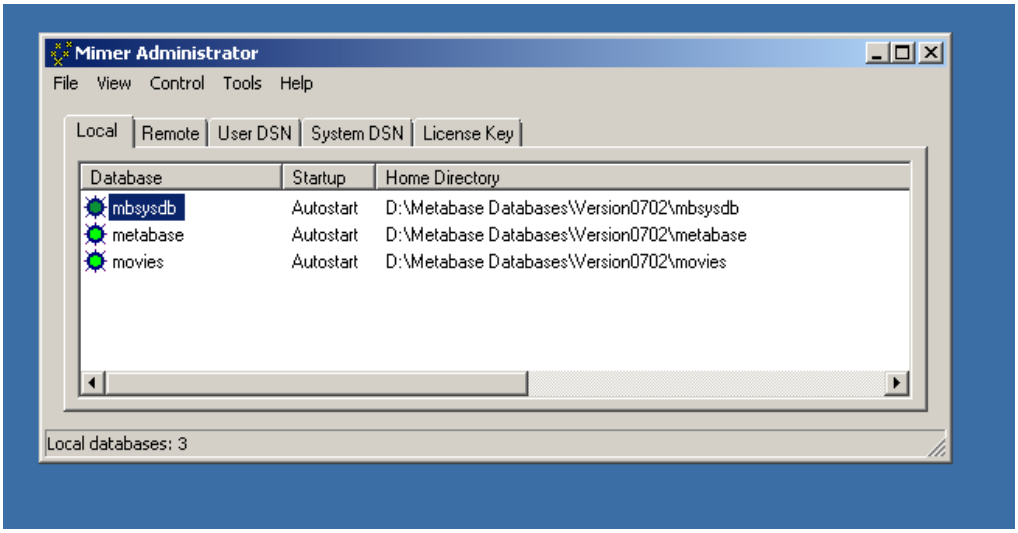

Figure 4. Mimer Administrator on Server ODBC screen.

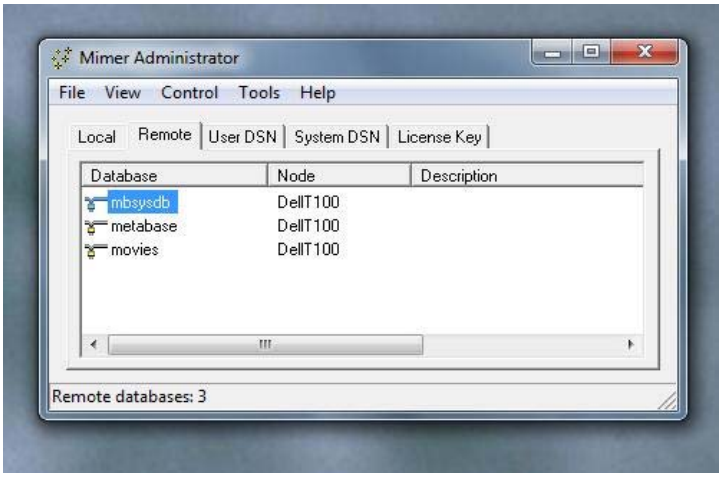

Figure 5. Mimer Administrator Client ODBC screen.

Before this screen can successfully process, the client component of Mimer has to be installed on the client machine. The option for Mimer client-only installation is presented when Mimer is being installed. When this client-only option is chosen, the ODBC software that enables an interconnection from the client machine through ODBC to the server machine and the subsequent use of the Mimer DBMS to retrieve and update data. Once this Mimer client software is installed, the Mimer Administrator software, when executed on the client machine produces a figure like what is produced in Figure 5. The Add button enables the naming of the database, e.g., mbsysdb, and the naming of the server, e.g., DellT100 on which the database resides.

3.2 The Microsoft MS/ SQL Engine

Employing MS SQL with the Metabase System involves:

1. Downloading the MS SQL DBMS
2. Creating MS SQL ODBC Connections

3.2.1 Downloading the Microsoft MS/ SQL Engine

This quick start guide does not provide any instruction for the procurement, installation or configuration of the MS/SQL engine. These activities should be easily accomplished by your database administration group.

3.2.2 Create Microsoft MS/ SQL ODBC Connections

The Whitemarsh Metabase Software System installation process automatically installs three MS/SQL databases: msmbsysdb, msmovies, and msmetabase. These databases are installed in the subdirectory, MsSQL within the directory Metabase Databases.

Each of these has to be connected to MS SQL's ODBC through the operating system's ODBC connection software. There is nothing unique about this process and your MS/SQL administrator should be able to accomplish this for you in just a few hours. When this is completed, a figure similar to the one shown in Figure 6 should be visible. Note these databases are shown through MS SQL 2012 R2.

The MS SQL databases have all been saved in the MS SQL 2000 format and your database administrator should know how to "Attach" these databases. These databases were designated as being available for access through the MS SQL administrator. They are: msmbsysdb, msmetabase, and msmovies. Again, your database administrator should be able to help in this activity. From the client point of view, these same databases have to be made available.

Figure 7 presents, from within Windows 7, the ODBC database definitions screen, but from the client computer's vantage point. This screen is presented as a consequence of executing the ODBC action from within the Control Panel. Again, your computer administrator should be able to set this up in just a few minutes. From Figure 7 above, SQL Server is chosen to be the DBMS for the three databases: msmbsysdb, msmovies, and msmetabase. When the Configuration button is pressed, a screen like that on Figure 8 is shown. It enables the naming of the database, e.g.,mbsysdb, and the naming of the server, e.g., DellT100 on which the database resides.

4.0 Installing the Metabase Production System

This section of the Getting Started Guide is applicable only if you do NOT have the Free version. If you have the **Free** system, the various metabase system functional modules will ONLY recognize three Mimer databases: mbsysdb, movies, and metabase. Or, for MS SQL, the various metabase functional modules will ONLY recognize three MS SQL databases: msmbsysdb, msmovies, and msmetabase. If you have a paid license to the Metabase System then the material in this section of the Getting Started Guide is of value.

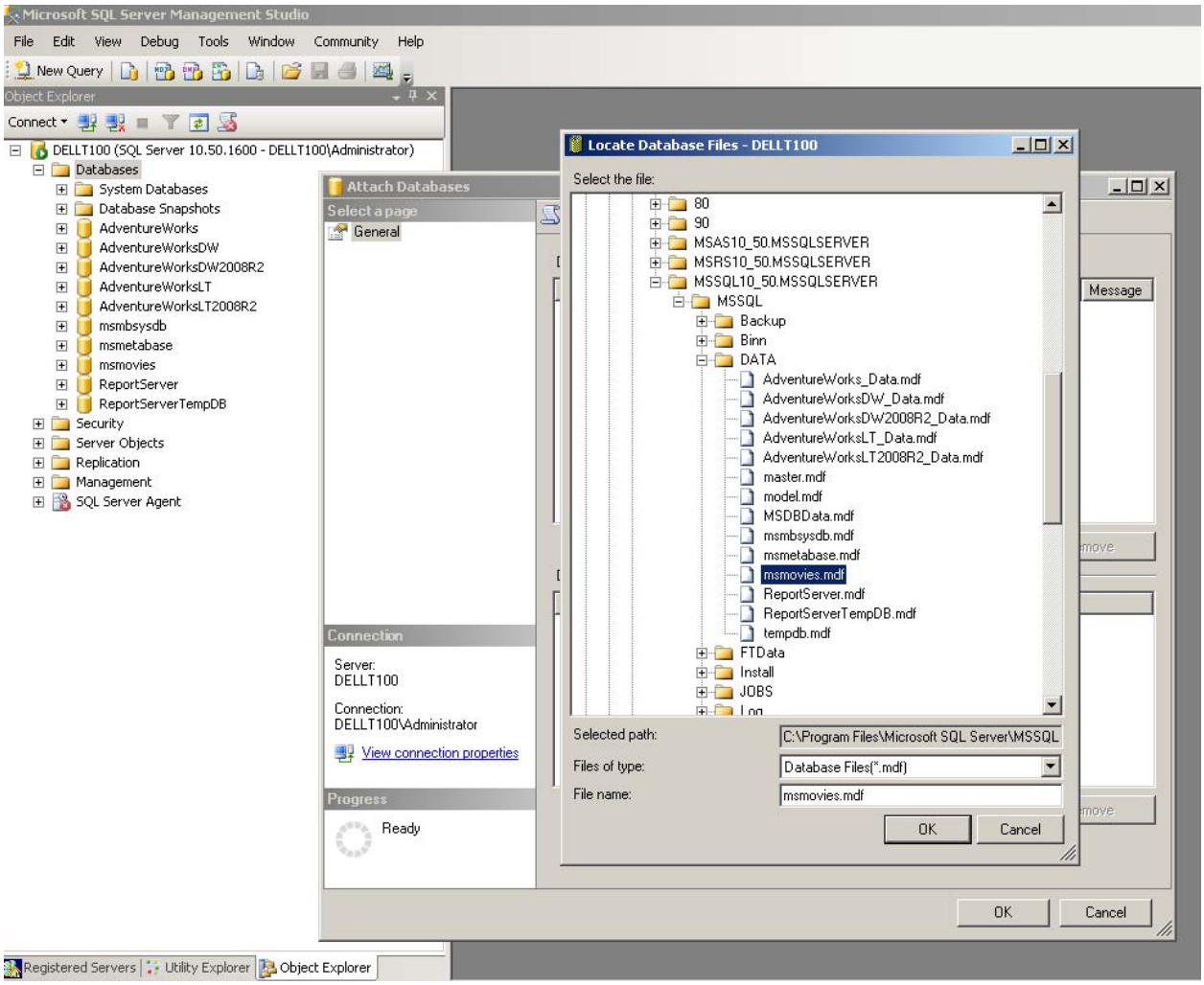

Figure 6. Attaching a Metabase Database, msmovies to MS SQL.

The material includes the following:

- Install Whitemarsh Metabase System Software (Section 4.1)
- Install the SQL Engine (Section 4.2)
- Understanding the two classes of Metabase System Databases (Section 4.3)
- Creating Mimer Metabase System Databases (Section 4.4)
- Creating MS SQL Metabase System Databases (Section 4.5)
- Making Metabase System Databases ready for use (Section 4.6)

Figure 7. MS SQL ODBC connection screen to Metabase databases.

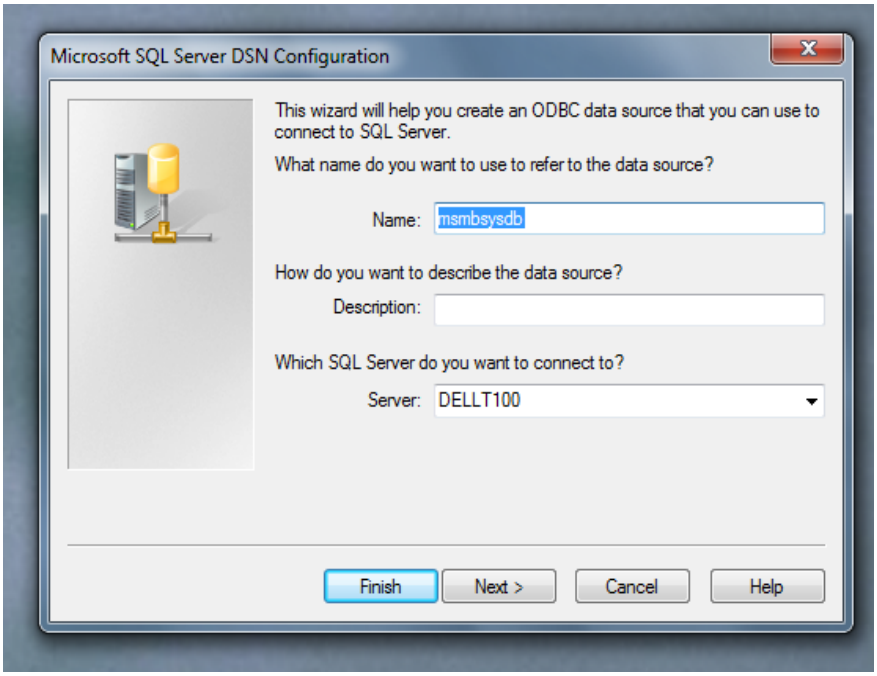

Figure 8. Configuring the ODBC connection to access the MSSQL database, mbsysdb.

4.1 Install Whitemarsh Metabase System Software

Run the downloaded executable to begin the Whitemarsh Metabase System software installation process. The installation is quite typical, during which the user will be prompted to supply a directory into which the installation process will install the Metabase system files. Depending which version of the Metabase you are installing, the install process may also prompt the user to select which administration components will be included in the installation.

4.2 Install the SQL Engine

The Mimer SQL engine is obtained from the Mimer website, www.mimer.com. The only requirement for downloading a "developers version" is filling out a registration screen.

Run the Mimer SQL Engine install downloaded from www.Mimer.com. The installation is quite typical, during which the user will be prompted to supply a directory into which the installation process will install the Mimer SQL Engine system files.

The MS/SQL engine from Microsoft is obtained from Microsoft through procedures established by your organization. Your system administrator and your database administrator should be familiar with those procedures.

4.3 Metabase Databases

After the SQL DBMS is installed, two types of databases need to be created. The first type is the Metabase System's Database (mbsysdb). There will only one instance of this database type for each different SQL engine. If you only use MS/SQL then there will only be one instance of mbsysdb. If you use both Mimer and MS SQL, there will be two instances of mbsysdb, one for Mimer and the other for MS SQL. Each mbsysdb contains metabase system defaults, enables the active management of all metabase functional module users, and contains the metabase administrator's constructed connect strings that enable ODBC access.

The second type of database is functionally named and contains all the metadata that is being captured for that function. For example, if there is to be only one metabase for the entire enterprise, that metabase might be called Enterprise. But if there is to be one for the Eastern Division, it might be called Eastern. For the Free metabase version that is described in Sections 2.1 and 3.1, the functional metabase that contains all the metadata for the Movies Rental Corporation is called movies. Each of these functional metabases contains the metadata in support of the all the analyzed areas of the enterprise. These are described in the Metabase Overview and Common Functions Guide.

In short, this second type of database is a database that holds the metadata created as a result of using the Metabase System functional modules. While every SQL database engine contains

different physical files that comprise database instances. Mimer, for example, contains five distinct operating system files. Each collection of five O/S files is called a Mimer Data Set.

Every metabase system installation requires one database, mbsysdb, for each different SQL DBMS, and any number of other metabase databases for each set of metadata that is to be captured and separately maintained within the enterprise.

ODBC, Microsoft's Open Database Connectivity, enables metabase database instances to be accessed by ODBC utilities like WinSQL, or to be reported from through systems like Crystal reports. The overall metabase system architecture is illustrated in Figure 4.

After the DBMS installation is complete, the next step is creating two different types of DBMS-specific metabase databases. The first, of course, is the mbsysdb database. The second , are the metadata databases, for example, Movies, Eastern, Marketing, etc that stores the organization's metadata under the control of a specific SQL DBMS engine. This step causes the creation of the SQL database which, in the case of Mimer, are the five operating system files that comprise each SQL database instance. In the case of MS SQL, there is only one database file and also a transaction file.

4.4 Creating Mimer Metabase System Databases

The steps required to create a Mimer metabase database consists of two major steps:

- Create Mimer Databases
  - ♦ Create a mbsysdb Mimer database Instance (Section 4.4.1)
  - ♦ Create one or more additional metabase database Instances (Section 4.4.2)
- Make mbsysdb Ready for use
  - ♦ Run the SQL DDL files for the mbsysdb database. These files are located in the DOC subdirectory.
- Make Each Metabase Database Instance Ready for Use
  - ♦ Through an ODBC SQL utility, Run SQL DDL Table Building Step (DDL file #1 from within the DOC directory)
  - ♦ Through an ODBC SQL utility, Run SQL DDL Index Building Step (DDL file #3 from within the DOC directory)
  - ♦ From within the Metabase's Admin module, Run Default Data Step
  - ♦ From within the Metabase's Admin module, Run Default Data Types Step
  - ♦ Through an ODBC SQL utility, Import the extended reference data via its Insert Values format
  - ♦ From within the Metabase's Admin module, generate the maximum identify values scripts.
  - ♦ Through an ODBC SQL utility run the maximum identify value scripts.
  - ♦ Run SQL DDL Foreign Keys Building Step (DDL file #4 from within the DOC directory)

Running the Mimer Administrator- located in the installed to directory (e.g. C:\Program Files\Mimer\mimadmin.exe) brings up a window with multiple tabs. The Metabase software *requires* at least two databases on the Mimer DBMS backend.

> ***It is best to NOT specifically identify the paths for any of the five files. This way these file can be located on any disk drive and under any directory. If paths are identified then these are "baked in" the files. While not difficult, it is tedious to remove these baked-in paths.***

4.4.1 Create a Mimer mbsysdb Database

There are two steps:

- Define mbsysdb to Mimer (Section 4.4.1.1)
- Use an ODBC utility to define the mbsysdb tables and indexes (Section 4.4.1.2)

4.4.1.1 Define mbsysdb to Mimer

To create the mbsysdb, first switch to the Local tab on the Mimer Administrator (mimadmin. exe), and press the Add button. A window will pop up into which specific information MUST be provided. In the entry field labeled Database Name enter: "mbsysdb" (without the quotes). In the Home Directory entry field, browse to the folder on your server that will contain the Mimer database files for the Metabase System Database. Leave the Create System Data Source checkbox checked. This enables the automatic creation of an ODBC entry in the Window's ODBC Manager, associating this DSN with Mimer's ODBC drivers. Pressing OK will return a message window that states: "No system databanks found in C:\Your\Selected\Folder\ Do you want to create them?" Press "Yes." The Mimer wizard will automatically create a databank, with the necessary data files for the metabase database. *Remember, it is best not to have explicit paths in the file names dialogs.*

This first step of creating new databanks includes setting the locations of the various system datafiles for each particular Mimer dataset. For performance reasons, Mimer distributes the various system database files to different hard drives. If you are merely evaluating the Metabase software, it is acceptable to change these drive values, consolidating the files to one folder. Press next after finishing any changes. *And remember, once again, it is best not to have explicit paths in the file names dialogs.*

The next step is to add the password for the mbsysdb database. For example, it might be "foodo2." ***Note Well: There is NO WAY to discover a Mimer database password. There is no "cracking tool." So, choose wisely and don't forget it.*** When finished, press the next button.

The wizard will then create the various datafiles for the mbsysdb, in the locations that you specified.

When the creation of the datafiles is completed, the wizard will prompt the user whether or not to “Start Database Server,”and whether or not to “Start Wizard for Development User and Databank and Example. ”Leave enabled the “Start Server” check box, but disable the second checkbox. The wizard for creating example data, and a developmental user is not necessary for the Metabase Software. Press Finish to complete the wizard.

Upon completion of the wizard that creates the Metabase System Database (mbsysdb), the Mimer Database Controller will appear, showing that the database’s databanks have been successfully created and is now currently running with logins enabled. Close this window when finished confirming this information. The first of the two requisite databanks are now created. The second required databank, the Metabase dataset itself, is next.

4.4.1.2 Use ODBC Utility to Define mbsysdb Tables and Indexes

This step makes mbsysdb ready for use. This involves running a SQL DDL script. Any ODBC client will accomplish this. There are many SQL ODBC clients available, including QBE Vision’s SQL Query Builder (qbebuilder. exe) that was installed with the Mimer SQL Engine server system files. Another tool named wSQL is available from the Mimer. com website. Both are free. You can also use a very popular ODBC client, WinSQL. This is very feature rich.

To accomplish running the SQL DDL script step, use the ODBC SQL client to login in to the mbSysdb using the SYSADM username, and the supplied password: “foodo2”. Then, run the commands in the file, MB72_MbSysDB_Mimer_SQLDDL, that is located in the installed to directory’s (e. g. C:\Program Files\Wiscorp\Metabase\DOC\) folder. After running the commands, disconnect the wSQL client from the Mimer mbsysdb database. At this point, the mbsysdb is ready for use.

4.4.2 Create Mimer Metadata Databases

A Metabase Database instance houses the various tables of metadata that are created through the Whitemarsh Metabase functional modules such as Data Elements or Business Information Systems.. As with the mbsysdb database there are two steps:

- Define a metadata database to Mimer (Section 4.4.2.1)
- Use an ODBC utility to define the metabase tables, and indexes foreign keys (Section 4.4.2.2)
- Use the ODBC utility to create default and extended reference data (Section 4.4.2.3)

- Use ODBC utility to define foreign keys (Section 4.4.2.4)

4.4.2.1 Define a Metadata Database to Mimer

Running the Mimer Administrator brings up a window with multiple tabs. To create a Metabase dataset instance, first switch to the Local tab, and press the Add button. A window pops up into which information must be provided. In the entry field labeled Database Name enter for example: “metabase” (without the quotes). If you are creating a metabase for Marketing metadata or for metadata for the Eastern division, you might make the metabase’s name, marketing or eastern. For this example, the metabase’s name is Marketing.

In the Home Directory entry field, browse to the folder on your server that is to contain the Mimer database files for this Marketing dataset instance. Leaving the Create System Data Source checkbox enabled automatically creates an ODBC entry in the Window’s ODBC Manager, associating this DSN with Mimer’s ODBC drivers. Pressing OK will return a message window that states: “No system databanks found in C:\Your\Selected\Folder\ Do you want to create them?” Pressing *Yes* begins the wizard to create the databank’s datafiles for this new Marketing metadata dataset. If you are merely evaluating the Metabase software, it is acceptable to change these drive values, consolidating the files to one folder. Press next after finishing any changes. *And remember, once again, it is best not to have explicit paths in the file names dialogs.* Again, it is best not to have explicit paths in the file names dialogs.

The first step of creating new databanks includes setting the locations of the various system datafiles for each particular mimer dataset. For performance reasons, Mimer defaults to distributing the databank’s various database files to multiple drives. If you are merely evaluating the Metabase software, it is acceptable to change these drive values, consolidating the files to one folder. Press next after finishing any changes. *And remember, once again, it is best not to have explicit paths in the file names dialogs.* Again, it is best not to have explicit paths in the file names dialogs.

The next step is to add the password for the Metabase Database. For example, it might be “foodo2.” ***Note Well: There is NO WAY to discover a Mimer database password. There is no “cracking tool.” So, choose wisely and don’t forget it.*** Enter a password value and confirm. When finished, press the Next button. The wizard then creates the various datafiles for the Metabase Database dataset, in the locations that you specified.

When the creation of the databank’s datafiles is completed, the wizard prompts the user whether or not to “Start Database Server,”and whether or not to “Start Wizard for Development User and Databank and Example. ”Leave enabled the “Start Server” check box, but disable the second checkbox. The wizard for creating example data, and developmental user is not necessary for the Metabase Software. Press Finish to complete the wizard.

The last step prior to running the SQL DDL scripts is to create various databanks within the Mimer databases. This is done by running the SQL statement, “Create Databank <mimer database name>;” In this case, the database names are mbsysdb, movies, or metabase.

4.4.2.2 Use ODBC Utility to Define Metabase Tables, and Indexes

The first step in getting a Metabase Database instance ready for use it to run a SQL DDL script. This involves three different SQL DDL files, which are:

- MB01_MimerSQL. SQL: Create table statements. (File #1, from within the DOC subdirectory)

- MB03_MimerSQL. SQL: Alter table statements (primary keys, sequences, and indexes) (File #3, from within the DOC subdirectory)

These files are included in a directory within

- C:\Program Files\Wiscorp\Metabase\DOC.

Use the ODBC SQL client that was used to build the mbsysdb. After running the three scripts, disconnect the wSQL client from the Mimer Metabase database. These Run SQL steps have to be accomplished for every Metabase Database Instance. Upon completion, a Metabase database fully exists, although completely empty of all data.

This step makes mbsysdb ready for use. This involves running a SQL DDL script. Any ODBC client will accomplish this. There are many SQL ODBC clients available, including QBE Vision's SQL Query Builder (qbebuilder. exe) that was installed with the Mimer SQL Engine server system files. Another tool named wSQL is available from the Mimer.com website. Both are free. You can also use the ODBC toolset, WinSQL, from Synametrics. This is very feature rich.

To accomplish running the SQL DDL script step, use the ODBC SQL client to login in to the mbSysdb using the SYSADM username, and the supplied password: "foodo2". Then, run the commands in the file, mbsysdb.ddl that is located in the installed to directory's (e. g. C:\Program Files\Wiscorp\Metabase\DOC\) folder. After running the commands, disconnect the wSQL client from the Mimer mbsysdb database. At this point, the mbsysdb is ready for use.

4.4.2.3 Use ODBC Utility to Install Default and Extended Reference Data

The utility that Whitemarsh uses is WinSQL (www.synametrics.com). The process is fundamentally this:

- Pick and make sure you have access to your source database and DBMS.
- Pick and make sure you have access to your target database and DBMS.
- Map source to target tables for a table-centered transfer.

- Save the created profile.
- Execute the profile.

In the case of Mimer, the necessary syntax for handling auto-incrementing primary key then requires:

- Execute the Admin module.
- Press the DataSet button, enter User Name and Password
- Choose and select the metadata database. For example, Metabase
- Press the GenMaxId button and select the appropriate output file and then generate the Max ID SQL script file.
- Use wSQL to execute the script against the metadata database. E.g., Metabase.

The alternative to this process is to accomplish the data transfer directly through the use of SQL Insert Values Scripts. This process causes the loading of both default and extended reference data. The files for this step are included in the reference data directory:

- C:\Program Files\Wiscorp\Metabase\DOC

The extended reference data is stored in one file for Mimer and another for MS SQL:

- Extended Reference Data Scripts_Mimer.txt
- Extended Reference Data Scripts_MS SQL.txt

Use an ODBC utility to load all these Insert Values statements into the appropriate database tables. The size of this file is about 60K bytes and there are about 900. Records.

If there is a need to completely reload the Movies database from SQL scripts, the two data files, which are also stored in the DOC subdirectory are:

- MimerMoviesBulkLoadScript.sql
- MSSQLMoviesBulkLoadScript.sql

Use an ODBC utility to load all these Insert Values statements into the 275+ Metabase System database tables. The size of this file is about 5 megabytes and there are about 65,000. Records.

Once this is complete, use the process set out above to re-set all the high values for the Identity columns associated with each database table. That file is: mbSeqMax.txt. Once this file is executed through an ODBC utility, the final and next step, setting the metabase's foreign keys

can be accomplished through the application of File 4 from within the DOC directory. There is one such file for Mimer and another for MS SQL.

You can use an ODBC SQL utility such as wSQL or WinSQL to generate these Insert Values scripts and then use the generated script for loading into different metadata databases.

4.4.2.4 Use ODBC Utility to Define Metabase Foreign Keys

- MB04_MimerSQL. SQL : Foreign Key statements (File #4)

4.5 Creating MS SQL Metabase System Databases

The steps required to create a MS SQL metabase database are:

- Create MS SQL Databases (Section 4.5.1)
  - Create a mbsysdb MS SQL database Instance
  - Create one or more metabase database Instances
- Make msmbsysdb Ready for use (Section 4.5.3)
  - Run SQL DDL Table Building Step
- Make Each Metabase Database Instance Ready for Use (Section 4.5.4)
  - Through an ODBC SQL utility, Run SQL DDL Table Building Step (DDL file #1)
  - Through an ODBC SQL utility, Run SQL DDL Index Building Step (DDL file #3)
  - From within the Metabase's Admin module, Run Default Data Step
  - From within the Metabase's Admin module, Run Default Data Types Step
  - Through an ODBC SQL utility, Import the extended reference data via its Insert Values format
  - From within the Metabase's Admin module, generate the maximum identify values scripts.
  - Through an ODBC SQL utility run the maximum identify value scripts.
  - Run SQL DDL Foreign Keys Building Step (DDL file #4)

The process shown in this Getting Started Guide is from MS SQL 2010 R2. The screens are from the MS SQL tool set, SQL Server Management Studio. This installs along with MS SQL 2010 R2. SQL commands can be accomplished either through the SQL Server Management Studio or through WinSQL.

4.5.1 Create a MS SQL mbsysdb Database

There are two steps:

- Define mbsysdb to MS SQL
- Use SQL Server Management Studio to define the mbsysdb tables and indexes

4.5.1.1 Define msmbsysdb to MS SQL

The msmbsysdb database should be created from within SQL Server Management Studio. In Figure 9, the database name is msmbsysdb. The suggested password can be "foodo2."

From other new database creation screens, the growth of the database has to be specified for both the database files and for the transaction files. Because the transaction volume for msmbsysdb will be close to nothing, you can safely set these values to 1 megabyte each.

Once the msmbsysdb is completely defined and the server for it started, the next step,

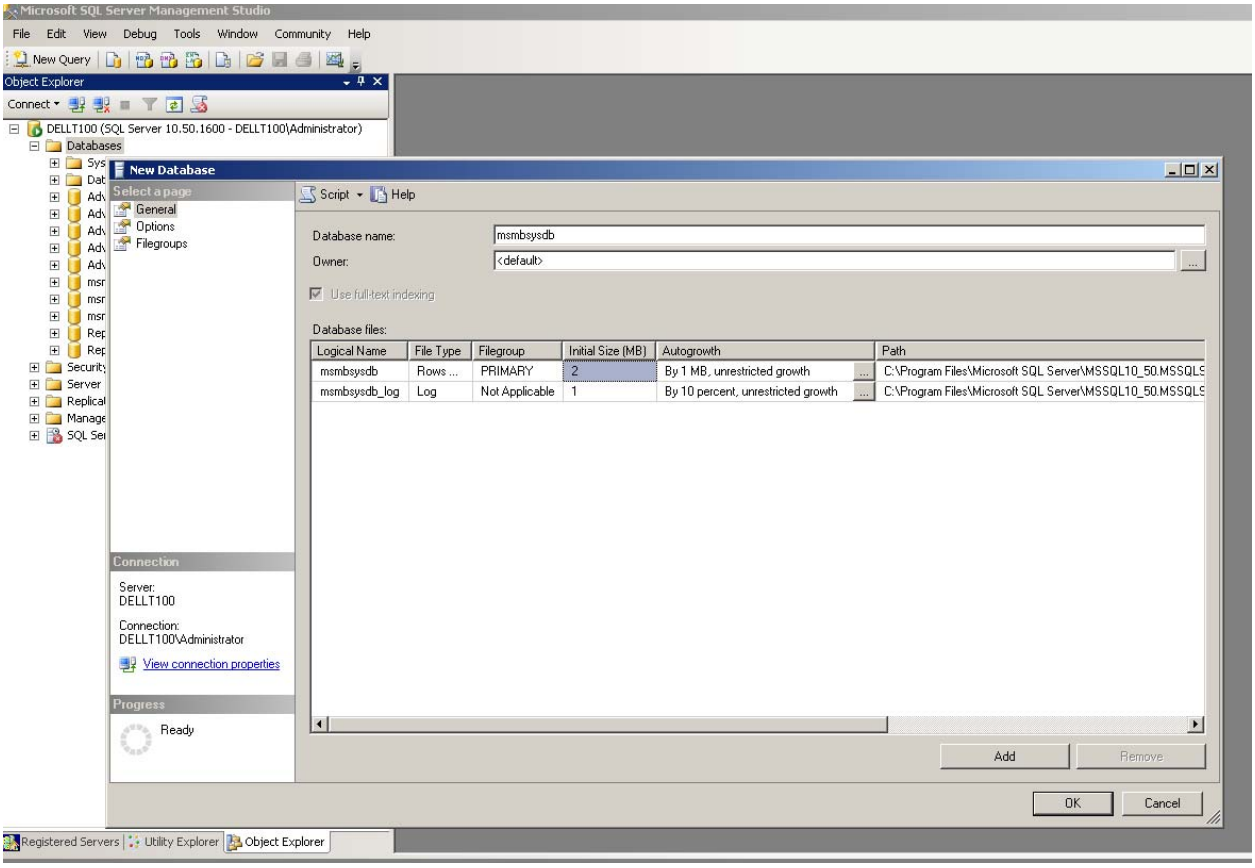

Figure 9. Defining a new MS SQL Database with SQL Server Management Studio.

defining the tables and indexes for msmbsysdb can be accomplished.

4.5.1.2 Use MS SQL to Define mbsysdb Tables and Indexes

This step makes msmbsysdb ready for use. That is, the SQL DDL script can be executed. This script exists within one single file that is stored in the DOC directory within the default wiscorp directory when the Metabase System was installed. This SQL DDL script file contains both table definition and index definition commands.

Figure 10 shows these SQL commands within the screen of a new query from within the SQL Server Management Studio. This command file should run to completion without any error. Once complete the database directory structure can be refreshed.

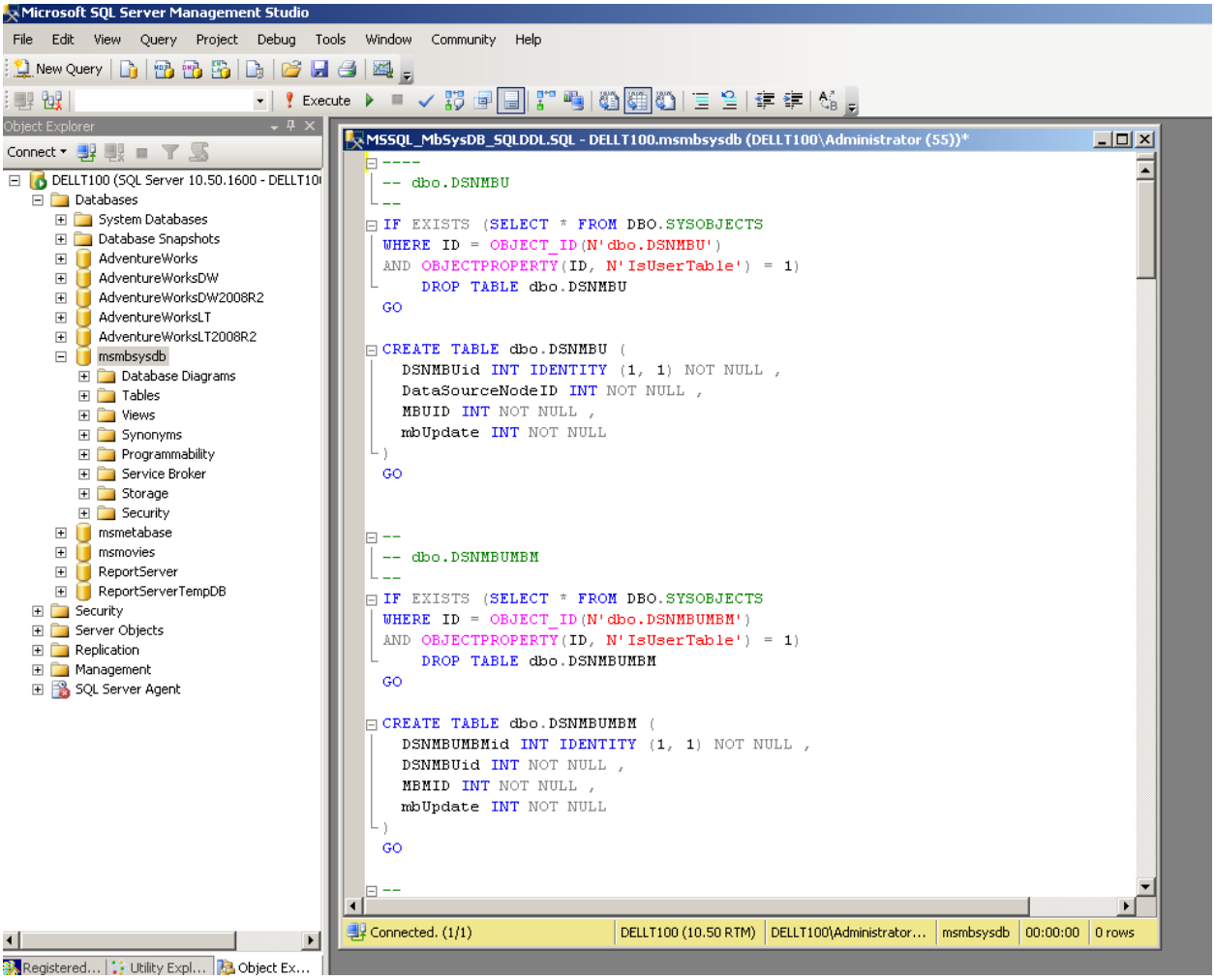

Figure 10. Defining Tables and Indexes for msmbsysdb Metabase Database.

This involves running a SQL DDL script. Any ODBC client will accomplish this. There are many SQL ODBC clients available, including QBE Vision's SQL Query Builder (qbebuilder. exe) that was installed with the Mimer SQL Engine server system files. Another tool named wSQL is available from the Mimer. com website. Both are free. You can also use a very popular ODBC client, WinSQL. This is very feature rich.

To accomplish running the SQL DDL script step, use the ODBC SQL client to login in to the mbSysdb using the SYSADM username, and the supplied password: "foodo2". Then, run the commands in the file, MB72_MbSysDB_MSSQL_SQLDDL that is located in the installed to directory's (e. g. C:\Program Files\Wiscorp\Metabase\DOC\) folder. After running the commands, disconnect the wSQL client from the Mimer mbsysdb database. At this point, the mbsysdb is ready for use.

4.5.2 Create MS SQL Metadata Databases

A Metabase Database instance houses the various tables of metadata that are created through the Whitemarsh Metabase functional modules such as Data Elements or Business Information Systems.. As with the mbsysdb database there are two steps:

- Define metabase to Mimer (Section 4.5.2.1)
- Use an ODBC utility to define the metabase tables, indexes and foreign keys (Section 4.5.2.2)
- Use ODBC utility to create default and extended reference data (Section 4.5.2.3)
- Use ODBC utility to define foreign keys (Section 4.5.2.4)

4.5.2.1 Define a Metabase Database to MS SQL

Running the Mimer Administrator brings up a window with multiple tabs. To create a Metabase dataset instance, first switch to the Local tab, and press the Add button. A window pops up into which information must be provided. In the entry field labeled Database Name enter for example: "metabase" (without the quotes). If you are creating a metabase for Marketing metadata or for metadata for the Eastern division, you might make the metabase's name, marketing or eastern. For this example, the metabase's name is Marketing.

In the Home Directory entry field, browse to the folder on your server that is to contain the Mimer database files for this Marketing dataset instance. Leaving the Create System Data Source checkbox enabled automatically creates an ODBC entry in the Window's ODBC Manager, associating this DSN with Mimer's ODBC drivers. Pressing OK will return a message window

that states: “No system databanks found in C:\Your\Selected\Folder\ Do you want to create them?” Pressing *Yes* begins the wizard to create the databank’s datafiles for this new Marketing metadata dataset. If you are merely evaluating the Metabase software, it is acceptable to change these drive values, consolidating the files to one folder. Press next after finishing any changes. *And remember, once again, it is best not to have explicit paths in the file names dialogs.* Again, it is best not to have explicit paths in the file names dialogs.

The first step of creating new databanks includes setting the locations of the various system datafiles for each particular mimer dataset. For performance reasons, Mimer defaults to distributing the databank’s various database files to multiple drives. If you are merely evaluating the Metabase software, it is acceptable to change these drive values, consolidating the files to one folder. Press next after finishing any changes. *And remember, once again, it is best not to have explicit paths in the file names dialogs.* Again, it is best not to have explicit paths in the file names dialogs.

The next step is to add the password for the Metabase Database. For example, it might be “foodo2.” ***Note Well: There is NO WAY to discover a Mimer database password. There is no “cracking tool.” So, choose wisely and don’t forget it.*** Enter a password value and confirm. When finished, press the Next button. The wizard then creates the various datafiles for the Metabase Database dataset, in the locations that you specified.

When the creation of the databank’s datafiles is completed, the wizard prompts the user whether or not to “Start Database Server,”and whether or not to “Start Wizard for Development User and Databank and Example. ”Leave enabled the “Start Server” check box, but disable the second checkbox. The wizard for creating example data, and developmental user is not necessary for the Metabase Software. Press Finish to complete the wizard.

The last step prior to running the SQL DDL scripts is to create various databanks within the Mimer databases. This is done by running the SQL statement, “Create Databank <mimer database name>;” In this case, the database names are mbsysdb, movies, or metabase.

4.5.2.2 Use ODBC Utility to Define Metabase Tables, and Indexes

The first step in getting a Metabase Database instance ready for use it to run a SQL DDL script. This involves three different SQL DDL files, which are:

- MB01_MSSQL. SQL: Create table statements.
- MB03_MSSQL. SQL: Alter table statements (primary keys, sequences, and indexes)

These files are included in the directory:

- C:\Program Files\Wiscorp\Metabase\DOC

Use the ODBC SQL client, wSQL, that was used to build the mbsysdb. After running the Create Tables and Create Indexes scripts, disconnect the wSQL client from the Mimer Metabase database. These two SQL steps have to be accomplished for every Metabase Database Instance. Upon completion, a Metabase database fully exists, although completely empty of all data.

NOTE: you must NOT run the Foreign Keys script (MB04_MSSQL.SQL) at this time.

This step makes mbsysdb ready for use. This involves running a SQL DDL script. Any ODBC client will accomplish this. There are many SQL ODBC clients available, including QBE Vision's SQL Query Builder (qbebuilder. exe) that was installed with the Mimer SQL Engine server system files. Another tool named wSQL is available from the Mimer. com website. Both are free. You can also use a very popular ODBC client, WinSQL. This is very feature rich.

To accomplish running the SQL DDL script step, use the ODBC SQL client to login in to the mbSysdb using the SYSADM username, and the supplied password: "foodo2". Then, run the commands in the file, mbsysdb.ddl that is located in the installed to directory's (e. g. C:\Program Files\Wiscorp\Metabase\DOC\) folder. After running the commands, disconnect the wSQL client from the Mimer mbsysdb database. At this point, the mbsysdb is ready for use.

4.5.2.3 Use ODBC Utility to Create Default and Extended Reference Data

There are two strategies for migrating SQL Data. Use a utility, or directly execute SQL Insert Values scripts.

The utility that Whitemarsh uses is Data Management Center (DMC). The website is:

- http://dmc-fr.com/

The process is fundamentally this:

1. Pick and make sure you have access to your source database and DBMS.
2. Pick and make sure you have access to your target database and DBMS.
3. Map source to target tables for a table-centered transfer.
4. Save the created profile.
5. Execute the profile.

In the case of MS SQL, the necessary syntax for handling Identity is performed.

The three data transfer profiles needed to accomplish the data transfer are:

- Transfer Mimer Movies to MS SQL Movies

- Transfer Reference Data Tables from Mimer Movies to Mimer Metabase
- Transfer Mimer Metabase to MS SQL Metabase

The alternative to this automated process is to accomplish the data transfer directly through the use of SQL Insert Values Scripts. This process causes the loading of both default and extended reference data. The files for this step are included in the reference data directory:

- C:\Program Files\Wiscorp\Metabase\DOC

Prior to inserting these data, the Identity setting for MS SQL has to be changed. The file for accomplishing this is:

- SetIdentityOnSQLDDL.txt

This file needs to be executed prior to inserting all the extended reference data. The extended reference data is stored in a single file:

- Extended Reference Data Scripts_MS SQL.txt

Use an ODBC utility to load all these Insert Values statements into the 180+ Metabase System database tables. The size of this file is about 135K and there are about 2500. Records.

Once these rows of data are inserted into the metadata database, another SQL DDL file needs to be executed to reverse the IdentityOn setting. That file is:

- SetIdentityOFFSQLDDL.txt

Once this file is executed through an ODBC utility, the final and next step, setting the metabase's foreign keys can be accomplished.

4.5.2.4 Use ODBC Utility to Define Metabase Foreign Keys

The final SQL DDL file contains all the specifications for the foreign keys. This establishes all the relationships among the metabase tables and sets all the referential integrity specifications and actions. The file for MS SQL is:

- MB04_MSSQL. SQL : Foreign Key statements

5.0 Migrating a Metabase Database to another SQL Engi ne

If you have downloaded the Free version of the Metabase System, have installed it using, for example, the SQL DBMS Mimer, extensively built metadata within Mimer-engine based metabases and then want to, for example, convert those Mimer metabase databases to MS SQL, there are a number of different ODBC based tool sets available to accomplish this.

In short, if you download, install and use Mimer you should not become worried that your metabase generation efforts will be trapped in Mimer. The data migration process is simple and straight forward. This is not to say, however, that there is data interoperability between some metadata that may be stored using the Mimer SQL engine and other metadata that may be stored using MS SQL.

There is only the ability to migrate the data from being managed by one SQL engine to being managed by another SQL engine. Whitemarsh uses WinSQL, and ODBC SQL utility (www.synametics.com) and has used its facilities to export Metabase data into the SQL Insert Values format string, merged the exported files together, and then used that data to load into an empty Metabase database. The general process that clearly works are:

- Export the Metabase data into an Insert Values format using, for example, WinSQL.
- Create a new (but empty) SQL database instance.
- Load in the SQL DDL that just accomplishes Create Tables (file 1).
- Load the exported Insert Values formatted data for all the Metabase database tables
- Load in the SQL DDL file that installs all primary keys and other indexes (file 3)
- Load in the SQL DDL file that accomplishes all the foreign keys (file 4)

Note finally, the installation of this software requires the user being logged into their computer with Administrator privileges.

6.0 Using the Metabase For Real Work

There are two documents that describe use of the metabase. These are:

- Data Modeler: Architecture and Concept of Operations
- Metabase Reverse and Forward Engineering

Both these documents are available for downloading from the Whitemarsh website, Metabase page. Check back to this page from time to time for additional Metabase For Real Work documents.

7. 0 Other SQL Engine Backends

The metabase system has been engineered to access SQL backends through Microsoft's ODBC drivers. The only requirements that the metabase imposes on the back end are:

- Referential Integrity
- Automatic Incrementing Primary Keys

There are a few places in the Metabase software where SQL statements are explicitly created. For example, the resetting of the high-value of an auto-incrementing primary key.

Metabase users desiring to use a SQL engine different from Mimer are requested to contact Whitemarsh for assistance.