



Whitemarsh
Information Systems Corporation

Whitemarsh Metabase Admin Module User Guide

March 2011

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

1	Installation	1
2	Administrator Functions	7
2.1	Creation and Maintenance of SQL Engine Connection Strings	7
2.2	The New/Mod Data Group	14
	2.2.1.1 Create Default/Unknown Data Records	15
	2.2.1.2 Create Default SQL and DBMS Datatypes	16
	2.2.2 GenMaxID	16
2.3	The Data Type Management Group	19
	2.3.1 Manage Data Types	19
	2.3.1.1 Value Domain Data Types	19
	2.3.1.2 SQL Data Types	19
	2.3.1.3 DBMS Data Types	19
	2.3.1.4 SQL Data Type to Value Domain Data Type Reassignment ...	19
	2.3.2 Export Data Types	29
2.4	The User Management Group	30
	2.4.1 Users	30
	2.4.2 DSN Users	30
	2.4.3 User Modules	33
	2.4.4 Current Users, Messages, and Auto-Logout	33
2.5	Abbreviation Management	37
	2.5.1 Abbreviation Generation	38
	2.5.2 Import Abbreviations	41
	2.5.3 Abbreviation Business Domain Reassignment	45
	2.5.4 Abbreviation Mining	46



Figure 1. Opening Screen of the Metabase System Admin Module.	3
Figure 2. Entering the User Name and Password for the Metabase System Admin Module. ...	4
Figure 3. Metabase System’s Admin Module Administrator Functions.	5
Figure 4. Metabase Database Bootstrap Functions.	6
Figure 5. Metabase System Architecture	8
Figure 6. SQL DBMS Selection Screen.	9
Figure 7. Connection String fields for Connecting to MbSysDB.	10
Figure 8. Entering the Sysadm User Name and Password.	11
Figure 9. Databases and Connection Strings for SQL Engine’s MbSysDb Data Source Node table.	12
Figure 10. Connection String for Metabase Database within MbSysDB Data Source Node table.	13
Figure 11. New/Mod Data Functions.	14
Figure 12. Auto Data Process.	15
Figure 13. Steps in Generating Table Maximum Sequence Numbers.	17
Figure 14. Capitalization process of metadata data models.	18
Figure 15. Data Type management processes.	20
Figure 16. Value Domain Data Types.	21
Figure 17. Value Domain Data Type update screen.	22
Figure 18. SQL Data Type list.	23
Figure 19. SQL Data Type update screen.	24
Figure 20. DBMS Data Type list.	25
Figure 21. DBMS Data Type update screen.	26
Figure 22. Reassignment of SQL Data Types to Value Domain Data Types.	27
Figure 23. Reassigning DBMS Data Types to SQL Data Types.	28
Figure 24. Export Data Types process.	29
Figure 25. List of Metabase users.	31
Figure 26. Metabase User update screen.	32
Figure 27. Allocating DSN Users to Metabase Modules.	34
Figure 28. Current Metabase Users.	35
Figure 29. Metabase Administrator Message to Current User update screen.	36
Figure 30. Abbreviation management functions.	37
Figure 31. Abbreviation Generation screen.	38
Figure 32. Abbreviation update screen.	39
Figure 33. Abbreviations import screen.	41
Figure 34. Abbreviations Import. Step 1, file selection and option setting.	42
Figure 35. Abbreviation Import. Step 2, Duplicate options.	43
Figure 36. Abbreviations Import. Step 3, field mapping and “Start.”	44
Figure 37. Re-Assignment of one or more abbreviations to a different business domain.	45
Figure 38. Abbreviation mining screen.	47



1 Installation

All metabase applications, including the metabase admin module are 32 bit applications that run on Windows operating systems platforms with ODBC access to an SQL engine that supports the metabase database instances. The Metabase System's installation process, as described in the Getting Started Guide, installs the metabase software, example data, as well as other various help files, and several admin utilities. The modules that are placed into the install directory are:

Metabase Module	Module purpose
Metabase (mb.exe)	Main module with contained submenus
Metabase Admin (mbadmin.exe)	Contains references to metabase administrator functions
Business Information Systems (mbbis.exe)	Allows inventorying of business information systems
Database Objects (mbdbo.exe)	Allows creation of database objects from tables contained in the implemented data model
Data Modeler Data Elements (mbdmde.exe)	Allows definition of data semantics, data elements, data element domains, compound data elements, and derived data elements
Data Modeler View Data Model (mbdmvdm.exe)	Allows definition of views and the mapping of view elements to DBMS columns of the operational data model, or to compound and derived data elements.
Data Modeler Implemented Data Model (mbdmidm.exe)	Allows definition of schemas, tables, columns, primary and foreign keys, and the creation of operational data models. A column map to both a data element and to attributes from specified data models.
Data Modeler Operational data model (mbdmmodm.exe)	Allows definition of DBMS schemas, DBMS tables, DBMS columns, DBMS primary and DBMS foreign keys, and the creation of operational data models.
Data Modeler Specified data model (mbdmsdm.exe)	Allows definition of subjects, entities, attributes, primary and foreign keys, and the creation of implemented data models. An attributes map to both a data element and to columns from the implemented data model.



Metabase Module	Module purpose
Document and Form (mbDocFrm.exe)	Allows the specification of Documents, interrelationship among Documents, Document Sections and Subsections, Document Cells, and the relationship between Document Section Cells and Mission Organization Form, and View Columns. This module allows creation of the same materials for Forms, interrelationship among Forms, Form Sections and Subsections, Form Cells, and the relationship between Form Section Cells and Mission Organization Form, and View Columns
Information needs analysis (mbina.exe)	Allows the capture of information needs of an enterprise and the characterization of those information needs. Information needs are then allocated in a ranked order to persons fulfilling functions within an organization in service of a mission.
Mission, Organization, Function Person Assignment (mbmofpa.exe)	Allows the capture of enterprise missions, organizations, functions, and persons. Additionally the assignment of organizations to missions, functions to mission-organizations, and persons to mission-organizations-functions.
Requirements Management	Allows for the specification of requirements and the allocation of those requirements to a number of different components in metabase databases. Requirements are able to be collected into network structures, and then within these contexts, assigned to any one or more of the following: Business Events, Business Information Systems, Database Objects, Data Integrity Rules, DBMS Columns, Mission-Organization-Functions, Resource Life Cycle Nodes, Use Case Events, and User Acceptance Test Steps. Note: As of version 7.2 of the Metabase System, the following functional modules do not exist: Data Integrity Rules, Use Cases, and User Acceptance Tests. These will be created during the second half of 2010.
Resource Life Cycle Analysis (mbrlc.exe)	Allows the capture of the essential resources of the enterprise, the development of the resource life cycles, the interrelationships between different nodes of different life cycles, and the assignment of information needs to particular resource life cycle nodes. Business information systems and databases (which are then related to implemented data models) are also assigned to resource life cycle nodes.



Metabase Module	Module purpose
Use Cases (mbUseCase.exe)	Allows the capture of the use cases and the interrelationship of use cases within network structures. Each use case contains pre- post- and special conditions. It also contains hierarchical collections of use case events. Use case events identify use case event actors and the roles they play, and can be interrelated with the business information systems that implement them. Use cases contain facts that are related to database table columns. The use case facts can be allocated to specific use case pre and post conditions and also to use case events.
Miscellaneous DLLs	Various run-time DLLs required by the executable modules

In addition to the main metabase directory (i.e. C:\Program Files\wiscorp\metabase), other directories are created store certain metabase databases, SQL DDL reference data, and some of the documentation. The bulk of the Metabase System User Guides are available for downloading from the Metabase's webpage on the Whitemarsh website, www.wiscorp.com.

The metabase administrator's module can be run from the main metabase menu. The Metabase is activated through the Windows Start Menu folder, created for the Whitemarsh Metabase during the install process.

When the **mbadmin.exe** is started, a window, shown in Figure 1, is presented that provides a choice of which SQL DBMS is to be employed as the engine for all the metabase databases. Once that choice is made and the window is closed, another window, shown in Figure 2, comes up prompting the user to enter a username and password. The default username/password set of 'sysadm'/'sysadm'). Once the Send button is pressed and after the Log On button is pressed, the Admin module is ready to use under two different modes:

- Administrator Functions
- Metabase Database Bootstrap Functions

The first mode is the default mode and is shown by the string, "Administrator Functions" just to the right of the button Dataset that is to the top right of the screen shown in Figure 3.



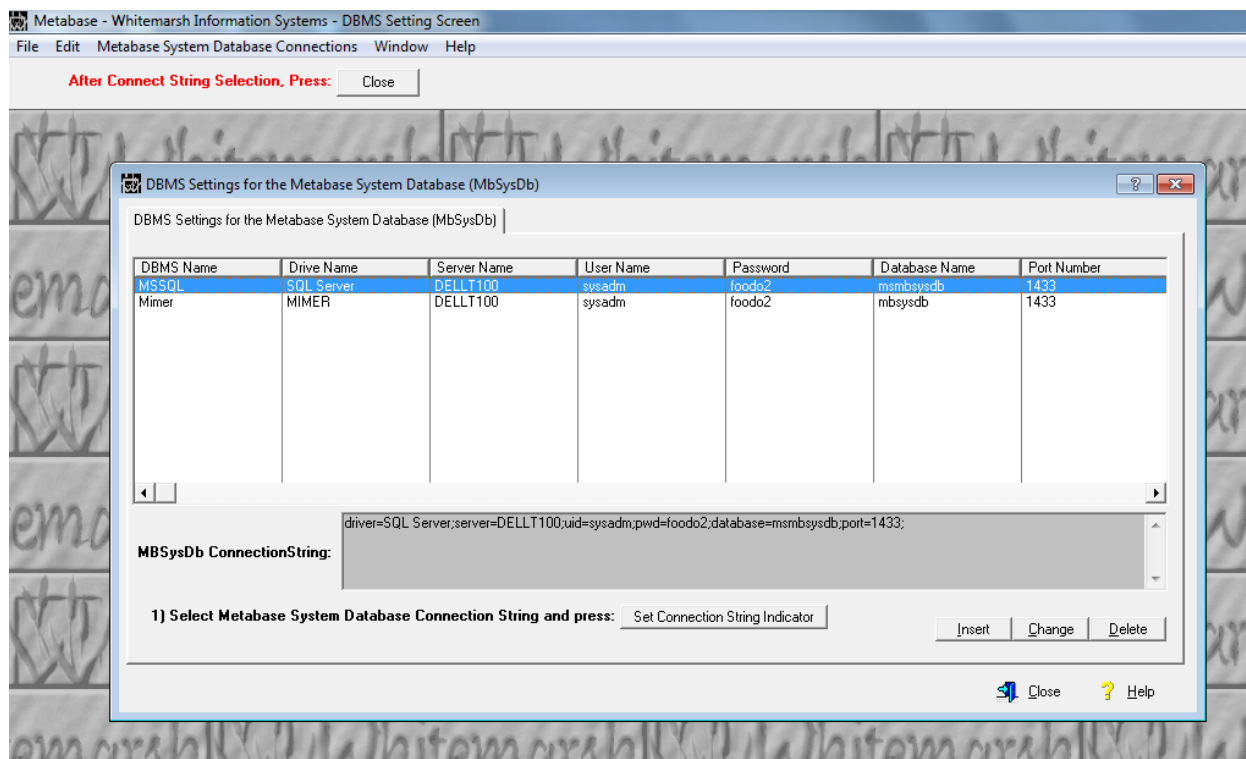


Figure 1. Opening Screen of the Metabase System Admin Module.



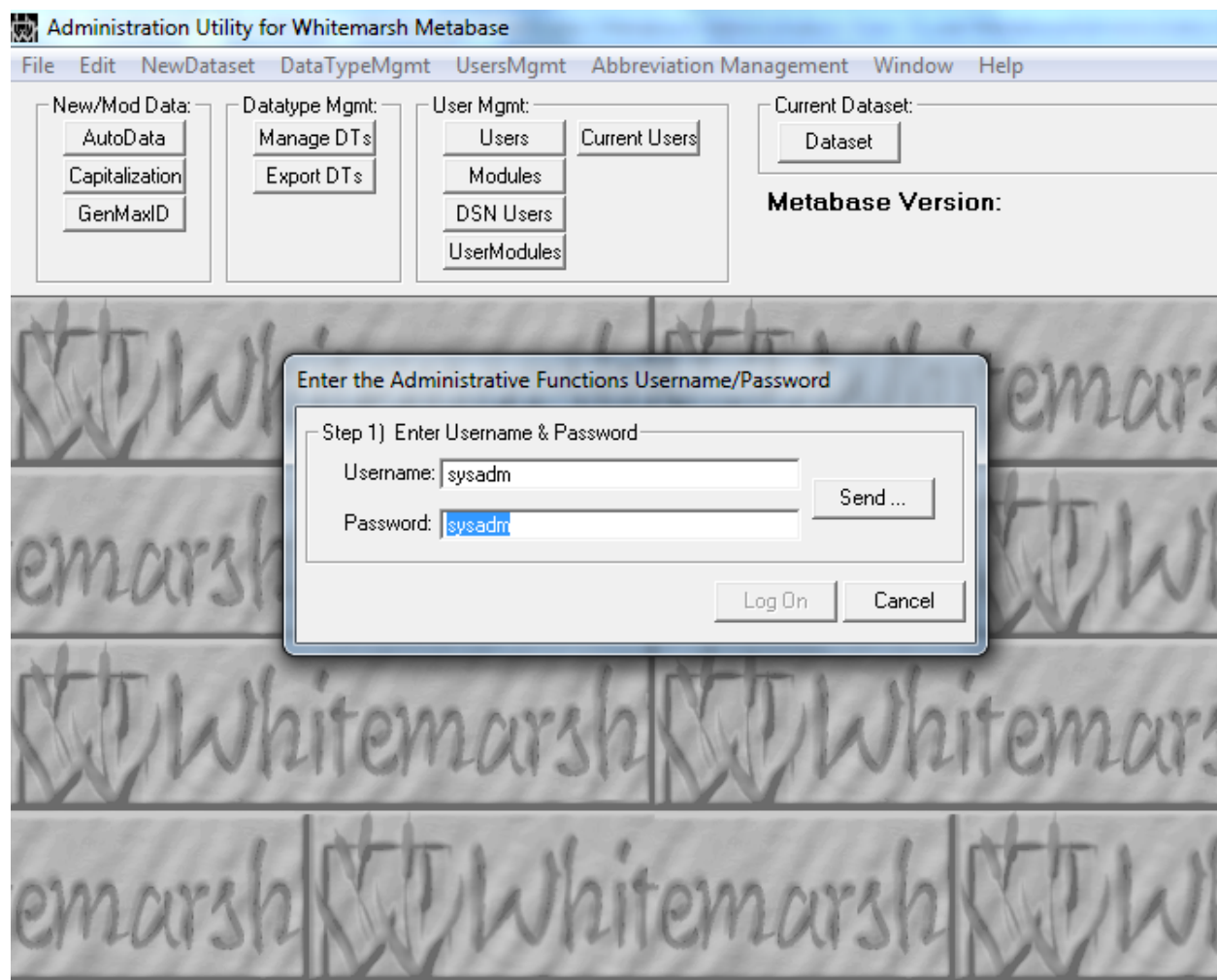


Figure 2. Entering the User Name and Password for the Metabase System Admin Module.



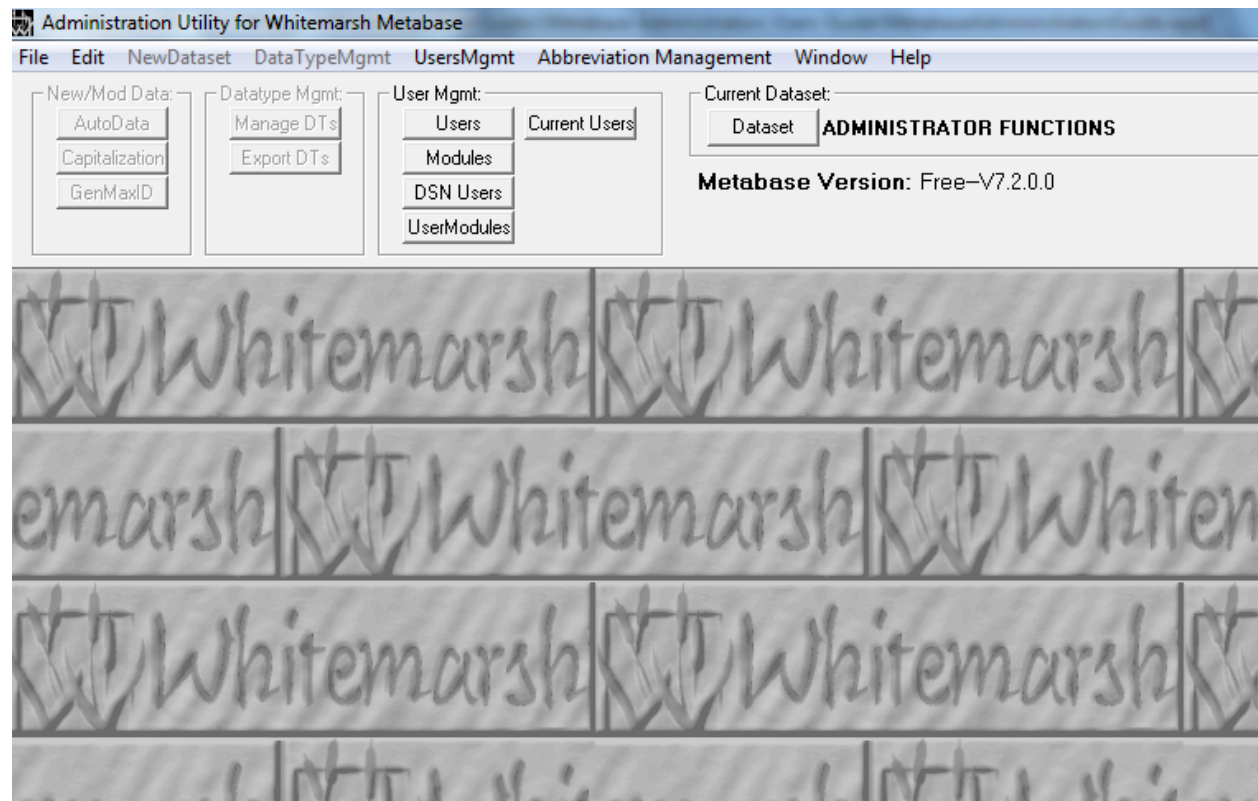


Figure 3. Metabase System's Admin Module Administrator Functions.

The Metabase Administrator can now activate any of the following User Management functions:

- Users
- Modules
- DSN Users
- User Modules
- Current Users

These functions are all describe in Section 2.4 of this Admin Module Users Guide.

The second mode, Metabase Database Bootstrap Functions, is activated by pressing the Dataset button. This brings up another screen that is shown in Figure 4. The screen allows for the selection of the Metabase database, that is, msmovies, msmetabase, or msmetabase2. This choice is predicated on the selection of the SQL DBMS that was performed in Figure 1. Under this mode the following functions are allowed:

- AutoData



- GenMaxId
- Capitalization
- Manage DTs
- Export DTs

These functions are detailed in Sections 2.2 and 2.3 of this Admin Module guide.

Note: *As of August 1, 2010, you cannot reselect the Metabase Database by re-pressing the Dataset button and choosing another. Additionally, while you can press the User Management buttons, an error will occur. These two problems will be fixed in the next interim release of the Metabase System.*



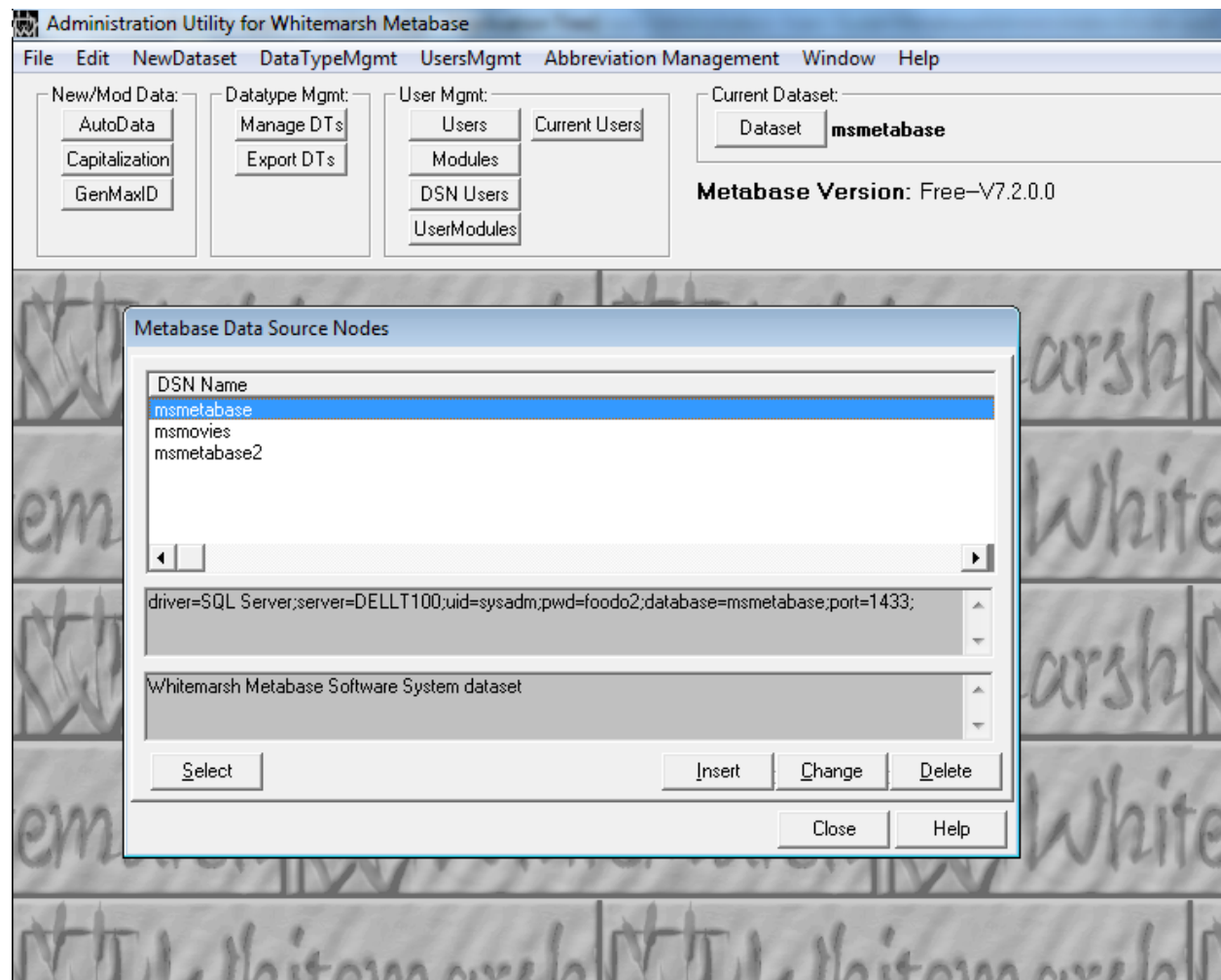


Figure 4. Metabase Database Bootstrap Functions.



The Metabase Administrator module plays three main roles. The first role is to aid the Metabase DBA in creating a new Metabase Dataset. The second role is to maintain a dataset, utilizing several specific Metabase functions. Lastly, the Metabase Admin Module controls the access of Metabase users to Metabase datasets for specific Metabase modules.

2 Administrator Functions

There are four main sets of administrator functionality:

Administrator Functionality	Description
SQL Engine Connection String Definition	This capability enables the Metabase System's administrator to create the boot-strapping connection strings to the Metabase System Databases, and once used to then create Connection Strings to the specific Metabase Databases.
New/Mod Data	The New or Modify Data functionality causes the creation of default data within a new metabase instance, the generation of maximum values for row keys, and changing capitalized metadata to be CamelCase.
DataType Mgmt	The Data Type Management functionality enables the metabase administrator to create and manage data types across the three classes of data types.
UserMgmt	The User Management functionality enables the creation of users, modules, users of data sets, and the allocation of users to specific user modules.
Abbreviation Management	Abbreviation management enables the creation of words and phrases that will ultimately be subject to abbreviation within the Operational Data Model.

2.1 Creation and Maintenance of SQL Engine Connection Strings

The Metabase System was designed to be independent of specific SQL engines. That's because it uses Microsoft's ODBC. Throughout the entirety of the Metabase System there is only one place where an explicit SQL command is used. This one place is contained within the DLL, MbUser. The purpose of this one use is to query the SQL DBMS's schema information tables to get the connected-to data source name. The programming logic of MbUser knows which SQL DBMS is being employed and branches around to the correct SQL DBMS's schema information table.



Regardless of the SQL DBMS, the overall architecture, as depicted in Figure 5, is the same. 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.

Because the Metabase System is largely independent of the SQL DBMS 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.

When the Admin module is executed, the SQL DBMS choice screen is presented. It is depicted in Figure 6. This screen shows the current set of Connection Strings to the MbSysDB data sets that are known to the Admin module. These connection strings are stored in a Clarion Access Method file, DBMSSetting.tps. The fields on the screen are those that are required to be known by Microsoft's ODBC interface mechanism so that the MbSysDB database can be accessed. Since it is an almost certainty that these Connection Strings are not correct for your

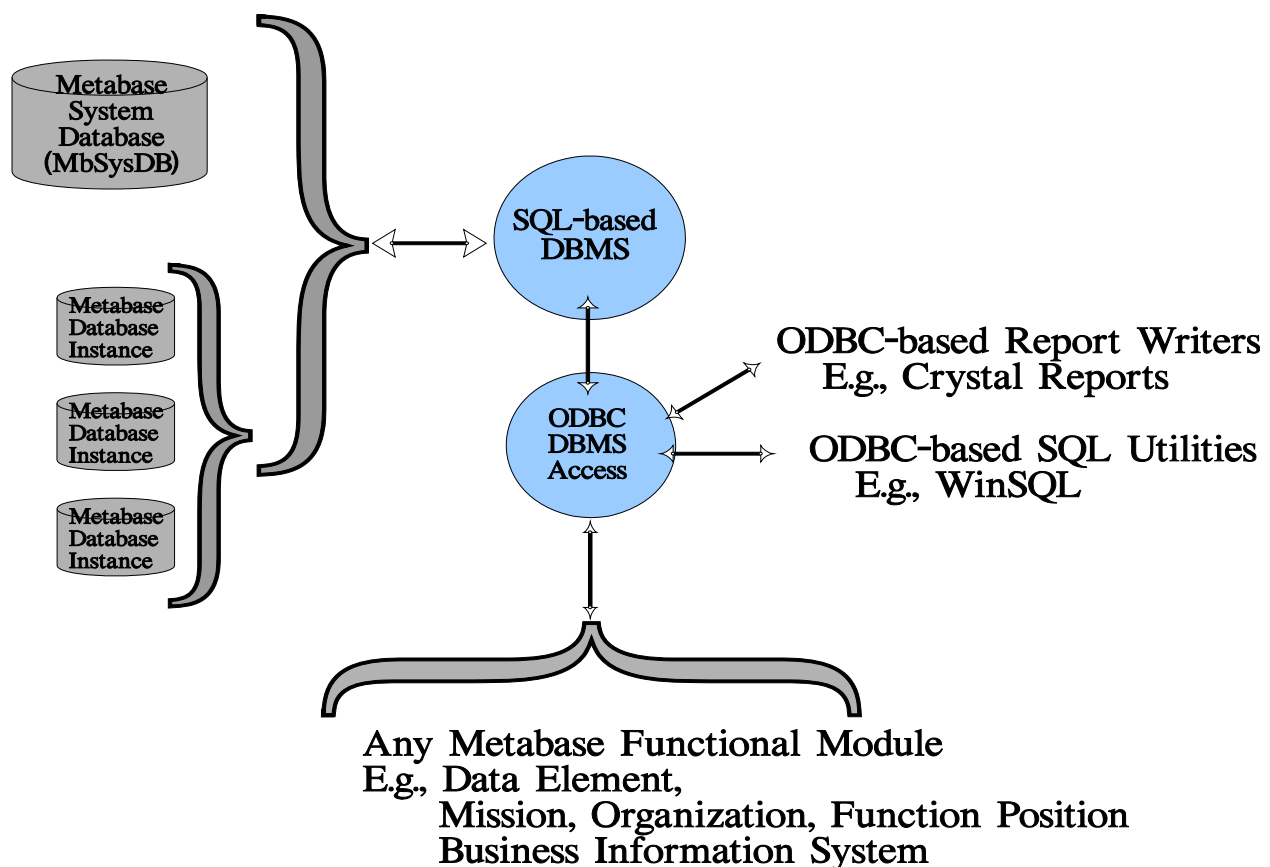


Figure 5. Metabase System Architecture



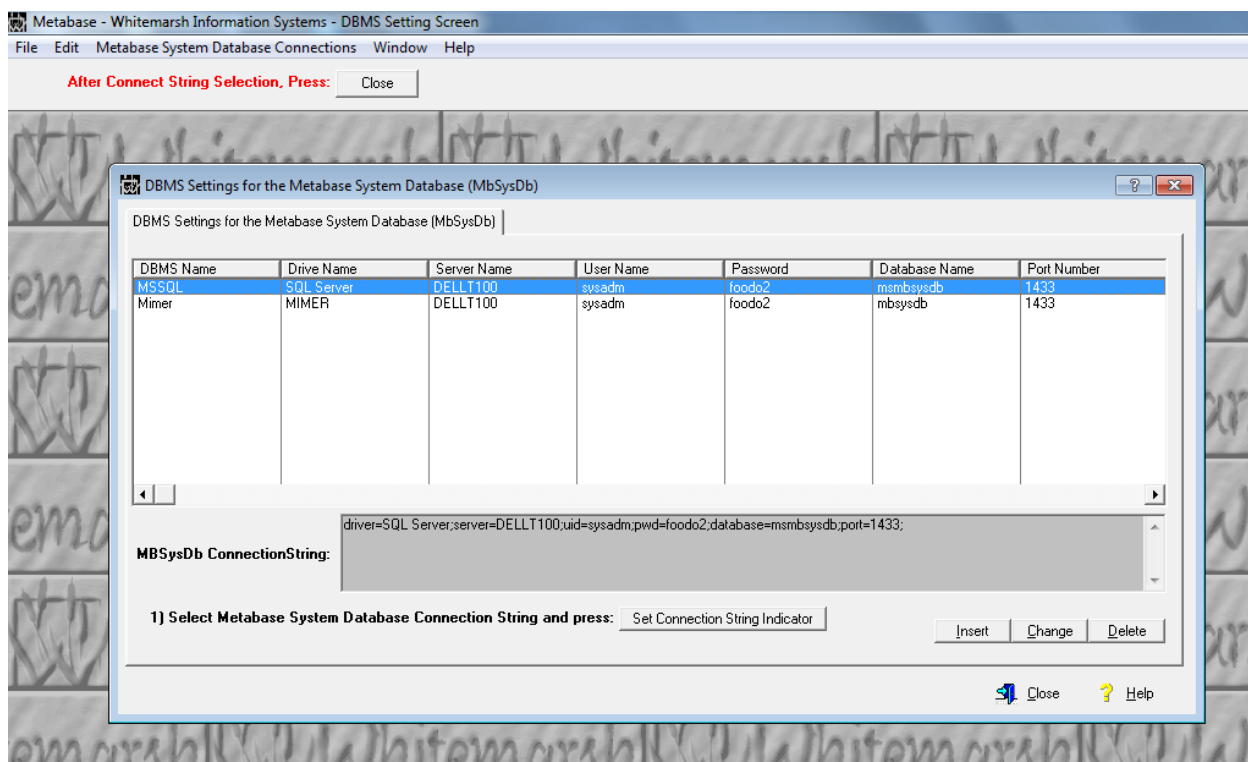


Figure 6. SQL DBMS Selection Screen.

Metabase System installation they need to be changed. Hence the existence of the Insert-Change-Delete buttons.

Figure 7 shows the Change screen. This screen contains all the fields necessary to establish a connection between a client machine and a server on which the MbSysDB database is stored. When this screen is completed, click OK. At this point a record in the DBMSSetting.tps table is created or updated.

Note: Once the Connection String records are created, the Metabase Administrator MUST distribute this file (i.e., DBMSSetting.tps) to every Metabase System client installation and must place this file in the directory where all the other Metabase System executables are stored. This is NOT an optional step. This is a MUST-DO step.

After this is done, close the Admin Module and restart Admin. When the opening screen is presented, choose the SQL DBMS and press the Select Connection String Indicator button. If the connection is made, an error will not be posted. Thereafter, press the Close button at the top of this screen.

At that point, Figure 8 is presented. Enter the administrator's user name and password, which is sysadm in both cases. Press Send, and then Log On. That screen closes. Press the Dataset button. When this is done, a new screen, Figure 9 opens. This screen is presenting the



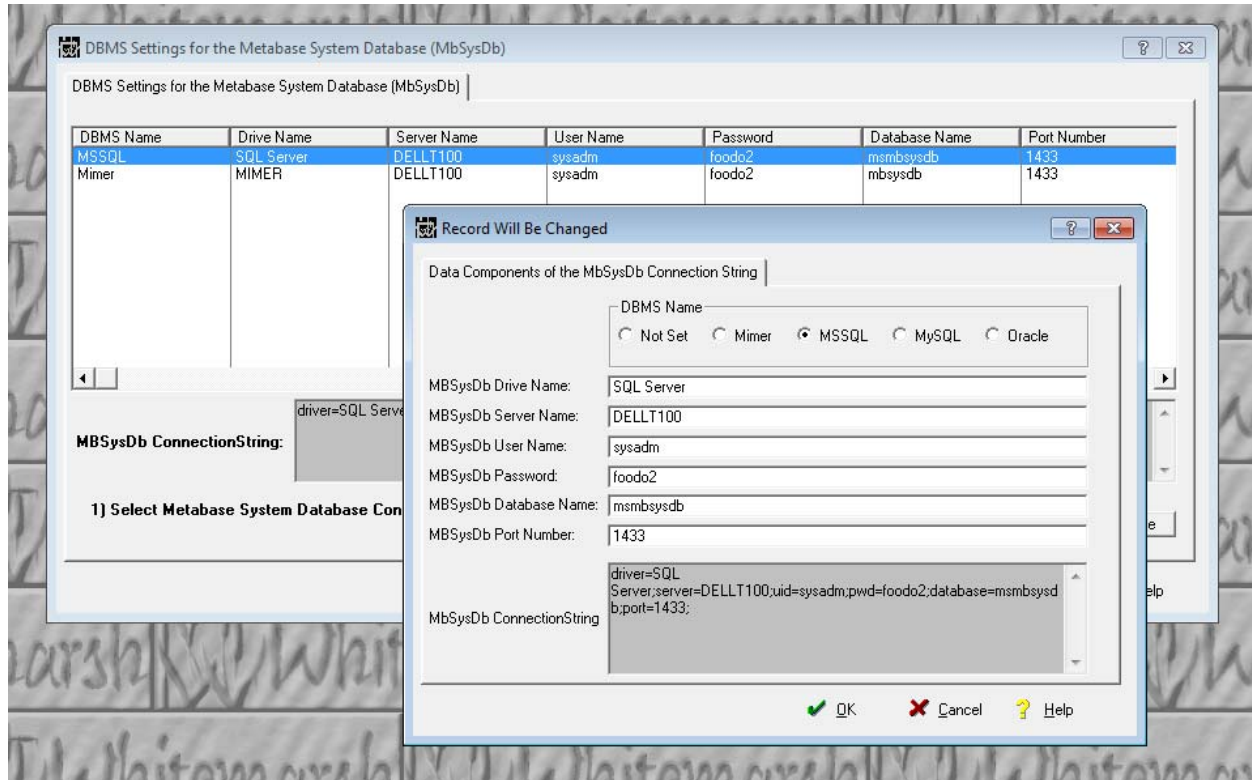


Figure 7. Connection String fields for Connecting to MbSysDB.

Connection Strings for all the databases that exist within the MbSysDB data source node table. From this figure, there are three databases: msmovies, msmetabase, and msmetabase2.

The metabase administrator is now able to change the Connection String fields for each of these databases. To do that, press, Change. Figure 10 is then displayed. Enter the values as appropriate so that there are then valid connection strings to the various Metabase Databases. Once this is done, close the window.

At this point, all the Metabase System functional modules will be able to access their proper Metabase Databases through the SQL DBMSs that are then installed and set up to run the Metabase System.



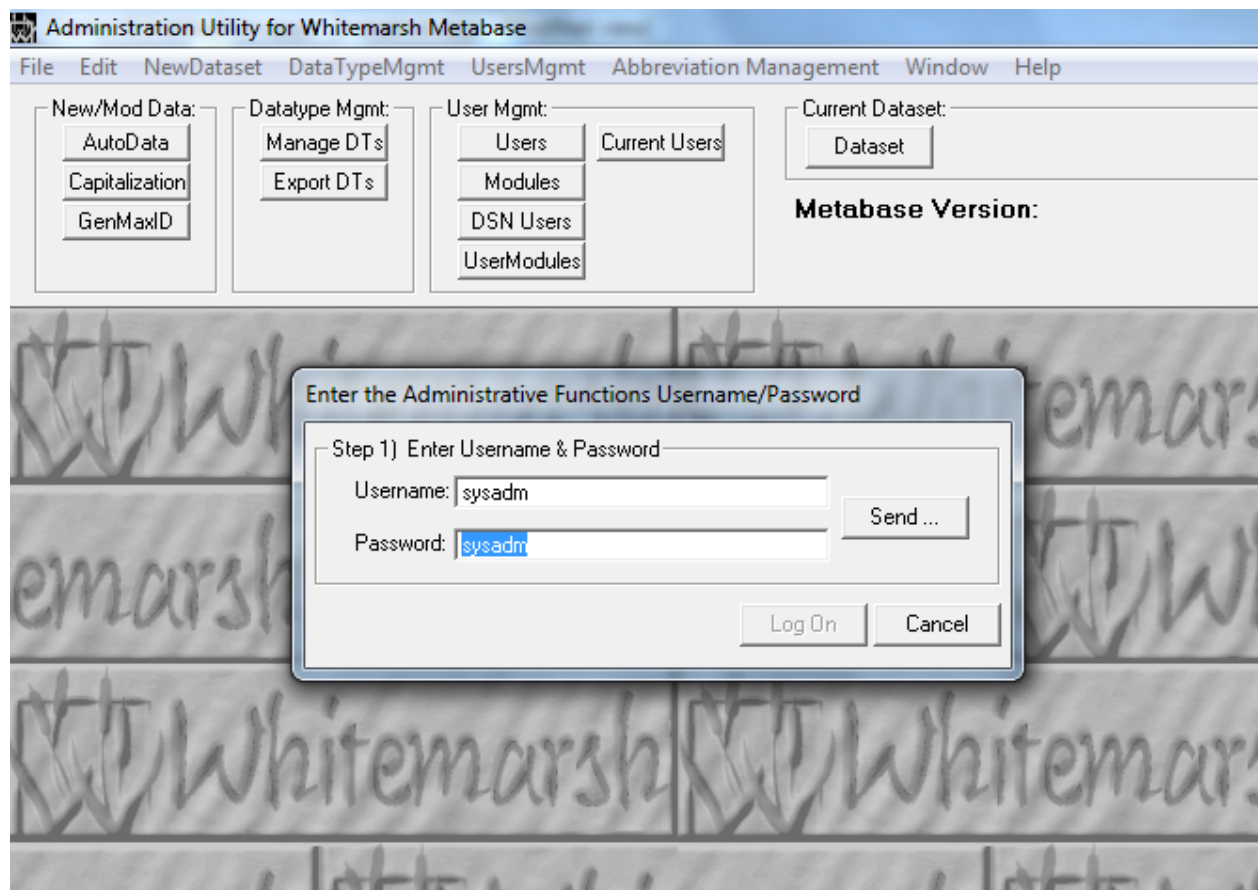


Figure 8. Entering the Sysadm User Name and Password.



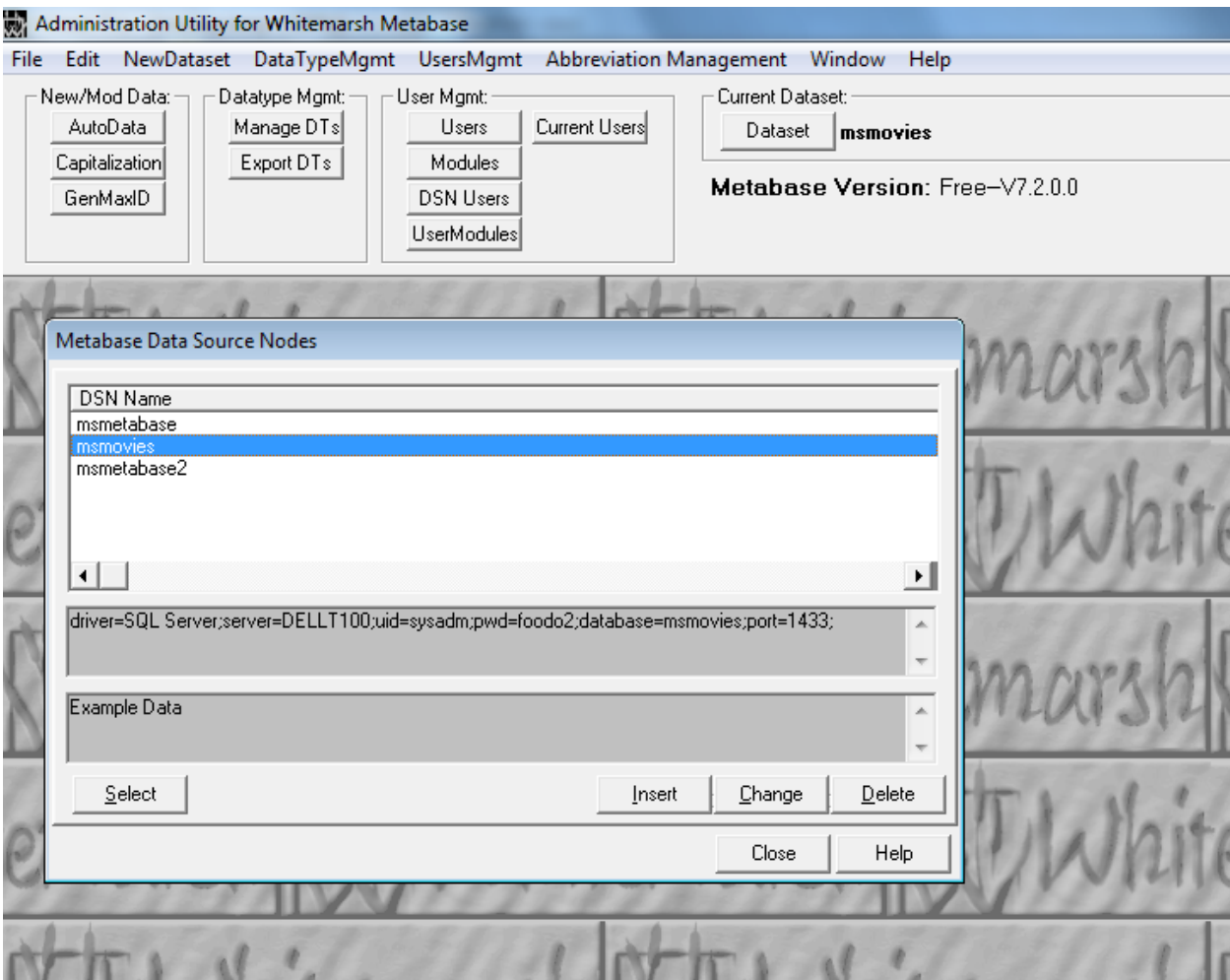


Figure 9. Databases and Connection Strings for SQL Engine's MbSysDb Data Source Node table.



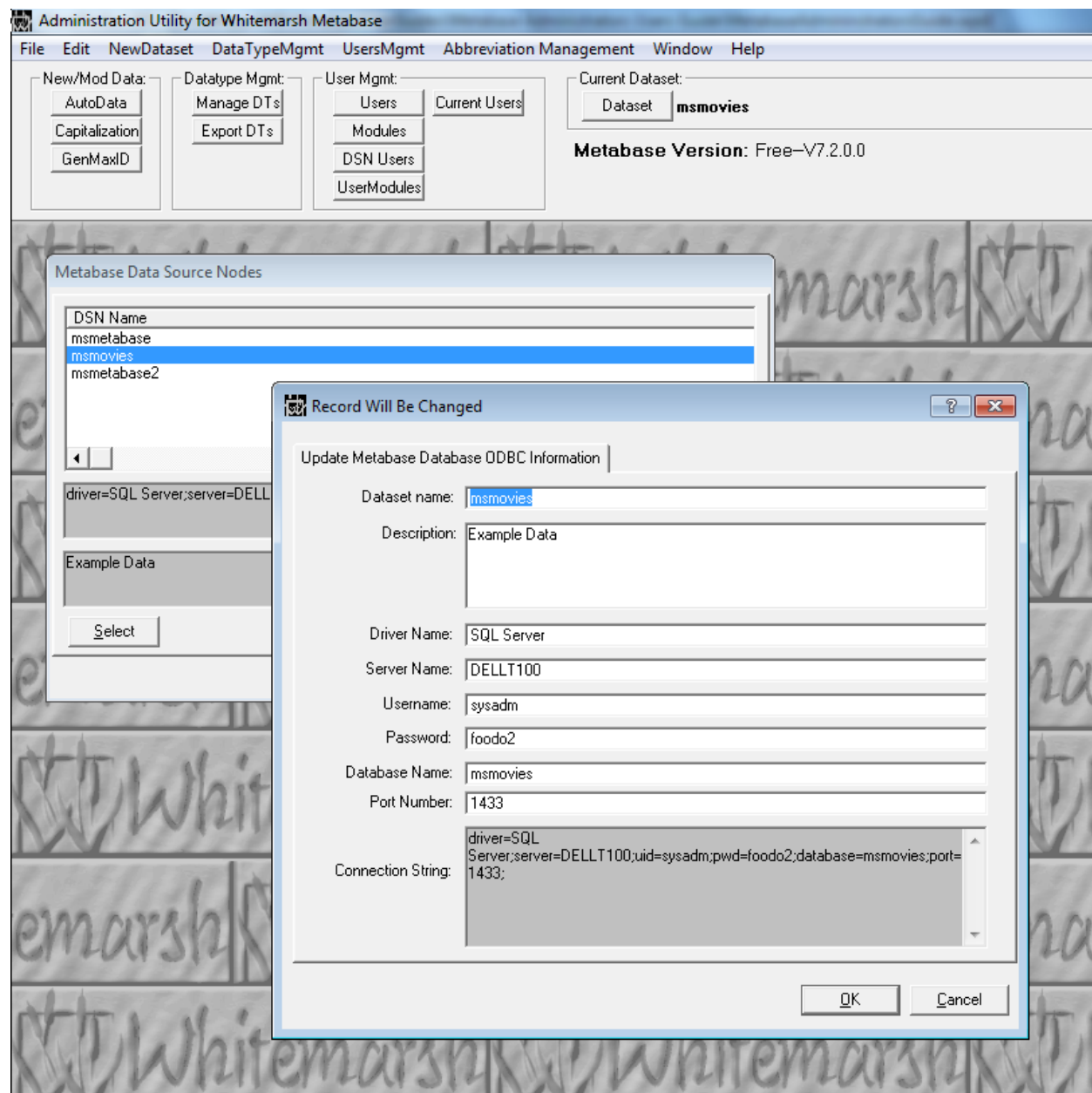


Figure 10. Connection String for Metabase Database within MbSysDB Data Source Node table.



2.2 The New/Mod Data Group

There are three buttons that create and/or manage initial data within a metabase instance. These groups are AutoData, GenMaxID, and Capitalization. These are illustrated in Figure 11.

2.2.1 Auto Data

The AutoData, button brings up a window that has two buttons, Create Default/Unknown Data Records, and Create Default SQL and DBMS Datatypes serving two functions, as shown in Figure 12.

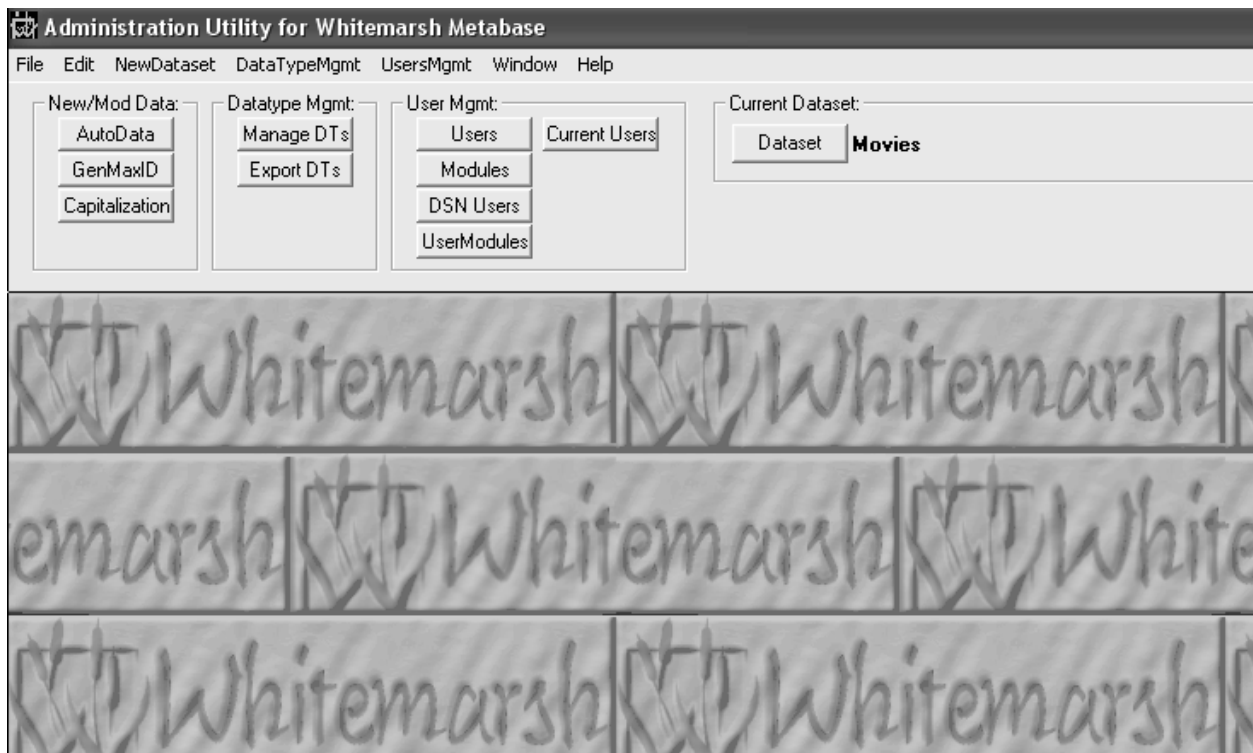


Figure 11. New/Mod Data Functions.



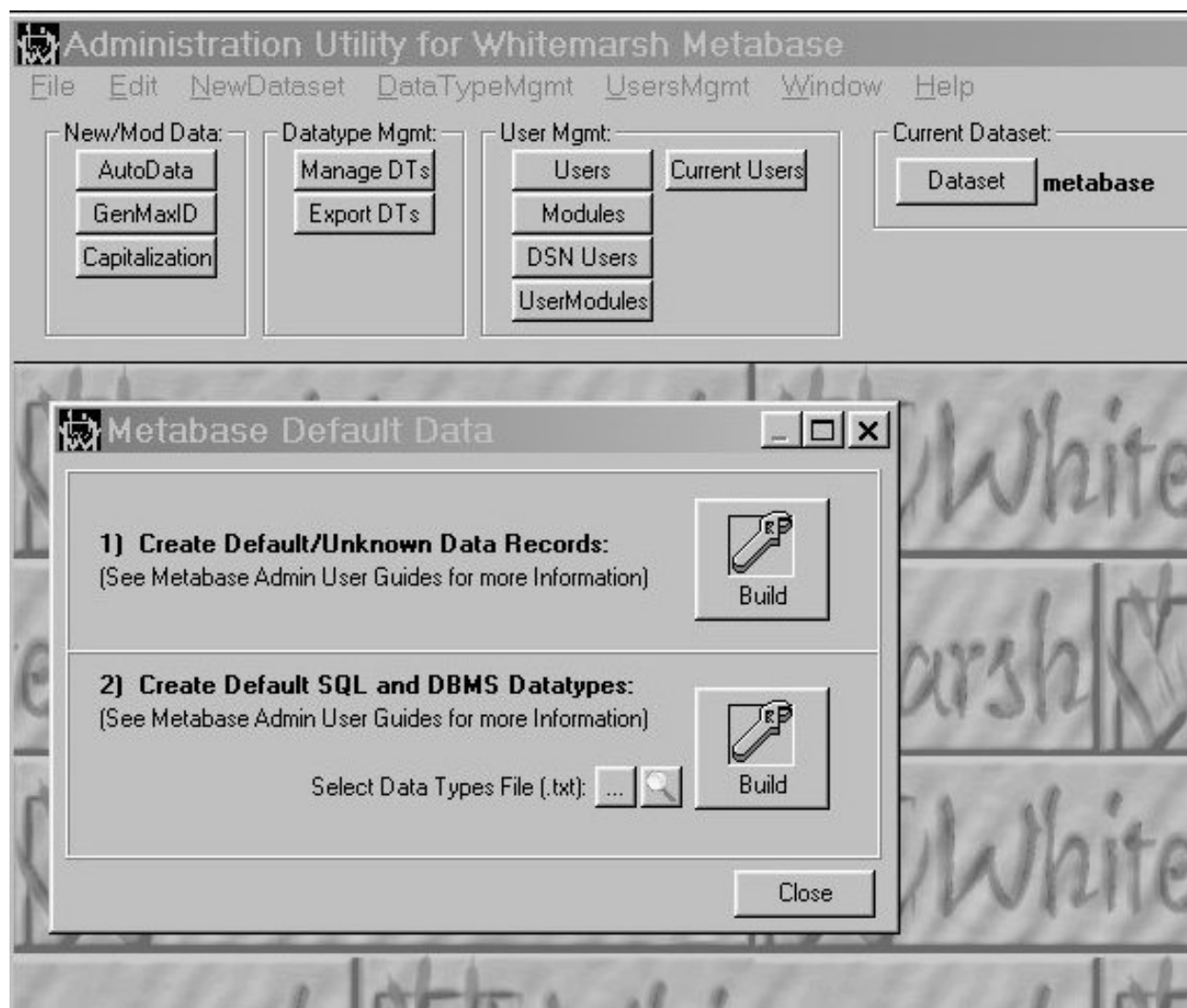


Figure 12. Auto Data Process.

2.2.1.1 Create Default/Unknown Data Records

The first process, Create Default/Unknown Data Records, creates the default data records that are requisite for a new Metabase Dataset. This consists of adding records with the value “Unknown” to most of the Metabase Dataset’s tables. If a dataset already has any number data records in it, then a message will pop up reminding the Metabase DBA that the default data process is for new data sets only.

Metabase database instances will never work properly if the default-data creation process has not been run. In short, this step is important, It is necessary. It is critical. If not done, then which ever values represent “1” and “2” in most of the meta entities will then be considered by the metabase system as the “null” or “unknown” values.



2.2.1.2 Create Default SQL and DBMS Datatypes

The second process, Create Default SQL and DBMS Datatypes, causes the creation of a set of reference data in three tables, Value Domain Data Types, SQL Data Types, and DBMS Data Types. The first is associated with the Data Element module. SQL Data Types are associated with the Implemented Data Model, and DBMS Data Types are associated with the Operational Data Model.

2.2.2 GenMaxID

The second button in the New/Mod Data group from Figure 11 is GenMaxID. This process creates a series of SQL statements that, when executed through an ODBC access tool

The Metabase System Getting Started guide should be consulted here because the process is different for each SQL DBMS. That is, the strategy for manipulating autogenerated primary key values is different between Mimer and MS SQL. Consequently, this button generates a different set of SQL statements for each different metabase compatible SQL DBMS.

The first step is to browse to select an output file for the generated SQL DDL statements. The second step is to run the routine that generates the SQL statements. The third step allows the viewing of the newly created statements. Figure 13 illustrated these steps.

2.2.3 Capitalization

The third button in the New/Mod Data group is labeled Capitalization. This brings up a window, see Figure 14, that contains four buttons. These routines format the various Metabase Data Modeler module's data. The non reversible formatting sets the CaPiTaLiZaTiOn to a CamelCaseFormat from a lower case format. The enhances the readability of the names in tables, columns, and elements in the Metabase. Pressing the various model's button will run this routine. These routines format the data records for attributes and entities in SDM, columns and tables in IDM, and DBMS columns and DBMS tables in ODM, and finally, elements in the DE module.



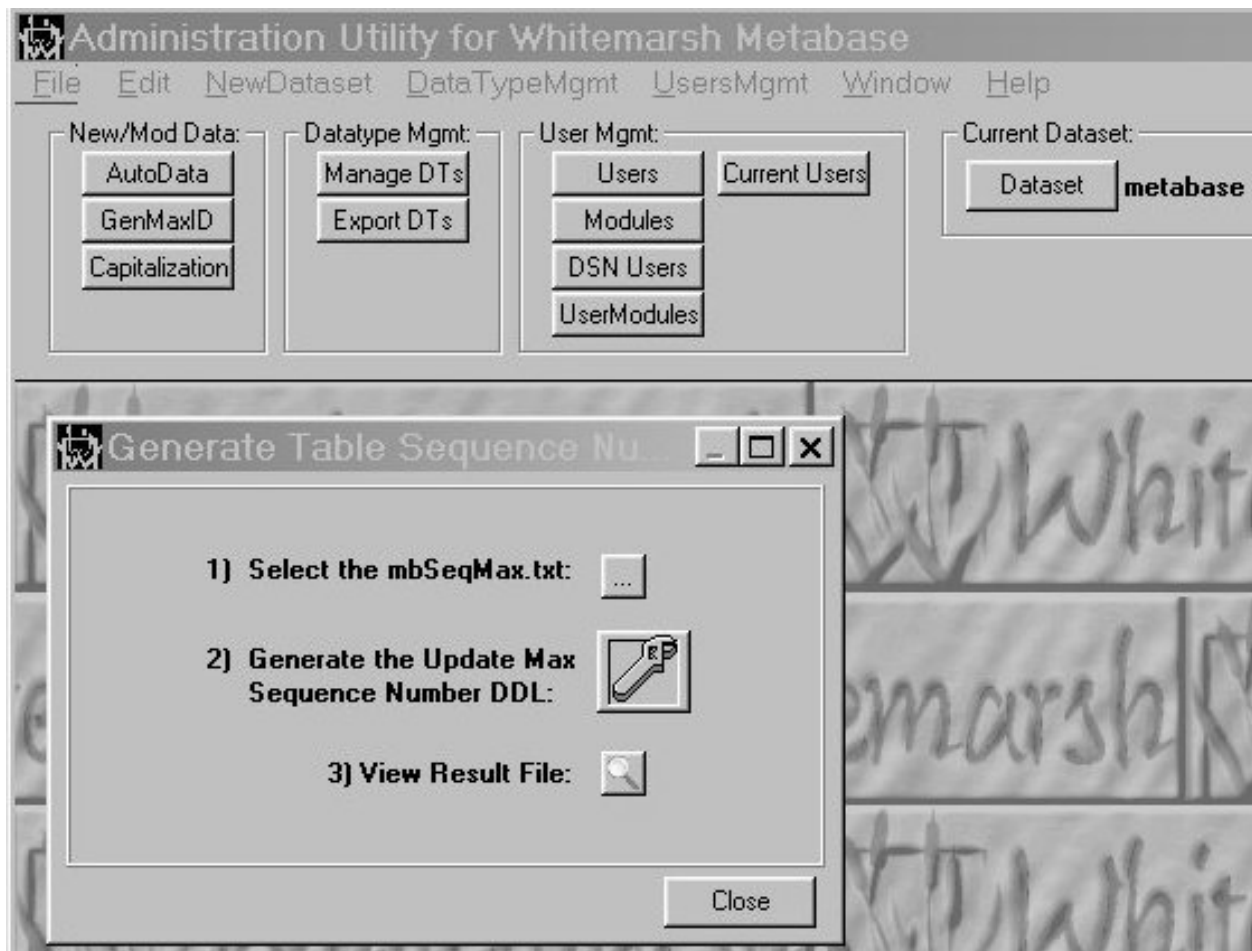


Figure 13. Steps in Generating Table Maximum Sequence Numbers.



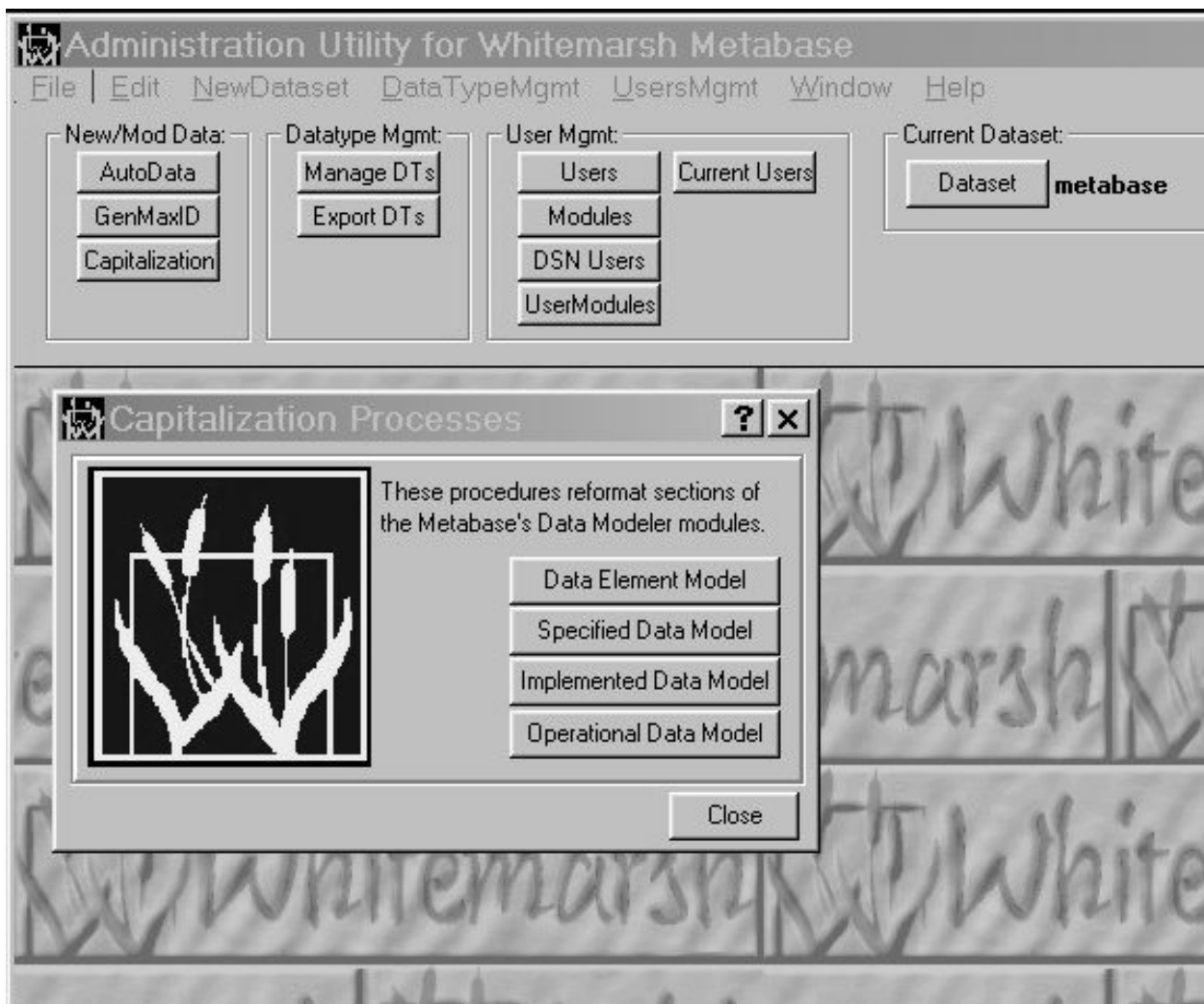


Figure 14. Capitalization process of metadata data models.



2.3 The Data Type Management Group

The Data Type Management Group, Datatype Mgmt, contains two buttons. These are ManageDTs (datatypes) and ExportDTs.

2.3.1 Manage Data Types

Pressing the ManageDTs button brings up a window, as illustrated in Figure 15, that will allow the Metabase DBA consolidated access to the IDM's, ODM's, and VDM's datatypes. At the bottom of Figure 15 there is also two data type reassignment processes. These allow the Metabase DBA to reassign the relationships between datatypes among the various Metabase modules.

2.3.1.1 Value Domain Data Types

Figure 16 presents the list of Value Domain Data Types. To add, change, or delete a Value Domain Data Type press the appropriate process button and create the Value Domain Data Type. The update process is shown in Figure 17.

2.3.1.2 SQL Data Types

Figure 18 presents the list of SQL Data Types. To add, change, or delete a SQL Data Type press the appropriate process button and create the SQL Data Type. The update process is shown in Figure 19.

2.3.1.3 DBMS Data Types

Figure 20 presents the list of Value Domain Data Types. To add, change, or delete a Value Domain Data Type press the appropriate process button and create the Value Domain Data Type. The update process is shown in Figure 21.

2.3.1.4 SQL Data Type to Value Domain Data Type Reassignment

The process of reassigning a SQL Data Type to a different Value Domain Data Type is illustrated in Figure 22. Tag one or more SQL Data Types and tag one Value Domain Data Type. Then press the reassign button.



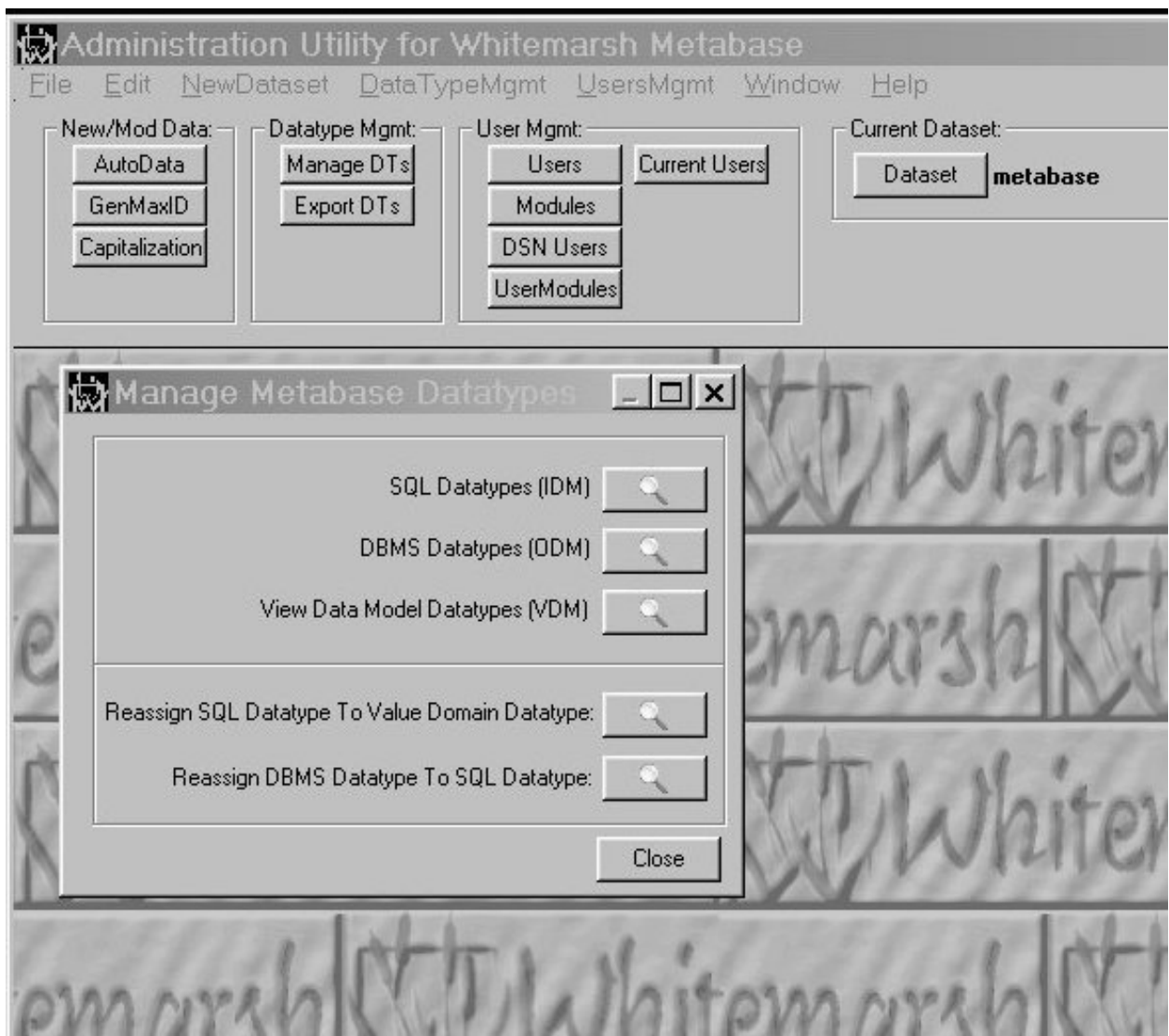


Figure 15. Data Type management processes.



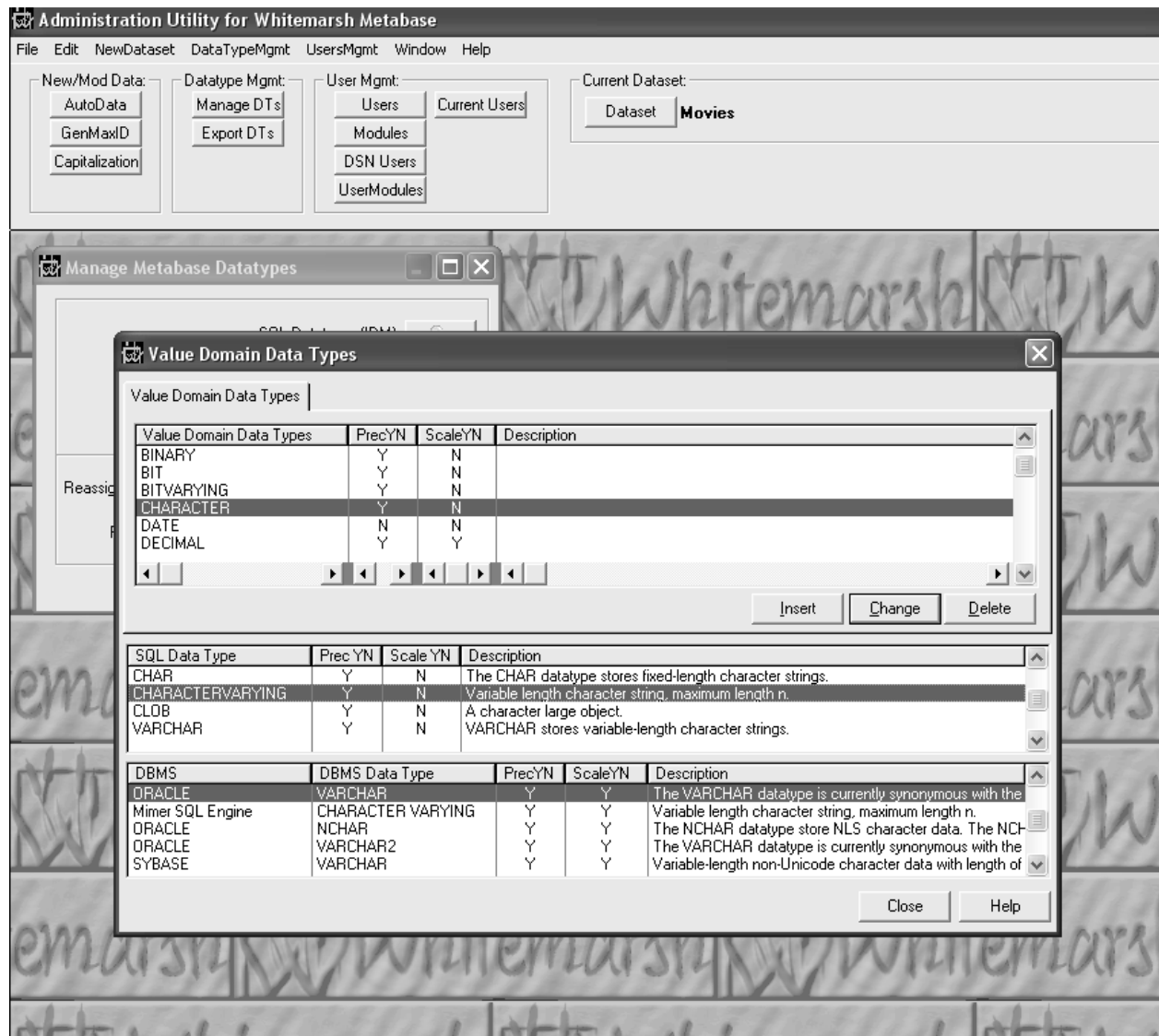


Figure 16. Value Domain Data Types.



2.3.1.5 DBMS Data Type to SQL Data Types

The process of reassigning a DBMS Data Type to a different SQL Data Type is illustrated in Figure 23. Tag one or more DBMS Data Types and tag one SQL Data Type. Then press the reassign button.

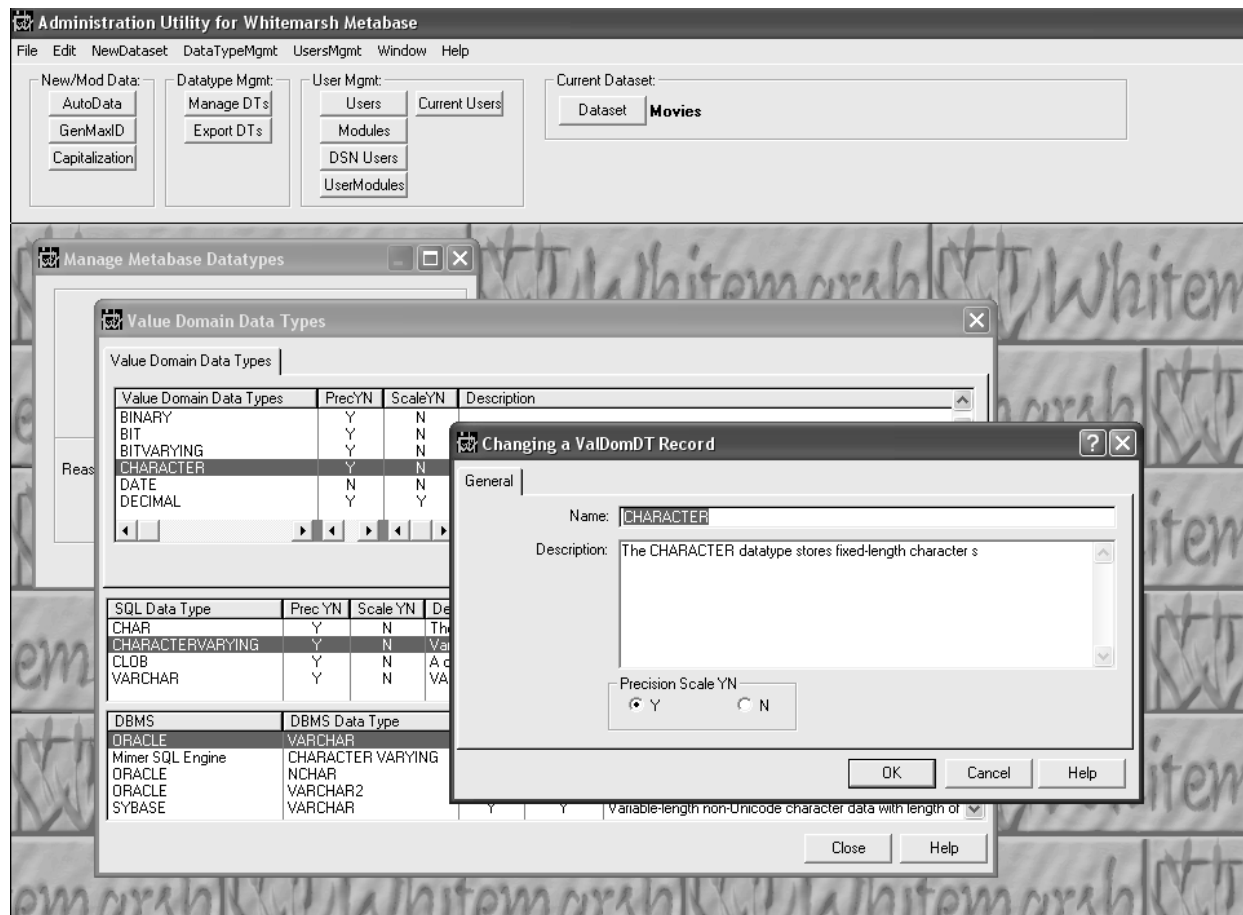


Figure 17. Value Domain Data Type update screen.



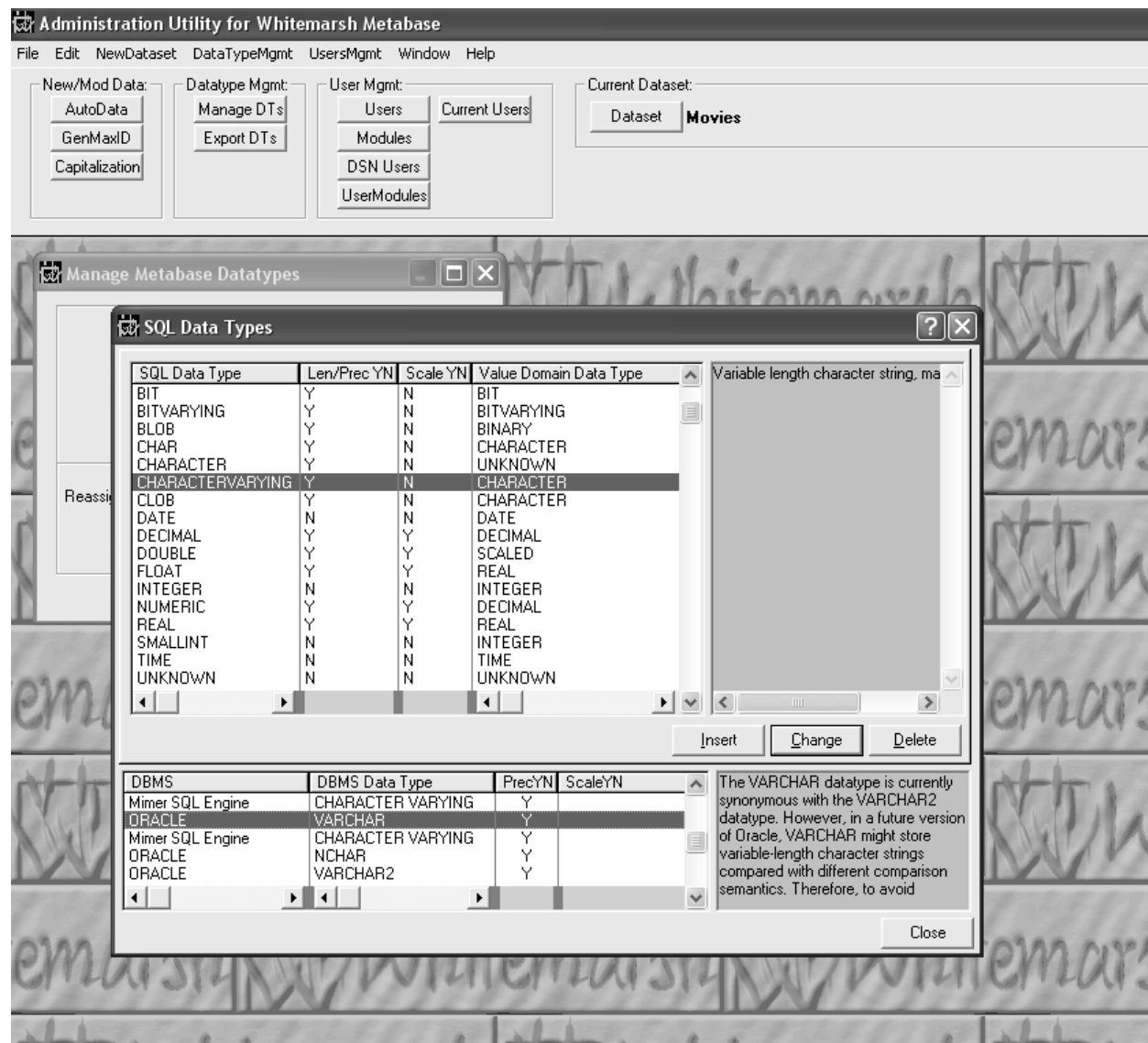


Figure 18. SQL Data Type list.



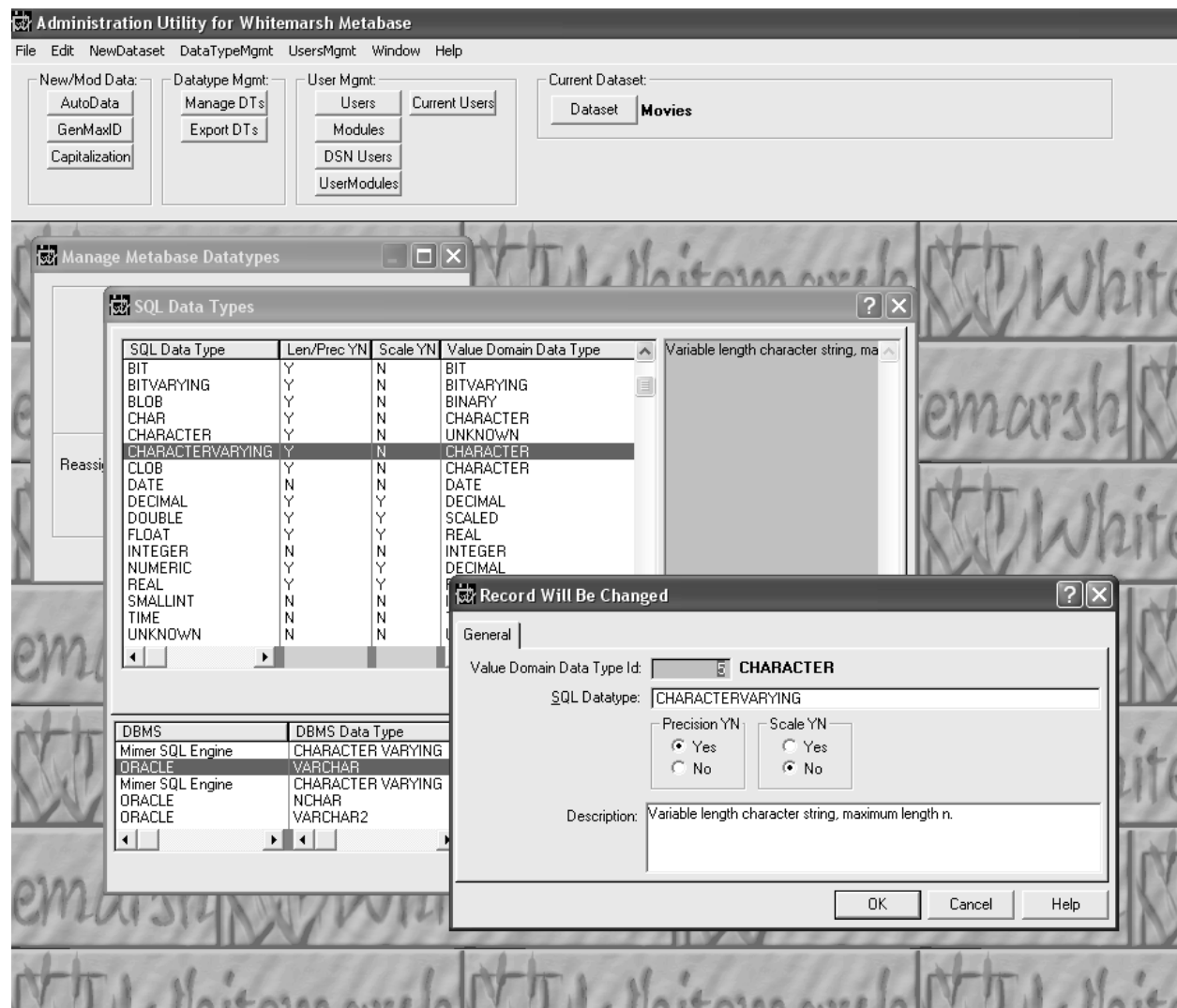


Figure 19. SQL Data Type update screen.



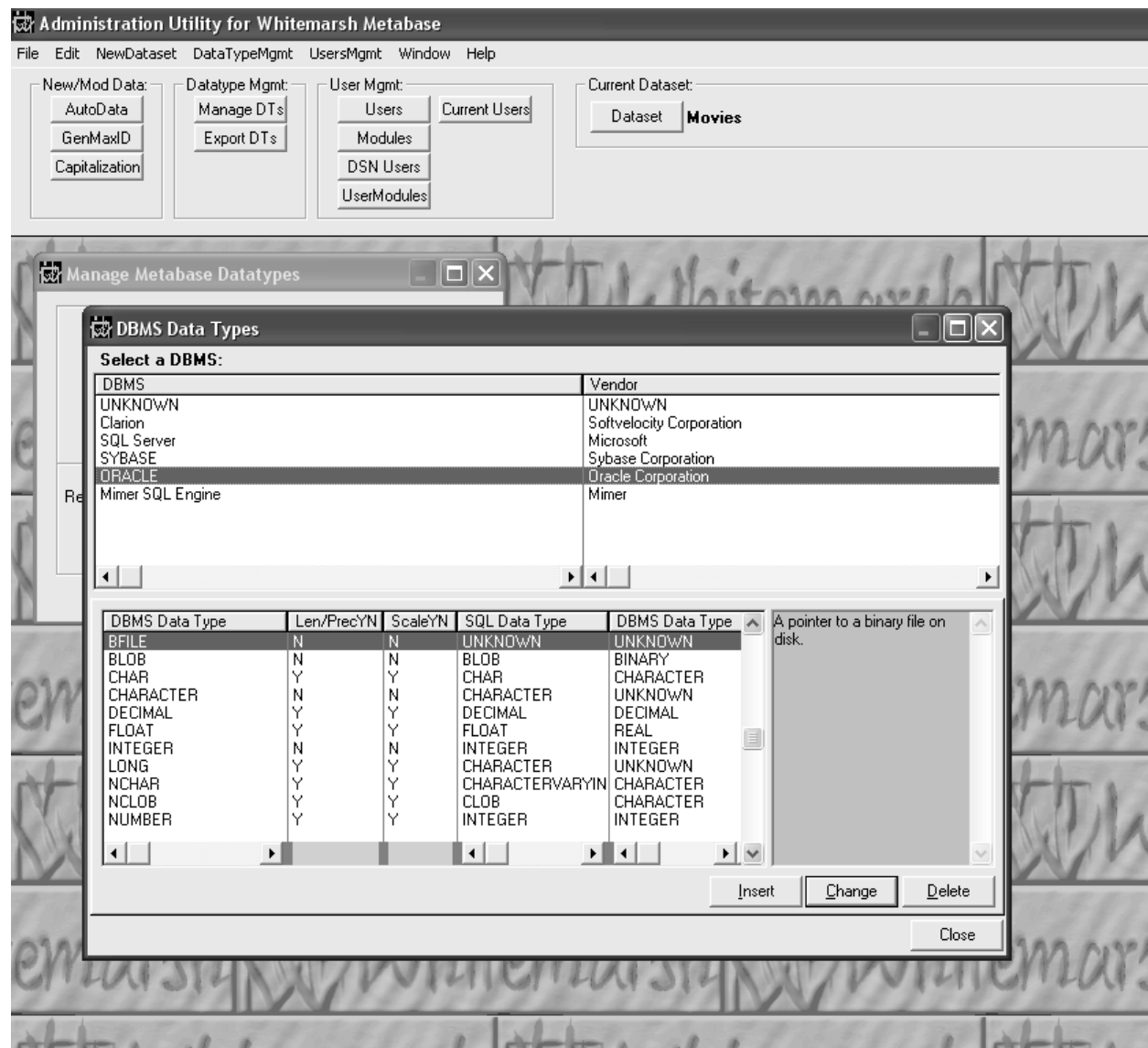


Figure 20. DBMS Data Type list.



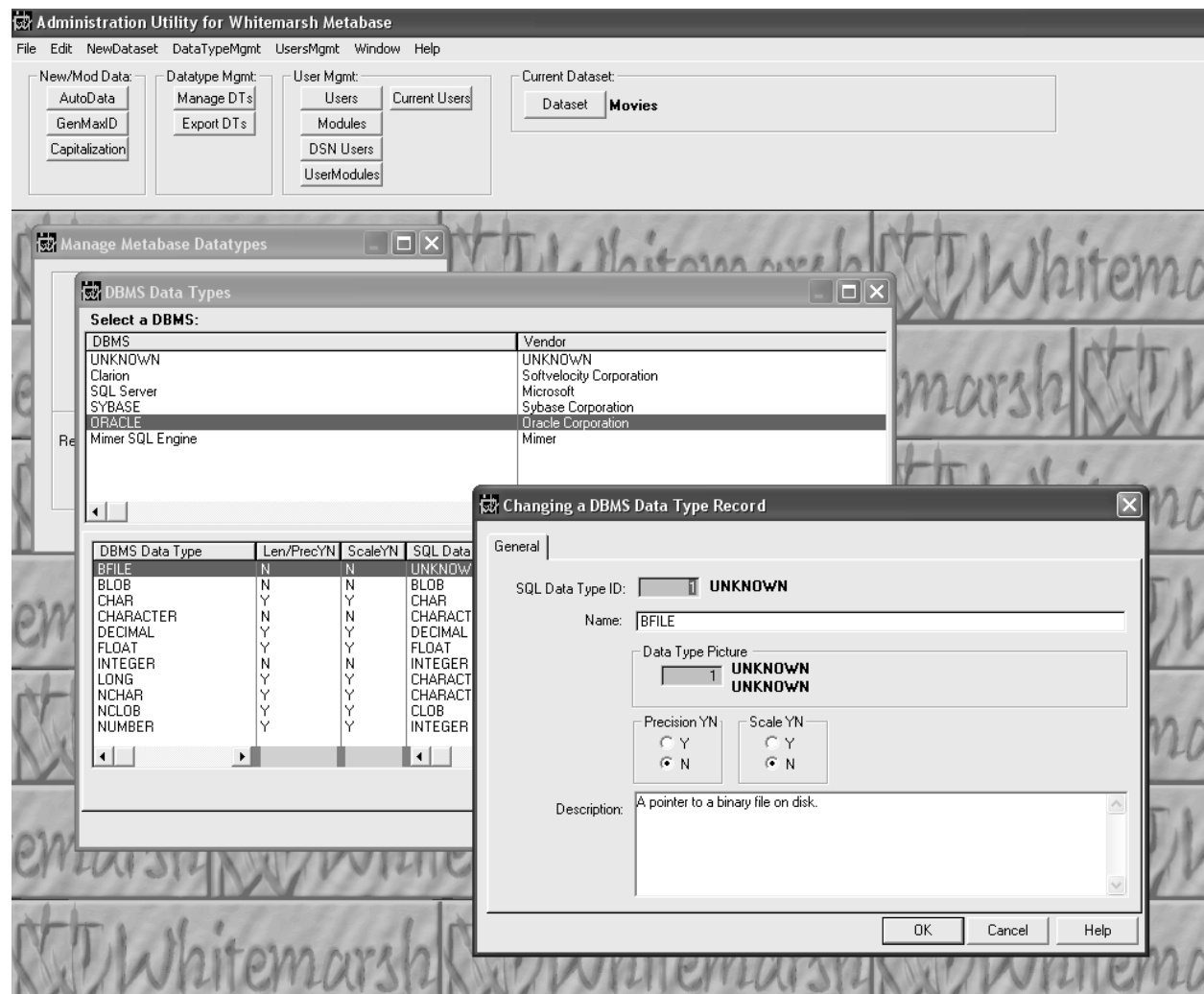


Figure 21. DBMS Data Type update screen.



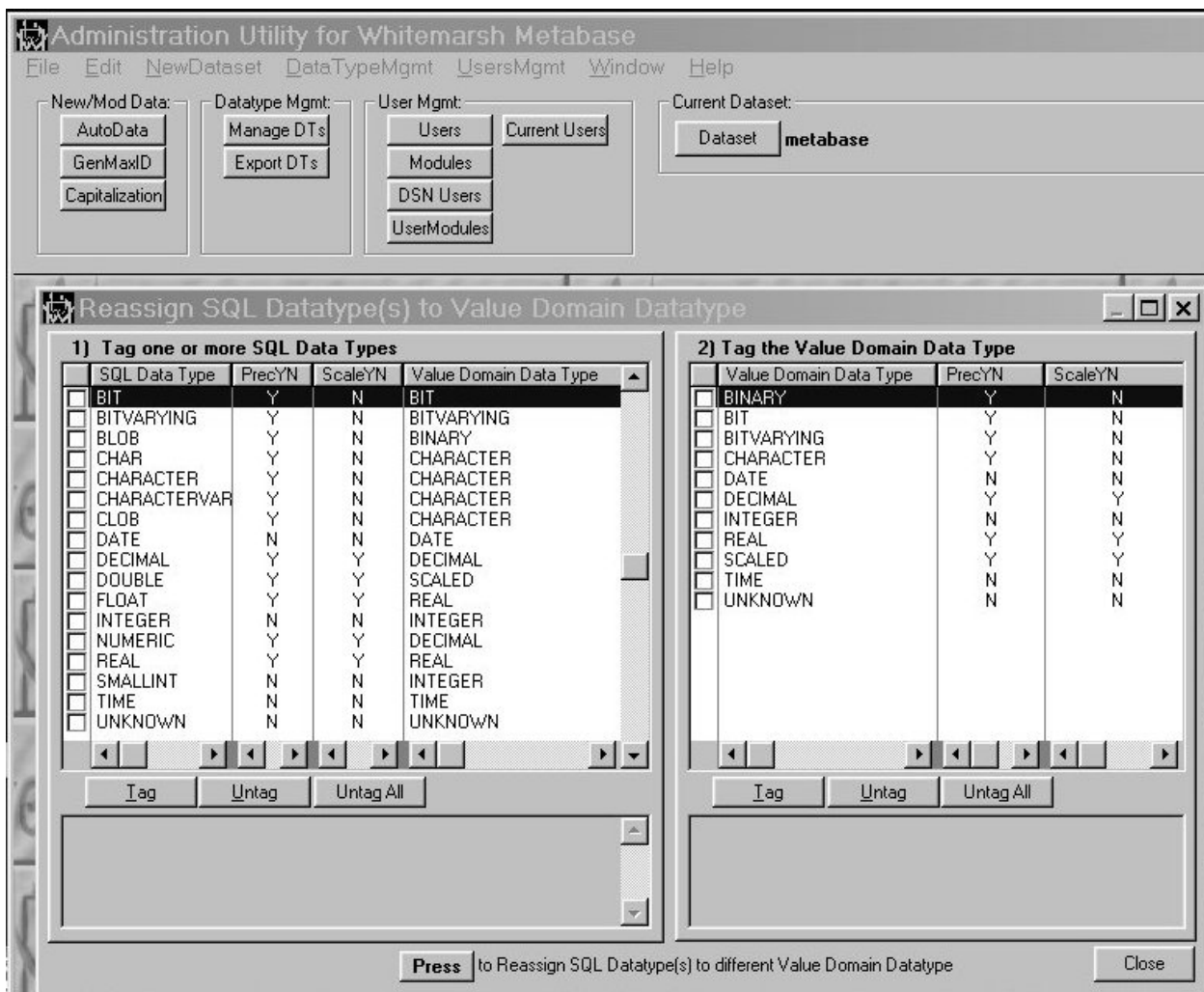


Figure 22. Reassignment of SQL Data Types to Value Domain Data Types.



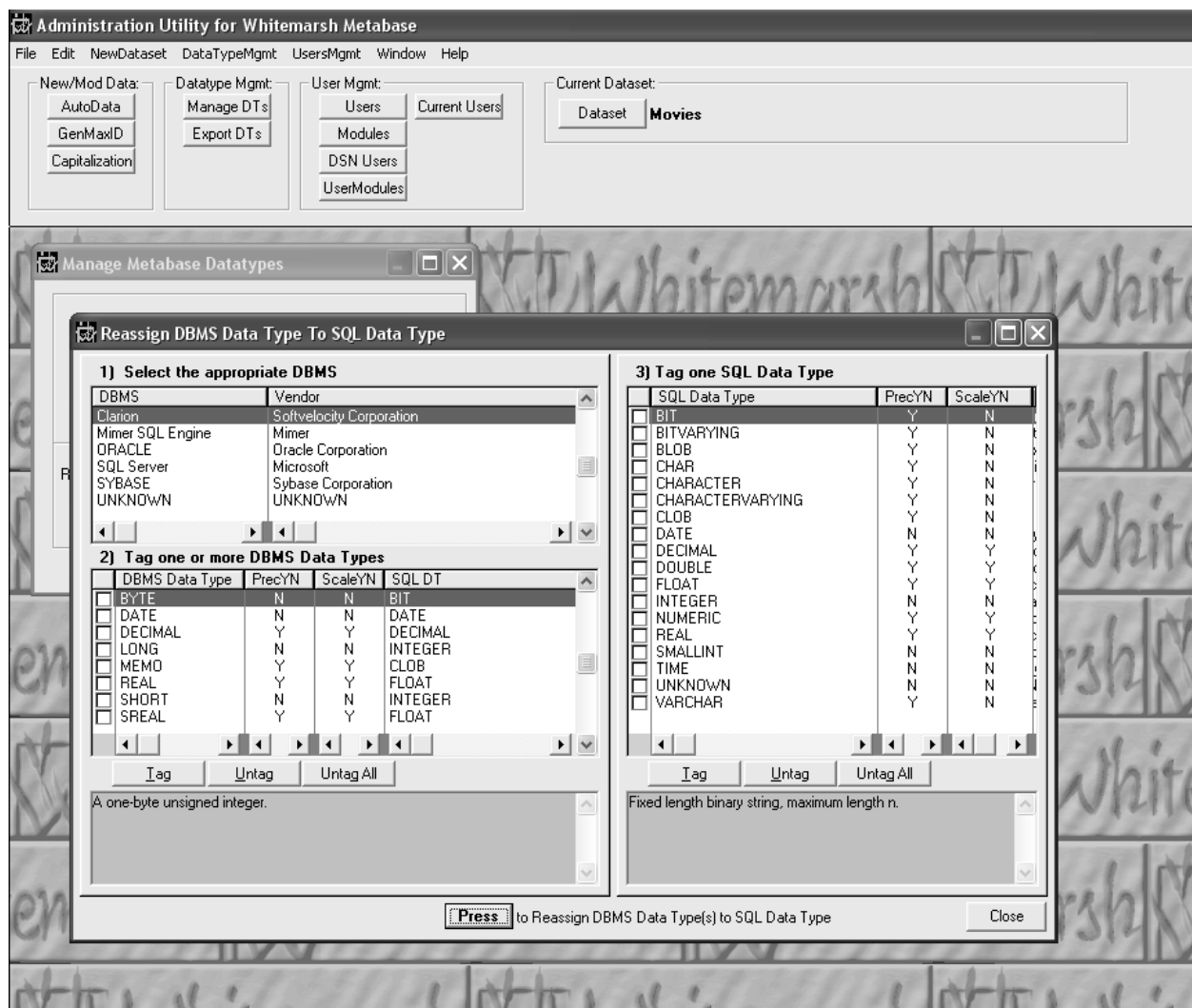


Figure 23. Reassigning DBMS Data Types to SQL Data Types.



2.3.2 Export Data Types

The second button in the Datatype Mgmt group is labeled ExportDTs. This process, illustrated in Figure 24 brings up a window that allows the Metabase DBA to export to a comma delimited the datatypes for a Metabase dataset. This is useful when a new dataset is created and a like set of datatypes are to be used. Used in conjunction with the import datatypes functionality, this can save the Metabase DBA time and decreases the risk of inaccurate input when working with multiple datasets.

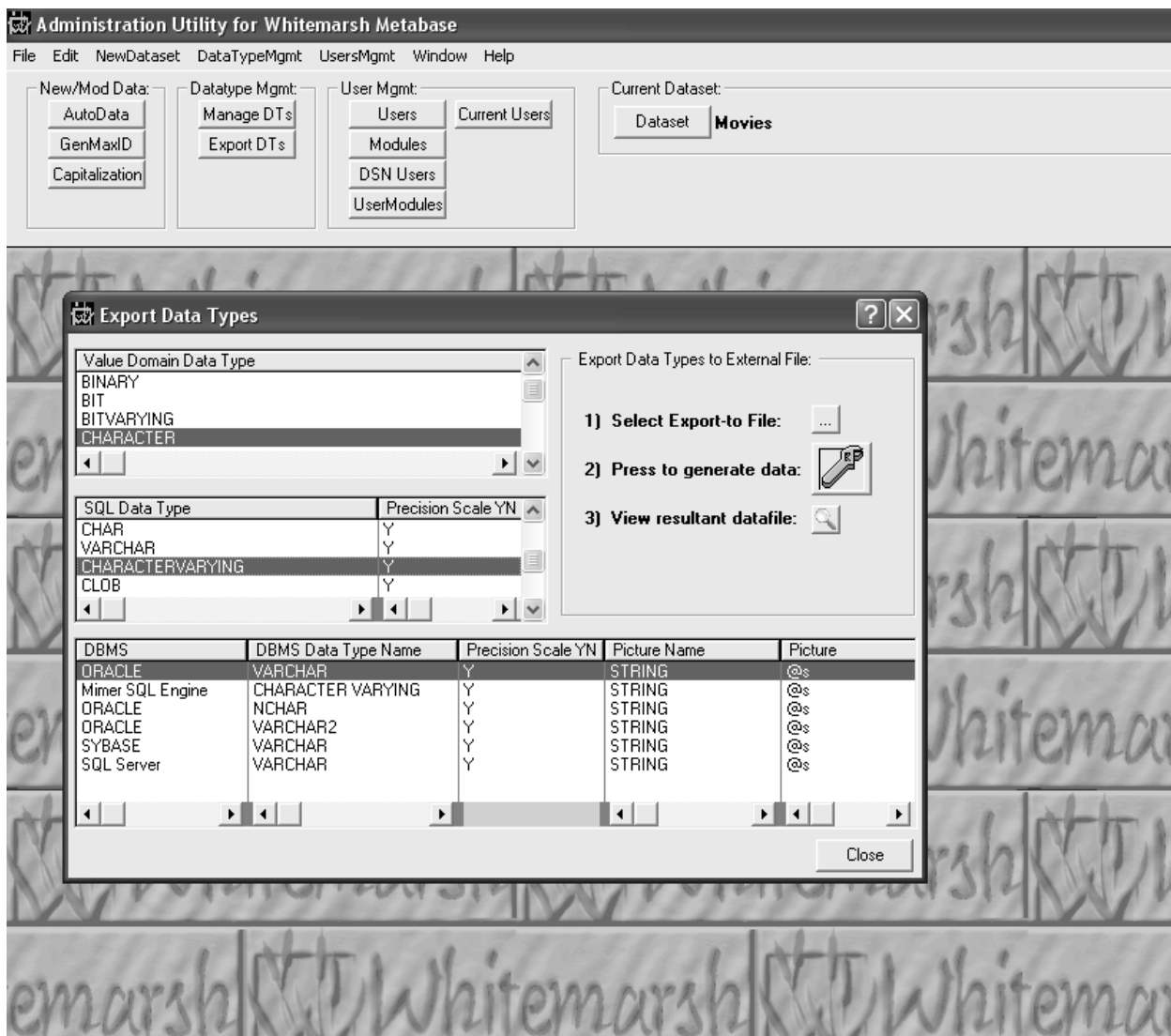


Figure 24. Export Data Types process.



2.4 The User Management Group

The User Management group controls the access of Metabase Users and the modules they can use with specified datasets. Fundamentally the strategy is this. There are users. There are metabase modules. There are metabase database instances, that is, data sets. User management consists of assigning certain users to certain data sets. Then assigning certain data set users to certain metabase modules. When a user activates a metabase functional module, say, Database Objects, selects the SQL DBMS, and then signs on with their user name and password, the metabase system accesses the metabase system administrators database (MbSysDB) and finds for that user the data sets that have been allocated to that user. If none have been allocated the select DSN screen, as presented in Figure 9 is empty.

2.4.1 Users

The Users button allows the add/modify/deleting of allowed Metabase users. The users are shown in Figure 25. The Modules button lists the Metabase Modules that are installed into the Metabase Software System. Adding, changing, and deleting users is accomplished through the update buttons. The information required for a user is presented in Figure 26. The email address is requested so that you can send the user a message through normal email processes. Users can, however be posted messages by the metabase administrator directly because the metabase administrator knows, at all times, the complete set of current metabase users. See Section 2.3.4 below.

2.4.2 DSN Users

The DSN Users button controls the allocation of specified users to access specific datasets (Data source nodes) when using the Metabase Software. The data sets are shown in the upper left browse. The users are shown in the upper right browse. DSN users are created by tagging one data set and then one or more users. The press the Build button. The resultant DSN users are shown in the bottom browse of Figure 27. DSN users do not yet have access to any specific metabase module. This is now done in the User Modules step.

NB: The creation of DSN Users DOES NOT in any way set/create security restrictions on the backend DBMS that is being accessed via ODBC. Backend SQL security that would control, for example access to an ODBC utility or to a report writer such as Crystal Reports must be done by a database administrator.



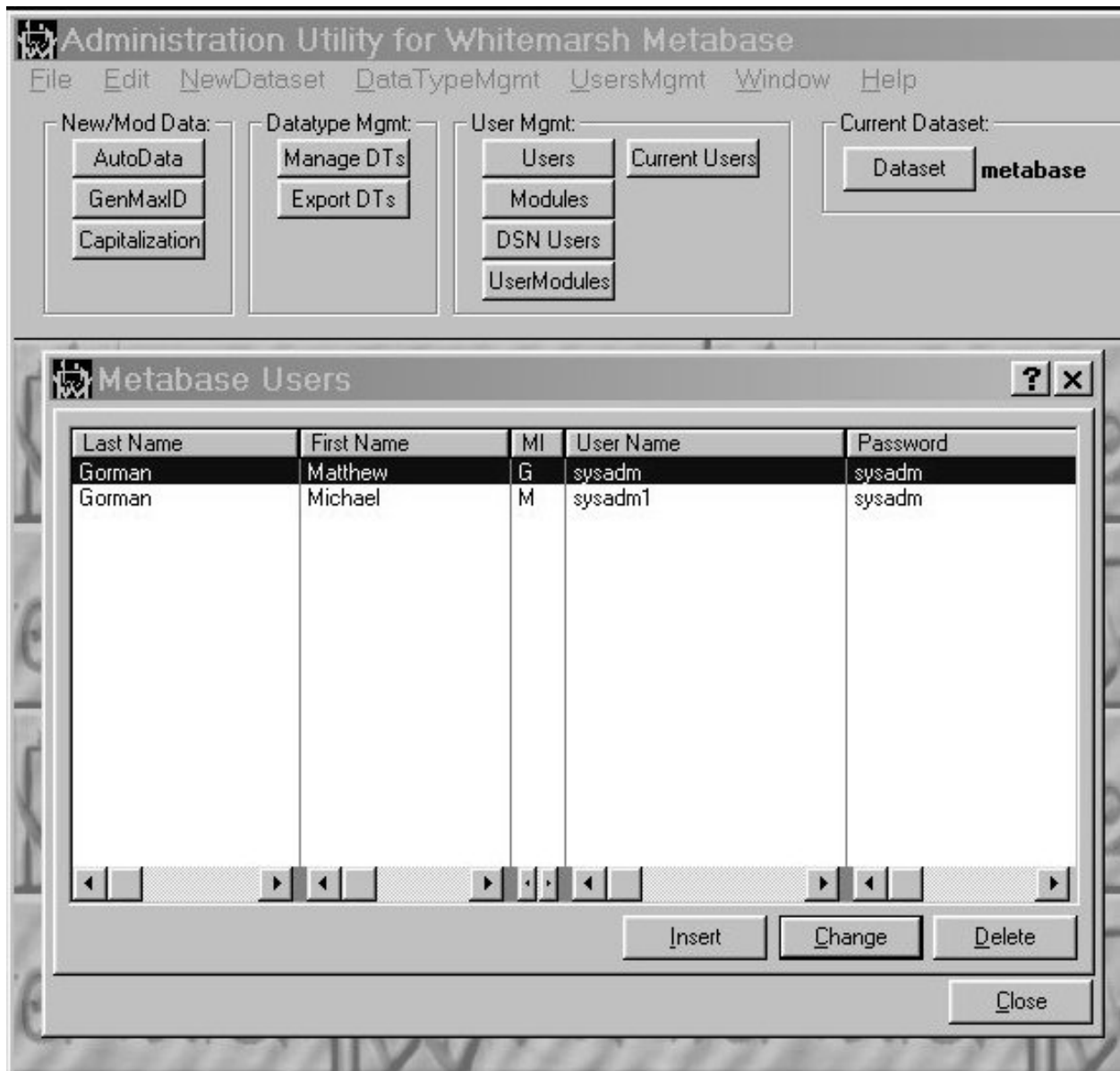


Figure 25. List of Metabase users.



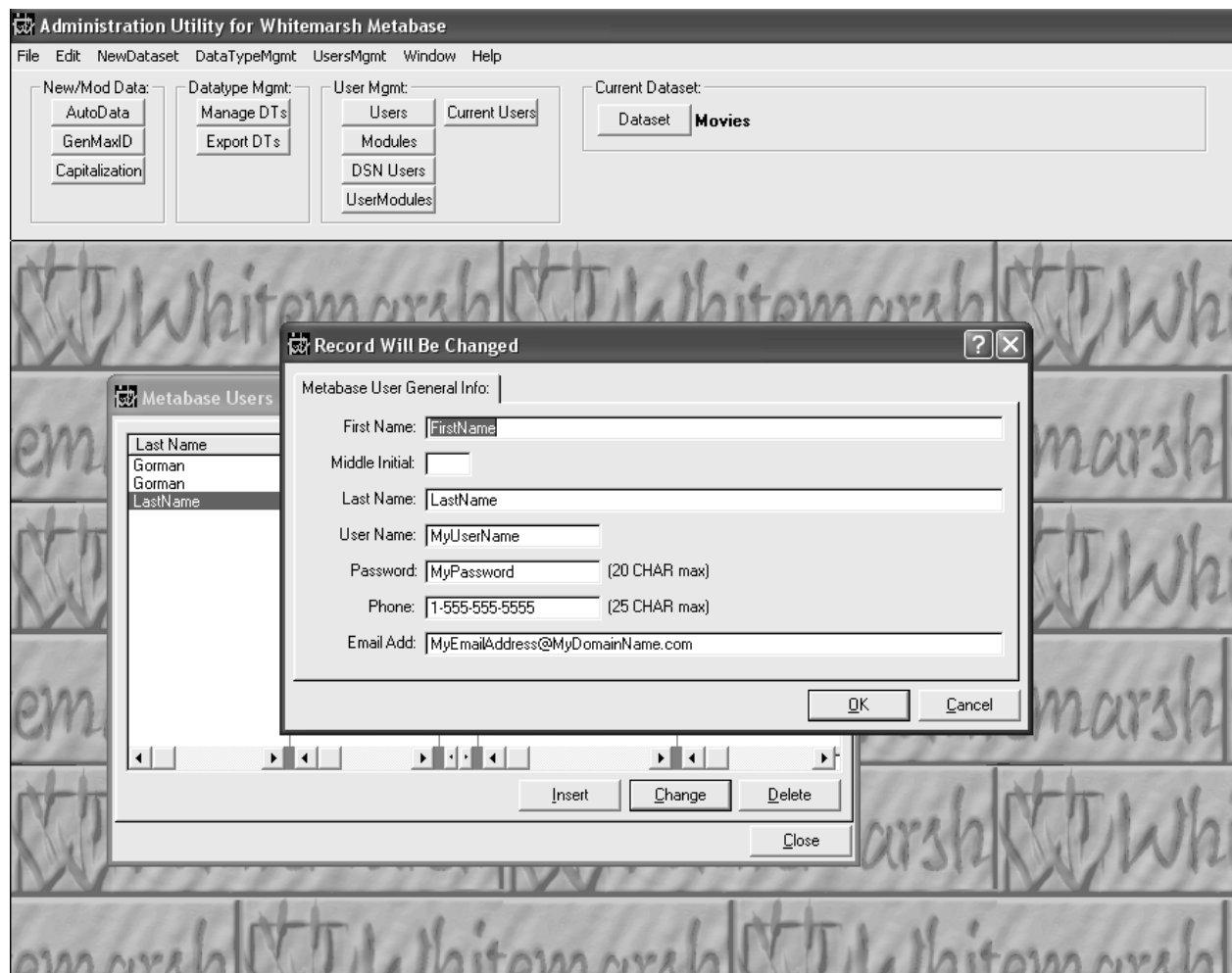


Figure 26. Metabase User update screen.



2.4.3 User Modules

The User Modules button, takes this access a step further and allows the DSN User to use a specific Metabase Software module on a specific dataset. E.g. After adding a specific user to the Users section, the Metabase DBA creates a DSN User record that allows access to the specified DSN/Metabase Dataset. Then, after giving the user general access to that DSN/Metabase dataset, the metabase administrator creates one or more DSN-user-module records that allows the DSN User access to specific Metabase Software Modules.

This user module step is illustrated in Figure 28. The set of DSN Users, that is, users allocated to data sets is shown in the upper left browse. The set of all metabase functional modules is shown in the upper right browse. Tag one DSN user and then as many metabase modules as appropriate. Press the Build button. The resultant DSN metabase module user records then appear in the bottom browse.

2.4.4 Current Users, Messages, and Auto-Logout

The Current Users button lists the Metabase Software users that are currently logged into the Metabase System, as shown in Figure 29. This window also allows the Metabase DBA to Flag a selected user, either merely sending the user a message, or additionally logging that user out after that user receives that message. Figure 30 shows the message composition window.

Users receive messages in the following manner. Every four or so minutes, each active metabase module “calls back” to the Metabase System’s database and does an automatic re-login. If at that time a message is waiting for that user then it is displayed. In this way, the metabase system administrator can be assured that every current user has received the message. If the message is a log-out message, the user is automatically prevented from re-logging in.

The metabase administrator removes a user from accessing a metabase module by deleting one or more of the User-DSN-Module records. Then the metabase administrator can remove one or more User-DSN records. Finally the metabase administrator can remove one or more user records.



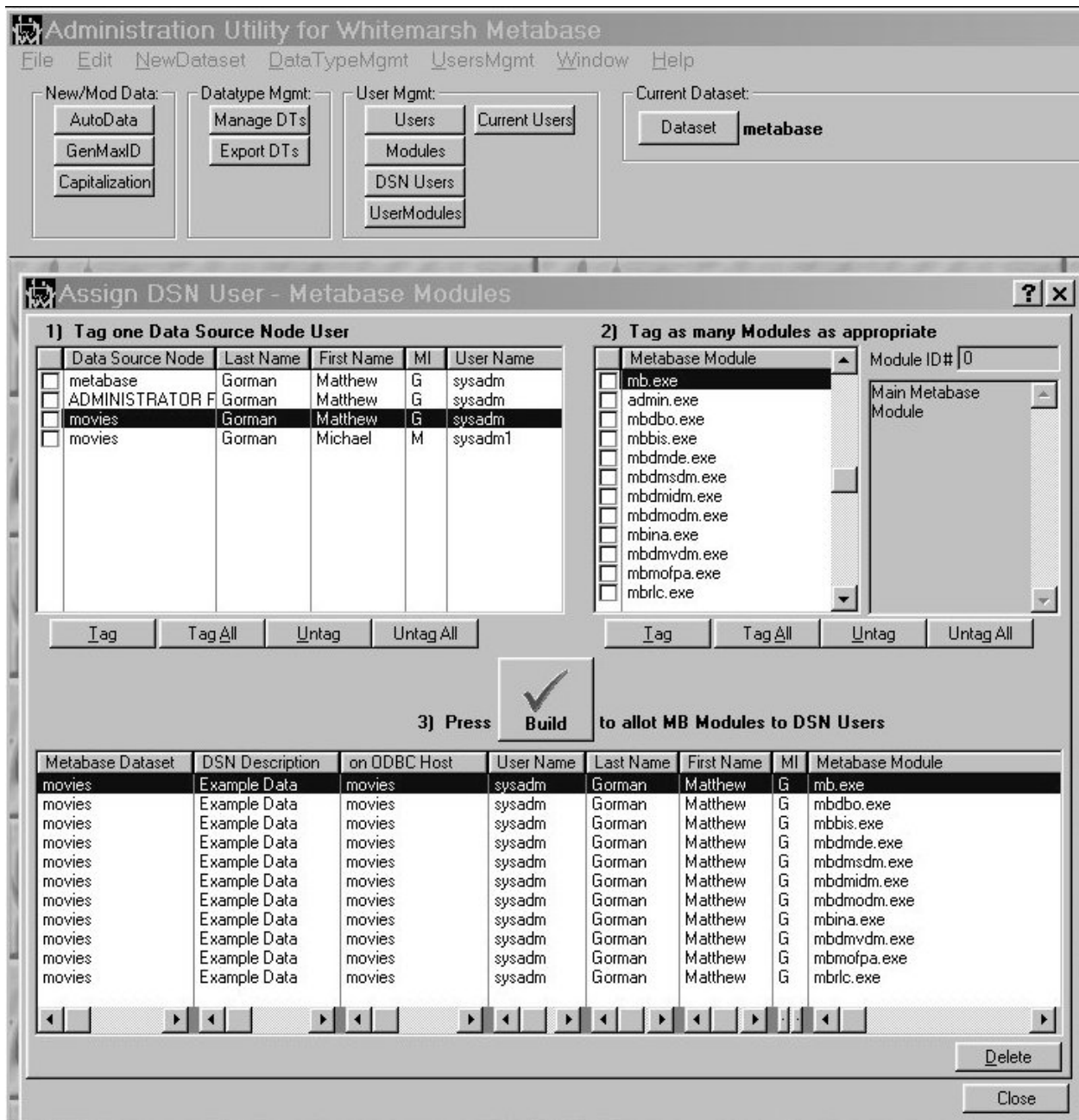


Figure 27. Allocating DSN Users to Metabase Modules.



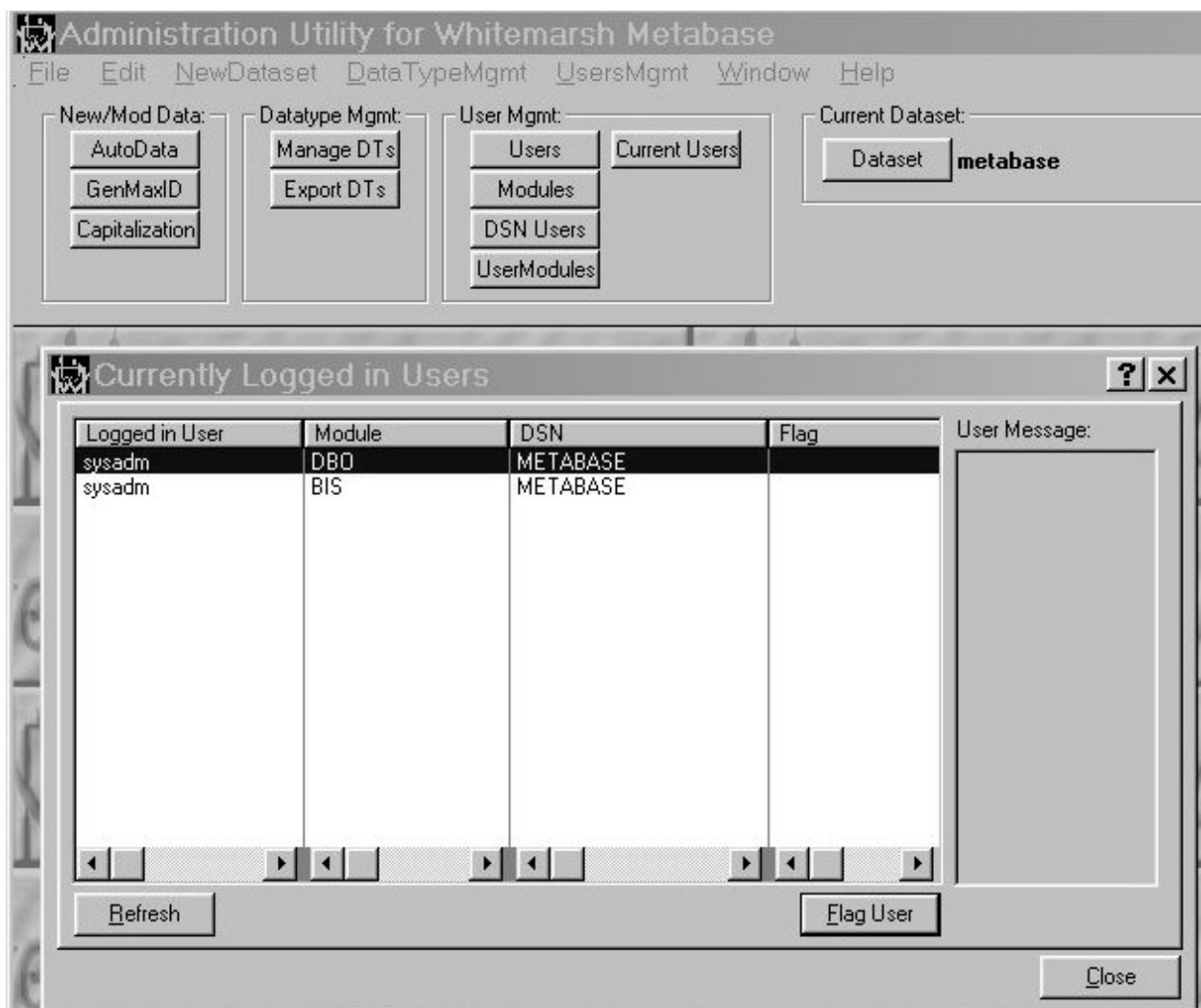


Figure 28. Current Metabase Users.



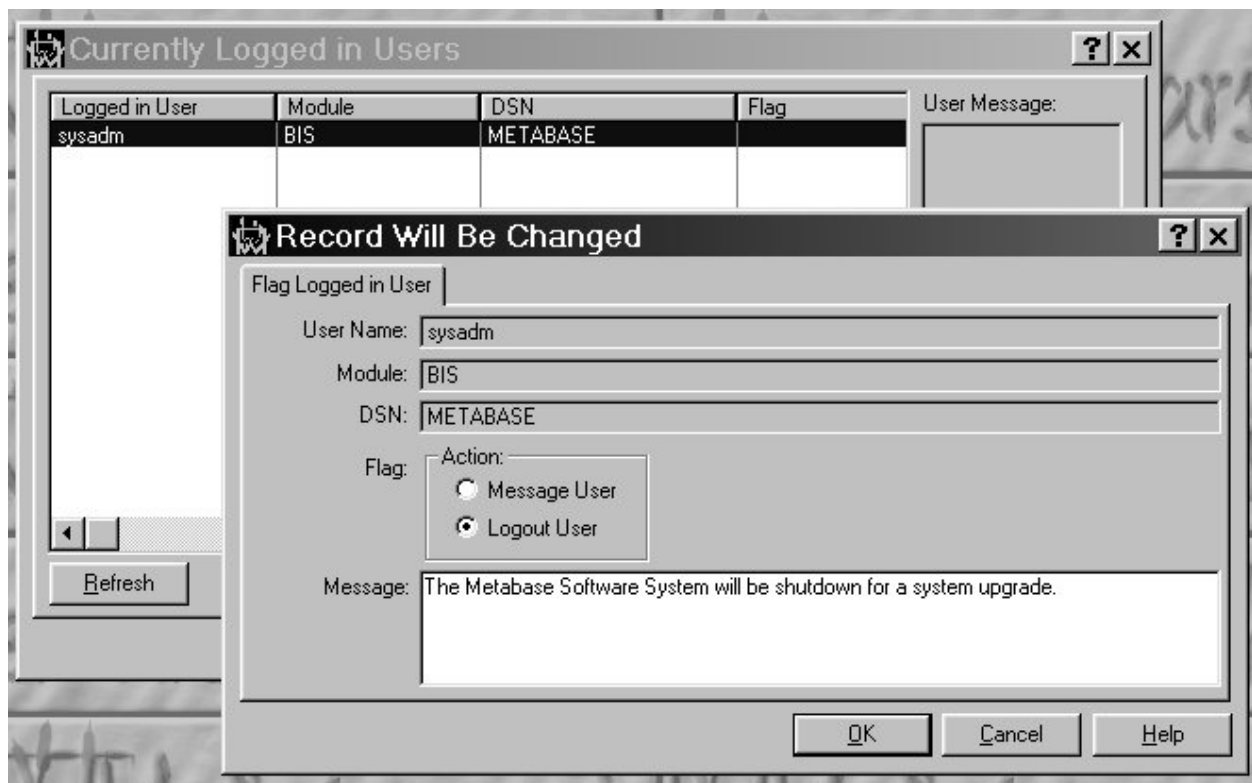


Figure 29. Metabase Administrator Message to Current User update screen.



2.5 Abbreviation Management

In data models, abbreviations are often employed to make names short to conform to the limitations of the DBMS. The Metabase supports a comprehensive approach to abbreviations and their management.

From Figure 31, there are four abbreviation management function. Each is presented in a separate section. Collectively these four functions enable an effective way to define, import, manage, separate by business function, and “mine” the metabase itself for abbreviations.

The actual use of abbreviations is within the Operational Data model when SQL data definition language scripts are created. Different SQL DBMSs have different restrictions on lengths. The strategy contain in the Administration module enables comprehensive management of the abbreviations that are employed.

SQL data definition scripts can be imported into the metabase in any of the three modules: Specified, Implemented, and Operational data models. The scripts may be full of abbreviated column names. Because of this, if there has been a comprehensive creation of abbreviated terms, then the process, UnAbbrev can be employes in the SQL DDL import processes to transform abbreviations back to English versions.



Figure 30. Abbreviation management functions.



2.5.1 Abbreviation Generation

The abbreviation generation screen is shown in Figure 32. This screen has two browse. The top is for business domains, and the bottom is for the abbreviation currently assigned within that business domain. This specific window shows that the abbreviations are all assigned to the Unknown business domain. Unknown here means, none specific. Individual or collections of abbreviations can be assigned to different business domain. For example, SOB, that is, Shortness of Breath might be assigned to the Health business domain.

The bottom browse has three columns. The first column contains the source word about which an abbreviation is created. For example, accommodation. The second column contains the currently accepted abbreviation for accommodation. In this example, it is, accm. The third column is the generated abbreviation. It is, acmdtn. This abbreviation resulted by removing all



Figure 31. Abbreviation Generation screen.



vowels and then removing adjacent consonants.

At the bottom of Figure 32 are five different buttons. These buttons are:

- Generate Abbreviations
- Generate All Abbreviations
- Make Generated Abbreviations Standard Abbreviations
- Recast Multi-Word Abbreviations to Single Words
- Lower Case all Abbreviations

Before describing any of the buttons, the Business Domain must be selected except for the Generate All Abbreviations button. If the Business Domain is not selected then an error message is presented and the operation stops.

The Generate Abbreviation button will generate a single abbreviation for the Selected Abbreviation Full Word. The process is simple. All vowels (a, e, i, o, u, and y) are removed starting from the second letter. All double consonants are changed to single consonants (mm becomes m). This strategy has very common application and was described in computer science literature almost 50 years ago. See for example, Abbreviating Words Systematically by June A Barrett, Mandalay Grems, in the Communications of the ACM (Volume 3 , Issue 5 (May 1960)) or A Study of Methods for Systematically Abbreviating English Words and Names by Charles Bourne, and Donald F. Ford. Volume 8 , Issue 4 (October 1961).

Using techniques that “appeal to common sense” results in the transformation, for example, of Accommodation to “Acmdtn.” It also results in “Mississippi” being transformed to Msp. One result is that abbreviations are not unique. That has ramifications in the “UnAbbreviation” process that is also supported in the Metabase.

If a given generated abbreviation is not satisfactory, then just press the Change button and enter the Abbreviation. For example, you might make the abbreviation for “Accommodation,” Accm. The authors of the Metabase recommend that you employ the generated abbreviation if at all possible because it follows a well established strategy that also has roots in computer science literature.

The Generate All Abbreviations button will generate abbreviations for all the terms across all the business domains.

The Make Generated Abbreviation Standard Abbreviation will take the selected Generated Abbreviation and copy it over into the Abbreviation column.

The Recast MultiWord Abbreviation to Single Words will take all instances of multi-word abbreviations and separate them into single words and cause abbreviations. This way, if there is a DBMS column that contains each of these abbreviated words then they will be abbreviated one at a time.

The final button, Lower Case all Words is employed to make all words lower case. This increases consistency throughout.

The three buttons, Insert, Change, and Delete perform their normal functions. The update screen is shown in Figure 33. In this screen the columns that are available for update are the full word, accommodation, the abbreviation, and the description. The Business Domain Id is not able to be updated because that would be a “Reassignment” process that is accomplished by a different Abbreviation process function. The generated abbreviation is also not available for update because its value is computed automatically. A quick look at the definition might leave



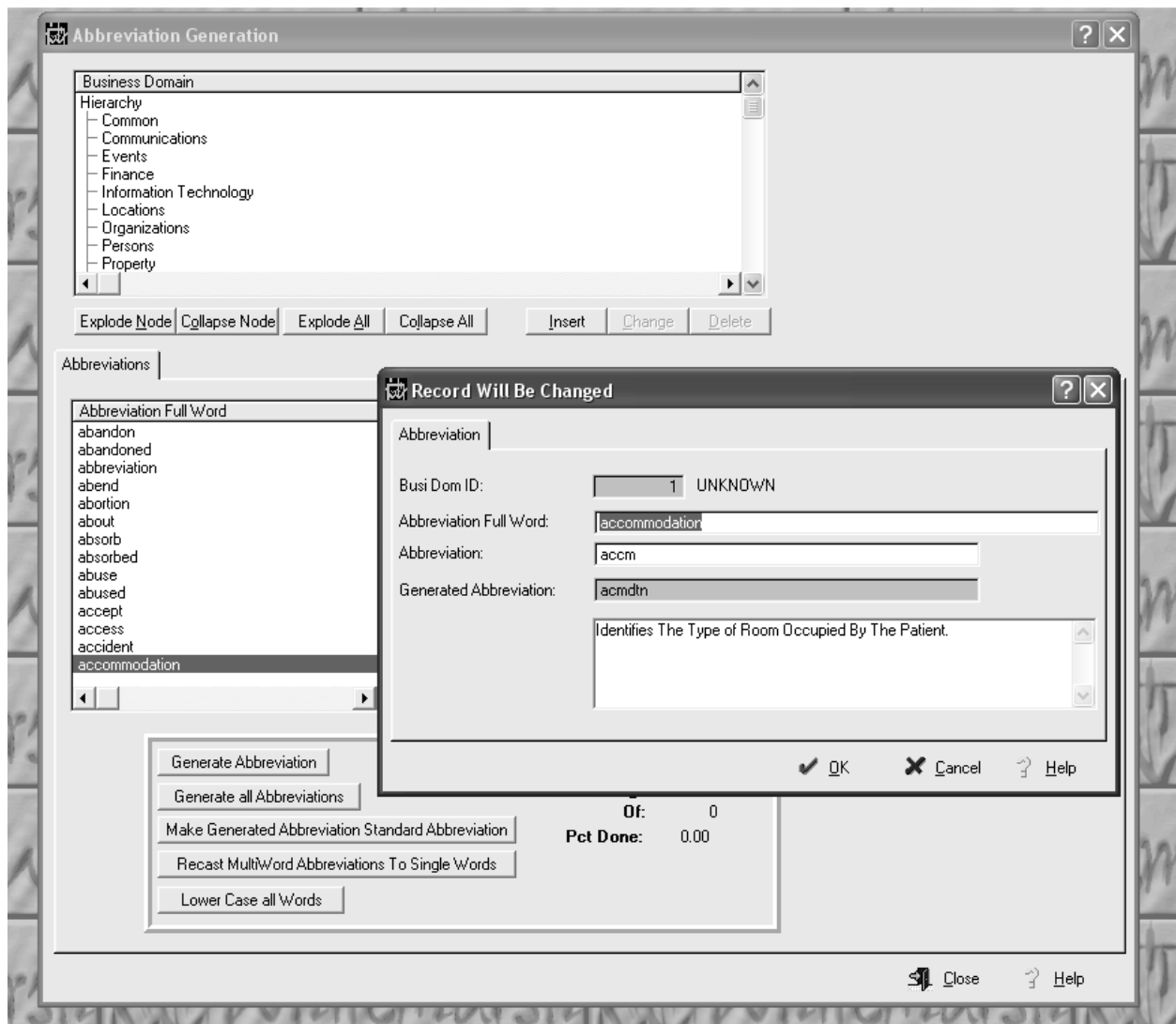


Figure 32. Abbreviation update screen.

you in wonderment. Where did such an description of Accommodation come from. The answer is regrettably simple. Someone designed a “stove-pipe” database that was related to the medical or insurance industry and needed a term for the patient’s room type. Hence the word and its definition. The problem with this definition is that the term might be employed in many other DBMS column names, such as “Accommodation Reached” or some other use. Three solutions come to mind. First, under the Abbreviation Management process of allocating abbreviations to specific domain, move this one to Health Insurance. Second, change the definition to something more generic. Or three, leave the definition blank because this is NOT the place for definitions, rather it’s the place for abbreviating words.



2.5.2 Import Abbreviations

This function is very important. Getting abbreviations into the metabase can be a very laborious task. Entity Relationship modeling tools such as Erwin have the capability of creating an Abbreviations file. This file can be imported into the metabase by first converting the file to a column based ASCII table, mapping the fields to the Metabase's two columns of the Full term, Abbreviation, and Abbreviation description (if any exist). Figure 33 shows the first screen in this process. In general the process consists of:

- Starting the process by pressing the Import Abbreviations Button
- Selecting the file to import
- Selecting the duplicate term options
- Mapping the import file's fields to the database fields

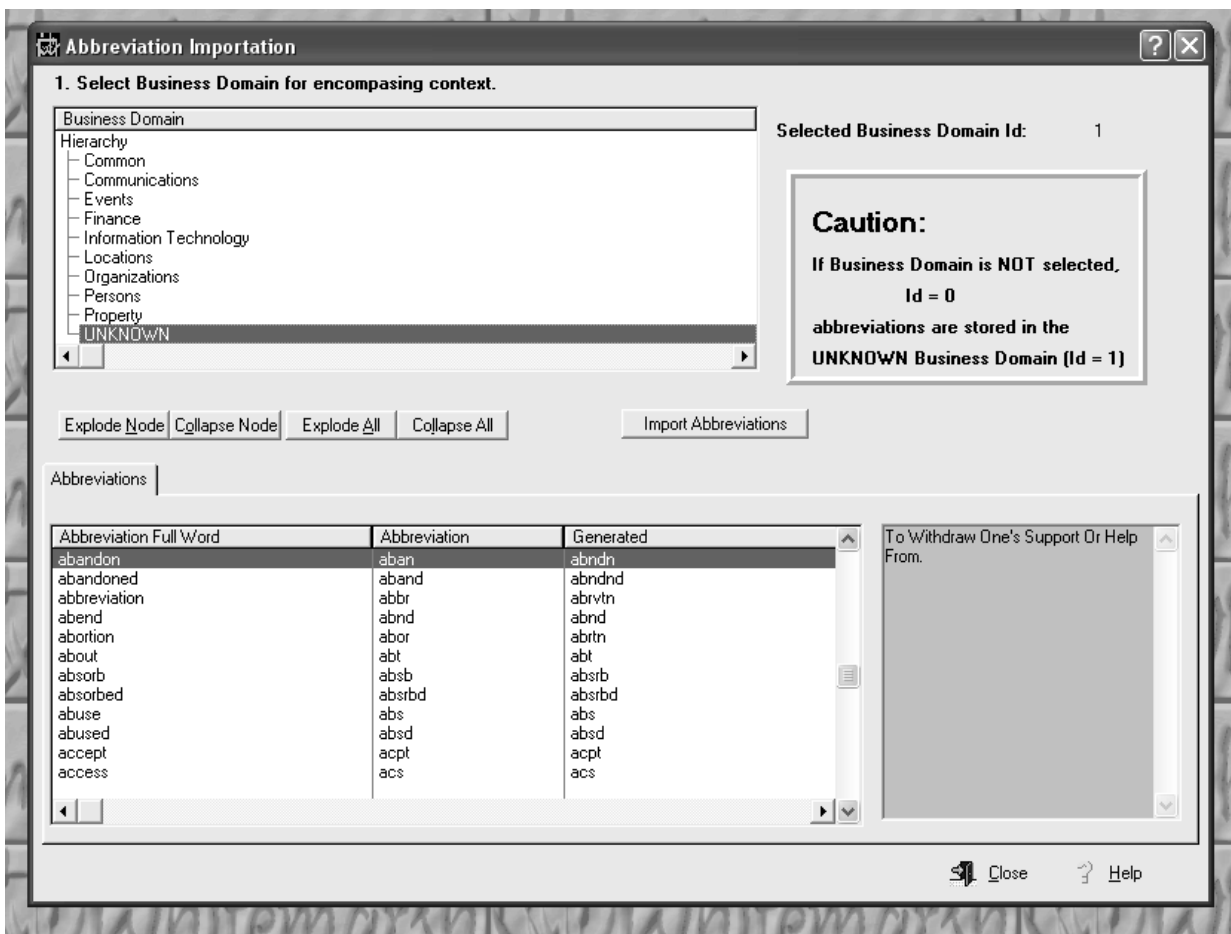


Figure 33. Abbreviations import screen.



Figure 34 shows the window that starts this process. Press the Import Abbreviations button AFTER selecting the business domain into which the abbreviations will go. If you fail to select a Business Domain, then the abbreviations will go under the UNKNOWN Business Domain. The current value of the Business Domain Id is shown on the screen. 0 means not selected. Any number not zero then shows the selected Business Domain

Figure 34 shows the first Import abbreviations window. Enter the file name that contains the source abbreviations. In this case, the first record had headers so it was skipped. Each column was separated by Tabs, hence the tab setting. After this you can proceed to Step 2, Duplicates.

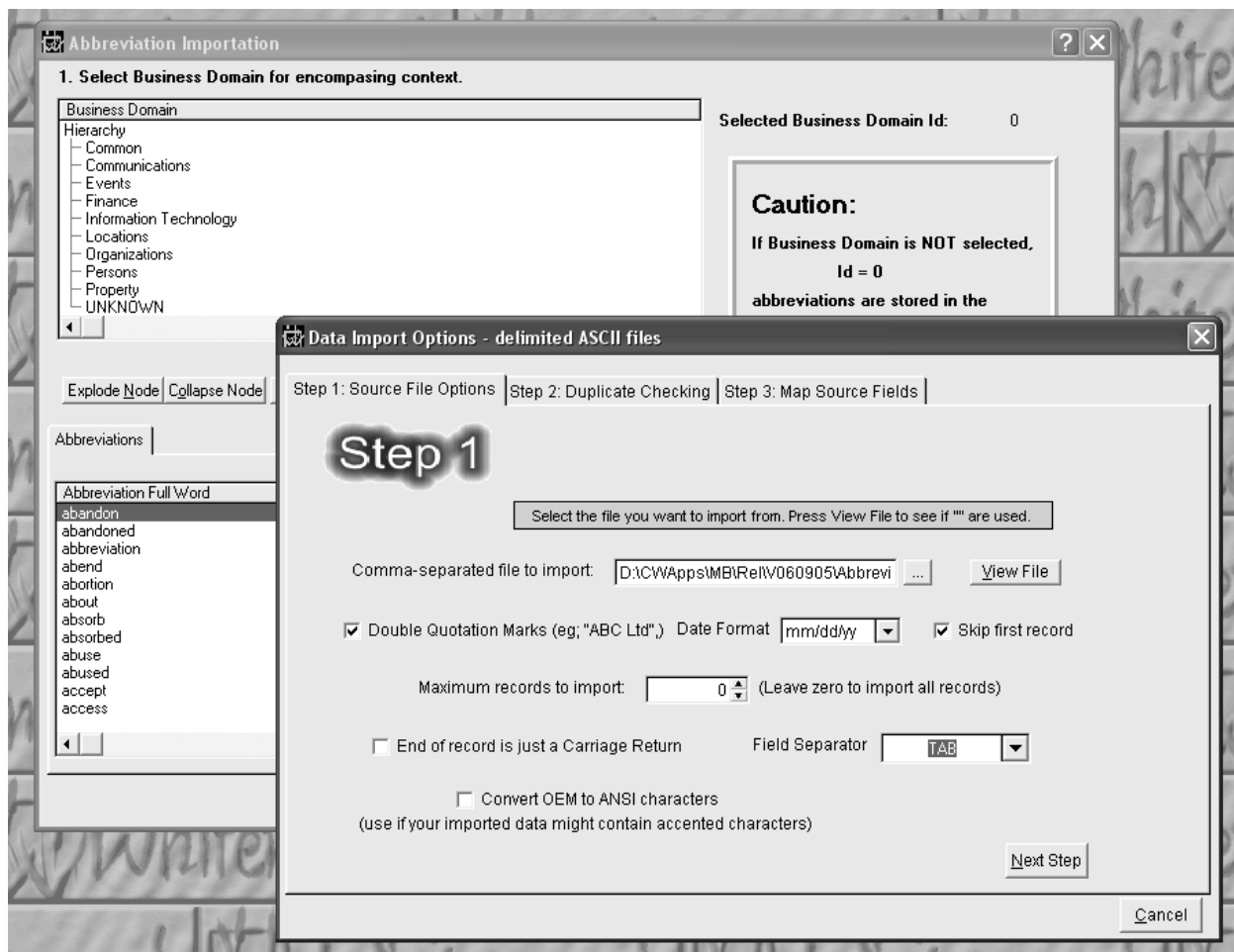


Figure 34. Abbreviations Import. Step 1, file selection and option setting.



Step 2, shown in Figure 35, in the Abbreviations import process consists of deciding what to do about duplicate full terms. Given the ability to distinguish full-terms by Business Domains, there should never be a need to have duplicate full terms. On this window, check the “check for duplicates” box. Then pick what to do for duplicates. Chosen is “Ignore it.” Finally checked is the “Show duplicates” in the default log file, Dupes.txt. At that point, click Next Step.

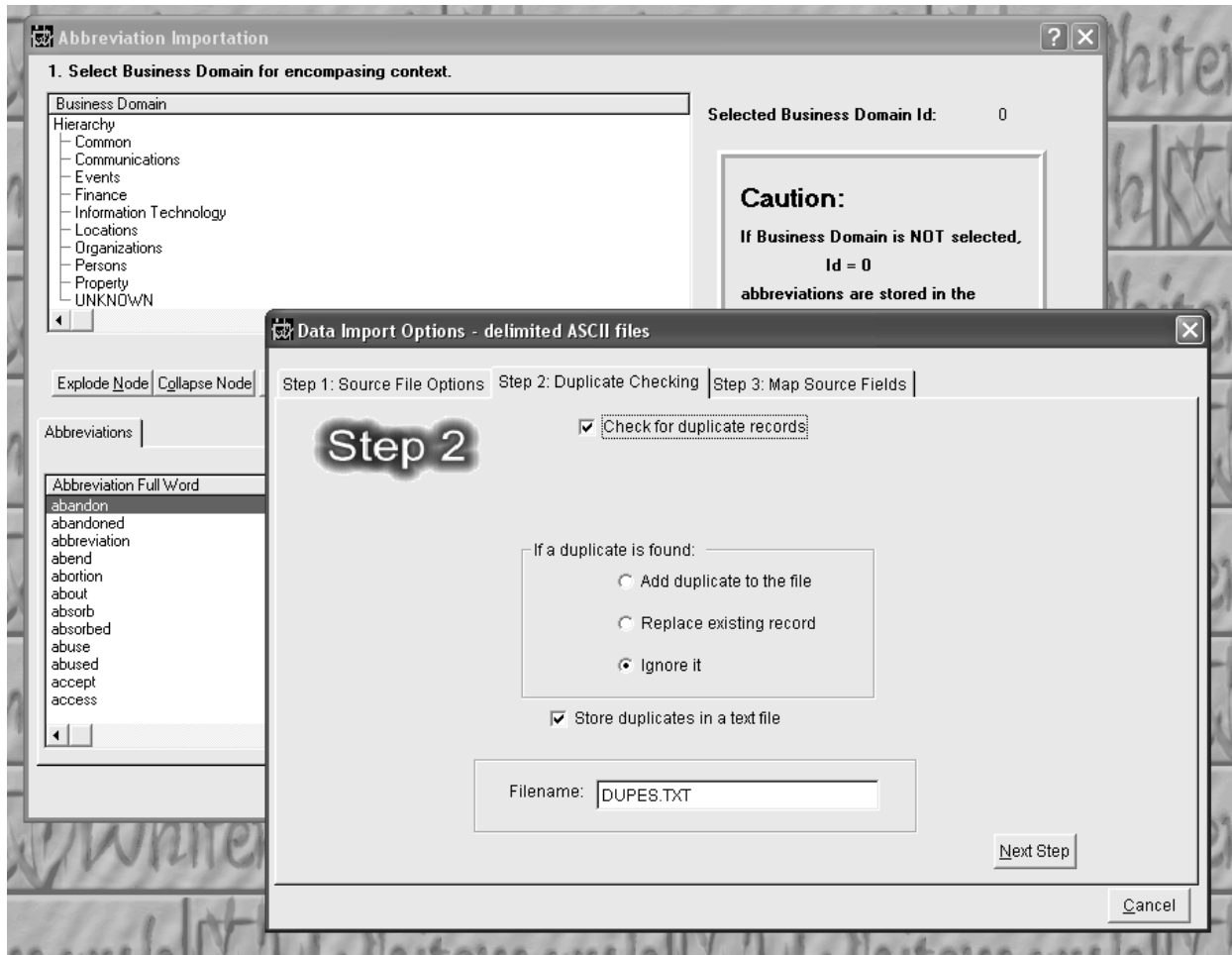


Figure 35. Abbreviation Import. Step 2, Duplicate options.



The last step is to map the fields. This is shown in Figure 36. The process is to press the “First” button. That shows a record of abbreviations. In this case it’s the header record that will be skipped in the actual load. The process of mapping is to Select the field in the left pane and “drag it” over the field in the right pane. When that happens the source field number shows in the right pane. Once the mapping is done, just press the Start button.

This window also provide the ability to save the mapping map through the Save button, or to select a saved one through the Retrieve button.

Once the records are all loaded they show up in the Abbreviations window.

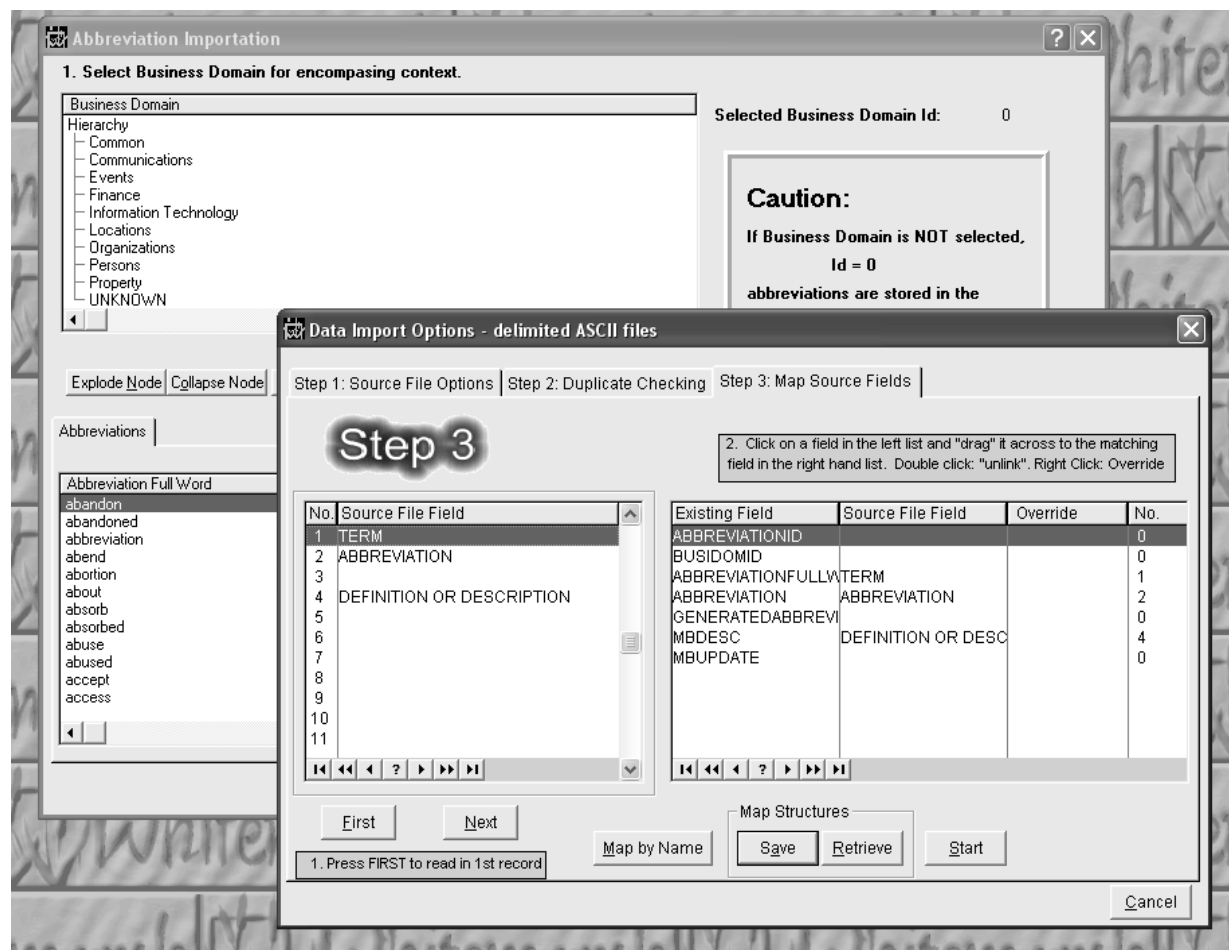


Figure 36. Abbreviations Import. Step 3, field mapping and “Start.”



2.5.3 Abbreviation Business Domain Reassignment

Figure 37 shows the reassignment window. Existing assignments are shown in the left browse. In the Movies version of the metabase that is distributed there are about 5,000 abbreviations, so when this windows opens, it might take a minute to show the records.

The process of reassignment consists of tagging one or abbreviations records in the left browse, and then tagging the “reassigns to” business domain on the right. The button, Re-Assign Abbreviation to Different Business Domain” accomplishes the re-assignment.

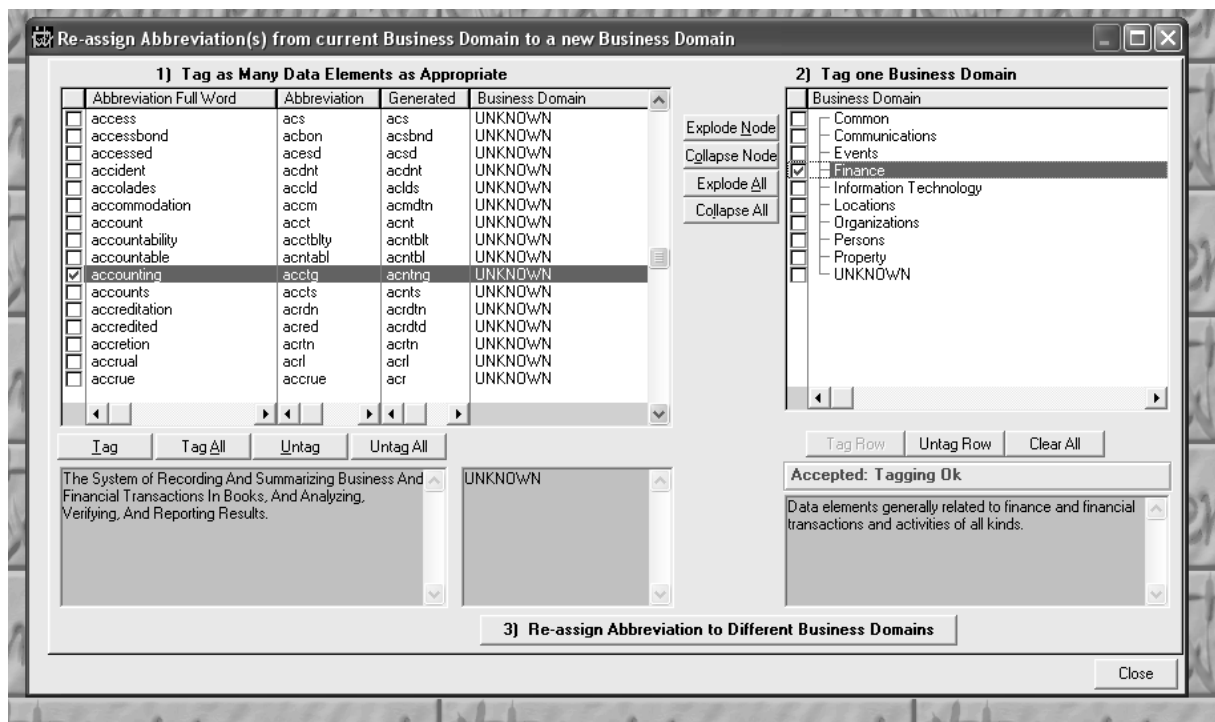


Figure 37. Re-Assignment of one or more abbreviations to a different business domain.



2.5.4 Abbreviation Mining

The final abbreviation management function is focused on gathering all the words that already exist among the various stores of metadata within the metabase. When the idea for this source of metadata words seemed to be valuable, it was presumed that there would be 5, 10, or maybe 15 different sources. Not so. There were 87 different sources of names, within which very valuable metadata names are stored. Here's the list.

Attribute	DBMS Table Secondary Key
Business Domain	Derived Data Element
Business Event Calendar	Document
Business Event Calendar Structure Type	Document Cell
Business Information System	Document Section
Calendar Cycle	Entity
Calendar Cycle Structure Type	Entity Candidate Key
Characteristic	Entity Foreign Key
Characteristic Type	Entity Primary Key
Column	Environment Type
Compound Data Element Structure Type	Form
Concept	Form Cell
Concept Structure Type	Form Structure Type
Conceptual Value Domain	Function
Conceptual Value Domain Structure Type	Information Need
Construction Method	Information Need Type
Data Architecture Class	Level
Data Element	Management
Data Element Classification	Meta Category Value Type
Data Element Classification Structure Type	Meta Category Value Type Class
Data Element Concept	Meta Category Value
Data Element Concept Structure Type	Mission
Database	Mission Organization Function Position
Database Domain	Modules
Database Management System	Nature
Database Object	Org
Database Object Information System	Position
Database Object State	Predominant User Class
Database Object Table Process	Production Status
DBMS Column	Programming Language
DBMS Data Type	Rank
DBMS Schema	Rationale
DBMS Table	Res
DBMS Table Candidate Key	Resource Life Cycle Node
DBMS Table Foreign Key	Resource Life Cycle Node Structure Type
DBMS Table Primary Key	Resource Type



Schema	Value Domain
Sql Data Type	Value Domain Data Type
Subject	Value Domain Structure Type
Table	Value Domain Value Structure Type
Table Candidate Key	View
Table Foreign Key	View Column
Table Primary Key	View Column Structure Type
Users	

Figure 38 presents the Abbreviation Mining screen. The process is simple. Select the Business domain into which all the discovered terms will go and press the button, Mine Metabase for Abbreviations. Discovered and added to the list of all abbreviations are the metadata specific words that may exist in any of the metadata objects list above.



Figure 38. Abbreviation mining screen.

