Dynamic Knowledgebase™ v2.3

Documentation for the System Administrator

Prepared by:

Spatial Knowledge Engineering Inc.
August 2001

SKE

Contents

Who Should Read This Manual

This manual is for systems administrators, web site managers running Dynamic Web Maps, stand-alone users of Dynamic Maps, and anyone who needs to register and / or manage tabular data, map layers, and Dynamic Web Maps Topics with an understanding of Dynamic Maps and / or Dynamic Web Maps. If you are a Database Administrator responsible for the data and setting up all the tools, then you should understand geographic information systems and database management concepts, the use of professional GIS tools like ArcInfo and/or ArcView, as well as Microsoft Access, other DBMS and Windows.

> To set-up the system on a network you may need to work with your Network Administrator if you are not familiar with networking concepts and terminology.

In order for map layers to be registered with the Dynamic Knowledgebase tool, they must first be properly prepared using the appropriate GIS technology and tools. Likewise tabular data needs to be first cleaned, properly geo-coded and then put into or accessed through Microsoft Access (M/S Access'97 or 2000), Excel (.xls), dBase IV (.dbf) or Oracle DBMS. See **Appendix 6. – Warehouse Data Preparation** for more detailed information.

Dynamic Knowledgebase is used wherever the data being accessed by Dynamic Web Maps and / or Dynamic Maps need to be set up and maintained.

The Dynamic Knowledgebase Tool Concepts

Most of us are familiar with using the computer to create documents, spreadsheets, presentations, and graphics. Some of us have used databases and distributed systems to share information locally and over the Internet. With the advent of Dynamic Knowledgeware, data are not only more accessible and shareable, but users have easy-to-use mapping capability as well. The Dynamic Knowledgebase tool makes this possible by setting up and "registering" the data and related components that Dynamic Maps and Dynamic Web Maps use. Importantly, the complexity associated with mapping data and relational databases has been removed from the end-user.

Dynamic Knowledgebase enables you to:

- Create one or more "warehouses" of data – organizing data with common characteristics together (a data warehouse is a set of logically related spatial and tabular data that are in the same geo-referencing coordinate system, along with related information objects);
- Import an existing data warehouse that has already been built;
- Register spatial data libraries their associated tiling scheme for very large spatial data sets;
- Register spatial data, i.e. map layers, and assign their rendering and feature selection capabilities;
- Register tabular data sets so that they are easily accessible and their records are tied to the relevant map features;
- Describe map layers and tabular data sets in a consistent manner;
- Set up the Dynamic Web Maps "Topics" for the web site – publish data warehouses to the web.

The Key Concepts

Map Layers

A Map Layer is a container for spatial data of the same feature type. There are five feature types: points, lines, polygons, labels and images. As well, a map layer usually represents a specific theme, for example – environmentally sensitive areas (polygon layer), roads (line layer), village locations (point layer), village names (labels), landcover (image layer).

A map layer can be stored as one continuous coverage representing all the features, or it can be broken down into tiles (see ***Tiles and Tiling*** below). The system places no limits on the number of map layers that can be registered. However, all map layers must be in shape file or ESRI ArcInfo coverage (PC or Workstation) format (see ***Shape Files*** below) or if they are image / raster layers, they cab be in .bmp, .sid, .bil or .tif format with an associated ESRI World file.

Depending on how the map layer is to be registered into the data warehouse, it should have the following fields:

- **Feature ID**. If the map layer is being made selectable, it will need to have a feature ID field in its .dbf file (primary attribute table). This ID is used to uniquely identify each feature in a map layer and is also used to join any related tabular data sets that are registered to the map layer. Feature IDs can be alphanumeric such as names, or numeric (Integer or Real numbers). Note with real numbers, rounding of the numbers may cause mismatches. Always try to use double-precision numbers if they are real.

- **Feature name field**. If the map layer is being made selectable, it will also need a feature name field in the .dbf file (primary attribute table) that accurately names each feature in the map layer. For example, a layer of villages would need a field in the .dbf file containing the relevant village name. The Feature Name will be used when the user wants to locate a feature or when they want to identify a feature in Dynamic Maps and Dynamic Web Maps. Even if the layer is not being made selectable, the feature name field is required if you register a map layer as a **Label Layer**. The feature name then becomes the "label" associated with the map layer features. Note, the Feature name field could be the same field as the Feature ID field.

- **Feature value field**. If the map layer will be “value rendered”, it will need a feature value field in its .dbf file. Value rendering helps take various features on the same map layer and display them differently (in different colors / characteristics). So, for example, a forest cover map layer may have several cover-types signifying locations of different single species stands and mixed stands. If these cover types have an appropriate name and value, then the map layer can be registered in the system with each value being assigned a different rendering property. Feature value fields can be numeric or alphanumeric data types.

In these three cases the map layer dbf file must contain the following fields.

.dbf File Fields	Data Type
Feature ID - for selectable map layers	Numeric or Alphanumeric
Feature Name - for selectable map layers	Numeric or Alphanumeric
Feature Value Rendering - required if map layer will be value rendered)	Numeric or Alphanumeric

If the map layer is to be registered as a background map layer (its non-selectable) or if it’s an image map layer, then no specific fields are required for the registration.

Shape Files

A shape file is a non-topologically structured graphic file developed by ESRI that consists of three main files: the .shp file which contains the geographic extent and shape information for each of the geographic features; the .dbf file that contains the related primary feature description and identification information; and the .shx file that is a spatial index file for the information in the .shp file.

Arc/Info Coverages

Coverages are also a geographic data format that is topologically based. For a description of coverages please consult ESRI’s web site (www.esri.com) and other literature. Dynamic Knowledgebase can register PC based and Workstation based coverages for polygons, lines, points and labels.

Tiles and Tiling

A "**tile**" is an area representing part of a map layer's extent. A "**tiling scheme**" is an orderly and logical way of dividing the extent of a map layer into smaller pieces. Each tile can represent logical boundaries (such as provinces) or geometric boundaries (such as squares 100 miles across) and can number into the thousands.

Tiling is used to increase the efficiency of the system by breaking up a large map layer into easily manageable chunks – smaller map layer files. When a layer is tiled, the system only needs to consider those tiles within the extent of the area of interest -- rather than the entire map layer (all tiles). For example, transportation layers, drainage layers and image / raster layers are good examples of map layers that could be broken up into tiles because they often contain a lot of information and could cover the entire extent of a database. If the information were to be stored into one large file, the ability of the system to perform efficiently would be degraded. Thus the reason for tiling.

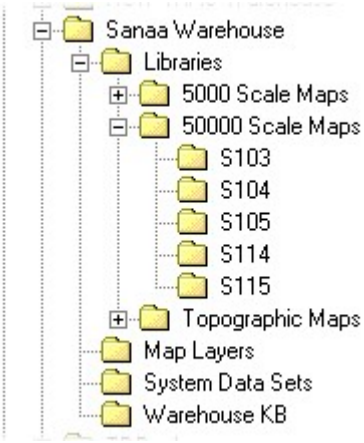
> Tiled map layers cannot be made selectable and therefore cannot have their features identified by the system, nor have tabular data sets and related information objects associated with them. Tiled map layers are treated as "background" information and are designed to provide additional information to help orient the user's geographic view.

For the system to work properly with tiles the DBA must create a Spatial Library.

Spatial Libraries

The system and data architectures are based on the concept of handling both tiled and seamless map layers. Seamless map layers can be considered as having one tile. The tiled databases can have a variety of geographic extents (covering the country, province, a region, a watershed, whatever) whose extent is "tiled" or broken into smaller manageable chunks in a structured and meaningful way. Each tiling scheme is managed and defined in a Spatial Library. The database administrator can set up multiple libraries to handle different types of data for different areas using different tiling schemes (indexes).

A spatial library contains the tiles and the index for the "tiled" map layers, all with the same tiling scheme. Therefore, it is possible to set-up multiple Spatial Libraries, each with its own tiling scheme. The index describes the tiling structure and needs to be created by a GIS technician with appropriate software – such as ArcInfo and saved in a shape file format.
Libraries do not support Arc/Info Coverages.



Physically, the Spatial Library is a hierarchical file structure. At the root is the Warehouse directory. Below the Warehouse directory are the Libraries, Map Layers, System Data Sets, Related Information, and Warehouse KB subdirectories. The Libraries directory consists of subdirectories that represent each Library. In each Library there will be the library index files (this is a shape file that maps out the tiling scheme and is called "Index.shp" (.dbf/ /shx) and subdirectories that represent each tile. Each tile directory contains the individual shape files for each tiled map layer. This is illustrated in the diagram to the right. So with a tiling scheme based on 1:50000 scale maps, then each "map" would be represented by a directory – e.g., S103, S104, S105, etc. -- in which each of the relevant tiles the associated portion of the map layer's .shp, .dbf, and .shx files would be stored. The index files for this tiling scheme would be placed only in the Library directory -- the "50000 Scale Maps" directory in this example.

There are no limits to the number of map layers that can be in a Spatial Library. Likewise there are no limits in terms of the number of tiles in a Spatial Library. To optimize performance the system allows the tiles to be distributed anywhere. Individual tiles can be stored on one computer hard drive or across several hard drives, or across several computer hard drives. . It is recommended, however, that the files reside within the directory structure of the warehouse – especially if that warehouse is to be moved and shared with other users outside of the computing environment in which it is created – if it is to be exported and then imported somewhere else.

The **Index file** consists of three files related to the tiling structure. The .dbf file contains a record for each of the tiles. All that is required in the .dbf file apart from the standard polygon feature attributes is a tile name field that identifies each tile for the Library. The Library Tiles function in Dynamic Knowledgebase keeps track of the location of the tiles. The following table provides an example of an Index.dbf file for a province-based library, where PROV29NAME is the field used to identify each tile in the library.

INDEXWD_ID	PROV29ID	PROV29NAME
1	11	BADAKHSHAN
2	12	TAKHAR
3	17	JAWZJAN
4	16	BALKH
5	14	KUNDUZ
6	15	SAMANGAN
7	18	FARYAB

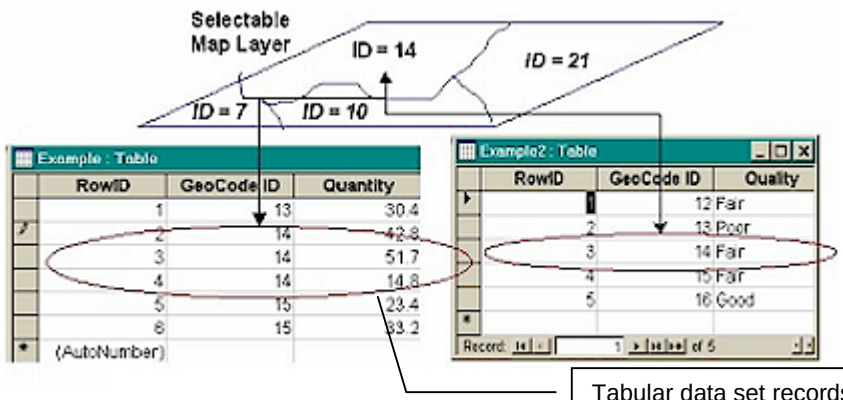
When a blank warehouse is defined, only the five root directories are defined (Libraries, Map Layers, System Data Sets, Related Information, Warehouse KB). It is up to the DBA to specify Library subdirectories and to copy the appropriate data into these.

Tabular Data Sets

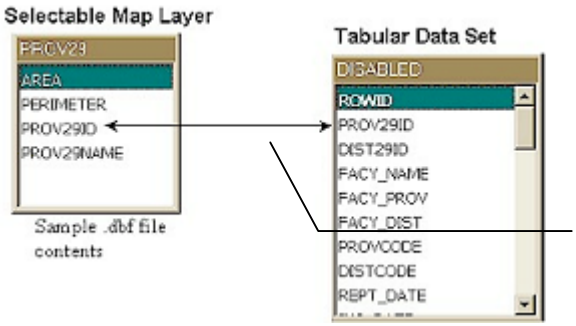
Tabular data sets can be in Microsoft Access' .mdb, Microsoft Excel (.xls), dBase IV (.dbf) or Oracle format for the system to recognize them. This includes M/S Access Link Tables to external ODBC compliant databases. As a result, the system is very open to virtually any ODBC-compliant database, since those not on the list can be access through an ODBC connection to a view.

Although the tabular data set can represent any type of information and contain one or more tables, they must contain at least one link field that matches a relevant map layers' feature ID field. For example, if the data in the tabular data set were collected on a province-by-province basis, then you would want to register the data based to the "province" map layer. The Feature ID field in the province map layer that uniquely identifies each of the provinces would need to be reflected in an appropriate Feature ID field in each of the records in the tabular database. This way, each record can be related through Dynamic Maps or Dynamic Web Maps to its appropriate map layer feature.

The following two diagrams illustrate this relationship between a selectable map layer and tabular data set records.



Tabular data set records are related to their feature on a selectable map layer through a matching Feature ID.



Fields of a sample map layer .dbf file and a tabular data sets .mdb file showing related Feature ID codes.

Feature IDs can be numeric or alphanumeric.

When dealing with tabular data sets, it isn't always clear what "Field Size" and "Data Type" the data are stored in. For example, a field of numbers may be a "Text" data type and this would not be clear just by looking at it. In order for the data to be useable, you would need to use a program like Microsoft Access to change the data type so they are the same. (You can make these changes in the M/S Access' design mode and by editing the column if they contain an extraneous character (such as a comma). Dynamic Knowledgebase also allows you to look at the records in the map layer's .dbf file and in the associated tabular data set and identify their type and relevance for a match.

All the records of a tabular data set do not necessarily need to have a matching geographic feature ID. If a record(s) has no spatial match, this record(s) will be displayed along with the others in the Dynamic Maps and Dynamic Web Maps spreadsheet but will not have an associated feature on the map. As a result, some of the numeric analysis possible in Dynamic Maps (and Excel) will still be possible for the entire set of records.

dBase files (.dbf) must be in an 8.3 format (i.e. the file name must be eight characters or less in length) in order to be registered

Warehouse

The Dynamic Knowledgebase tool enables you to maintain the "warehouses" in which map layers and system data sets are grouped. Each warehouse reflects a logical grouping of map layers or a grouping based on physical reasons. For example, a logical grouping of map layers may be needed for map layers covering vastly different scales -- such as local data and data at a national scale. Different physical groupings would result if you have map layers with different coordinate systems – since map layers in a warehouse must have the same coordinate system.

Dynamic Maps, Dynamic Web Maps and Dynamic Knowledgebase can access multiple warehouses, but not at the same time. With Dynamic Web Maps the warehouse map layers, tabular data sets and related information objects are organized into one or more Topics. Users of Dynamic Web Maps on the web site can select any Topic regardless of the warehouse, and in fact will not even be aware of the fact that they are changing "warehouses".

There are three databases related to the management of a warehouse in the system:

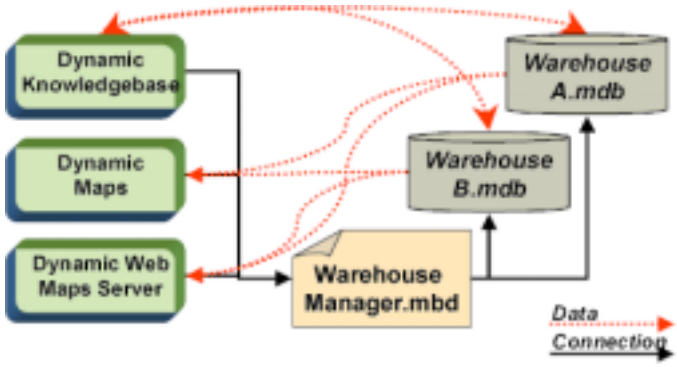
1. The ***Warehouse Manager.mdb*** keeps track of all the warehouses on the system, including the path name, the warehouse ID, the color schemes, and the warehouse password. This is the file that the Dynamic Knowledgebase, Dynamic Maps and Dynamic Web Maps Server all point to when they are launched for the first time. As a result, the file should be stored on the server/common hard drive and accessible to users of Dynamic Maps and to Dynamic Web Maps Server. If this is a standalone computer, then the Warehouse Manager.mdb file will be installed by default on C: drive in a new directory called "Warehouse Manager".

   The ***Warehouse Manager.mdb*** file is installed with the ***Dynamic Knowledgeware Warehouse Manager 2.x.exe*** installation program.

Although you usually should only have one Warehouse Manager.mdb file, it is possible to have more than one. A small executable program called "***Reset dB Connection.exe***" is in the Dynamic Knowledgebase installation directory (<drive>:/Program Files/Dynamic Knowledgeware/Dynamic Knowledgebase 2.3) and will enable you to point to a different Warehouse Manager database.

2. The "<Warehouse Name> ***Warehouse KB.mdb***" is the database that Dynamic Knowledgebase manages. This file stores all the information about the warehouse – the Libraries, map layers' names and location, their rendering properties, their metadata, the topics, the system data sets, etc. The ***Warehouse Manager.mdb*** points to this so that users of Dynamic Maps can access the warehouse and therefore the map layers and data sets. Likewise, Dynamic Web Maps users can access the various Topics and therefore the map layers / data sets in the warehouses.
3. The ***Related Information.mdb*** is a database in each warehouse that keeps track of all the information for any related information objects that are linked to one or more selectable map layer features.

The graphic shows the connections made by the three systems and the data flow, as the systems first connect to the ***Warehouse Manager.mdb*** to get information about the warehouses, so that the user can then connect to the actual warehouse containing the data. Dynamic Web Maps is a bit different since the user doesn't connect to the warehouse directly, but indirectly through their choice of a "Topic".



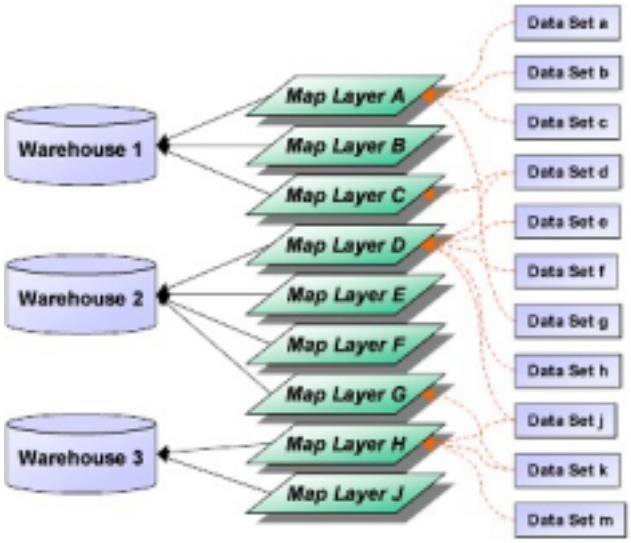
You start each Dynamic Knowledgebase session by first selecting the warehouse with which you want to work. You don't have to manually point to the ***Warehouse Manager.mdb*** every time you launch Dynamic Knowledgebase. This connection to the ***Warehouse Manager.mdb*** is established the first time you launch Dynamic Knowledgebase.

If for some reason Dynamic Knowledgebase ends up pointing to the wrong Warehouse Manager file, you can correct the problem by using the "Reset dB Connection.exe" program in the {root}Program Files/Dynamic Knowledgeware/Dynamic Knowledgebase directory. A similar program is also found for Dynamic Maps and Dynamic Web Maps Server in their program directories.

If the Warehouse Manager.mdb file is moved for some reason, then each Dynamic Knowledgeware program will indicate that it cannot find the file in its originally registered location and prompt the user to locate it. If the file is properly located, the new location path will be stored and subsequently used.

The Overall Data Architecture

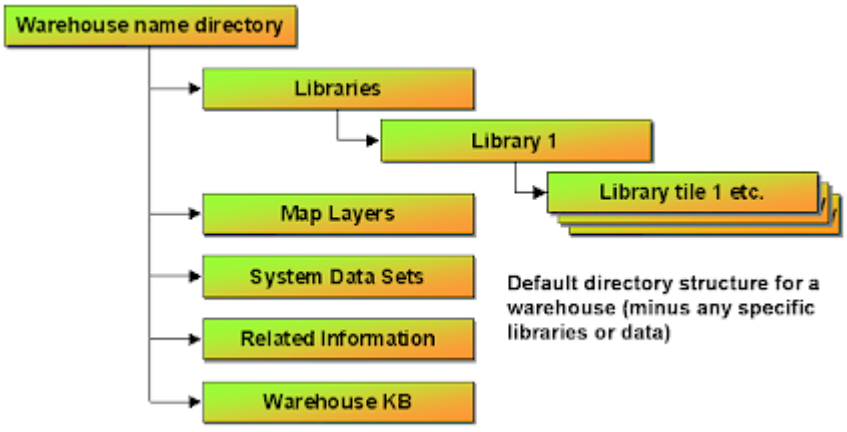
The conceptual data architecture is partially illustrated in the diagram below – where several tabular data sets can be related to one or more map layers, which can be registered in one or more warehouses. Even if a map layer or a tabular data set is related to more than one warehouse, it still needs to be registered with each one separately.



Map layers and data sets can reside anywhere on the data computer or across multiple computers – however, it is best to store them in the default warehouse directory structure.

If the same map layer or tabular data set is being used in more than one warehouse, it is often a good idea if a copy is made for each warehouse so that it can be stored in the directory structure noted below. This makes it easier to manage the warehouse for distribution to other sites. A default directory structure is set up automatically when you define a Warehouse using the Dynamic Knowledgebase Add function.

The Warehouse name directory contains five subdirectories: Libraries, Map Layers, System Data Sets, Related Information, and Warehouse KB.



Default directory structure for a warehouse (minus any specific libraries or data)

- The Libraries directory contains one or more "Library name" subdirectory(s). Each library name directory contains the index files for the library plus all the individual tile directories. The index files keep track of the various tiles and characteristics. So if the library contains 30 tiles, there would be 30 directories representing each one of the tiles. The tile directories each contain the relevant .shp, .dbf. and .shx files (in the case of a Shape file) for each of the tiled map layers. However, a map layer does not have to exist in each tile, i.e., there can be holes in the coverage. For example, if there are three tiled map layers associated with this library (such as drainage, transportation, and land cover) then there can be three .shp, .dbf, .shx files in each directory. Note: setting up the tiling scheme and directory structure should be done by an experienced GIS technician with the appropriate GIS tools.
- The Map Layers directory contains the files associated with each of the non-tiled map layers in the warehouse.
- The Tabular Data Sets directory contains all the system tabular data sets (.mdb, .dbf, and .xls files).
- The Related Information directory contains all the "documents" that get registered against the warehouse and its map features. **NOTE: This directory must have appropriate operating system permissions set to enable users to write to it.** The directory also contains a "***Related Information.mdb***" database that keeps track of the related information registered in the warehouse.
- The Warehouse KB directory contains the "<Warehouse name> ***WarehouseKB.mdb***" database that keeps track of the warehouse and all its components.

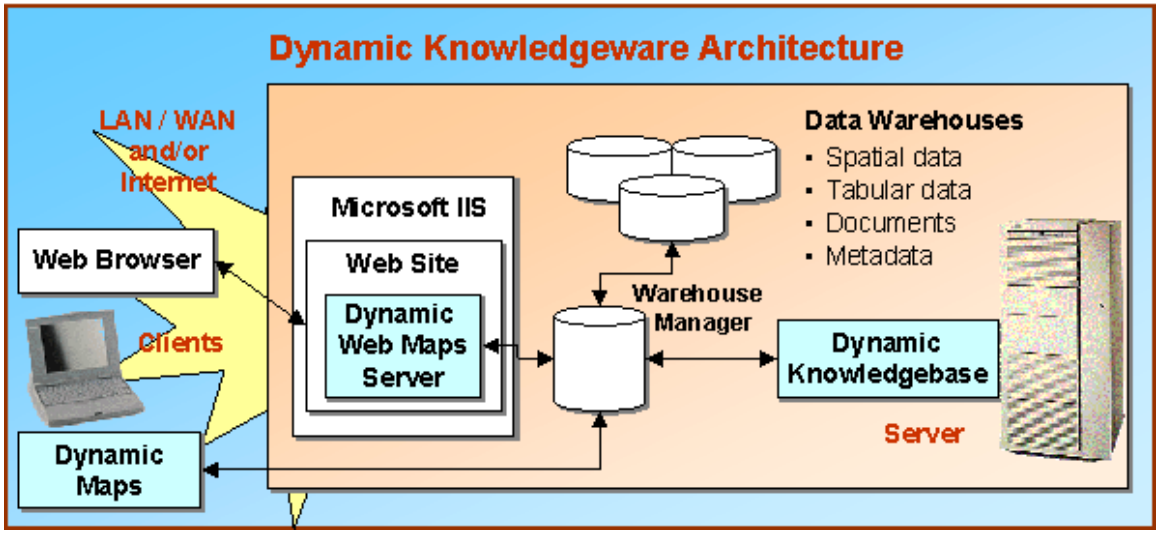
Why Have Different Warehouses?

Example 1: If a country has two ways of defining its provincial boundaries (pre-amalgamation and post-amalgamation, say) it would take two different map layers to describe one and the other. Chances are, other thematic layers, such as a layer showing districts within the provinces would only be relevant for one of those map layers. As a result, the two sets of map layers could each be put into separate warehouses to keep things clear. This would also affect tabular data sets – since their records' provincial identifier would likely be tied to one province identification or the other. The warehouse enables you to take into account these distinctions by separating the data into logical groupings.

Example 2. Some map layers may have been collected using the UTM mapping coordinate system, while others may have used geographic coordinates. Or, some may be national in scope while others are very local in scope (say, for a city). In these instances, relevant map layers can be kept in separate warehouses. In this example a warehouse must be defined for those map layers that use the UTM coordinate system, another warehouse for those map that use geographic coordinates (a warehouse must have all its map layers in the same coordinate system). Warehouses could also be set-up to handle the national data sets and a separate one for the city level data sets.

The Overall System Architecture

Dynamic Knowledgebase manages the corporate database, which is made up of one or more warehouses in support of Dynamic Maps and Dynamic Web Maps Server. This is illustrated below.



Although not shown in the diagram, the Systems Administrator can install the Dynamic Knowledgebase on his or her own Windows 2000, NT, 98, or 95 workstation. They would then manage the warehouse(s) for the organization by creating them on the server or a shared hard drive.

Although this diagram illustrates the Dynamic Knowledgebase tool in a networked environment, it can also be used in a standalone environment.

Although Dynamic Knowledgebase is designed to work over a network it does NOT allow MORE THAN ONE database administrator to access the warehouse at the same time. Doing so will lead to unexpected errors and may corrupt the Warehouse KB file. Only one person can use Dynamic Knowledgebase at any one time.

Summary of Dynamic Knowledgebase Functions:

Many organizations have highly dynamic databases and frequently need to incorporate new data sets and find their current data continually changing in quality, coverage, and accuracy. The Dynamic Knowledgebase helps the organization's database administrator maintain and describe the data sets so that the data are immediately and easily accessible to the user through Dynamic Maps and through Dynamic Web Maps.

The Dynamic Knowledgebase has the following primary functions:

1. Open / Add / Update / Delete / Distribute Warehouses

The opening form of the system enables the system administrator to:

- Select and open an existing warehouse.
- Add / create a new warehouse directory structure.
- Modify the description of a warehouse.
- Assign a warehouse password.
- Define the Web Server and alias for related information.
- Delete a warehouse.
- Import a warehouse created on another system.

2. Add / Update / Delete Map Layers

This component of the system enables the systems administrator to:

- Add and update map layers in the warehouse.
- Identify the map layer name.
- Identify the file type (.shp or coverage files for vector map layers, and tiff, .sid, .bil or .bmp for image / raster layers).
- Identify the library layer with which the layer is associated (and therefore the tiling scheme upon which the layer is based) if applicable.
- Define the map scales at which the layer is made visible.
- Define the feature name and ID fields for use in identifying and finding features, and integrating with tabular data.
- List the contents and attributes of the map layer's .dbf file.
- Define whether or not the layer is selectable.
- Define which layers are "value" rendered so that each of its feature class is rendered in a specific way.
- Define the rendering properties of the layer features – their color, shape, fill type, etc.
- Create, import, and use color schemes.
- Define the layer display order.
- Delete selected map layers from the database.

3. Maintain Spatial Library(s)

When using the Spatial Library component of the Dynamic Knowledgebase the system administrator can:

- Add a new library by choosing a name and defining its file path, the name of the associated index file (shape file) and the database field for the tile.
- Define and associate map layers that have the same tiling scheme with the library by identifying their appropriate shape files.
- Specify the path/location(s) of the various tiles (.shp files and image / raster files) associated with the library.

4. Add / Update / Delete Tabular Data Sets

This component enables you to register the various tabular (system) data sets to be made available by recording the name, location, and specifics of a data set. The system administrator can:

- Define the data set's name for addition, update, or deletion.
- Define the appropriate table or view name.
- Define the table field that links it to the Feature ID in a selectable map layer.
- Define the selectable map layer with which the data set is associated.

5. Add / Input / Delete Metadata

The Dynamic Knowledgebase tool enables the management of metadata for map layers and tabular data sets in the system. The Map Layers' metadata are managed through the metadata function in the Map Layers section of the system, while tabular data sets' metadata are managed in the Data Sets section of the system. Both management tools are very similar – reflecting only minor differences in the unique characteristics of spatial and tabular data.

6. Define Topics - how data will be organized for presentation on the Internet through Dynamic Web Maps

"Topics" are used to organize relevant map layers and tabular data into logical thematic groups for presentation on the web site. The system administrator can:

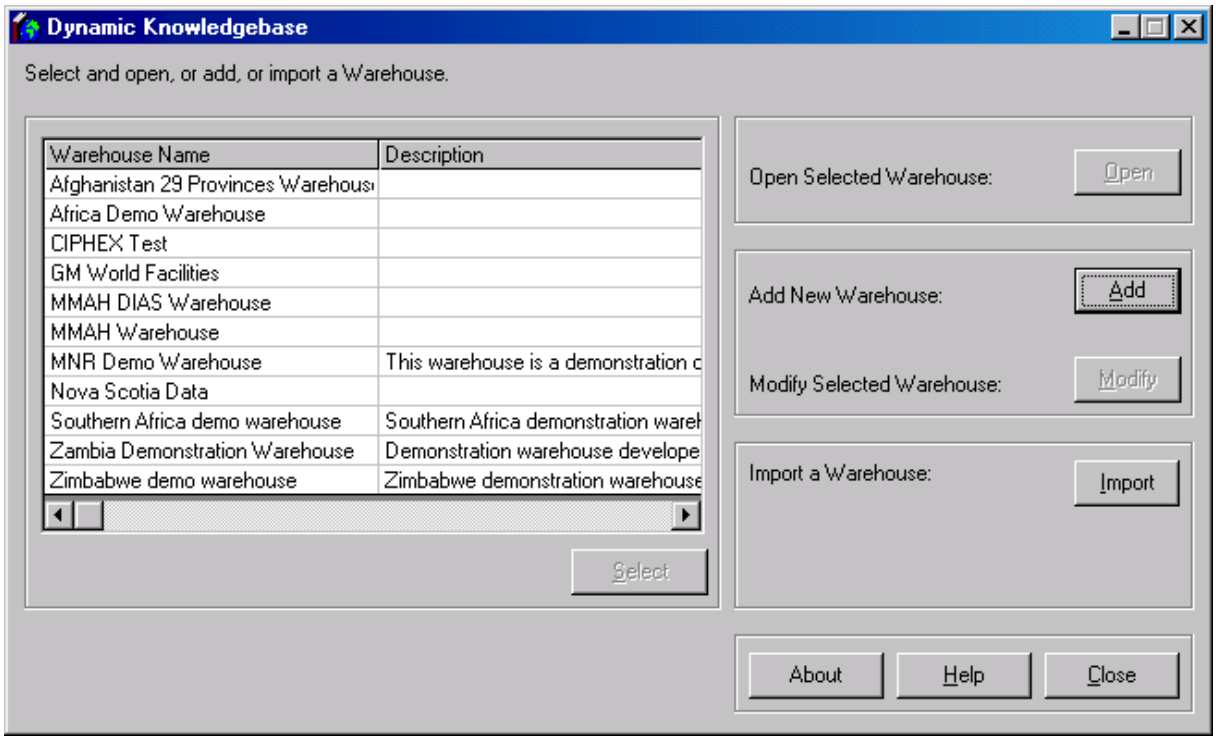
- Define, update, and delete Dynamic Web Maps Topics.
- Associate Map Layers to Topics and remove them.
- Define the order in which map layers are rendered.
- Specify which map layers are displayed as base layers (layers that are always displayed when at their appropriate viewing scale).

Functional Summaries

Working with the Forms

When working with the forms, use the ESC key on the keyboard to clear an individual field. Use **Clear** to clear the entire form. Use **Close** to exit.

The Warehouse Form – Starting Point

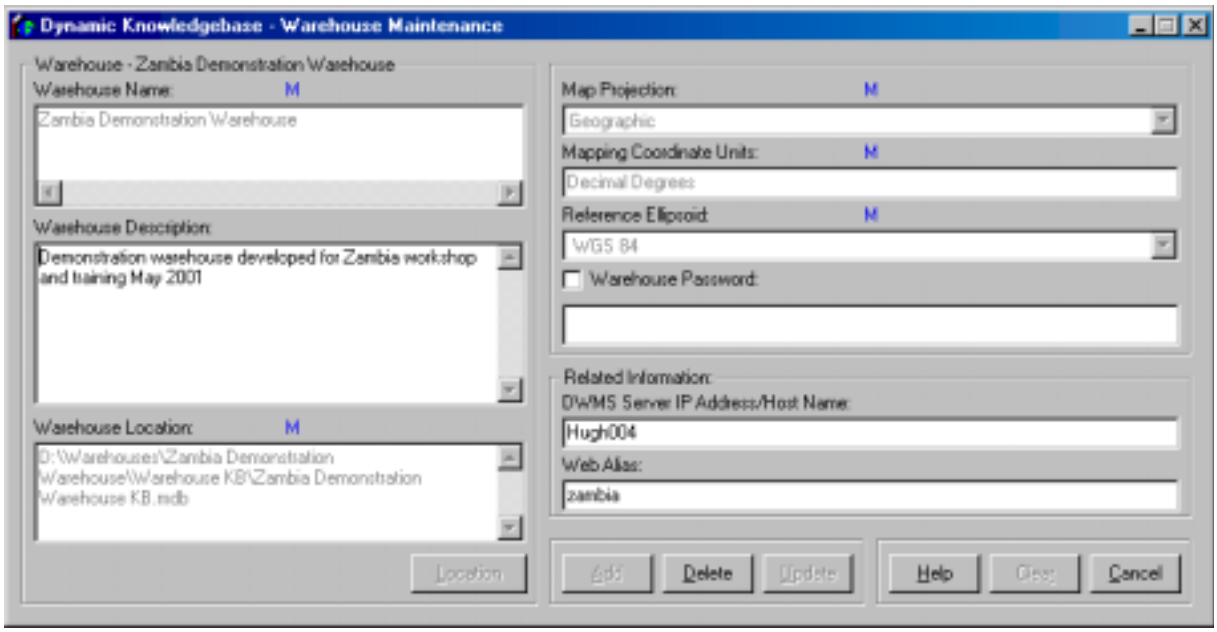


When you launch Dynamic Knowledgebase, the introductory form will enable you to:

- Select and work with a warehouse that is already created using **Open**.
- Add a new warehouse using **Add.**
- Modify an existing warehouse using **Modify** – including deleting a warehouse.
- Import a warehouse with **Import**.

To select an existing warehouse for either opening or modifying double-click on its name in the list, or highlight it and click **Select**.

Warehouse Maintenance Form



If you choose to either **Add** a new warehouse or **Modify** the description of a warehouse, the above form will appear.

Warehouse Name

This is the descriptive name you provide. All users of Dynamic Maps will need to refer to this name when deciding what warehouse they need to use, so ensure the warehouse name accurately, yet succinctly reflects the contents.

Warehouse Description

The warehouse description field enables you to more specifically describe the warehouse and its contents. The warehouse description will appear to users and can be used as a quick verification that the warehouse is the correct one. The warehouse description field is limited to five lines of text and is optional.

Warehouse Location

This is the path to the ***<Warehouse Name>Warehouse KB.mdb*** file. The path will be a network path if the warehouse is used on a network, or a local path if used in stand-alone. Navigate through the Browse for Folder form to where you want to place the warehouse folder.

Map Projection

The map projection identifies which map projection the map layers are stored in. There are many possible projections, including: Geographic, Mercator, UTM and several others.

Mapping Coordinate Units

The Mapping Coordinate Units represent the reference system used to measure locations on a map. The choices are Decimal Degrees and Meters. This field is filled in automatically by the system once the projection is chosen.

Reference Ellipsoid

The Reference Ellipsoid is a mathematical representation that describes the approximate surface of the earth and is the basis for a mapping coordinate system. There are several ellipsoids including: GRS 1980, Clarke 1866 (for North America), WGS 84, etc.

Warehouse Password

Giving a warehouse a password is optional. If you give the warehouse password protection, users of Dynamic Maps will need to input the password before they are allowed to add related information.

Related Information

When the warehouse is being published on a web server using Dynamic Web Maps, then these fields need to be populated to ensure appropriate pathnames are maintained and the related information files can be found.

DWMS Server IP Address / Host Name

The "**Server**" is the URL/location of the Internet or Intranet service where Dynamic Web Maps Server (DWMS) is installed and running. You can enter a physical IP address like 193.221.12.34; or a server name like SKE002. If entering a server host name, make sure the name is entered into your Domain Name Service so that the web map server can be found.

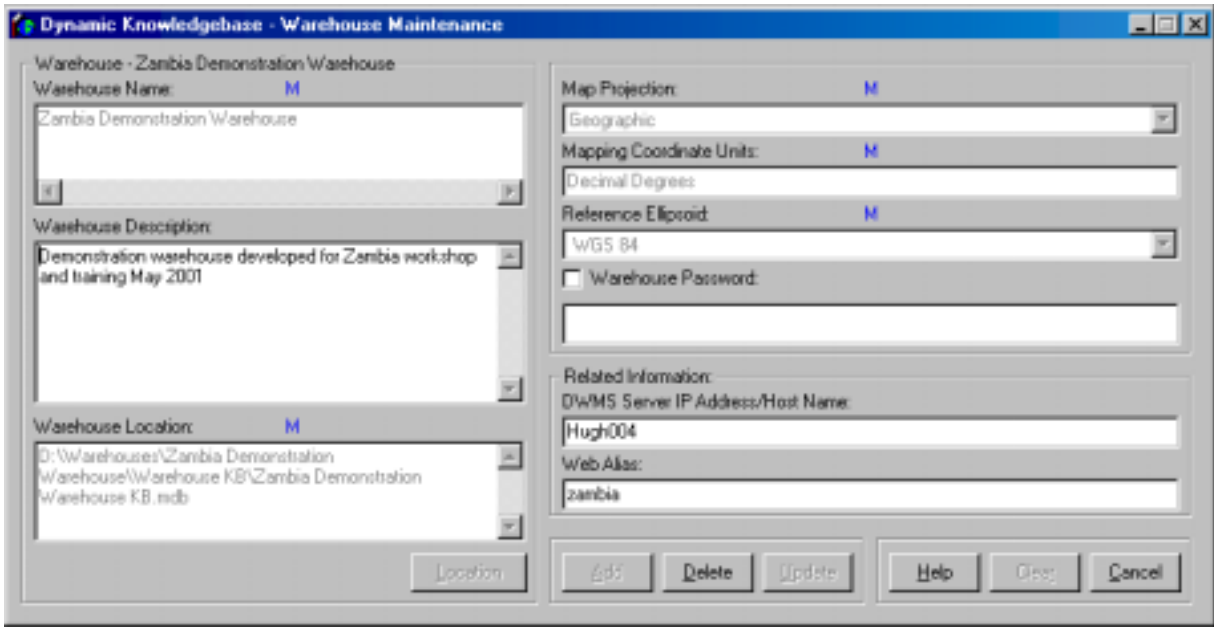
Web Alias

The "**Web Alias**" is the Virtual Directory name that needs to be set up on the web server using Internet Information Server which points to the Related Information directory in the warehouse. This allows for any related information objects to be accessible to the web map server. Please see the instructions provided in the Dynamic Web Maps Server manual for setting up the alias in M/S Internet Information Server.

Quick Guide to Defining a Warehouse

Adding a warehouse creates the warehouse directory structure including the Libraries, Map Layers, System Data Sets, Related Information, and Warehouse KB directories. A blank *<warehouse name> KB.mdb* file is created in the Warehouse KB directory.

1. Click **Add** on the Dynamic Knowledgebase Warehouse form to open the Warehouse Maintenance form.



2. Give the warehouse an easily identifiable name and a description.
3. Point to the directory where the new warehouse structure will be created by clicking on **Location** and navigating via Network Neighborhood to the directory. If this is a standalone installation, then choose a local path.
4. Since all map layers in the warehouse must have the same projection and reference ellipsoid, describe these using the drop down list fields.
5. You may want to set a warehouse password to ensure only authorized users of Dynamic Maps can register related information. Click the "Warehouse Password" checkbox and input a password name.
6. If related information will be accessible through Dynamic Web Maps, input the web map server IP Address or Host Name and the warehouse related information folder alias if they are known. If they aren't, they can be input later.
7. Click **Add** and the new warehouse will be added to the list of warehouses on the Warehouse form. Also, the associated directory structure for the warehouse will be created at the specified "Warehouse Location".
8. From Windows Explorer, populate the Libraries, Map Layers, System Data Sets directories with appropriate map layer and data set files and subdirectories.
9. If you now want to register the map layers, system data sets, topics using Dynamic Knowledgebase for the warehouse, select the warehouse from the list and click **Open**.

Importing a Warehouse

If you have created a warehouse elsewhere, or have a warehouse from another system, the import function automatically resets the proper pathnames in the **<warehouse name>*Warehouse KB.mdb*** database to various map layers, library tiles, related information and system data sets so that they can be recognized and used on your system. This is also useful if you are changing from a local install to a network install. You can't just copy a warehouse from one machine to another machine or from one network to another network and have it work because the system would not know where to look for the data.

There are generally three types of warehouses that can be imported:

a) If the original warehouse being imported had all its files contained within the common warehouse directory structure. This is the simplest and most straightforward type of import, and should proceed without any errors or warnings.

b) If the original warehouse being imported had map layers and/or system data sets outside the directory structure, but contained on one computer. If the data are moved onto the new computer with the same relative directory structure, the warehouse should import without warnings or errors. If those map layers / system data sets are not copied onto the new machine in the same relative directory structure, the system will not find them, and will:

   a. Provide a report of the data sets / map layers it could not access, and,
   b. Provide you with the option of either stopping the import process, or continuing the import and deleting the references to those map layers / data sets from the warehouse kb database.

   NOTE: All data must be at or below the directory level of the warehouse in order for the system to "find" the data.

c) If the original warehouse being imported had map layers and/or system data sets located on another computer on a network. During the import process, Dynamic Knowledgebase will try to access those map layers / data sets / related information objects on the network. If they are successfully "found", the import process will continue without warnings or errors. If one or more map layers / data sets on the network cannot be located, or if the network is not working, the system will:

   a. Provide a report of the data sets / map layers it could not access, and,

b. Provide you with the option of either stopping the import process, or continuing the import and deleting the references to those map layers / data sets / related objects from the warehouse kb file.

NOTE: If Windows share names were used to describe the path / location in registering map layers and data sets on the original computer (e.g. “warehouses” instead of “C:/data/warehouses”), these must be consistent for all map layers / data sets /related information and the same share name must be available when importing on a new computer.

Quick Guide to Importing a Warehouse

1. To move a warehouse to Computer B that was created on Computer A, copy the entire warehouse over to Computer B using Windows Explorer, making sure you maintain the exact directory and file structure. If the two computers are not on the same network, you can copy the warehouse onto a ZIP drive or burn onto a CD-ROM.
2. From the Import Warehouse form, click **Locate** and navigate to the ***<warehouse name>WarehouseKB.mdb*** file in the Warehouse KB directory. If you are running Dynamic Knowledgebase on a network and wish to have the warehouse accessible to other computers, navigate via Network Neighborhood. Click **Open** when you have located the file.
3. Confirm the path name in the WAREHOUSE PATH on the Report. If correct, click **Import** and the warehouse will be imported. This might take a few moments, depending on the size and complexity of the warehouse.
4. Once imported, you may wish to modify the warehouse description. If so, select the warehouse from the list and click **Modify**. The Warehouse Maintenance form will appear and you can change the description and then click **Update**.

Main Interface Functions

Once you have selected a warehouse you want to work with and opened it, you will have the following options:

1. **Add / Update / Delete Map Layers**
2. **Maintain Spatial Library(s)**
3. **Add / Update / Delete Tabular Data Sets**
4. **Define Topics - how data will be organized for presentation on the Internet through Dynamic Web Maps**
5. **Change Warehouses**
6. **Get Help, and**
7. **Exit.**

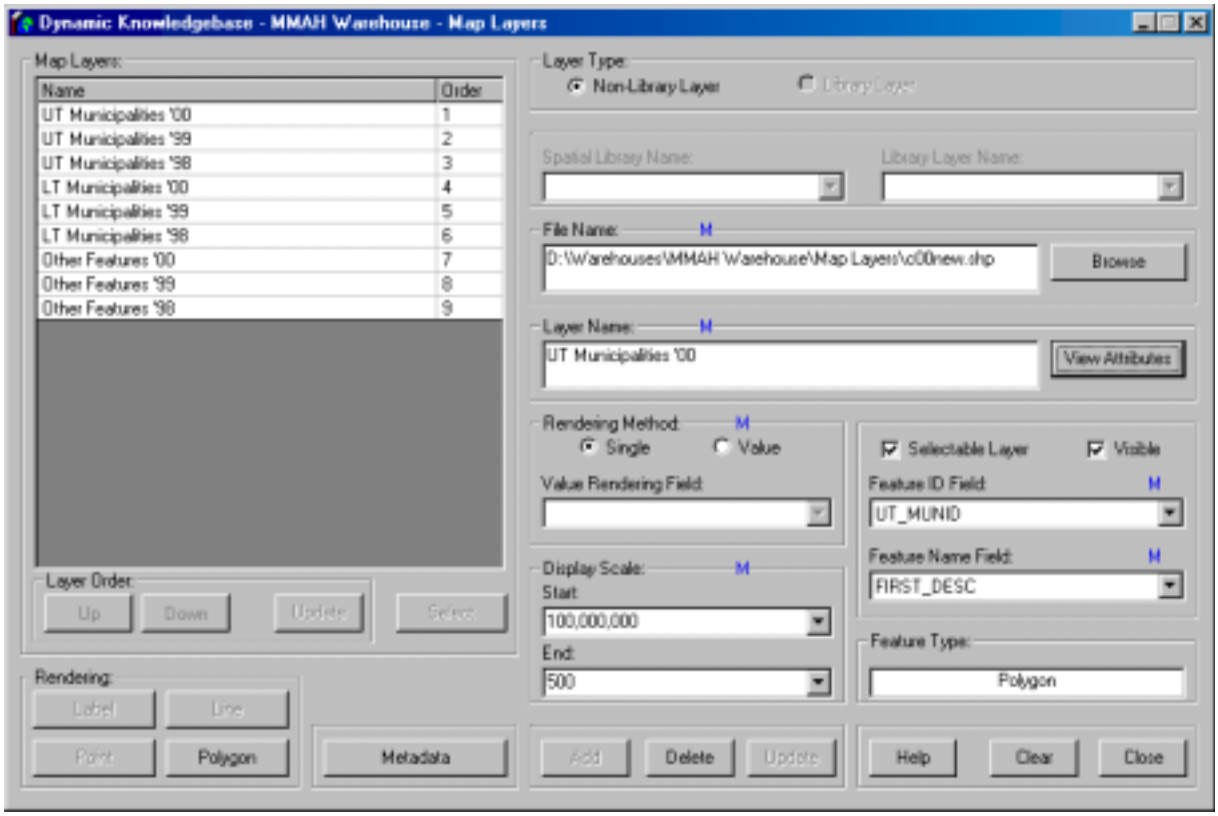
Map Layers

The Map Layers function enables you to set up the map layers that are available to the user and how / when they will appear. There are several different types of map layers that you can register and each has a slightly different process. See the ***"Quick Guides"*** for step-by-step instructions on registering:

- **Non-tiled, non-selectable map layers**. Non-selectable layers are considered "background" layers, since the user can only use them as a reference and cannot identify / select features or register tabular data against them. These are the most straightforward type of map layer to register.
- **Selectable map layers**. A selectable map layer has the following key characteristics:
  - Its features can be selected.
  - Its features can be located using a "Find" function and identified.
  - Its features can have related information objects and record(s) from a tabular data set(s) associated with them.
  - As a result, selectable map layers must have an appropriate Feature ID and Feature Name in their .dbf file (primary attribute table).
- **Label layers**. Point, line, and polygon layer shape files can also be registered as a "Label" layer. A label layer is a map layer with only its feature name(s) rendered (such as a place name). This is useful if, for example, you have a number of geographic features in a particular area and wanted to show their names once the user has zoomed in to an appropriate scale. This way, the user would see the geographic features at a small scale and then as the scale got larger and there was more rendering room on the map, the feature names would appear as labels.
- **Tiled map layers**. Tiled Map Layers are stored as spatial libraries. They cannot be selectable. See the discussion about tiles and tiling in the Dynamic Knowledgebase Concepts section above for more information about spatial libraries.
- **Image / Raster layers**. Image layers can be set-up in a spatial library structure if they have been tiled into smaller image files or they can registered as a single image map layer. In both cases these types of map layers are always treated as background layers and should appear at the bottom of all the other vector map layers, i.e., image layers are not transparent and will cover-over anything that is shown below them.

Each map layer can be registered multiple times and used in different ways. For example, a layer could be registered as a value rendered layer to show only certain features at a gross scale, then registered again as a value rendered layer to show all the features at a detailed scale, and again as a label layer. The system does not copy the actual map layer data every time the map layer is registered, it only stores the appropriate parameters about how to display and work with the layer in the Warehouse KB.

Map Layer Form Field Definitions



Map Layers field (Name and Order)

The Map Layers field lists currently registered map layers and the order in which they are drawn in Dynamic Maps. Use this list to select a layer to work on.

To select a map layer: click on the layer and click **Select**. Or, double-click on the layer name. The various fields will be populated on the form.

Layer Type

The Layer Type field enables you to tell the system you want to register a Non-Library Layer (default), or a Library Layer. If you select Library Layer, the Spatial Library Name and Library Layer Name fields become active.

Adding a map layer that is part of a library is a bit different than adding other map layers. When you select the Library Layer function, you are telling the system that the map layer is part of a Spatial Library and is tiled based on a particular indexing scheme. Therefore, this type of map layer must ***first be set up as a Library Layer*** using the Dynamic Knowledgebase tool's **Spatial Libraries** function on the main program interface. Once set, they will be ready to be registered as map layers within the warehouse and made available to users.

Spatial Library Name

If you are working with a Library Layer, use this drop-down list to select the Library with which the map layer is associated.

Library Layer Name

If you are working with a Library Layer, use this pull-down list to select the particular map layer from the associated Spatial Library. The File Name field and the Layer Name field will be populated automatically, based on the library layer name that you choose.

File Name

When adding a new map layer, click **Browse** beside the File Name field and navigate to the file. For map layers that already exist, this field will be populated with the actual path to the database for the map layer file (.shp or ArcInfo coverage files for vector map layers, and .tif, .sid, .bil or .bmp for image / raster layers).

Make sure you use network path names if the warehouse is being accessed on a network.

Layer Name

The Layer Name is the logical name that you give the map layer to succinctly describe it. The Layer Name will be on the list of available layers in Dynamic Maps and in Dynamic Web Maps and it will **also be the name used in the map legend**. Therefore, keep the name relatively short so that it will fit easily in a legend, and make sure it adequately describes the features the map layer represents.

When you are adding a layer, the name field will be initially populated for you. If you want to change the name, you can by simply typing over the existing name.

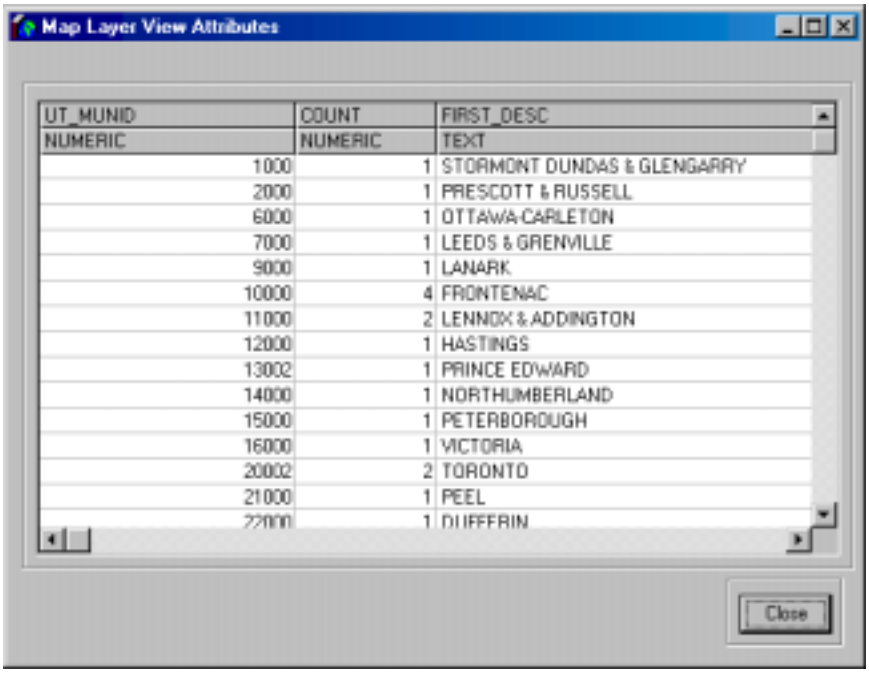
You may want to give two map layers the same name if they show the same features but at different scales. For example, a transportation layer may show roads at a national scale, and another might have a local scale with much more detail. By using the same name for each and turning one layer "on" and the other "off" at complementary scales, the user will not need to worry about what they should be looking at and when.

View Attributes

The **View Attributes** button brings up a form that lists in tabular format the name of all the fields in the .dbf file, their data type (e.g. numeric, alphanumeric), and the records. This enables you to quickly identify what field should be the Feature ID field and Feature Name field for any map layer that you want to make selectable, as well as what field to use for value rendering.

From the View Attributes form, you have a lot of functionality. For each field, you can right-click on the field names to get a variety of functions. For each field you can:

- **Sort Ascending and Sort Descending.** Choose one of these to sort the records in the attribute table based on the column over which your cursor was clicked. This helps identify the high and low extent of the records in that field and it works on both numeric and text fields.
- **Set as Value Rendering Field**: This is a quick way to select the field that you want to use as the Value Rendering Field (see ***Value Rendering*** field described below)
- **Set as Feature ID Field**: This is a quick way to populate the Feature ID field (see ***Feature ID*** field described below).
- **Set as Feature Name Field**: This is a quick way to populate the Feature Name field (see ***Feature Name*** field described below).



Rendering Method

You must assign a rendering method to a new map layer.

Single: Probably most of the map layers you register will have "Single" rendering. When you select "Single" rendering you are telling the system to render all the features of the map layer in the same way. So, for example, a rivers layer would have all its features rendered with the same color and line width and style.

Value: For some map layers, you may want to render groups of features that have a common value in one of the .dbf fields (value rendering field) with specific rendering properties. To do so, select Value. The Value Rendering Field will now become a mandatory field and will need to be selected.

To assign the rendering properties for a new map layer, you must first register it, then select it from the list of Map Layers, then choose the Rendering option for Polygon, Line, Point, or Label – depending on the type of map layer.

Once you've assigned a layer as Value-rendered, you cannot change it back to Single rendered without re-registering the layer.

Value Rendering Field

The "Value Rendering Field" is used when you have assigned the Rendering Method as "Value". Select from the pull-down list, the field in the map layer that contains the unique list of values to be used for rendering the layer features. You can also use the View Attributes function to both identify and select the Value Rendering Field (also see ***View Attributes***). See "***Using Value Rendering***" for more information about how to assign rendering values to the individual layer features.

Once you select the field value, you may get a message that says there are more than 20 unique values. This message appears as a warning suggesting you will probably need to do a lot of work assigning the rendering properties for the layer – since there are more than 20 values to be assigned. It may also be an indication that you have chosen the wrong value-rendering field.

Display Scale / Start and End

Assigning a display scale is mandatory for all map layers. Defining the display scale is the way you indicate to the system between what scales the map layer should be made visible. The Start scale is the scale at which the map layer first appears. The End scale value is the scale at which the layer is no longer visible and available.

> The display scale works by taking a distance on the user's computer monitor and providing a ratio to the actual distance on the ground as displayed on the map.

For map layers that provide the same type of information only at more generalized or at higher resolution, the Display Scale function makes it possible to show these layers one after another as the level of detail and scale get larger (as you zoom-in). For example, several layers of road networks could be set up to display at differing scales in such a way the user does not need to know that there is actually more than one map layer of information being accessed and rendered.

> **Tips for setting the Display Scale:**
> - Do not display detailed map layers at small scales (when zoomed-out at full warehouse extents). If you do, the user will need to turn off the map layer to prevent it from cluttering up the Map View.
> - Consider that Dynamic Maps users may want to overlap layers – especially selectable map layers so that they can easily do the "select by intersecting" function and so that they can zoom out without losing selections.
> - Consider that when zooming-out from a selectable map layer in Dynamic Maps, the map layer scale may be insufficient to retain visibility, and therefore any selected features would be lost.

It is very important that you set the appropriate maximum and minimum scales as this will also affect how the Find and Select functions work in Dynamic Maps and Dynamic Web Maps. If the scales are set incorrect (beyond what is actually needed to see the map layer) then these two functions may not work properly when they try to zoom-in on the found/selected feature(s).

> **Tip:**
> When registering a map layer for the first time, specify the maximum and minimum scales and define the rendering properties. Using Dynamic Maps, zoom in and look at the scale display (bottom right corner of the map view) to see what scale you want the map layer to show up and then what scale you want the map layer to disappear. Then go back to Dynamic Knowledgebase and set the correct scales for the map layer.
>
> Keep in-mind that the scale displayed in Dynamic Maps will change depending on whether the Map View is displayed in a maximized (full screen) window or a partially minimized window.

If your map layer is not showing up when you first access it with Dynamic Maps, it may be because:

- The Start scale is not set high enough. This can happen when you are trying to display continental or world data sets. Our experience has shown that a continental map requires a scale of at least 1:130,000,000.
- You may not have assigned as yet any rendering properties for the map layer so the system does not know how to display the map layer features.

Selectable Layer

Click the checkbox to make the map layer a Selectable Layer. A selectable layer is a layer that can be made into an Active Layer in Dynamic Maps and one whose features can be identified in Dynamic Maps and Dynamic Web Maps. Selectable layers are the only type of map layers that can have tabular data sets and related information registered against them and their features.

> Tiled layers (library layers) cannot be designated as selectable layers.

> "Label" layers should not be made selectable.

Visible

By selecting the "Visible" checkbox, you are indicating that the layer will be shown by default without the user having to turn it on in Dynamic Maps. If the Visible checkbox is not on, the layer name will appear in the list of available map layers, however, the user will have to turn it on in order for it to be displayed. For Dynamic Web Maps, visibility is determined by the Topics function and specifically which map layers are assigned as "Base Layers". By default Visible layers are assigned as Base Layers, however you can change this when building the web Topic.

> Image / raster layers should typically not be made visible by default, because they can take a long time to draw, especially if the user's map view covers a large area.

Feature ID Field

The Feature ID Field is used when you are registering a **selectable** map layer. This field provides a unique identifier for each feature in the map layer and the link to any associated tabular databases. This field must have a unique set of values and has a value for every map layer feature. Select the Feature ID Field from the pull-down list.

If you need to look at the records in the map layer's .dbf file, or if you want to ensure the Feature ID field is the expected data type, click on the "**View Attributes**" button above.

> The field type in the .dbf file should be and Integer or Long Integer for map layers with many features if the field is numeric, otherwise you may not get proper matching if there are decimals. The tabular database and the .dbf file may treat the decimal differently (e.g. rounding differences). Feature ID fields can also be alphanumeric.

> You cannot change the Feature ID field on a map layer if there are already Tabular Data Sets associated with this map layer. Re-register the map layer with the different ID, or delete the associated tabular data sets.

Feature Name Field

The Feature Name Field must be used when you are registering a selectable map layer. This field contains the values that is used to name each feature and is used throughout Dynamic Maps and Dynamic Web Maps for finding features and selection.

The Feature Name Field is also used when you are registering a Label layer. The values in the Feature Name Field from the .dbf file are the ones that are displayed on the map as labels.

If you need to look at the Feature Name records in the map layer's .dbf file click on the "**View Attributes**" button above.

Feature Type

The system will automatically display the type of vector-based map layer – point, line, or polygon – in the Feature Type field once you have selected the file in the File Name field.

Layer Order

To change the rendering order for users of Dynamic Maps, select the map layer. Use **Up** and/or **Down** to move the layer up and down in the list. The layer rendering order will change accordingly. Click "**Update**" to update the database so that the change in layer order is reflected in the system.

> The rendering order of map layers in Dynamic Web Maps is set when you define a Topic.

Map Layer Order Tips

It is important when adding layers to a warehouse to consider how each layer will appear relative to other layers. By controlling the layer order in the Map Layers list users will see the map layers displayed in the best possible way. Some rules-of-thumb to follow:

1. Label Features should appear first on the list since if turned on, it is important that the entire label is displayed and not covered up by a point, line or polygon layer.
    - Some thought should be given to the priority of each label layer (i.e. should Provincial Capital Names appear on top of District Names?)
2. Point Layers should appear below Labels and above Line Layers.
    - Whenever point symbols in one layer might coincide with point symbols from another layer, place the point layer with the smaller symbol on top.
3. Line Features should appear below Point Layers and above Polygon Layers.
    - Whenever line features in one layer might coincide with line features from another layer, place the layer with the smaller line width on top.
4. Polygon Features should appear below Line Layers and above Imagery.
    - All transparent Polygon Layers should be at the top of the group of Polygon Layers. Translucent (Highly Transparent, Somewhat Transparent, Slightly Transparent) filled or patterned layers should be next. The Solid Fill Polygon layers should be last in the order.
5. Image / raster layers are like Solid Fill Polygon layers. They should appear at the bottom of the list.

Of course, the order of map layers may also depend somewhat on logical sequencing and scales at which map layers are displayed. You may, for example, want to have the labels for a layer next to the layer itself – just be aware of the difficulties this may cause.

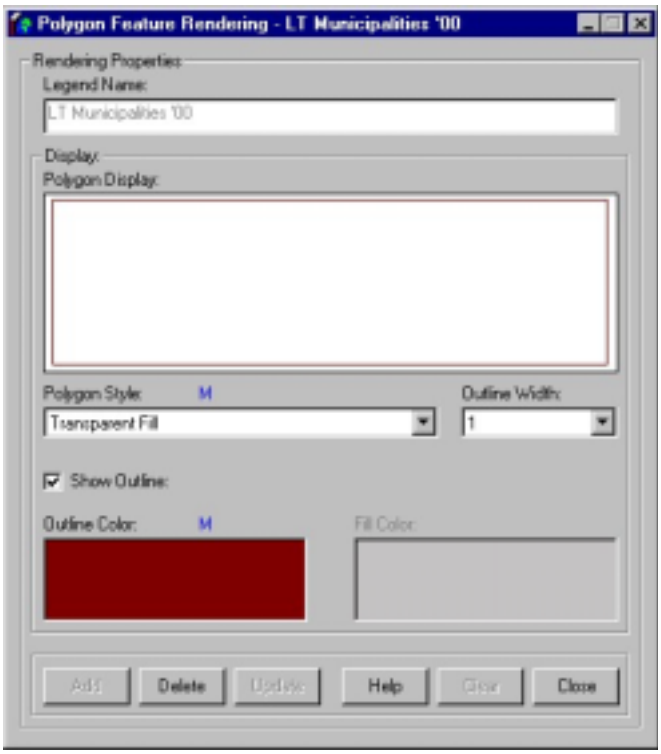
Introduction to Rendering

The rendering properties you assign determine how the layer's features will appear on the map to the user. Dynamic Maps users are able to reassign these rendering properties. For web-based applications, the user cannot change the rendering properties for the layers.

There may be two instances when you want to delete the rendering properties:

- Deleting resets the layer in the Map Layers form so that either Label rendering or Point / Line / Polygon rendering can be assigned to the map layer.
- If you no longer want the layer to appear to the users of Dynamic Maps / Dynamic Web Maps, but you want to keep the layer registered in the database.

Rendering Polygon Layers



Legend Name

The legend name reflects the map layer name and is set on the main Map Layers form. The name will appear in the legend of the map for Dynamic Maps and Dynamic Web Maps users.

Polygon Style

The Polygon style includes the various types of fills available from solid to transparent including hatching – type fills and various levels of transparency.

Outline Color, Width and Show Outline

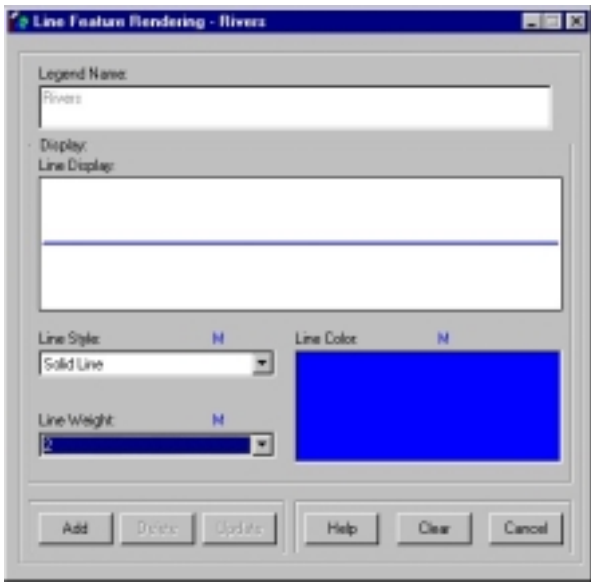
The outline color reflects color of the boundary line for each polygon. The Outline Width field sets how wide the outline will be and the Show Outline checkbox indicates whether or not the outline appears at all. If you have a transparent fill, you cannot turn "show outline" off.

Fill Color

The fill color reflects the color used for the solid fill and any hatched-type fill. For transparent-type fills, the color is made highly, somewhat or slightly transparent.

There is a bug in Windows 98 that prevents line widths greater than 1 to be displayed properly in the Map Composition and when copying a View. If the warehouse is being used by Dynamic Maps on a Windows 98 machine, then use an outline width of 1 for all polygon map layers. This is not a problem under Windows'95 / NT / 2000.

Rendering Line Layers



Legend Name

The legend name reflects the map layer name and is set on the main Map Layers form. The name will appear in the legend of the map for Dynamic Maps and Dynamic Web Maps users.

Line Display

The Line Display field shows you how the line feature will appear once you have set the Line Style, Line Weight, and Line Color.

Line Style

A line feature can appear as a solid line, a dash line, a dotted line, a dash-dot line, or a dash-dot-dot line. All lines on the layer will appear with the same style.

Line Weight

The line weight sets the thickness or width of a line in pixels. All lines on the layer will appear with the same weight.

Line Color

Click on the color swatch to set the color of the feature. You have an unlimited choice of colors via the standard Windows color palette. All lines on the layer will appear with the same color.

There is a bug in Windows 98 that prevents line styles other than solid with a width greater than 1 to be displayed properly in the Map Composition and when copying a View. They will all appear as solid lines. If the warehouse is being used by Dynamic Maps on a Windows 98 machine, then use an outline width of 1 for all lines that are not solid. This is not a problem under Windows'95 / NT / 2000.

Rendering Point Layers

Legend Name

The legend name reflects the map layer name and is set on the main Map Layers form. The name will appear in the legend of the map for Dynamic Maps and Dynamic Web Maps users.

Symbol Type

Available symbols include: circle, square, triangle, cross, true type font. When you use a True Type font, you will need to define the font, rotation, and symbol characteristics.

Point Size

Point sizes range from 6 up to 24.

Color

Click on the color swatch to set the color of the feature. You have an unlimited choice of colors via standard Windows color palette. All points on the layer will appear this color.

Font

The Font field is available when Symbol Type = “True Type Font”. The available fonts reflect all the fonts registered on your system, such as: Arial, Courier New, Monotype Sorts, MS Sans Serif, Times New Roman and Wingdings.

If you are registering a point layer in a warehouse that will be broadly distributed, make sure you use a font that will be available to all users. If the font you choose is not available to someone accessing the warehouse with Dynamic Maps, the resulting display will be unpredictable (the system will use another font that is available locally).

Rotation

Rotation is available when Symbol Type = “True Type Font”. Use this field to define any rotation for the true type font symbol. Rotation can vary from 0 to 345 in 15 degree intervals and rotation occurs counter-clockwise.

Symbol

Symbol is available when Symbol Type = “True Type Font”. Use this field to select the symbol that will be used. The various symbols change based on the font chosen.

Rendering Label Layers

Legend Name

The name will appear in the legend of the map for Dynamic Maps and Dynamic Web Maps users. Same as Layer Name.

Font Name

The fonts available reflect the fonts on your system, such as: Arial, Courier, Courier New, MS Serif, MS Sans Serif, System, Terminal, Times New Roman.

If you are registering a point layer in a warehouse that will be broadly distributed, make sure you use a font that will be available to all users. If the font you choose is not available to a user, the resulting display will be unpredictable (the system will use another font that is available locally).

Font Size

Select the size of the label from 6 points to 24.

Font Color

Click on the color swatch to set the color of the feature. You have a virtually unlimited choice of colors.

Alignment

- **Horizontal**: Use the Center, Left, or Right horizontal alignment to determine where the label will appear relative to the feature.
- **Vertical**: Use the Center, Top, Bottom vertical alignment to determine where the label will appear relative to the feature.
- **Rotation**: You can choose a rotation from 0 to 345 degrees in 15-degree intervals counter-clockwise.

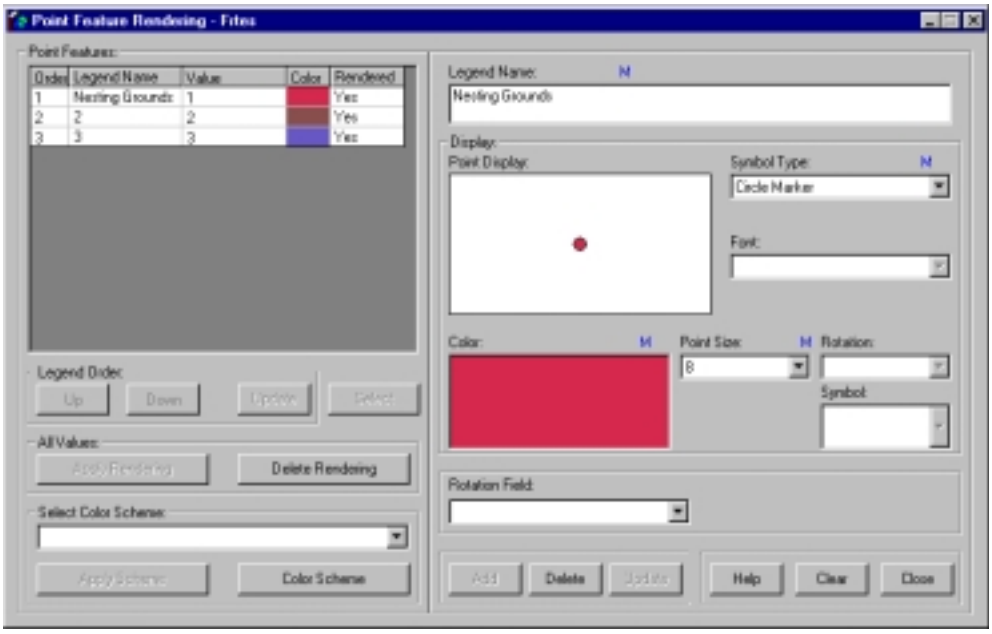
The following four fields are used in special cases in which the map layer has been created with special rendering fields for labels as part of its .dbf file.

- **X Offset Field**: Works based in map units (either decimal degrees or meters – depending on the warehouse type and whether the map layers have been projected in meters or decimal degrees.) The X-Offset describes the number of map units to the left or right of the feature center that the label will be placed. Select the .dbf table field name from the drop-down list that contains the offset values for each feature label.
- **Y Offset Field**: As with the X-Offset, the Y-Offset describes the number of map units above or below the center of the feature that the label will be placed. Select the .dbf table field name from the drop-down list that contains the offset values for each feature label.
- **Rotation Field**: The rotation field is the number of degrees rotation identified for the label layer. Select the .dbf table field name from the drop-down list that contains the rotation values for each feature label.

- **Text Height Field**: The text height field describes the size of the text in the label. When the Text Height field is used the Font Size is ignored. Select the .dbf table field name from the drop-down list that contains the text height values for each feature label.

To clear any of these fields, use the ESC key.

Using Value Rendering



Value rendering can take time to set up, but the results are often very effective and greatly enhance the user's understanding of the data. Value Rendering enables you to show each of the map layer's unique feature groups (values) with a different color and legend name.

> Once you've assigned a layer as Value-rendered, you cannot change it back to Single rendered without re-registering the layer.

To use Value Rendering on a map layer, you must first set the Rendering Method to "Values" and then select the appropriate field from the Value Rendering Field.

When you do, the Rendering forms look a bit different – since you can now assign individual colors, shapes, rotations, etc. for each of the features in the layer. Below is a sample screen capture of the value rendering form for a point layer.

To apply value rendering to a map layer, select the value from the Feature (the above form is for a Point feature map layer) list by clicking on it and then clicking **Select**. Type a name in the Legend Name field, and then assign the rendering properties as you would normally. When you click **Add**, that feature will now appear in the list with its associated legend name, color, and order number. Simply repeat this process until all the values that you want to have rendered have been assigned rendering properties.

You can change the order in which they appear in the legend by selecting a feature and using **Up** and **Down**. Once they are in the order you would like, remember to click **Update**.

> You may not always want to have all the different values rendered, i.e., you do not have to specify rendering properties for all the values. For example, there may be a specific type of feature, such as particular species habitat, in a map layer of various habitats that you do not want to have rendered. If you decide not to render a particular value(s), Dynamic Maps and Dynamic Web Maps will not display the associated features.

To turn off a feature type and prevent it from being rendered, select it and hit the “Delete” button. This only deletes the currently assigned rendering properties and not the actual features from the map layer.

Quick Value Rendering Approach

A quick way to do value rendering is to use the default colors assigned by system or select a colour scheme and apply the scheme. Define the **Polygon (or Line) Style,** and click the **Apply Rendering** button. This will apply these rendering characteristics to all the values. You can then edit the **Legend Name** values individually if you need to.

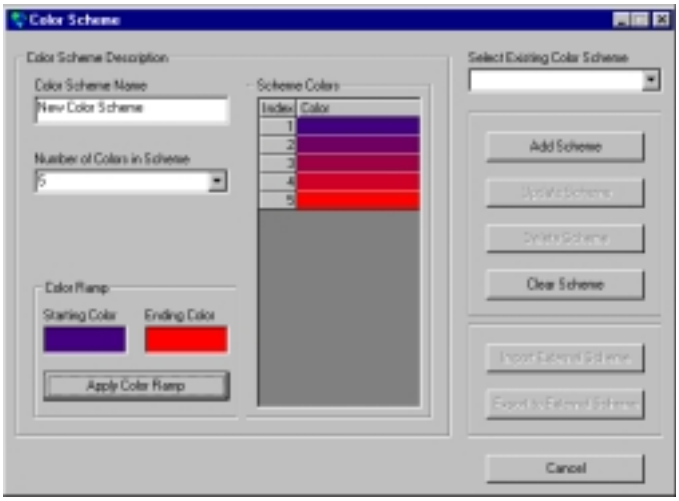
Color Schemes

To help value render Point, Line and Polygon layers, you can set up and use Color Schemes. Either select a color scheme from the list and click **Apply Scheme**, or click the Color Scheme button to define your own. Remember to click Apply Scheme in order for changes to take effect – otherwise you will only be previewing the changes.

Defining a Color Scheme

Dynamic Knowledgebase enables you to manage the color schemes you use for creating value rendered maps for **polygon map layers**. There are several functional components and options for creating, editing, and managing your color schemes.

The Color Scheme functionality is available from the Polygon Feature Rendering form by clicking on the Color Scheme button.



Creating a New Color Scheme

Give the scheme a name in the **Color Scheme Name** field. Identify how many colors you want in the scheme. When you do this, the **Scheme Colors** field gets filled with black swatches.

The fastest approach to creating a scheme is to use the **Color Ramp** feature. By clicking in the **Starting Color** and **Ending Color** fields you can define a ramp's beginning and end colors. Clicking **Apply Color Ramp** populates the other colors in the **Scheme Colors** display list.

If you want to change a color in the **Scheme Colors** display list simply click on the color and choose a new one.

When you are satisfied with the colors chosen, click Add Scheme.

Use Clear Scheme to clear the information on the form and start again. Any unsaved changes will be lost.

Note: The color scheme function only sets the fill color parameter for the rendering properties of each class. Other rendering properties, such as transparency and outline need to be set manually for each class if you want them changed from their default setting.

Managing the Color Scheme:

If you want to edit an existing color scheme, use the **Select Existing Color Scheme** dropdown list to select the color scheme. Each color in the scheme can be edited simply by selecting it in the **Scheme**

Colors field. If you need to update the number of colors in the scheme, you must update the number of colors first and then re-select the scheme before you can assign values to those new colors. If you want to update the name of the color scheme, type a new name in the **Color Scheme Name** field. After editing, select **Update Scheme**.

Delete Scheme

To delete a scheme from the list of available color schemes, select the scheme and then click **Delete Scheme**.

Importing a Color Scheme

To import a color scheme file, clear the form and click **Import External Scheme** button. Locate the <scheme name>**.clr** file on your computer or on the network and click Open. The Color Scheme will have to have a unique name -- which may mean renaming an existing color scheme before importing if there are two schemes with the same names.

Exporting a Color Scheme

You can share the color schemes you create with your colleagues. To export a color scheme to an external file, select the color scheme you want to export and click "**Export Color Scheme**". Give the theme a file name and location and click Save. You do not need to specify the file extension of .clr – the system does that automatically.

Map Layer Metadata

> See Appendix 4 for a description of metadata, the standard used in this system, the metadata capture / input process.

The metadata for map layers are divided among three tabs – the Identifier Information tab, the Content Information tab, and the Spatial Information tab. Although metadata are not mandatory, they are HIGHLY RECOMMENDED.

When you input metadata, you must populate each mandatory field before the system will accept the record i.e. the system will not accept an incomplete record. If you do not have all the information you need readily available, then use placeholder text in the field and come back to complete the metadata later.

It is always a good idea to ensure the owner or creator of the map layer supplies you with the appropriate metadata for that map layer. Make sure they are aware of the mandatory fields, and have them provide you with this information.

You can copy (Ctrl-C) and paste (Ctrl-V) information from another source into a metadata tab field. Try to do so if there is a lot of information to incorporate and that information is available in another electronic format elsewhere from the map layer owner / creator.

Identification Information:

This Identifier Information tab fields describe properties of the map layer such as the owner / originator, and contact information. The mandatory fields include Contact Name / Position, Contact Organization and Telephone.

> The *Email address* is particularly useful in the web site, since it provide a direct link to users to the contact person and the information.

Content Information:

The content information tab fields describe the characteristics of the map layer. One key field on this tab is the "Abstract". Since it is a free-text field, descriptive information can be cut from other relevant documents and pasted here to help save time in inputting. The other mandatory field is the Creation Modification Date (year).

> The *Abstract* field in the Content Tab is one of the most important fields in the system. Use it to describe the content of the map layer and provide other relevant information about the methodology used to collect the data. Since you may not know this information, make sure the owner / creator of the map layer provides it to you.

Spatial Information:

The spatial information tab provides fields for describing the geographic extent and positional accuracy of the map layer. The Mapping Coordinate Units, Reference Ellipsoid and Map Projection fields will be input by the system automatically – reflecting the warehouse settings.

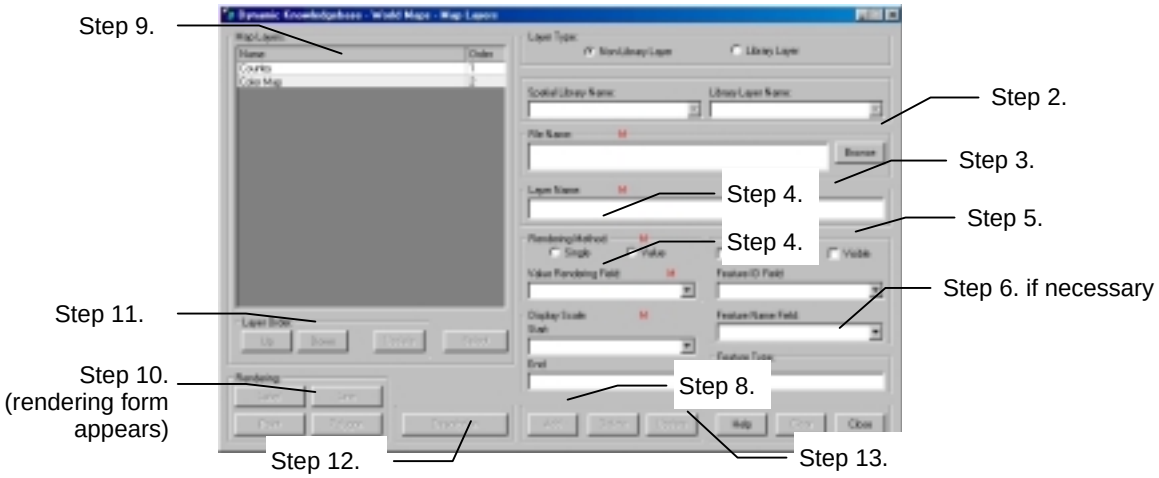
Quick Guide to Registering a Non-selectable, Non-tiled Map Layer

1. Using Windows Explorer move the map layer's files into the Map Layers directory under the appropriate warehouse directory if they are not already there. Launch the Dynamic Knowledgebase tool and select the appropriate warehouse name and then the "Map Layers" function.
2. Click **Browse** in the File Name field to navigate to the appropriate map layer file via Network Neighborhood or via My Computer if a local install.
3. Give the layer an appropriate Layer Name.
4. Set the Rendering Method field to Single or Value, and if value rendering, identify the appropriate Value Rendering Field. To see the fields in the map layer's .dbf file, click the View Attributes button.
5. Indicate whether or not the layer will be visible by default to users of Dynamic Maps in the "Visible" check box.
6. If you intend to use this layer as a Label layer, select the appropriate field from the Feature Name Field drop-down box. Otherwise this step is unnecessary. To see the fields in the map layer's .dbf file, click the View Attributes button.
7. In the Display Scale fields, input the scale at which the map layer will be displayed and then the scale at which the map layer will no longer be displayed. Test these settings once the map layer is registered using Dynamic Maps. Check the scale readings at the bottom of the Map View at the point at which you want the map layer to appear and/or disappear. Then go back and edit and update the Display Scale appropriately.
8. Click **Add** to add the layer to the list of Map Layers.
9. Select the layer from the Map Layer list, by double-clicking on it or by clicking on it and choosing **Select**.
10. Choose the appropriate **Polygon**, **Line**, **Point** or **Label** rendering properties button and define the layer's rendering properties.

    If VALUE RENDERING: Select each value in the list of possible values one at a time and give each a Legend Name and define each one's rendering characteristics. A quick way to do value rendering is to use the default colors assigned by system or select a colour scheme and apply the scheme. Define the **Polygon (or Line) Style,** and click the **Apply Rendering** button. This will apply these rendering characteristics to all the features. You can then edit the **Legend Name** values individually if you need to.

Once you have defined how the map layer should look, click **Add** to close the form and return to the Map Layers form.

11. Define the layer's rendering order by moving it **Up** and/or **Down** in the list of map layers. Click **Update** once the Layer Order is correct.
12. Click **Metadata** and describe the metadata for the layer.
13. Click **Update** to complete the layer registration.
14. Test the settings using Dynamic Maps and update as necessary.

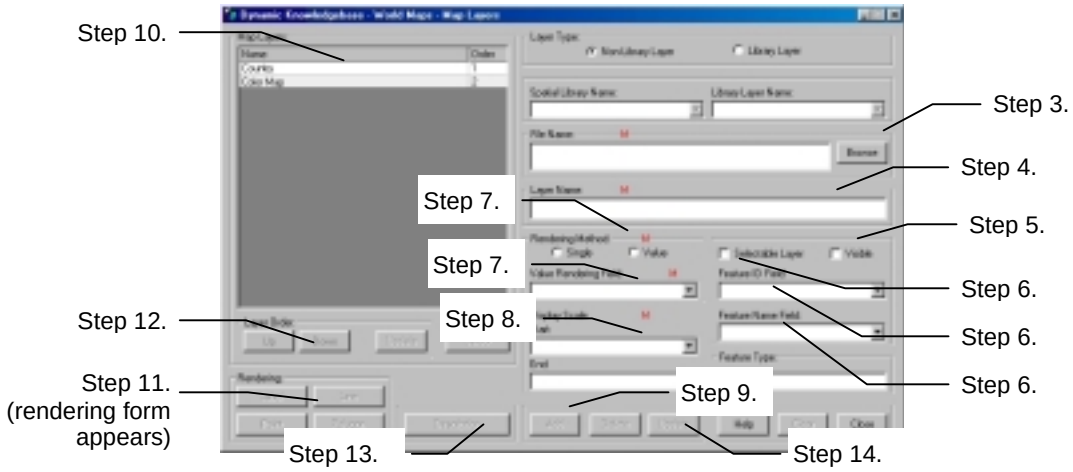


Quick Guide to Registering a Selectable Map Layer

1. Using Windows Explorer, move the map layer's files into the ***<warehouse name>Warehouse\Map Layers*** directory if these files are not already there. Feel free to put them in their own sub-directory so that they are more easily found and organized.
2. Launch Dynamic Knowledgebase and select the appropriate warehouse and then the Map Layers function.
3. Use **Browse** in the File Name field to navigate via Network Neighborhood (or via My Computer if a stand-alone application) to the map layer's file.
4. Give the map layer an appropriate Layer Name. It will appear in the list of available map layers in Dynamic Web Maps and Dynamic Maps and it is the default legend name.
5. Indicate whether or not the layer will be visible by default to users of Dynamic Maps in the "Visible" check box.
6. Check the "Selectable Layer" check box. Select the Feature ID Field and the Feature Name Field from the drop-down lists. To see the fields in the map layer's .dbf file, click the **View Attributes** button.
7. Set the Rendering Method field to Single or to Value, and if value rendering, identify the appropriate Value Rendering Field.
8. In the Display Scale fields, input the scale at which the map layer will be rendered and then the scale at which the map layer will no longer be rendered. Test these settings once the map layer is registered using Dynamic Maps. Check the scale readings at the bottom of the Map View at the point at which you want the map layer to appear and/or disappear. Then go back and edit and update the Display Scale appropriately.
9. Click **Add** to add the layer to the list of Map Layers.

10. Select the layer from the Map Layers list, by double-clicking on it or by clicking on it and choosing **Select**.
11. Depending on what type of layer it is, choose the **Polygon**, **Line**, or **Point** rendering properties button and define the layer's rendering properties.

    If VALUE RENDERING: Select each value in the list of possible values one at a time and give each a Legend Name and define its rendering characteristics. A quick way to do value rendering is to use the default colors assigned by system or select a color scheme and apply the scheme; the define the **Polygon (or Line) Style**; and click the **Apply Rendering** button. This will apply these rendering characteristics to all the features. You can then edit the Legend values individually if you need to. Click **Add** to close the form.
12. Define the layer's rendering order by moving it **Up** and **Down** in the Map Layers list. Click **Update** once the Layer Order is correct.
13. Click **Metadata** and describe the layer. This is <u>particularly important</u> for selectable map layers.
14. Click **Update** to complete the layer registration.
15. Test the settings using Dynamic Maps and update as necessary.



Quick Guide to Registering a <u>Label</u> Layer

> You can register a layer as a "Label" layer only if that map layer does not have its rendering properties set. You can, however, delete the rendering properties associated with a point, line, or polygon map layer so that you can then register the layer as a label layer.

1. Add the map layer as you would for any non-selectable polygon, line, or point layer, but don't define Rendering properties. Include the **Feature Name Field** when you define the map layer.
2. Select the layer from the Map Layers list. The **Label** button will now be active. Click on it and complete the Label Rendering form to assign the appropriate rendering properties:
   - provide a Legend Name,
   - define the font that will be used when rendering the label,
   - define the color and size of the font,
   - define the Alignment of the label and its position relative to the feature it describes.
   - Click **Add** to complete the definition.
3. Define the layer's rendering order by moving it **Up** and **Down** in the Map Layers list. Click **Update** once the Layer Order is correct.

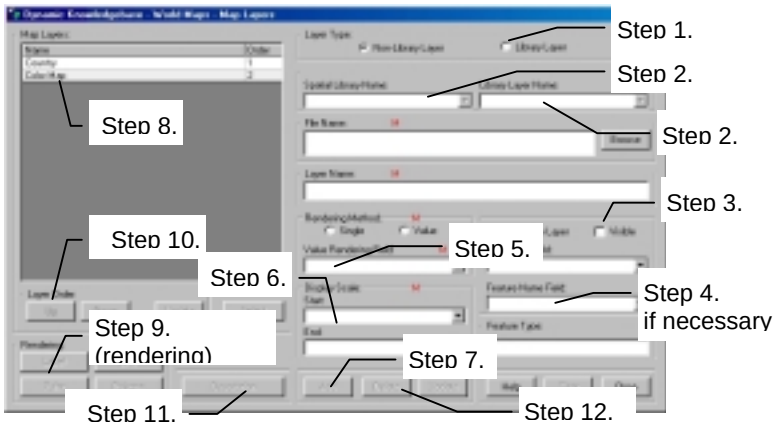
4. Click **Metadata** and describe the layer.
5. Click **Update** to complete the layer registration.
6. Test the settings using Dynamic Maps and update as necessary.

> Tip:
> Give the label layer a name that reflects the fact that the map layer contains only names/labels e.g. River Names, District Capital Names, etc.

Quick Guide to Registering a Tiled Map Layer

The layer must first be set up using the Dynamic Knowledgebase **Spatial Library** function.

1. On the Map Layers form, click the Library Layer check box.
2. Select the name of the library from the drop-down Spatial Library Name list, and then select the name of the layer you are adding from the Library Layer Name list.
3. Indicate whether or not the layer will be visible by default in the "Visible" check box.
4. Optional: If this will be a Label Layer select the Feature Name Field from the drop-down listings. This will be the "label" as it appears on the map.
5. Set the Rendering Method field to Single or Value, and if value rendering, identify the appropriate Value Rendering Field.
6. In the Display Scale fields, input the scale at which the map layer will be rendered and then the scale at which the map layer will no longer be rendered. Test these settings once the map layer is registered using Dynamic Maps. Check the scale readings at the bottom of the Map View at the point at which you want the map layer to appear and/or disappear. Then go back and edit and update the Display Scale appropriately.
7. Click **Add** to add the layer to the list of Map Layers.
8. Select the layer from the Map Layers list, by double-clicking on it or by clicking on it and choosing **Select**.
9. Depending on what type of layer it is, choose the **Polygon**, **Line**, or **Point** rendering properties button and define the layer's rendering properties. IF VALUE RENDERING: Select each value in the list of possible values one at a time and give each a Legend Name and define its rendering characteristics. Click **Add** to leave the form and return to the Map Layers form.
10. Define the layer's rendering order by moving it **Up** and/or **Down** in the list of map layers. Click **Update** once the Layer Order is correct.
11. Click **Metadata** and describe the layer.
12. Click **Update** to complete the layer registration.
13. Test the settings using Dynamic Maps and update as necessary.



Quick Guide to Registering an Image / Raster Layer

Image / raster layers are registered in the same way as a tiled map layer, with the following special considerations:

- When you "add" the layer, you are not looking for a shape or ArcInfo coverage file, but a tiff, .bil, .sid or .bmp file.
- There are no rendering qualities associated with image / raster layers. As a result, the Rendering section of the form is disabled.
- If the image / raster layer is not tiled (i.e. it is simply one image covering part of the geographic extent), then you do not need to identify the spatial library.

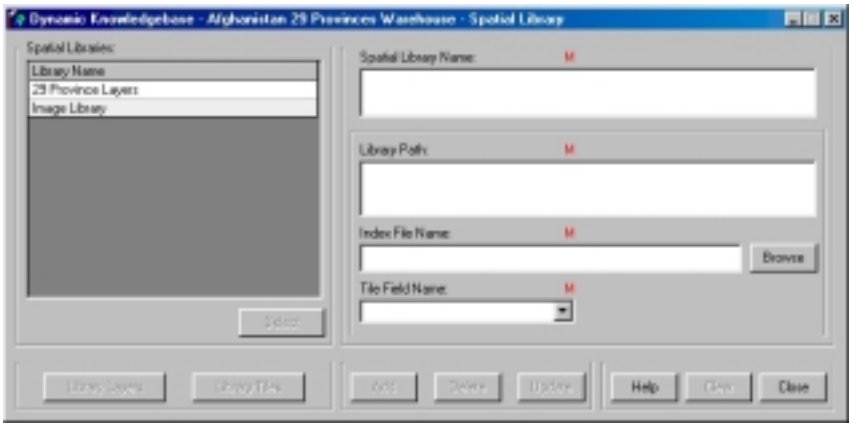
Tips:

- The Display Scale is particularly important since you do not want to zoom in beyond the image's recognition capability, and you do not want the image being displayed at too small a scale so that it is unrecognizable. Test the settings once the map layer is registered using Dynamic Maps. Check the scale readings at the bottom of the Map View at the point at which you want the image to appear and/or disappear. Then go back and edit and update the Display Scale appropriately.
- Generally, you will want to set "Visible" off since an image / raster layer will obscure many other features and could take significant system resources to display. Also, if there is more than one image / raster layer registered, only one can be visible at the same time for any particular geographic area.
- Finally, ensure the image / raster layer is near or at the bottom of the Map Layers list so that other layers, especially vector map layers are rendered on top of it.
- There is no limit to the number of image / raster layers you can register, however, as noted above, only one image / raster layer should be visible at a time in any particular geographic area, i.e., image / raster layers are not transparent and therefore anything below them will not be visible.
- Metadata are particularly important for image / raster layers. Use metadata to describe the type of image (aerial photograph, interpreted remotely sensed image, etc.), how and when it was taken, at what resolution, etc.

Spatial Library

The Spatial Library function enables you to manage the spatial libraries for the tiled map layers in the warehouse.

Spatial Library Forms Field Definitions



Spatial Libraries

Lists the available spatial libraries by Library Name. To select a library, click on its name and click **Select**, or double-click on the name.

Spatial Library Name

The name you define for the library. This name will appear in the Dynamic Knowledgebase Map Layers component in the Spatial Library Name field when registering library layers.

Library Path

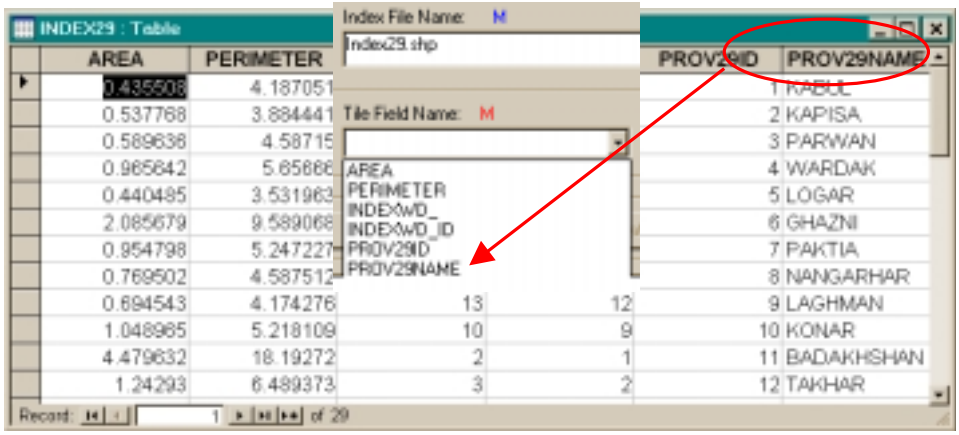
The path associated with the spatial library's index file. It will automatically appear once you have added the library File Name.

Index File Name

The name of the Index file. This ".shp" file contains the master registry identifying each tile and defining where all the map layers for each tile are located. Click **Browse** and navigate via Network Neighborhood when defining the file name – unless you are setting up the warehouse for local use, in which case use local path names.

Tile Field Name

Each polygon in the Index file's shape file stores a field used for the Tile Field Name. This drop-down list will contain the fields in the index file's .dbf file. The tile field name is the name that reflects the names of the directories in which the associated map layers' tiles are stored. See the graphic below for an illustration of this.



Updating and Deleting a Library

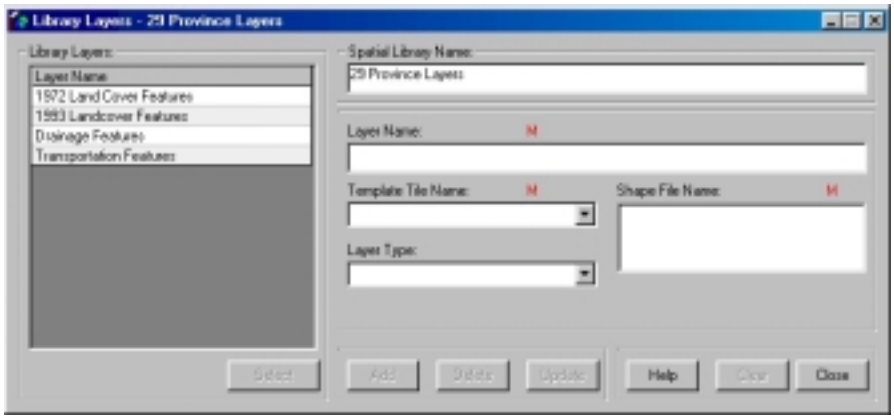
When you are updating a Spatial Library, select the library from the list. Make the necessary changes – which may involve a new Spatial Library Name, or location of the index file and click **Update**.

When you are deleting a Spatial Library, select the library from the list. Make sure this is the right library click **Delete**. Any references to tiled map layers associated with this library will also be deleted.

> Deleting a Spatial Library does not delete the files associated with the library, it only deletes the registration information in the warehouse. Therefore, these files are still available if you need to recreate the library. In order to delete the actual files, you could use Windows Explorer. If you do, YOU MUST delete the Library first from the system using the Dynamic Knowledgebase.

Library Layers

Once you have created a Library definition you now need to register the Library Layer. Select the **Library Layer** button on the main form.



Library Layer: Layer Name

This is the list of map layers associated with the library. To select a map layer from this list, double-click on it or highlight it and click **Select**.

Spatial Library Name

This field is simply a reminder of the spatial library with which you are working.

Layer Name

This is a name you assign to the layer. It will be the name used on the user's map legend and within the Dynamic Maps and Dynamic Web Maps system of available map layers.

Template Tile Name

The Template Tile Name is a drop-down listing of all the available tiles associated with the library. These tiles are actually set up as directories within the system and those directories contain the relevant tiled map layer shape files. You use this list to pick one as a "template" from which the layer's shape file is then chosen.

Your library may have tiles that contain shape files for several map layers. For example, you may have a tiled rivers layer, elevation layer, and roads layer. For those map layers that have complete coverage they will have relevant .shp files in each of the tiles. However, if some map layer do not cover the full extent of the library (there are holes in the coverage), you'll need to make sure you choose a template tile that contains the shape file you need, i.e., don't choose a tile where the map layer does not exist.

> You may encounter tiled map layers that do not have any data in the Template Tile Name that you selected – perhaps because that tile represents a geographic region that does not have data for that map layer. You will need to check / know an appropriate Template Tile for your tiled map layers.

Layer Type

The layer can be either a vector layer in .shp format, or an image / raster layer in .bmp, sid, .bil or .tiff format.

Shape File Name

The map layer has the same name in each tile with which it is associated. For example, a tiled national transportation map layer might have a file with the name "tnat.shp" in each of the tiles. Select the shape file that you want to associate with the named map layer.

Updating and Deleting a Tiled Layer in the Library

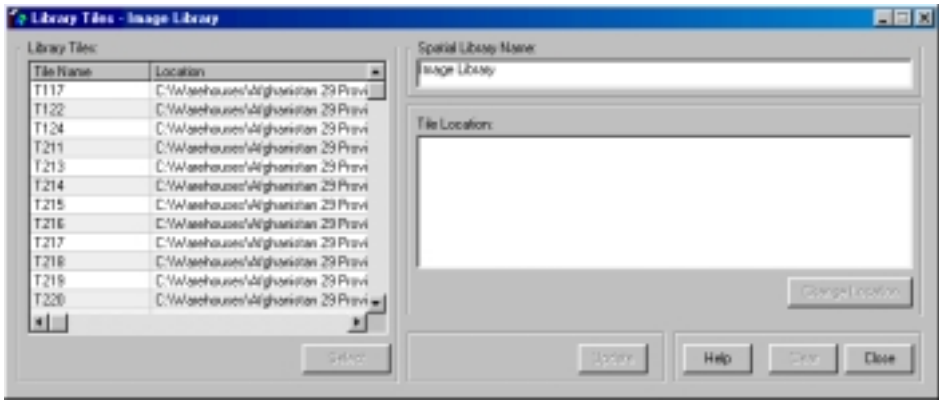
To update the tiled map layer, select it from the list. Make the appropriate changes to either the Layer Name or the Shape File Name and click **Update**.

To delete a tiled map layer, select it from the list. Click **Delete**.

> Deleting a tiled map layer does not delete the files associated with the layer, it only deletes the registration in the warehouse. Therefore, these files are still available if you need to re-reference the layer.

Library Tiles

The GIS technician has set up the tiles and given them names reflected by the directory in which the file(s) are put. Depending on the map layer extent and the tiling scheme used, there can be many tiles within a Library.



The **Library Tiles** function enables you to store tiles over multiple drives or in various locations on the server. This is important when the warehouse gets very large and the tiles need to be distributed across multiple drives. As a result, it is easy to split up a library without re-registering each layer. In general, though all these tiles will be under the same directory.

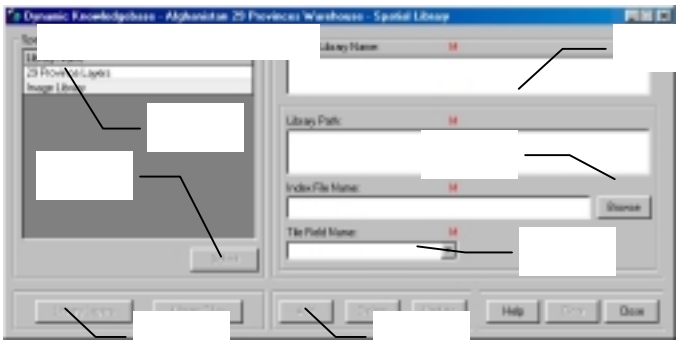
> If you are low in disk space and you want to distribute the tiles, you must move the tiles yourself using Windows Explorer. Dynamic Knowledgebase does not move the actual files, it enables you to point to the new location of the files if they have changed.

To indicate the location of tiles, you can select the tiles one at a time or in groups. To select one, click on it and it will become highlighted. Then click **Select**. To select more than one tile click on one name and then click again while holding the SHIFT key to select all tiles from the first selected tile to the tile you are now selecting. Then click **Select**.

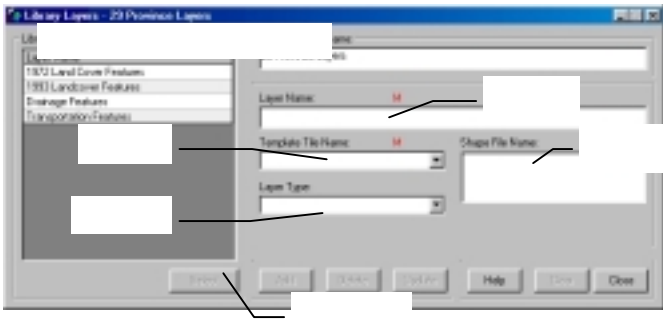
The Spatial Library Name field simply reminds you what Library you are working with. The Tile Location field will indicate the location of the tile directory as it was previously set. Use **Change Location** to navigate to the directory in which the tiles are stored. Note, you must Open one of the tile folders for the actual path to be identified properly – up to the library level.

Quick Guide to Adding a Spatial Library

You must have an appropriate library directory structure set up. The following instructions assume that all the Library Tiles are within the warehouse directory structure. If not, you will need to use the Library Tiles function before Step 6. so that the system knows where the tiles are physically located.



1. From the Main Menu, select "Spatial Library".
2. Type the name of the library in the Spatial Library Name field. If you aren't sure of the name or you want to see what library(s) is available, come back to this step after step 4.
3. Locate the library's index shape file by clicking **Browse** in the Index File Name field and navigating to the file via Network Neighborhood if setting up the warehouse for use across a LAN. Select the file and its path will appear in the Library Path field.
4. Select the Tile Field Name from the drop-down list.
5. Click **Add** and the library will appear in the Spatial Library list.

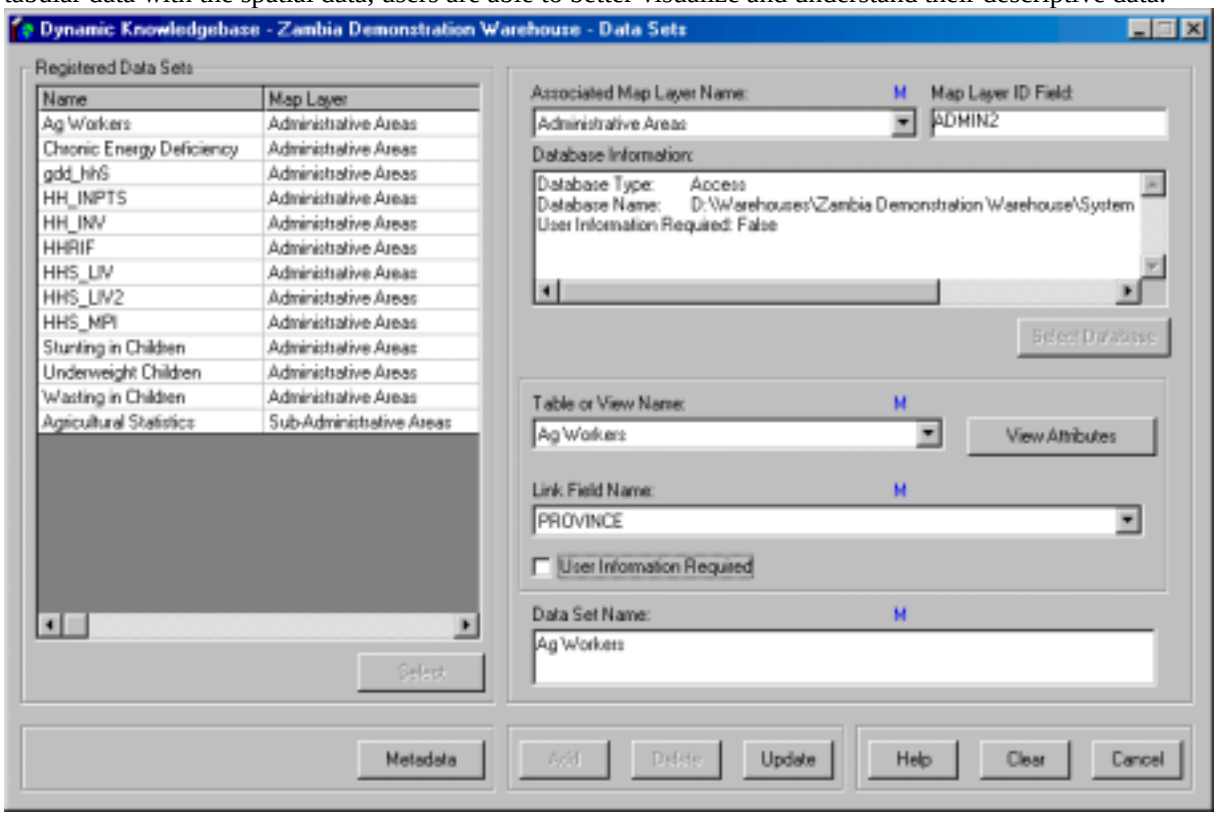


6. Select the library from the Library Name list by clicking on it and clicking **Select**. Select **Library Layers**.
7. Choose a Tile from the "Template Tile Name" drop down list. Make sure that you select a tile that has all the different tiled map layers for that Library, i.e., do not select a tile that has no data or just partial data. Then select the Layer Type. This can be a vector or image / raster layer.
8. The available tiled map layers for this template tile now appear in the Shape File Name field. Select the appropriate map layer.
9. Provide a descriptive name for this tiled map layer in the Layer Name field.
10. Select **Add** and the layer is now available to be registered with the system through the Map Layers function accessible in the main form.
11. Repeat this process if needed to register any remaining tiled map layers within this library.

SKE

Data Sets

The **Data Sets** function enables you to register tabular data sets with the system. You must first register an appropriate selectable map layer before a related tabular data set can be registered. By incorporating tabular data with the spatial data, users are able to better visualize and understand their descriptive data.



Data Sets Form Field Definitions

Registered Data Sets

The Registered Data Sets field lists the currently registered data sets available to the system along with their associated Map Layer. To select a data set from the list, double-click on its name or click on it and click **Select**.

Associated Map Layer Name

The Associated Map Layer Name is a drop-down list of the selectable map layers registered in the warehouse. When you associate the data set to the map layer, you must be sure that the map layer has the appropriate Feature ID Field set up to match the Link Field Name from the data set. If it does not, the spatial association of map features with tabular data records will not work. To help this matching process, when you select an Associated Map Layer, its Feature ID field will be displayed in the Map Layer ID Field.

Map Layer ID Field

A read-only field that displays the name of the Feature ID field for the selected map layer.

Database Information

The Database Information field contains a number of pieces of information: the Database Name (and path), the Database Type, and the “User Information Required” indicator.

Database Name

This is the actual file name and path for the data set. Use **Select Database** to begin the process of defining the database as one of an Access, Excel, dBase or Oracle database. (See below for a description).

Database Type

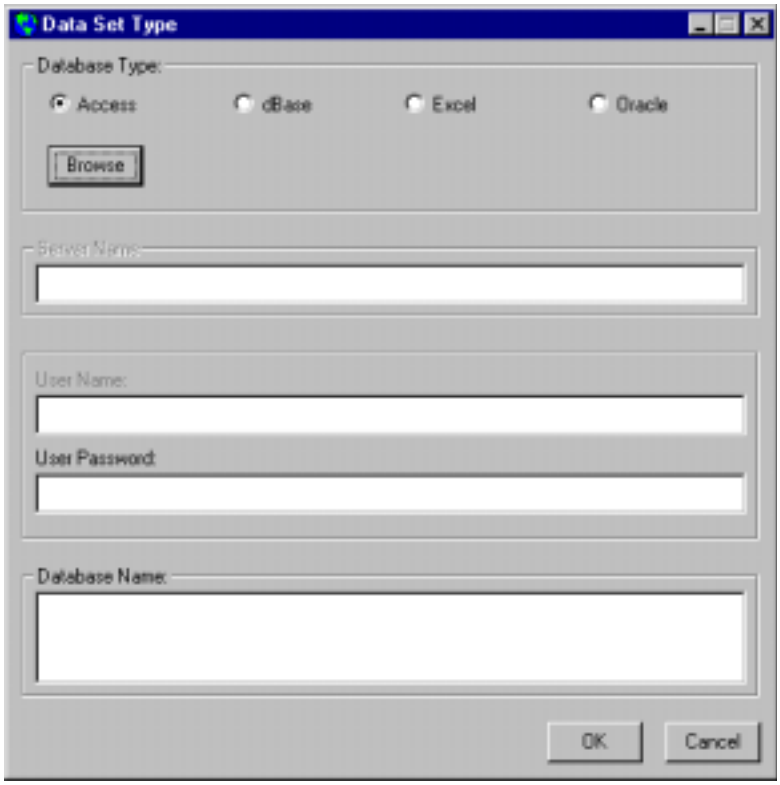
The database type indicates if the database is Access, Excel, dBase or Oracle.

Selecting the Database

Different types of data are registered slightly differently.

Access, Excel, dBase Databases

Select the database type and click **Browse** to locate the database on your machine or network. If an Access dB's has a password then you will also need to specify this in the **User Password** field. Click **OK** to go to the registration form.



Connecting to an Oracle Database

If you are defining a connection to an Oracle database, the Browse button will not be active, but you will need to input the Server (Service) Name, as well as your User Name and User Password.

Considerations for Importing an Excel Spreadsheet:

When importing an Excel spreadsheet, the system will assume that the first row of the spreadsheet contains your field headings.

Any OLE objects in the spreadsheet will not be shown in the grid – such as graphs.

When matching on a non-numeric field, sometimes a record in Excel will look like it's spelled correctly, but in fact there is a blank space before or after the name that you can't see. If so, the link with the map layer won't be successful. Sometimes the words are spelled slightly differently, and sometimes there is different capitalization. **Advanced Considerations**: In cases like this, you should actually import the excel spreadsheet into Access. You'll now see much more clearly if there is a blank space. Also in another table, import the map layer's .dbf file into the same database. If you do a join of the two ID fields (by creating a query) - depending on how the join is done you can tell what records are exactly the same and what aren't (and therefore what won't link).

Table or View Name

A data set can have several tables and views each covering a particular subject or type of information. When you have browsed to the database, Dynamic Knowledgebase will list the available tables and views within that data set in this field. Use this pull-down list to identify the table or view of interest.

> Dynamic Knowledgebase enables you to connect to database Views – so that you can get snapshots of data from multiple tables and even from other databases if the View has been set up with an appropriate ODBC connection (You can set-up an ODBC connection to an external database through M/S Access).

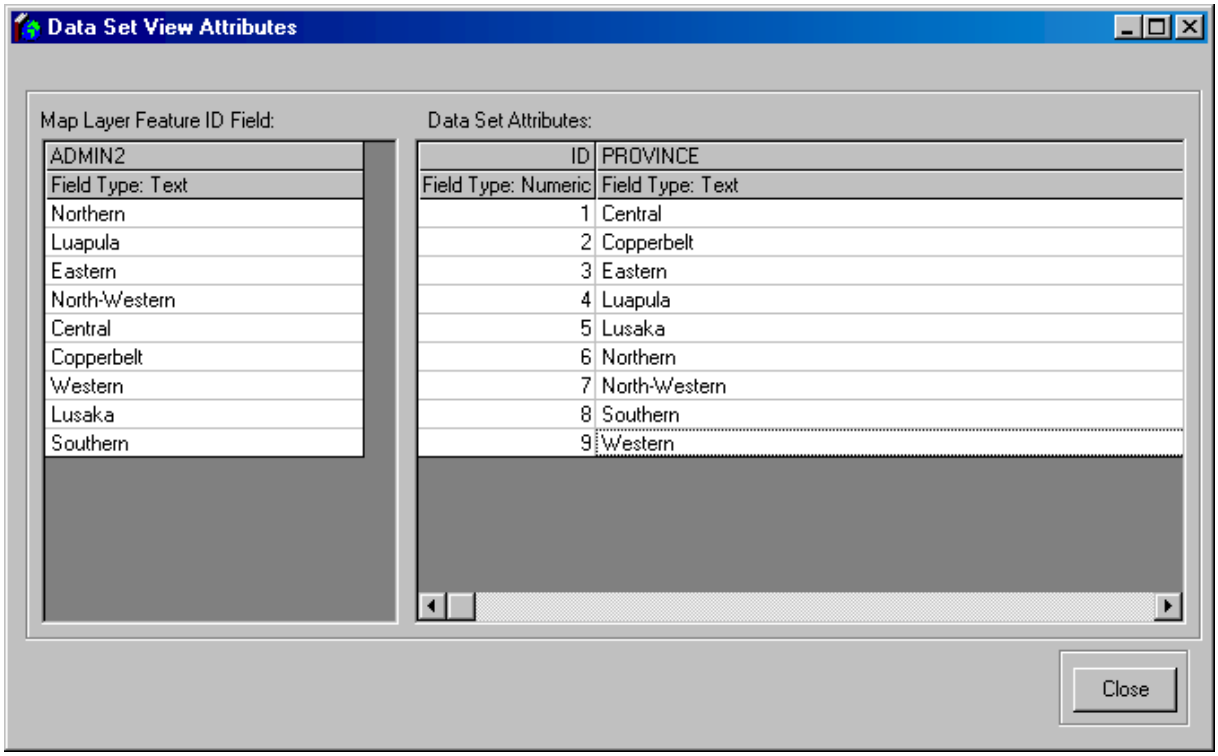
> You may find it easier to fill in this field prior to filling in the Data Set Name field.

SKE

View Attributes

The View Attributes button provides a view of both the Map Layer's Feature ID field its field type (e.g. numeric, text), and values; as well as the data set table's fields and their record values. This is very helpful to:

- Identify the appropriate field in the data set to use as the Link Field.
- Confirm a match between the Feature ID field in the Map Layer and the Link Field in the database.
- Identify / confirm that the number of features in the map layer match the number of features in the data set (won't always be the case).
- Identify errors, such as typos or other inconsistencies in the data that might prevent a match (e.g. the map layer may use "Northwestern" and the data set might use "North-Western").



For each field, you can right-click on the field names to get a variety of functions. For each field you can:

- **Sort Ascending and Sort Descending.** Choose one of these to sort the records in the Data Set Attributes table or the Map Layer Feature ID Field table based on the table and column over which your cursor was clicked. This helps identify the high and low extent of the records in that field and it works on both numeric and text fields.
- **Set Link Field**: When you right-click over the Data Set Attributes fields you can assign that field as the Link Field. This is a quick way to select the field that you want to use as the Link Field (see ***Link Field Name*** described below).

Link Field Name

The **Link Field Name** is the identification field that will tie each record to a selectable map layer. The link field values reflect the particular selectable map layer's Feature ID Field. So, for example, a data set that has information that reflects activities or findings in various provinces might have a "Province" ID field to match the record to the "Province" map layer's Feature ID Field. Use the **View Attributes** button to help ensure you assign the appropriate Link Field.

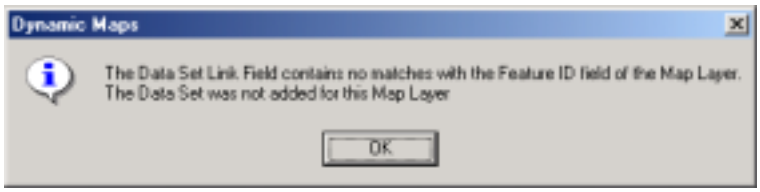
User Information Required

The **User Information Required** check box if activated will force the user to enter an appropriate username and password for the M/S Access database (only password required for Access) or Oracle service to which they are connecting.

Data Set Name

The Data Set Name is the name that you assign to the data set. This is the name that appears in the Dynamic Maps list of available data sets and in a Dynamic Web Maps data sets listing and report. Since one database might have several tables/views of interest and therefore might be registered more than once, choose a name that reflects the Table or View Name.

Note: Your data set can have some records that do not link to any features on the active map layer. This is perfectly acceptable. However, if there are no records in the data set that link to the active map layer features then the import will not be allowed. The following message is displayed:



Tabular Data Set Metadata

The data sets must first be registered in the system before their metadata can be input. The metadata for data sets are divided among three tabs – the Identifier Information tab, the Content Information tab, and the Spatial Information tab. Although metadata are not mandatory, they are VERY HIGHLY RECOMMENDED.

When you input metadata, you must populate each mandatory field before the system will accept the record i.e. the system will not accept an incomplete record. If you do not have all the information you need readily available, then use placeholder text in the field and come back to complete the metadata later.

It is always a good idea to ensure the owner or creator of the data set supplies you with the appropriate metadata. Make sure they are aware of the mandatory fields, and have them provide you with this information.

You can copy (Ctrl-C) and paste (Ctrl-V) information from another source into a metadata tab field. Try to do so if there is a lot of information to incorporate and that information is available in another electronic format elsewhere from the data set owner / creator.

See Appendix 4 below for a description of metadata, the input forms, and the importance of metadata in overall information management.

Identification Information Tab – Data Set

This Identifier Information tab fields describe properties of the map layer such as the owner / originator, and contact information. The mandatory fields include Contact Name / Position, Contact Organization and Telephone.

Content Information Tab – Data Set

The Content Information tab fields describe the characteristics of the data set. One key field on this tab is the “Abstract”. Since it is a free-text field, descriptive information can be cut from other relevant

documents and pasted here to help save time in inputting. The other mandatory field is the Creation / Modification date (year).

> The *Abstract* field in the Content Information Tab is one of the most important fields in the system. Use it to describe the content of the data set and provide other relevant information about the methodology used to collect the data. Since you may not know this information, make sure the owner / creator of the data set provides it to you.

Spatial Information Tab – Data Set

The spatial information tab provides fields for describing the geographic extent of the data set. In particular, you must explicitly identify the geographic features associated with the map layer. Do this by selecting the appropriate map layer from the list of Selectable Map Layers. A list of the map layer's features will then populate the form. Select from those by clicking on the features that represent the data set's geographic extent and then clicking **Add**.

Quick Guide to Registering a Tabular Data Set

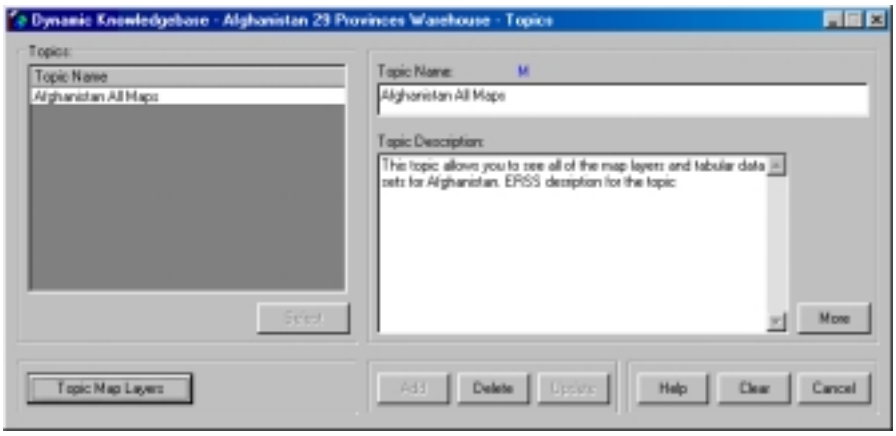
The associated map layer must be registered prior to adding a tabular data set.

1. Determine where the data set resides and if necessary move or copy it to the System Data Sets directory in the warehouse directory. For an Oracle database, make sure you know the appropriate service name, username and password.
2. Click on **Data Sets** in the Dynamic Knowledgebase Main Menu.
3. Select the Associated Map Layer Name from the list of selectable map layers. The Map Layer's ID field will be shown.
4. Click **Select Database** to get the Data Set Type form
5. Choose the appropriate Data Set type. If it's in Access, Excel or dBase format navigate via Network Neighborhood (or via My Computer if not on a network) to the appropriate database file. Once found, select the file. If it's an Oracle table or view, click the Oracle radio button and provide the appropriate Server Name, Username, and Password.
6. Close the Data Set Type form by clicking **OK**.
7. Select the Table or View Name in the Data Set that contains the appropriate records.
8. Select the appropriate Link Field Name -- i.e. the Feature ID. Click "View Attributes" if you need help determining what the appropriate field would be.
9. Identify if User Information Required -- i.e. if this is an Oracle database and the User of Dynamic Maps needs to provide a username and password to access the data.
10. Give the database a logical name in the Data Set Name field. This is the name the user will see in Dynamic Maps and Dynamic Web Maps.
11. Click **Add** to add the data set to the list.
12. Select the data set from the list and click **Metadata** to give a description of the data set.
13. Click **Update** to accept the new system data set.

Topics

Topics are logical groupings of map layers designed to present information clearly to Dynamic Web Maps users.

To manage the Topics, click **Topics** on the Dynamic Knowledgebase Main Menu to get to the "Topics" form. To select and work with a topic, double-click on its name or select it and click **Select**.



Creating a separate Dynamic Web Maps warehouse is sometimes a very effective way of enabling web access to map layers – since you may want map layers to appear at different map scales on the web than in Dynamic Maps or you may not want particular data sets made available.

All tabular data sets associated with map layers included in a Topic will be automatically made accessible through Dynamic Web Maps. If you do not want certain data sets to be accessible on the web, you should register the map layer(s) without associating tabular data set(s). This may mean, for example, registering the map layer twice (once for Dynamic Maps and once for Dynamic Web Maps) or creating separate warehouses for Dynamic Maps and Dynamic Web Maps.

Topics Form Field Definitions

Topic Name

The Topic Name is the name you assign the Topic. This name will appear in a drop-down pick list for the user on the web site beside the mapping frame. Users will select their Topic of interest before they can choose available map layers and build a map.

Description

The description will be available to users of the Topic on the web site -- so double-check spelling. The description should provide details as to what maps and tabular data sets are available for the Topic and some interpretation of them. The information could include, for example, a suggested area of the map to zoom in to see a particular feature or map layer.

It is important that the information in the description be written in clear, concise sentences. If the web site has any particular approvals process for content, make sure the description is vetted through these prior to inputting it. In any case, it is highly recommended that the description be written first in a word processor capable of spell checking, and then pasted into this field.

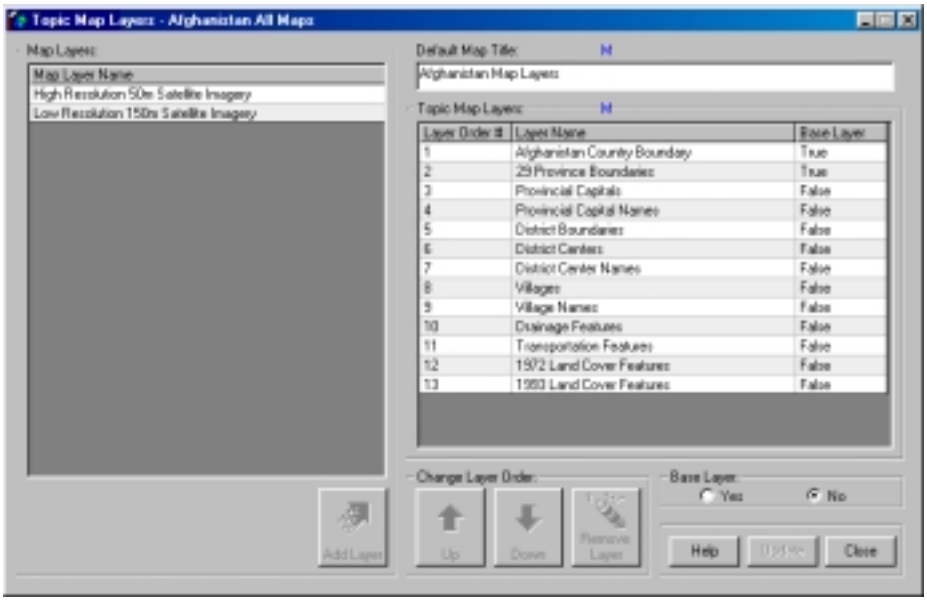
Topic Map Layers

The Topic Map Layers form enables you to assign various map layers to the Topic – so that the user has available to them only the most relevant map layers and in a particular order.

To set up the map layers for the topic you need to do four things:

SKE

1. Choose what map layers should be available for that topic.
2. Choose what order the map layers should be rendered in. All other aspects of the map layer – such as the scale at which they appear and their rendering properties are determined in the Map Layers component of the system.
3. Choose a Default Map Title for the Topic.
4. Select whether or not the layer will be a "Base" layer – i.e. if it will be always set as visible. By default any map layer set as visible when it was registered will also be defined as a base layer. You can change this if you want the web user to be able to turn layer visibility on/off at its viewing scale.



Map Layer Name

To choose what map layers should be available for that topic, select the map layer or layers from the list (left hand list) and click **Add Layer**.

Default Map Title

The Default Map Title is the title that any map is given when it is created in Dynamic Web Maps for this Topic. The system automatically puts the name of the Topic in this field – and that will frequently be the title you want. If not, you can assign the Default Map Title a new name. The user is also able to change the title on the map when they create it in Dynamic Web Maps.

Topic Map Layers

The map layers that will appear in the Topic are listed in the Topic Map Layers field. Initially, they are listed in the order in which they are chosen. Layers will be rendered in this order when a user creates a map using this Topic on the web site. The listing also reflects the order in which the map layers will be listed on the list of layers for that topic on the web site.

To select a layer, simply click on it to highlight it. You then have the option of moving it up or down, or removing it from the list.

Change Layer Order

The Change Layer Order function has three buttons: **Up**, **Down,** and **Remove Layer**. To change the layer rendering order, select a layer and use **Up** and/or **Down** to move the layer in the listing. Make sure Point-type map layers are first, followed by Line-type map layers, followed by Polygon-type map layers, and lastly followed by image / raster layers. Label layers should be ordered with the feature map layer. To remove a layer(s) from the list, select it and click **Remove Layer**.

Base Layer

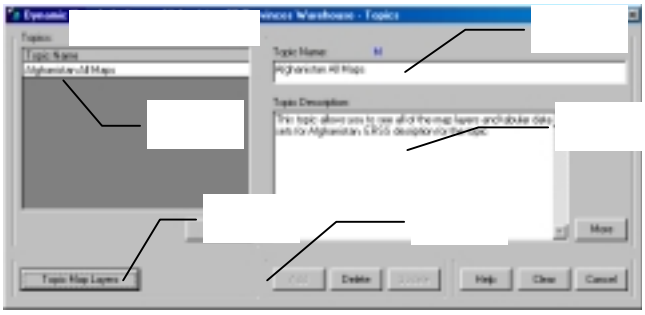
When a layer is set as a Base Layer ("Yes"), then it will always be visible to users of Dynamic Web Maps when they create a map depending of course on the extent of the view and the map scale. The

default setting is "No". If the layer is set as "**No**", the user will have the option of turning the layer on or off for viewing.

> In order for new Topics to be visible on the web to Dynamic Web Maps clients, you must re-initialize the Dynamic Web Maps Server.

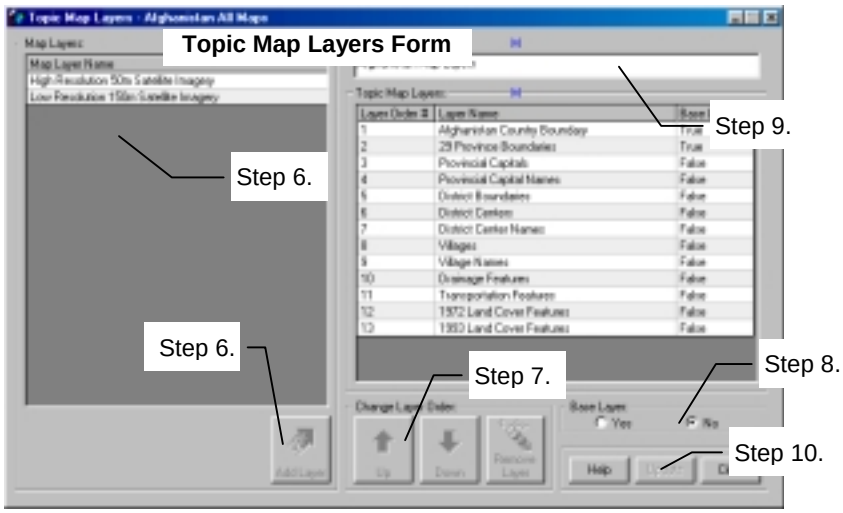
Quick Guide to Defining a Topic for Dynamic Web Maps

1. Click **Topics** in the Dynamic Knowledgebase Main Menu.
2. Type the name of the new Topic in the "Topic Name" field.



> **Tip:**
> Use a name that accurately describes the data, e.g., Environmental Hot Spots Mapping, Pest Control Maps, Refugee Repatriation from 1980 to 1990 Maps, etc.

3. Type a description for this topic.
4. Click **Add** and the Topic will be added to the list.
5. Now associate map layers to the Topic. Select the Topic from the Topic Name list. Click **Topic Map Layers**.
6. Select the layers you want from the Map Layer Name list one at a time.
7. Once the layers are chosen, use **Up** and/or **Down** to arrange them in an appropriate order.
8. For each layer, indicate if it is a Base Layer.
9. If you want, change the Default Map Title.
10. Click **Update** to accept the changes.



Appendix 1. Creating New Thematic Maps (Value Rendered Map Layers) From an Associated Database

This process enables you to use a selectable map layer as a template for new "thematic" map layers in Dynamic Maps and Dynamic Web Maps. This way, say, a municipal map layer and a database table with population figures for municipalities can be transformed into a value rendered map layer that shows the relative population of municipalities.

Very briefly, the process involves:

1. Creating a field in the database with appropriate Class codes associated with each feature. E.g. a city whose population is between 250,000-500,000 might get a code = 3 while a city above 500,000 might have a code = 4.
2. Importing the .dbf file from the associated map layer into an Access table and merging the class field with the .dbf table.
3. Exporting the new merged table back out to a .dbf file to become part of the map layer.

Check Feature ID

Make sure your tabular data can be related to a map layer. In other words, there needs to be a Feature ID field that can tie each record in the database to an identical Feature ID in the map layer's .dbf file.

Copy the Original

Creating a Value Rendered Map Layer usually means adding a new field with values to the .dbf file for a selectable map layer. Instead of this, it is better practice to create a new shape file by copying the original selectable map layer's shape file and then adding appropriate value rendering fields to it. Also, make a copy of the database if the database has already been registered as a System Data Set.

Deal with Duplicate Records

Sometimes map features might be reflected by more than one value in an associated data set. For example, a meteorological station might have several rainfall readings in a year. The Class Rendering functionality of Dynamic Maps enables you to deal with this easily by assigning the class to be determined based on the minimum, maximum or average of those values. Creating a Value Rendered map layer is different, though, because by definition, a feature can only be represented once in the shape file – so you can't have several unique values for the same feature.

As a result, if you are creating a new map layer for value rendering and some or all of the features have more than one record value associated with them, you need to clean this up first. In most cases, you should be able to quickly go through the data and eliminate the records you don't want to use by first sorting the data in the table.

> If you don't deal with duplicate records and do a Join with the related .dbf file (as outlined in Part 2) Access will do the join and carry over all values it encounters for each feature. As a result you will end up exporting a .dbf file (in Part 3) that has more feature attributes than there are map features. It is very important that the order of the records in the .dbf file not change. The order determines what map layer feature it links to.

Step 1. Creating the Class Code Field and Adding Values.

Open the database in Access and identify the tabular data set field you want to use for class rendering (e.g. Municipal Population). In Design Mode, add a new numeric (integer) field to the database table to store the class rendering codes.

Decide how many codes will be needed to best express / describe the data. If there is a wide range of values, and several spatial features, then you may want to have several codes. Also consider that there

may be logical and meaningful breakpoints for the data (e.g. 10,000, 100,000 and 1,000,000). **Make sure you WRITE DOWN where your breaks are and therefore what the codes represent. You will use this information later when you define the map layer's legend information.**

For example, to create classes for Municipal Population, look at the values in the Municipal Population field and decide a logical way of grouping them – so that, say, the code for municipalities with a population between 0-100,000 is 1; for 100,001–500,000 is 2; for 500,001 – 1,000,000 is 3; and for greater than 1,000,000 is 4.

Depending on the number of features in the map layer, defining the class codes in a database might be easy or somewhat more difficult. Three different methods are outlined below:

a) For Map Layers with only a Few Features

If the related database only has a few records (say less that twenty), you can probably just as quickly update each code value manually right in the database table itself.

Look at each record in the database and determine the class rendering codes for each based on the break points you have established for the field that is being class rendered. Put each value into the newly created code field.

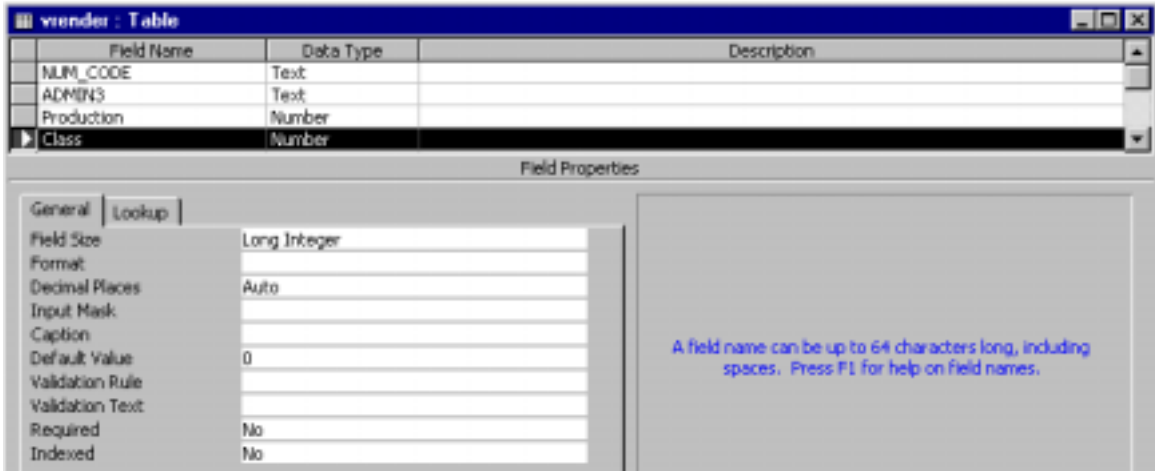
Go to Step 2. Merging the Class Field with the .dbf Field.

OR

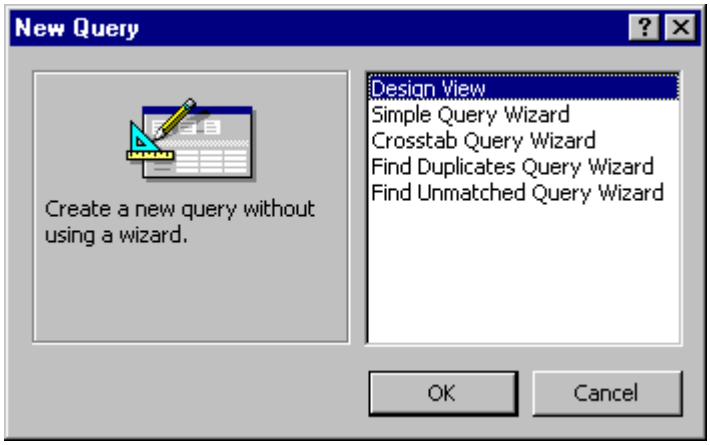
b) Using An Update Query For Map Layers with Many Features

The next "easiest" and probably preferred way to create class codes is to use a series of Update Queries. This method for created Class Rendering values in a table requires several steps, but is usually faster and easier than creating complex query statements (as outlined in option c). This uses the power of an M/S Access "Update" query to create new values in a field based on the value range from another field.

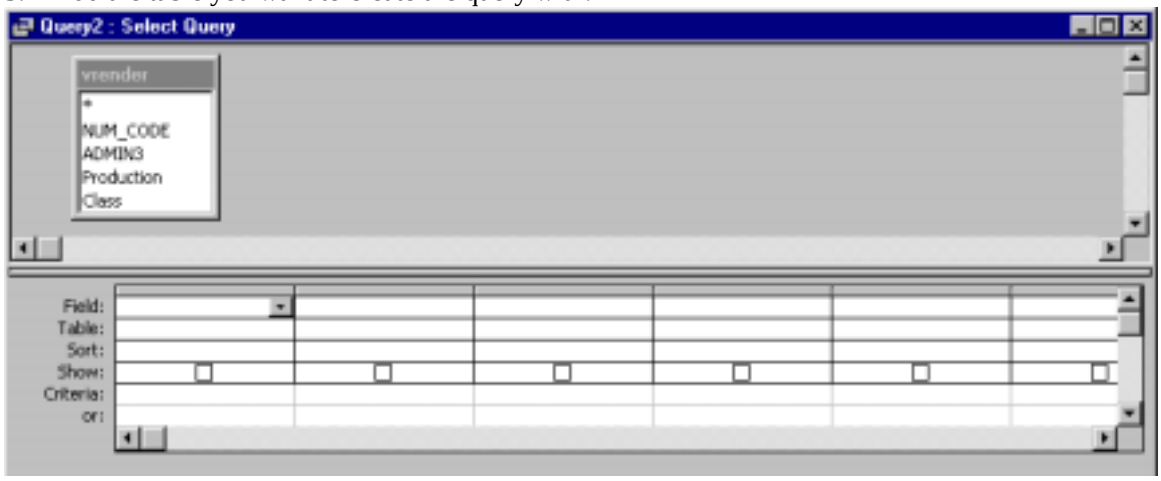
1. Create a new field in the table containing the values you want to base a value rendering on. In this example, the new field is called "Class" and it is numeric, and the field from which the values for Class will be based is called "Production".



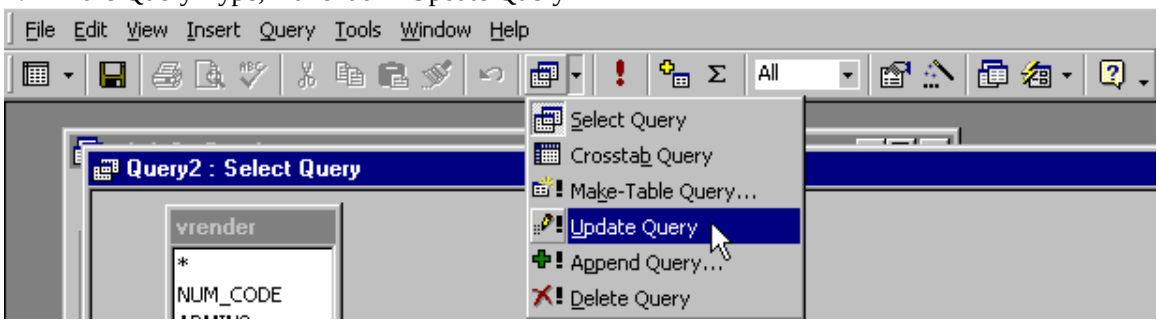
2. Create a new query, enter Design Mode.



3. Add the table you want to create the query with.



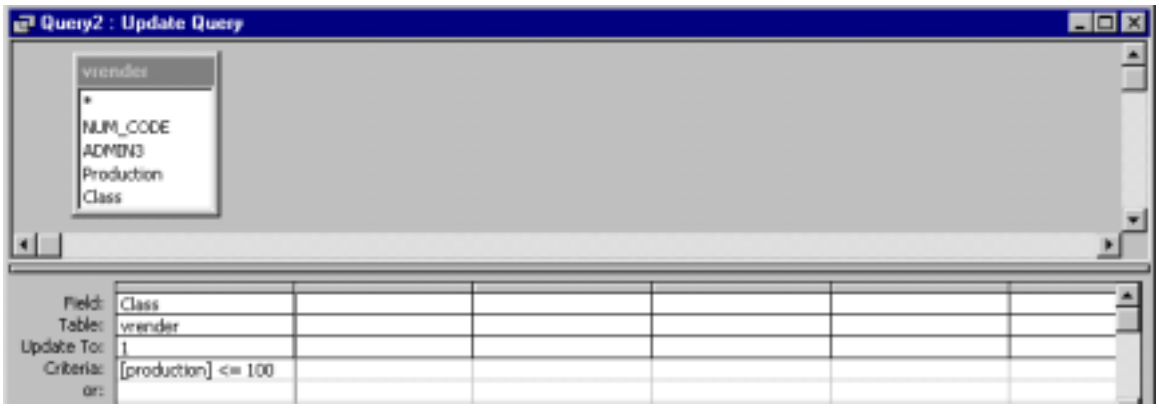
4. In the Query Type, make it an “Update Query”



5. Add the “Class” field to the query.

6. Under “Update To”, type 1.

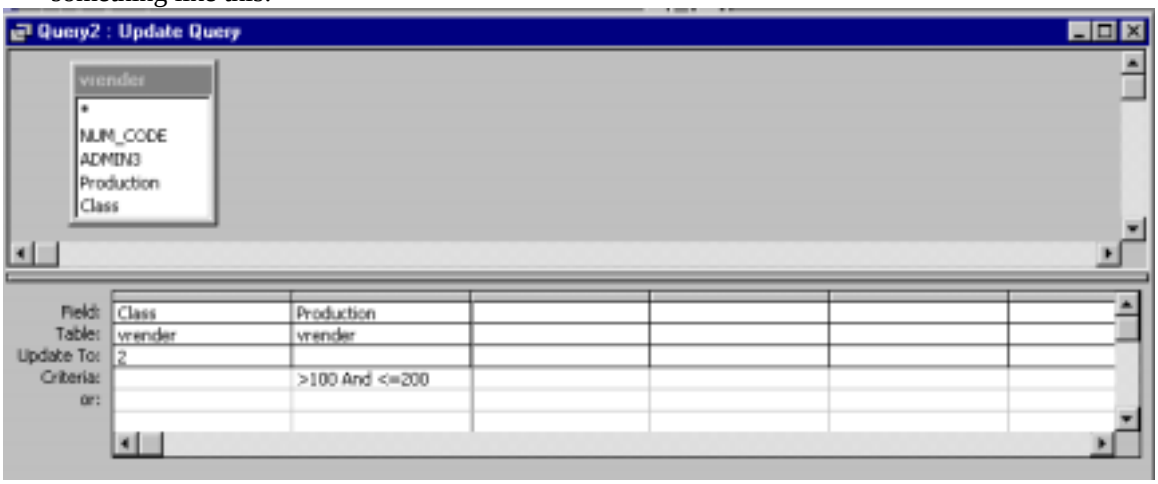
7. In the Criteria line, type the criteria that will create a value of 1. Sample syntax is: “[rendered field] >value1 And <value2” or “[rendered field] <value1”. (don’t use the quotes) Where [rendered field] is the field name of the field from which the value is being determined (in this case “Production”), and the “value1” and “value2” are the ranges that reflect a value of 1.



8. Now run the query and all the values for Class =1 will be created.
9. **Exit and save the query**.
   Check the table to ensure the value was appropriately assigned.

10. Repeat steps 6 - 8 with a new value for the Class field (2,3,4, etc.) and the appropriate value ranges for the "production" field until all the ranges have been set up.

    You will notice that Access separates the query into two columns automatically when you open the query for editing again as shown in the example below. Creating Class = 2 would look something like this:



For each new class in this example, you would change the "Update to" field in the first column; and, in the second column you'd change the "Criteria" field.

**** Remember to Run the query for each new value "Class" value.**

After you run the query, the second time, you don't necessarily need to exit the query and check the table. You can just keep changing the values for the Class field and its criteria and running the query.

Go to Step 2. Merging the Class Field with the .dbf Field.

OR

c) Using SQL Query Statements to Define Class Codes.

When you have numerous features the most direct although most difficult way is to use SQL expressions. The following two queries when set up and run together will generate rendering values in a

new field in the database table based on the column of values that you want to value render. *Pay careful attention to the named fields and what they represent.*

In the following example, "Class_Values" is the name of the table. "Code" is the name of the first SQL expression.

The "Class_Values" table that the queries are being run against will need to have at least three fields:

- "ID" is the primary field variable and it is the table's unique ID field.
- "Random_Values" is the variable / name of the field containing the values that you need to create a code for.
- "Code_Number" is the variable / name of the new code field that you create to store the generated codes.

Step 1 is to ensure the table has these three fields. You will probably need to create a "Code_Number" field in the table. Do so by selecting the Table and then going into design mode. The field should be numeric and can be Long Integer. Name the field something appropriate based on the code and the data it reflects.

Step 2 is to determine appropriate break points for the value rendering and WRITE THESE DOWN.

Step 3 is to design the Query. To do this, select the Query tab and go into Design mode and use the SQL view (rather than the Table view).

In the blank SQL form, write the following expression (using the actual field and table names in place of the variables in this script):

```
SELECT ID, IIf(Random_Values<100,1,
            IIf(Random_Values<200,2,
              IIf(Random_Values<300,3,
                IIf(Random_Values<400,4,
                  IIf(Random_Values<500,5,
                    IIf(Random_Values<600,6,
                     IIf(Random_Values<700,7,
                      IIf(Random_Values<800,8,
                        IIf(Random_Values<900,9,
                           IIf(Random_Values<1000,10,

IIf(Random_Values>999,11,0))))))))))) AS Code
FROM Class_Values;
```

You will notice that the IIf function has the format: IIf(test condition, true do this, false do this). The very last IIF statement has a false condition equal zero. This is very important, since any record that does not have a value in its Random_Values field will produce a nul result.

Also notice that the last IIF statement tells the query to define itself as "Code" – it gives itself an Alias, so that it can be referred to in the next query without having to retype all the text.

Step 4 is to save this query with a meaningful name (called "Code_Count_All" in the next query). Take a look at the query to see if it produced the results you expected. You should be looking at the ID field and the Code_Number field.

Step 5 is to create a second query to join this first view with the original table based on the common ID. Open up a new query in Design mode and SQL so that you can enter the coding:

```
UPDATE Class_Values INNER JOIN Code_Count_All
ON Class_Values.ID = Code_Count_All.ID SET Code_Number = Code;
```

Save this query with a unique name.

Step 6 is to run the query. Your original table should not have a new field with codes in it. You may wish at this time to go into design mode and quickly type into the description of this field what the codes mean (i.e. “1 = values less than 100; 2 = values between 100 and 200; 3 =”, and so on).

Go to Step 2. Merging the Class Field with the .dbf Field.

Step 2. Merging the Class Field with the .dbf Field.

You now have a database table that has a field in which codes have been populated using one of the three methods above. You now need to transfer these codes to the .dbf file of the selectable map layer and thus create a new map layer that you can register in Dynamic Knowledgebase as a Value-Rendered map layer.

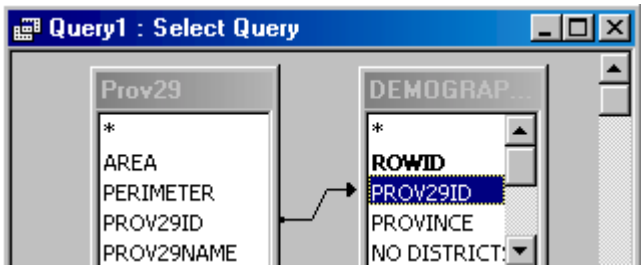
You should make a copy of the .dbf file so that it does not get confused with the current files being used by Dynamic Maps, Dynamic Knowledgebase, and Dynamic Web Maps Server. Import the map layer’s .dbf file into the database as a new table. To import:

- Go to File / Get External Data / Import on the file menu and open the Import form.
- On the Files of Type field, select dBase IV (which uses the .dbf extension as do shape files).
- Navigate to the appropriate selectable map layer and click **Import**. You should get a message that the table was successfully imported and the map layer’s .dbf file should now appear as a table in the database.

> If you get an error when importing the .dbf file, it’s probably because Access can’t import a .dbf file with a name that is more than eight characters in length. If yours is, simply make a copy of the map layer and rename the file. Remember to change the name again later to match the .shp and .shx files!

The imported .dbf file is now a table in your database. You will now create a new table that incorporates fields from the shape file’s .dbf file that you just imported and the Class Codes field that you created earlier. To do so:

- Click the Query tab and click **New**.
- Choose Design View.
- In the Show Table dialogue box, select the table that you just imported and click **Add**, and then select the table containing the new class codes and click **Add**.



- Join the two Feature ID fields in the two tables by dragging the ID field in the .dbf table over top of the ID field in the other table (as shown in the diagram). Select the join and right-click to access its properties. Ensure all records from the map layer table (.dbf) are included.
- Add all the fields from the .dbf file table and the new class rendering code field from the other table into the query list. You may also want to add the field that the class-rendering code is based on if you haven’t deleted that field.
- On the button bar, choose Query-Type and from the drop-down list select **Make Table**. This way when you run the query a new table will be created.
- Run the query to create the new table.
- When you exit the query, you will be asked if you want to save the changes to the design of “Query1”. There is no need to save the Query.

You should now have a table in the database that contains the fields of the .dbf file (map layer), plus the value rendering codes and perhaps the values themselves. You can now create the new shape file and register it with Dynamic Knowledgebase.

Step 3. Exporting the Merged Table the Map Layer's .dbf File.

Don't change the order of your records in a .dbf file! If you are adding fields or changing a .dbf file to replace an existing .dbf file, you must ensure the records in the new .dbf file are in the ***same sequential order*** *as the records in the original .dbf file. If not, the shape file's index will be incorrect and you will lose the relationship between the shapes and their attributes.*

To export your new table to a new .dbf file that the shape file will use, select the table in the database and then use File / Save As-Export. When prompted, select "To an External File or Database" and click **OK**. Now choose where you will be saving the file and choose from "Save as type" = dBase IV.

Copy the .shp and .shx files into this same directory as the new .dbf file that you just exported and, if necessary, rename the .shp and .shx files so that all files have the identical name before their extension. Now move all the files into the appropriate warehouse Map Layers directory.

Use Dynamic Knowledgebase to register this new map layer as a value-rendered map layer (see instructions in the manual above for registering a value-rendered map layer if you need help). Remember that when defining the legend, to indicate the values at which you created the breaks.

The map layer is now ready to be displayed in Dynamic Maps and Dynamic Web Maps as a thematic map (value rendered map layer).

You can add multiple value code fields to the <shape>.dbf for each associated tabular data set field that you want to class render. However, for each value code field that you want to display, you must re-register the map layer and identify the respective field for value rendering. This does not copy the map layer data only the reference information.

Library map layers can also be used for value rendering. In this case each tile that contains a portion of the map layer must have the associated value code field attached to its <shape>.dbf file. To do this efficiently it may be better to add the value field prior to tiling the map layer using a professional GIS tool.

Appendix 2. - Using ArcView with the System

Adding Attribute Data to a Shape File to make it Selectable

ArcView can be used to add values to the various shapes in a map layer so that the layer can then be made selectable and you can identify its features using Dynamic Maps.

Two key fields need to be added or edited in the Attribute table in ArcView.

1. There must be a new ID field created or edited. To do this, open the attribute table using the Open Theme Table button  on the button bar, and select Table / Start Editing. Select Edit / Add Field and create a new numeric or string type field with a field name that you can easily remember. Select the "Edit" pointer from the toolbar to change your cursor so that you can now add the new ID values to the shape file.

2. There must also be an attribute type associated with each of the shapes so that when you identify the feature, its "name" or value appears. If necessary, add another field to the attribute table for this feature name and make it a "string" type. Then add the feature name values. Each record should have a value and if you don't know the value, simply state "Unknown". Note: you may simply want to use the same field for both the Feature ID and for the Feature Name.

Now use "Table / Save Edits As" to save the new shape file somewhere where you can easily locate it. Copy the shape file into the Map Layers directory so that you can now register this map layer as a selectable map layer using the new Feature ID and Feature Name.

Improving Rendering, Find, and Identify Response

Some spatial data sets can be quite large and therefore may cause Dynamic Maps and Dynamic Web Maps to take a long time to load, identify features and even find its features. In cases, where these map layers do not need to be selectable, they should probably be tiled and set up in a spatial library (by a GIS expert using ArcInfo). However, if you are making a large map layer selectable, then you should use ArcView to create indexes for the Shape field, (called a Spatial Index) and the Feature ID field and the feature name field (called Attribute Indexes).

> When you create a spatial index a new file is added to the shape file with .sbn and .sbx extensions. When you create an attribute index, new files with an .ain and .aih extensions are added to the shape file. Make sure if you are copying a shape file that these additional files are included.

To create an index, set the map layer active, open the attribute table (.dbf), then make the field the "active field" by clicking on the field name so that it becomes shaded. Choose Create Index from the Field menu.

> **Tip:**
> Create indexes for all your map layers – regardless of their size and whether or not they are selectable. This will improve performance considerably!
> If a map layer is not selectable, just create a spatial index (on the Shape field).
> If you subsequently edit a map layer, make sure you re-create the indexes.

Appendix 3. - Linking to Other Data Sources with M/S Access

You can also link tables from Microsoft Access databases so that you can use them without opening the other database. This way Access can be set up to link to other data sources and thus provide Views to external data sources when you are registering the database.

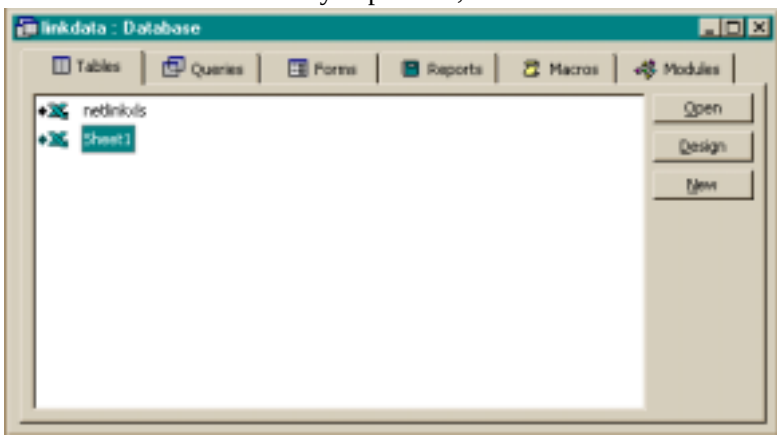
The following instructions reflect Access 97. Users of Access 2000 may find slight differences.

Decide first whether you should Link to the data or Import the data. Linking will rely on having a stable connection with the other data source, however it has the advantage of being maintainable elsewhere. It could also be a "live" data source that frequently changes. The instructions for ***importing*** data into an Access database are virtually the same as linking. The main difference, of course, is that the data themselves are actually moved.

If you need to export your warehouse, linked data sources will need to be recreated and re-registered when you import the warehouse – unless the linked source is also within the warehouse.

Linking to data from a spreadsheet

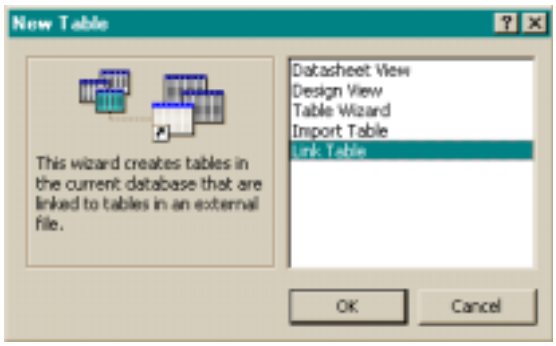
You can link data from a Microsoft Excel (versions 3.0, 4.0, 5.0, 7.0/95, 8.0/97 and 2000) spreadsheet. You can also link (read-only) data from a Lotus 1-2-3 spreadsheet. In both cases, the data in the spreadsheet must be arranged in an appropriate tabular format. Before you proceed, make sure that the spreadsheet has the same type of data in each field (column) and the same fields in every row.



You can link all the data from a spreadsheet, or just the data from a named range of cells. Although you normally create a new table in Microsoft Access for the data, you can also append the data to an existing table as long as your spreadsheet column headings match the table's field names, or its column order is the same.

1 Open a database, or switch to the Database window for the open database.

2 To link a spreadsheet click **New** and then on the New Table form, select Link Table and click **OK**.



3 In the Link dialog box, in the Files Of Type box, select Microsoft Excel or Lotus 1-2-3.

4 Click the arrow to the right of the Look In box, select the drive and directory where the spreadsheet file is located, and then double-click its icon.

5 Follow the directions in the Import Spreadsheet Wizard dialog boxes. If you are importing from a Microsoft Excel version 5.0, 7.0/95, or 8.0/97 workbook, you can import from one worksheet within a workbook. You can't import from any other multiple-spreadsheet files, such as Microsoft Excel version 4.0 workbooks or Lotus notebooks. To import from these files, you must first save each spreadsheet as an individual file.

SKE

- The Lotus 1-2-3 driver is not included in Setup, but it is available through the Office 97 ValuPack.
- If importing a spreadsheet takes an unexpectedly long time, it might be because many errors are occurring. To cancel importing, press CTRL+BREAK.
- Access attempts to assign the appropriate data type to imported fields, but you should check your fields to make sure they are set to the data type you want. For example, a phone number or postal code field might be imported as a Number field, but should be changed to a Text field in Microsoft Access because it is unlikely that you will perform any calculations on these types of fields. You should also check and set field properties, such as formatting, as necessary.
- You can import data from other programs' spreadsheets, as long as they are in the Microsoft Excel or Lotus 1-2-3 format.

Linking to SQL database tables or data from other ODBC data sources

Linking to ODBC data sources requires that the appropriate ODBC drivers be set up on your computer. This sometimes occurs when Access is installed, depending on the settings you selected. If you chose the Typical option when you installed Microsoft Access, the SQL Server driver and ODBC support files will not be installed, and you will not be able to import or link from any ODBC data source. In this case, rerun Setup to install the SQL Server driver. Additionally, for each ODBC data source that you want to use, you must have the appropriate ODBC driver installed on your computer before you can link this database. For more information on installing and configuring ODBC drivers, see the Access Help documentation.

1 Open a database, or switch to the Database window for the open database.

2 To link tables, on the File menu, point to Get External Data, and then click Link Tables.

3 In the Link dialog box, in the Files Of Type box, select ODBC Databases.

4 In the Select Data Source dialog box, do one of the following:
- Click the Machine Data Source tab to see a list of all ODBC machine data sources currently defined for the ODBC drivers installed on your computer. This list might include drivers that have not been tested and verified for use with Microsoft Access. Contact the driver vendor for verification.
- If you want to use a file data source, click the File Data Source tab and enter or browse for a file name. A file data source is useful for sharing ODBC driver connection information between different computers and does not require a Registry entry. For more information on file data sources, see the ODBC 3.0 Programmer's Reference Manual.
- If you want to define a new data source for any installed driver, click **New**, double-click the name of the ODBC driver, enter the data source definition (this will vary depending on the requirements of the ODBC driver), and then click **OK**.

5 In the Select Data Source dialog box, double-click the ODBC machine data source that contains the data you want to link.

6 If the ODBC data source that you selected requires you to log on, enter your logon ID and password (additional information might also be required), and then click **OK**. Microsoft Access connects to the ODBC data source and displays the list of tables that you can link.

7 If you're linking a table, select the Save The Login ID And Password check box if you want to save this information for the table in the current database and not require users to enter this information. If you leave the check box cleared, all users must enter the logon ID and password every time they open the table in each new session with Microsoft Access. The database administrator can also choose to disable this check box, requiring all users to enter the logon ID and password each time they connect to the SQL database. See the Access documentation for more information.

8 Click each table you want to import or link, and then click **OK**. If you're linking a table and it doesn't have an index that uniquely identifies each record, Microsoft Access displays a list of the fields in the linked table. Click a field or a combination of fields that will uniquely identify each record, and then click **OK**.

- After linking an SQL database table, you can set field properties for the table.

- If you encounter an error while linking or using an SQL database table, there might be a problem with your account on the SQL database server or with the database itself. If you can't access an SQL database table, contact the SQL database administrator.
- To edit a linked SQL database table, usually the table must contain a unique index on the server. If you want to edit a table that doesn't have a unique index or you want to edit an SQL view, you can create an index within Microsoft Access that the SQL database isn't aware of. You do that by creating a data-definition query, using the Create Index statement. Be aware, however, that the index must be created on a field or combination of fields in which each value is unique. If the field contains any duplicate values, all updates to the table will fail. To delete the index, use another data-definition query.
- If the structure of an SQL database table changes after you link it, use the Linked Table Manager to refresh the link.

Appendix 4. - Metadata

Introduction

Metadata are summary descriptions of data sets and information holdings. They are important because they help promote the use of data standards, leverage partnerships, and make the vision of easy, "one-window" access for spatial data more achievable. However, metadata management can be difficult. Spatial data often come in large complex database structures with numerous object types. They come in a variety of different data formats. They come from different sources, each with its own way of describing things. They vary in spatial and temporal resolution. They range widely in quality. All these and more contribute to making the documenting of spatial data in a consistent manner difficult. The Dynamic Knowledgebase tool enables your organization to catalog spatial data and tabular data sets.

Metadata are an important ingredient in the overall success of the system. Metadata will help management and staff who use the data to find the data they need and determine how best to use it. Metadata will benefit the organizations that produce the data as well. Undocumented data tend to lose their value over time, especially as people in an organization change. Staff later may not understand the contents and uses for the databases, and may not be able to trust results generated from these data. Not knowing about other organizations' data often means duplication of effort. So, although it means some work up front to generate metadata, it's worth it in the long run.

These complexities, though, also highlight the importance of metadata. Metadata can cut through the complexity to help describe what data sets and documents exist, their characteristics, and their availability. Several jurisdictions around the world are working towards a successful metadata implementation. To do so means having: an accepted metadata standard; appropriate metadata input and search technology tools; and a commitment for client and agency metadata collection and maintenance.

The Metadata Standard

The metadata standard describes the format and characteristics of data elements to be used when collecting and describing information holdings – such as spatial data, tabular data, and documents. It is not specific to a single organization or agency.

Having an accepted standard means:

- metadata are collected in a clear and systematic way;
- metadata for a particular data set remain consistent over time -- as the data set is updated or added to;
- similar data sets can be compared to determine their similarities and differences;
- the development of integrated access and input systems is encouraged; and,
- others can understand and easily determine the potential usefulness of the data.

What was considered in determining the standard:

Four principles were important when developing the standard.

- ***Ease of input and use***. If the system and the standard are too bulky, too complex, have too many mandatory components, or are too difficult to understand, they won't be used.
- ***Meet client need***. What would he/she need to know about a data set if searching for information – such as What data are out there?; What are they like?; and, How do I get them?
- ***Collect adequate detail***. Unfortunately, identifying just the name of the information holding and a contact phone number is not enough. Either the contact would be deluged with calls, or the system would not be used by anyone who doesn't already have a good idea of what the data are.

In other words, developing the standard meant compromise and client consideration.

Metadata Submission and Input Process

The typical process for metadata collection and input involves the following:

- Owners / custodians of the data set will use a form to do the initial metadata input for the information holdings with which they are most familiar. A copy of the input forms for map layers metadata and tabular data set metadata is available for distribution on the software installation CD.
- Their completed forms will be received by the system administrator – either via email, diskette or in paper form.
- The system administrator checks the forms for completeness. In particular, all mandatory fields will need to be completed.
- The system administrator inputs the metadata record with the Dynamic Knowledgebase tool. The map layers and tabular data sets must already be registered as part of the warehouse before their metadata can be input. In this way, the metadata is ensured to reflect actual data available.
- Once the metadata records are in the warehouse, they are then available to users of the Dynamic Maps and the Dynamic Web Maps.

Summary Components:

The following outlines the standard components:

Metadata Standard Summary Components

Standard Element	Map Layers	Tabular Data Sets
Data Set Name	**(M) Mandatory**	**M**
Contact Source (Contact Name / Position; Contact Organization; Telephone Number)	**M**	**M**
Contact Source (Postal Address; Email Address; Fax)	**(X) not mandatory**	**X**
Originating Organization (Author, Originating Unit, Parent Organisation)	**X**	**X**
Publication / Creation Date	**M**	**M**
Date Comments	**X**	**X**
Status Indicator	**X**	**X**
Link URL To Other Related Information	**X**	**X**
Other Information Description	**X**	**X**
Metadata Date	**M**	**M**
Abstract	**M**	**M**
Constraints	**X**	**X**
Data Set Size	**X**	**X**
Attribute Reliability		**X**
Geographic Extent	**X**	**M**
Positional Accuracy	**X**	
Mapping Coordinate Units	**X**	
Reference Ellipsoid	**X**	
Map Projection	**X**	

Appendix 5. - Backups

The need for frequent backups will depend somewhat on the number of new map layers and data sets added and registered with the system. Initially, when many are being added and there are frequent changes to the warehouse(s), daily backups are recommended. As the system matures, backups may be needed infrequently.

Since the Dynamic Knowledgebase software components can be reinstalled in case of a "fatal crash", the only components of the system that need to be backed-up are:

- The Warehouses
- The Warehouse Manager directory (i.e. the Warehouse Manager.mdb file)

It is particularly important that the Warehouse KB directory be backed up on a regular basis – since this is where all the characteristics of the warehouse are stored, including metadata.

The warehouse data – the Map Layers, the System Data Sets and Libraries – may be quite large and some or all of it may exist in prepared form on CD or backed-up already. With respect to the data, backups should ensure that any new data or changed data are captured, but not all data need to be backed-up each time.

Appendix 6. - Warehouse Data Preparation

Quick Tips

The data must be ready to be registered before Dynamic Knowledgebase can be used. Depending on the data, the effort required for data preparation could range from being very involved, time consuming and potentially costly -- for some large tiled spatial data library layers, or quite easily accomplished -- for some tabular data sets.

Below is a quick checklist of what you need to consider about your data and their location before using the Dynamic Knowledgebase.

Warehouse

- The map layers and system data sets need to be copied into the proper warehouse directories.

Library Layers

- Any tiled map layers must be properly set up in a directory structure that relates to the tiling scheme and that tiling scheme must have an appropriate index shape file.

Map Layers

- All map layers if they are in shape format must have an appropriate .shp file and related .shx and .dbf files (unless they are image / raster layers).
- If the layer is being made selectable / feature identifiable, it will need to have a ***feature geographic ID*** (Feature ID) field in its shape .dbf file.
- If the layer is being made selectable / feature identifiable it will also need a ***feature name field*** in the shape .dbf file that accurately names the related feature.
- If the layer is a Label Layer, a ***feature name field*** is required in its shape .dbf file.
- If the layer is value-rendered, a ***feature value field*** is required in its shape .dbf file.

> It is a good idea to create indexes on these fields using professional GIS tools. For example, in ArcView you can create indexes on the feature type field, the feature geographic ID field, and the feature name field.

Tabular Data

All tabular data sets must contain the ***linking field*** that matches the relevant map layers' Feature ID field. The coding must follow the standard feature coding scheme established for each Warehouse. All Feature ID's must have their Field Size = "Long Integer" when they are numeric. Feature IDs can also be alphanumeric.

Metadata

Try to have metadata available for the map layers and the data sets. Metadata are not mandatory but highly recommended. The metadata help the user of Dynamic Maps and Dynamic Web Maps understand what data they are using -- including the source and the reliability. Contact the "owner" / originator of the data for this information.

1. Warehouse Definition

Before proceeding, define:

a. **The data storage scheme for the project**. Each warehouse is a set of logically related information objects, related tabular data and spatial data that are in the same geo-referencing

coordinate system. Depending on your data, you may wish to or need to define more than one warehouse. For example, if you have data at widely varying scale you could create a global warehouse (1:5M scale), a national warehouse (1:1M and 1:250K scale), one or more provincial warehouses (1:100K or larger scale), or even site specific warehouse(s). Another approach would be to create subject-specific data warehouses with data at any scale. For example, you could have a natural resource warehouse, a political warehouse, an economic indicators warehouse, etc. The main criteria for choosing the data storage scheme is to consider what the user needs – and then put those "like" groups of data together in a logical way. There is no limit to the number of map layers and amount of data each warehouse can contain; the limit, however, is more a practical limit of how many map layers and related data sets you want to present to the user. Too much data can cause the usability of the warehouse to suffer.

b. **The library tiling scheme**. See the section in the help / manual that describes tiles and tiling for more information. Tiling structures for very large spatial data layers often follow logical boundaries, such as Provincial boundaries, or can be square/rectangular tiles.

c. **The warehouse(s) names**. If the same warehouse needs to be created in two or more languages, then create unique warehouses, one for each language.

d. **The warehouse(s) location and directory structure**. Although data can reside anywhere on the network (as long as it is visible to the system), it is still recommended that all the data be in the default warehouse directory format and that they reside together on the same computer. This ensures that the warehouse can be easily moved / exported if needed, and it also helps the DBA manage the files. The warehouse and all its data need to be in a location visible on the LAN to all users of Dynamic Maps, and, if the same warehouse is being published on the Internet, it needs to be located on the server running Dynamic Web Maps. Remember to always use network path names when defining the warehouse(s) location of the warehouse is being used across a network – unless the location is a "mapped" drive. In some cases, an organization may not want to or be able to have the same warehouse on the Web server that Dynamic Maps users use internally. In this case, Dynamic Knowledgebase should be used to manage the warehouse on the LAN for local use, and then a copy of the warehouse could be imported on the Web server for Internet use.

e. **The coordinate system for each warehouse**. Global data are often in geographic coordinates while larger scale map data will often use a map projection such as Lambert Conic Conformal or UTM. Each warehouse can only support data with the same coordinate system. If some data need to be included in a warehouse, but are not in the same coordinate system, then these would need to be transformed using appropriate GIS tools.

f. **One person should be designated as the Database Administrator (DBA).** The DBA who will be responsible for the warehouse(s) creation, management and maintenance.

2. Spatial Data Cleaning and Validation

Check to see that:

- Each map layer has a unique name.
- Each vector map layer is in shape file or Arc/Info coverage format (Library layers must be in shape format).
- Each map layer is topologically clean – no weird polygons, slivers, over shoots, undershoots.
- There are no Multipoint type map layers – they should all be converted to single point features.
- All map layers have valid business features – there are no unwanted graphic primitives from the digitizing or topological processing. This can be done by looking at the map layer dbf file and sorting on the area/length field to identify the smallest values. They can then be visually inspected to see if they are valid. Any features that are not valid business features should be deleted from the map layer or the map layer should be re-processed with larger geo-processing tolerances. All geo-processing should be done using appropriate GIS tools, i.e., Dynamic Knowledgebase does not provide these functions.
- For those map layers that will be rendered using a value or as a label, there must be appropriate value rendering field(s) or label rendering fields(s) appended to the map layer dbf table.

Identify the selectable map layers (those which can be used to relate tabular data and can have features identified).

For Selectable Map Layers:

- Each map layer dbf contains a business feature ID, business feature Name field. ID fields should have intuitive names, e.g., MUN_ID for a Municipal identifier. A good convention is <Map layer name>_ID and <Map layer name>_NAM. If necessary, the name and the ID fields can be the same field. All tabular data sets (system or user) must be registered using the same business feature ID type as used for the map layer ID.
- Each business feature ID value should be unique across the map layer. It is preferable to define the ID field first in numeric format, if that is not possible then as an alphanumeric code, and lastly as a text field. Note, in a lot of cases the business feature ID will be determined by the business program/area, e.g., each mine has a unique identifier defined by the Ministry of Mines and Metals – in this case this should be designated as the feature business identifier. In all cases the business feature identifier should be the ID field, unless it is not unique. In this case define a unique ID field and use the business identifier as the Name field or as an additional attribute in the dbf.
- Each map layer should have a spatial index created on the feature type field, and for selectable map layers on the business feature ID field, and the business feature Name field.

For Tiled Map Layers:

- Map layers have the same name for all tiles.
- All map layer features are clipped to the tile boundary.
- For those map layers that will be rendered using a value or as a label, there must be appropriate value rendering field(s) or label rendering fields(s).

Also for Tiled Selectable Map Layers:

- When a business feature (such as a road or wetland) spans more than one tile, its business feature ID values for each of its graphic primitives must be the same across all the tiles.
- As in a non-tiled map layer, all unique features on a selectable map layer must have a unique ID.
- Map layer business feature ID and business feature Name fields are named the same way and have the same field physical type across all tiles.

Copying/Moving Map Layer Files into the Warehouse Directory Structure

For Libraries:

- Define each library as a named directory under the Warehouse/Library directory.
- Each Library directory must have an Index shape file. The Index dbf file must contain a field that uniquely names each tile in the Library.
- Each tile in the Library should have its own subdirectory with the same name as defined in the Index file.
- For each tile, copy all the map layers in the tile subdirectory. Note, not all map layers have to exist in each tile, i.e., there can be islands/holes in the coverage.

For Seamless Map Layers:
Copy seamless map layers into the Warehouse/Map Layers directory. If required, create subdirectories to manage map layers that apply only to a specific geographic area or topic.

3. Tabular Data Cleaning and Validation

Sometimes the map layer's .dbf file contains information that would be better stored in a relational database. This section deals with how to clean up the .dbf file for the map layer – so that it is easier to use; and, how other fields in the .dbf file can be turned into a System Data Set.

Define one or more tabular databases in M/S Access'97 or '2000 format that will act as repositories for all tabular data. Tabular data can be organized by subject, time period, language, geography, or some other logical group. A database can contain one or more physical tables and one or more SQL Views defined on one or more physical tables. This same procedure can be done using other database management systems.

Each tabular database should be located under the Warehouse/System Data Sets directory. If necessary, use subdirectories to organize multiple databases (note: this is not applicable to an Oracle dB).

For each Selectable Map Layer's shape dbf file, import the contents into M/S Access. Note, the dbf file cannot already be open by any other application during the import otherwise M/S Access will report an error (this occurs because of file locking). Also, you may get an error importing a .dbf file if its name is more than eight characters (using Access 97). If the name is too long, make a copy and give it a new name (remember to rename it when you export it back as the new .dbf file for the map layer).

Once imported:

- Make a copy of the table – name it "<map layer name>Copy", i.e., you will now have two versions of the same table.
- To create a system data set from the contents of the .dbf file, open the table in Design View:
  - Add an ID field – first field in the table, AutoNumber type and defined as the primary key.
  - Delete all the information typically associated with a shape file, such as Area, Perimeter, etc. that do not have any relevance to the business features.
  - If necessary, define Look-Up tables for standard values. Note: if you define Look-up tables, these have the advantage of helping ensure data are consistent e.g. there are no

spelling mistakes; and, it makes it easier to change the contents of a data set that have several records with the same look-up value. If you use Look Up tables, you will need to also create a View in order to register the data so it shows the look-up value as opposed to the look-up code.

- To create a "clean" .dbf file for the selectable map layer, open the second table (<map layer name>Copy) in Design View:
  - Delete all the unnecessary attributes in the table except for the business feature ID field and the business feature Name field. If necessary, rename the business feature ID field using the following naming convention "<map layer name>_ID" and ensure that the field's physical type is the same as the business identifier in the System Data Set table.
  - Note - It is critical that the order of records does not change. If the record order changes then the link to the spatial features will be invalid.
  - Export the table from M/S Access format to dBase IV format – use the same name as the map layer (replace the original dbf file). Before overwriting the original dbf file make a temporary copy just in case there are problems. After validating the data and geographic links, delete the old dbf file.
  - Delete this table from the M/S Access dB.
  - Note - The process of deleting business related fields from the shape dbf file can also be done in ArcView or other GIS software tools.

Define any SQL Views on one or more or tables, as required to present various summaries or key indicators.

4. Spatial Data Registration

As each map layer is registered, you should note the following types of information: the ID field, Name field, Value Rendering field(s), Label fields. This information should be entered into the map metadata.

After each map layer is registered, define the feature rendering/labeling properties.

Add the metadata for each registered map layer. Common metadata components such as various common notes, codes and their values, should be standardized to maintain consistency.

As each map layer is registered, look at the results in Dynamic Maps. Ensure the map layer is being shown correctly (at the right scale) with the specified rendering properties. If it's a selectable map layer, make it active and check that the Identify, Find, Select by Name functions all work correctly.

5. Tabular Data Registration

As each tabular data set is registered, note the business identifier field name and the associated map layer. Add this information into the metadata for each registered tabular data set. Common metadata information such should be standardized to maintain consistency across all metadata.

As each table is registered, look at the results using Dynamic Maps. Ensure the tabular data set is being shown correctly - select one record at a time to see that it corresponds to the correct feature.

6. Geographically Linked Objects

Collect various objects (documents, photos, etc.) that can be associated with selectable map layer features.
Register them into the warehouse against selectable map layer features.

Appendix 7. – Installing the Warehouse Manager.mdb

Determine if Dynamic Knowledgebase is for Use in a Network or Standalone Scenario

Dynamic Knowledgebase can be used in either of two scenarios – in network mode in which you and potentially others are using Dynamic Maps, Dynamic Web Maps Server and Dynamic Knowledgebase to access data on a server, or in standalone mode where you are running Dynamic Maps on your machine and only need to access data on your machine.

You can't have the same warehouse for both local and network use.

In a network environment every warehouse and database connection and setting should be set up by using Network Neighborhood to specify a network pathname. In standalone mode everything should be set up using local pathnames – the warehouse location, the registration of map layers, data sets, etc. You can't mix and match! If you have set up a warehouse for a single computer (stand-alone mode) and then want to make it visible to other computers, the warehouse will have to be reset using network pathnames. This can be done by "Importing" each warehouse using a networked path name.

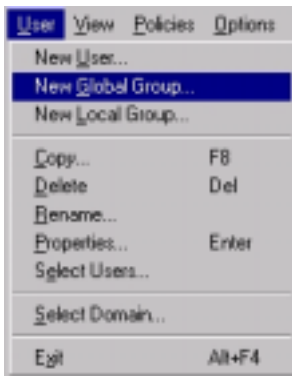
Install the Warehouse Manager.mdb

If this is a brand new install, the *Warehouse Manager.mdb* file needs to be installed **before you can run Dynamic Knowledgebase**. Run the ***Dynamic Knowledgeware Warehouse Manager 2.3.exe*** file and choose a directory that will be easily accessible to users on the network – if you intend to run Dynamic Maps over the network. The default location for the database is a directory called "Warehouse Manager" and it will be created at the root level on the drive that will be made visible to all the users (if Dynamic Web Maps is also accessing the warehouse then put the Warehouse Manager and the warehouses on one of the server drives).

See **Appendix 9** for assistance in installing Dynamic Knowledgebase.

Appendix 8. – Setting Up User Privileges for an NT Network

With an NT network installation of the Dynamic Maps and Dynamic Knowledgebase, the system administrator may find it necessary to set up a Dynamic Knowledgebase User Group and a Dynamic Maps User Group on the server where the data warehouse is located. In this way, appropriate access and permissions can be granted to users.



Set Up the Domain Group

On an NT Server that is the Domain Controller perform the following steps:

From the Start Menu and Programs, choose the Administrative Tools and then User Manager for Domains. Under the User menu, choose New Global Group.

Add a New Global Group and add the various domain users as members with the add button (this assumes that your users are already identified and have accounts on the system). Click **OK** to create the group. The Domain Users should already be added to the system. If they are not, each user can be added to this group later by editing the user properties.

Repeat this step twice, once for Dynamic Maps Users, and once for Dynamic Knowledgebase users.

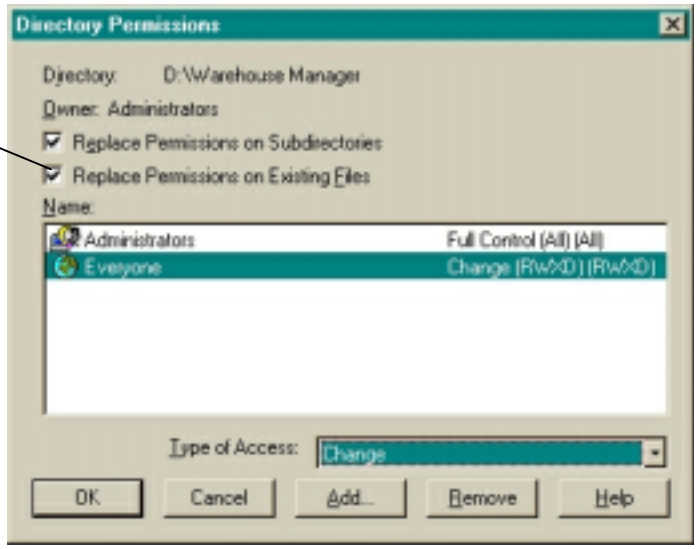
Next, make sure these new user groups have access to the computer, so under **Policies / User Rights**, grant "**Access this computer from network**" to the newly created groups.

Assign Appropriate Permissions

In order for each Group to be recognized as having authority to access the data, you may need to assign access privileges to the **Warehouse Manager** directory, and the **warehouse directory** and its subdirectories. Right-click on the directory name in Windows Explorer, then select Properties and the Security tab. We recommend that Dynamic Maps user group gets " Read and Execute" privileges to the directories, and that the Dynamic Knowledgebase user group has "Full Control" privileges to the directories.

You set permissions by selecting the user name from the Directory Permissions form list (shown below) and changing the access in the Type of Access drop-down menu.

Check on both these to replace permissions on files and sub-directories within the directory.



SKE

Setting Different Security Levels for Dynamic Maps Users for the Related Information Directory

Some organizations may want to restrict access to certain directories and files on their network while enabling some users of Dynamic Maps to publish related information. (Registering related information object is done through Dynamic Maps).

For Dynamic Maps users who only need to access the data in order to read and use it, they only require READ, EXECUTE privileges on ALL files and directories in each warehouse and to the Warehouse Manager directory and dB.

For Dynamic Maps users who need to be able to manage related information, they require:

- READ,WRITE,EXECUTE access to the "Related Information.mdb", and
- READ,EXECUTE,WRITE,DELETE access to the "Related Information" directory in each warehouse that they will be working in.

Make sure you test these scenarios in your environment to make sure they work.

> If you are also running Dynamic Web Maps Server, you will set up a special User Account (DWMS account) for those accessing the Warehouse from the web. See the Dynamic Web Maps Server manual for a detailed discussion on setting up the account and setting security.

Appendix 9. - Installing Dynamic Knowledgebase

Uninstall First

If you are installing a new version or re-installing Dynamic Knowledgebase, first remove the old program from the computer. To uninstall the old version, go to Control Panel / Add Remove Programs and find Dynamic Knowledgebase in the list of programs. Select it and click “Remove” and accept all the prompts. There may be a directory and a few files left in your “Program Files\Dynamic Knowledgebase” directory that are not removed by the uninstall process. These can be deleted manually.

*If the uninstall prompts whether you want to delete shared files, pick NO. They **should not** be deleted since they may affect other applications that depend on their shared components.*

Install Dynamic Knowledgebase

Dynamic Knowledgebase is installed using the installation program “***Dynamic Knowledgebase 2.x Install.exe***”. It is designed to work under Windows’95, Windows ’98, Windows 2000, and Windows NT Server/Workstation v4 with Service Pack 5 or better.

The Dynamic Knowledgebase tool should only be installed on a computer that will be used to administer the warehouse(s). If you are running in a stand-alone mode, all the software and data will reside on your machine. If this is a LAN environment, then the system administrator’s computer should act as the host for the Dynamic Knowledgebase application while the data and associated warehouse(s) must reside on the server. The other possibility in a LAN environment is to install the application on the server.

Insert the CD in the drive and navigate to the appropriate install file. Double-click on the file to begin installation, and accept any prompts.

> During the installation process, a couple of Microsoft upgrades may be installed on the computer as well. If Dynamic Knowledgebase is being installed on a Windows 95 machine, both DCOM and MDAC 2.1 will be automatically installed. If on a Windows 98, 2000 or NT machine, only MDAC will be installed. Both DCOM – Distributed Common Object Model, and MDAC – Multiple Data Access Components are required for the database and server interaction to work properly. If your machine

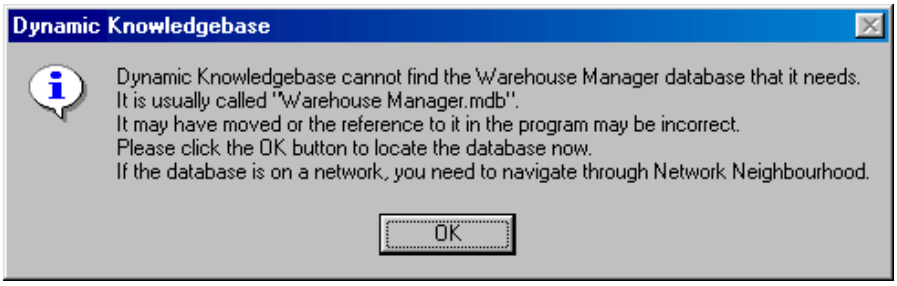
already has these components installed, their re-installation will not adversely affect any current program.

Once the installation process is complete, you MUST reboot your computer to ensure the appropriate registry settings are put in place.

Once installed, the Dynamic Knowledgebase program will appear on your **Start** menu.

Running Dynamic Knowledgebase for the First Time

The first time you run the program, it needs to connect to the warehouse manager database – called "***Warehouse Manager.mdb***" – so that it knows where to find the warehouse(s) that you are managing. When the program is launched for the first time, you will get a message saying that the program cannot find the database.



Click **OK** and, if you and others are accessing the warehouse(s) over a network, navigate via Network Neighborhood to the location of the database. This will mean navigating from the root level of the computer. Select the file, and then Dynamic Knowledgebase will run. If you point to the wrong file, the system will check and ask you to find the ***Warehouse Manager.mdb***. If you can't find it, try using the "Find" feature in Windows Explorer, or make sure that you have run the ***Dynamic Knowledgeware Warehouse Manager 2.3.exe*** file to install the database. The Warehouse Manager.mdb file will by default be installed in a directory called "Warehouse Manager" at the root level of the computer.