The Data Manager

Figure: Data Manager window screenshot with the following annotations:
- Add a sparse disk structure
- Add a Dimension
- Add a Compound structure
- Add an Alias View
- Add a private memory structure
- Click to view column mappings

Data Manager - ...P'1\toys.hdm

File Edit Node

Calculate Data	No
Caption	Structure TOYS_TBL
Consolidate Data	No Consolidate
Data File Store	HOSTFILE
Delete Underlying Dat	No
File Name	toys_tbl.hsh
Name	TOYS_TBL
Treat Data as Sparse	Yes

Details / Advanced / Cache

- Structure TOYS_TBL
  - Table TOYS_TBL
    - $ Column NUMBERS contains values
  - Dimension PRODUCTS
  - Dimension PROFLOSS
  - Dimension TIMEM
  - Dimension MEASURE

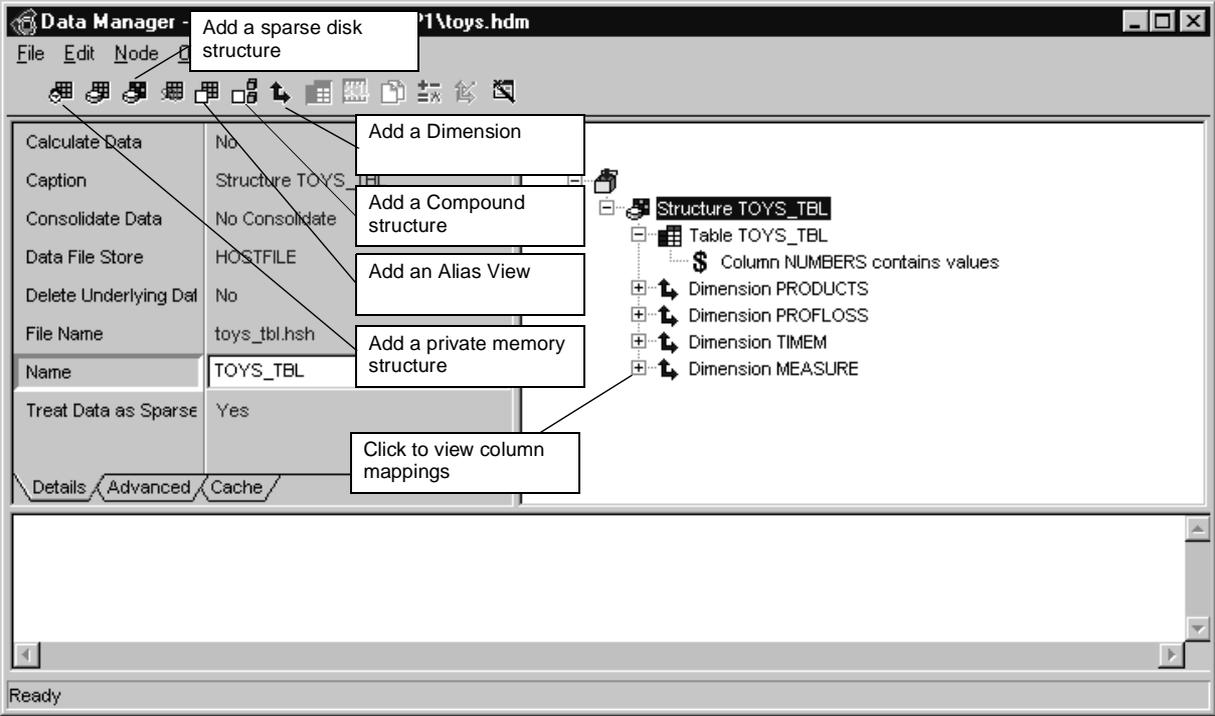Ready

The Data Manager (cont'd)

The difference between a create and a load script is...

The data manager creates two types of script, a 'create' script and a 'load' script.

The create script contains the code which is required to create objects from an external source (database tables or text files). If the name of a load script is also specified, the actions of making and executing the create script will also save copies of the objects created to disk for future use.

The load script makes use of the objects saved to disk by the create script to allow the same objects to be reloaded more quickly than the original creations.

In terms of a production application, the create script will often be used to create objects from a database in an overnight batch. To save time, the load script is used to load the created objects when the user logs on to the application.

How do I load from a simple (single) table?

To load data from a single table which contains all the dimension information as well as the data:

1. Open the Data Manager.
2. Add a structure object (of the required type).
3. Under the structure object add a database table object.
4. Choose the required database table name from the 'Name' property list.
5. Click the database table object button to show the table columns. Right click on each of the required columns in turn and choose one of the following:
   - **Field nam**e: if the column contains text which will become fields in a dimension.
   - **Longname**: if the column contains text which will become the longnames of a dimension.
   - **Parent**: if the column contains text which indicates the parent field of another column.

- **Value** - if the column contains values which will be used to populate the structure.
  (The 'Parent (AVG)' is the same as parent but creates an average hierarchy, and 'Fieldname from column name' should not be required for this type of load).
6. Click 'OK'. The Data Manager will now display a hierarchical representation of the structure you have specified. In many cases this will be all you need to do; however you may make some changes:
   - The name of the structure and its dimensions will have been taken from the names of the columns you have chosen. You can change these by altering their 'Name' properties.
   - The Data Manager will have attempted to match the fieldname columns with the longname and parent columns based on their relative position of the columns in the table. If these are not correct then you can simply drag and drop the appropriate nodes until they are correct.

You can now use the 'Make Script' option from the File menu to generate the code which will create and load the structure (this code will contain a LOAD block for each dimension, the appropriate STRUCTURE, SHARE STRUCTURE or ALIAS commands, and a LOAD block to populate the structure).

How do I load a structure from a star schema?

A simple table that contains all data and dimension information is rarely found in a production database, as it is rather inefficient. A standard method of storing multi-dimensional data more efficiently in a database is the 'star schema'. A basic star schema consists of a number of tables that separately hold the dimension information and the data.

The data is held in a table (called the fact table) which contains the data along with only the basic field information (usually coded). Then for each dimension there is a table (called a metadata table) containing information about its fields, longnames and hierarchy.

This is a simple two dimensional case:

Fact table

A	B	VAL
a1	b1	10
a2	b1	11
a2	b2	11

Meta table for dimension a

FLD	LONG	PARENT
a1	Field a1	at
a2	Field a2	at
at	Total	

Meta table for dimension b

FLD	LONG	PARENT
b1	Field b1	bt
b2	Field b2	bt
bt	Total	

This would represent:

```
DIMENSION a
    a1 'Field a1'
    a2 'Field a2'
    Total = a1+ a2
END DIMENSION
DIMENSION b
    b1 'Field b1'
    b2 'Field b2'
    Total = a1+ a2
END DIMENSION
STRUCTURE s a,b
```

Star schemas are often talked about as being very complex; in fact they are really a very simple concept.

To use the Data Manager to load data from a star schema, you need to create the dimensions first, followed by the structure.

To create a dimension on its own from a table:

1. Open the Data Manager.
2. Add a dimension object.
3. Add a table object below the dimension.
4. From the 'Name' property of the table object select the appropriate table.
5. Click on the table's button to view the table columns.
6. In the view of the table select each column in turn and give it the appropriate 'Type' property to indicate if the column contains fieldnames, longnames or parent field names.
7. Click OK and the Data Manager will give you a hierarchical view of the dimension.

If you need to, you can change the name of the dimension (using its 'Name' property) or the association of longnames and parents (by dragging and dropping the appropriate column objects).

The Data Manager (cont'd)

The Data Manager (cont'd)

It is possible to create the structure in the same Data Manager tree as the dimensions or in a separate one. One tree appears a little simpler, but doing them in separate trees gives more flexibility later. Whichever you choose, create the structure as follows:

1. Open the appropriate Data Manager object.
2. Click the button for the required structure type to add a new structure object.
3. Click the table button to add a table object below the structure.
4. Click on the table object's button to open the table viewer.
5. In the Table Viewer, use the 'Type' property of each of the required columns to identify each one as a field name or a value.
6. Click OK to see the Data Manager's hierarchical view of the structure.

What happens next depends on the names of the tables and columns from which you have created the dimensions and structure. There are two cases:

• If the names of dimension tables are the same as the names of the columns for the dimensions in the fact table, then the default names used by the Data Manager will be the same for the dimension creation and the structure creation and you should not need to do anything else. (In this case the Data Manager will have recognised that the dimensions in the structure already exist, which is why they have to be in memory, and will have changed their advanced 'Fields' property to 'Use existing'.)

• If the default names do not match, then for each of the affected dimensions, you should change its 'Name' property to the appropriate (existing) dimension. The Data Manager will automatically change the 'Fields' property from 'Create' to 'Use existing'.

You can now make and execute the script to create and load the structure.

How do I load a dimension from more than one table (snowflake schema)?

A complication that can be added to a star schema is to hold the information for a single dimension in more than one table. For example, consider the following dimension:

```
DIMENSION district
    d1 'district1'
    d2 'district2'
    d3 'district3'
    d4 'district4'
    r1 'region1' = d1 + d2
    r2 'region1' = d3 + d4
    'all regions' = r1 + r2
END DIMENSION
```

The meta information for this could naturally reside in the following two tables:

DISTRICT

FIELD	LONG	PARENT
d1	district1	r1
d2	district2	r1
d3	district3	r2
d4	district4	r2

REGION

FIELD	LONG	PARENT
r1	region1	all regions
r2	region2	all regions

A star schema with dimensions stored like this is often referred to as a 'snowflake schema'.

To create this dimension using the Data Manager, create two dimension nodes as if you were going to create a district and region dimension from their respective table (see the first part of 'How do I create a structure from a star structure?' for details).

Now all you need to do is change the 'Name' property of the second dimension to 'District' and change its advanced 'Fields' property to 'Add'. Making and executing this script will create the dimension you require.

How do I build a structure from existing dimensions?

It is possible to use the Data Manager to create a structure from existing dimensions without using an external data source:

1. Ensure that the dimensions you are going to use have been loaded into memory.
2. Open the Data Manager.
3. Add the appropriate type of structure object to your Data Manager tree.
4. Type the name of the structure into the 'Name' property.
5. Add as many dimension objects below the structure as you require.
6. For each of the dimensions, choose the appropriate dimension name from the drop-down list in its 'Name' property.
7. Change any of the other of the structure properties (such as the tuning parameters).
8. Make the script.

The script created will contain the appropriate structure creation syntax (STRUCTURE, SHARE STRUCTURE or ALIAS) and can be executed either in the Data Manager or in the Application Manager.

How do I change the filenames the Data Manager saves objects to?

It is not possible to change the filenames that the Data Manager uses to save objects to for future reloading. They are automatically saved to files of the form

`'object_name.extension'`

The Data Manager (cont'd)

e.g. `'region.dim'`, `'sales.str'` etc.

It is possible, however, to change the location to which the files are saved by changing the 'Common File Store' property of the Data Manager object in the Application Manager.

How do I use my own dimension for multiple values columns?

Consider the following table:

PRODUCT	REGION	SALES	COSTS
product1	region1	10	5
product2	region3	20	14
….			

If both the SALES and the COSTS columns were specified as 'Value' columns for a data load, then as well as a product and a region dimension, you would need to have another dimension into which the sales and costs values could be read. To make this possible, the Data Manager automatically creates just such a dimension. In this case the following dimension would be created:

```
DIMENSION DSALES
    SALES
    COSTS
END DIMENSION
```

The Data Manager (cont'd)

To use another dimension in place of the one created:

1. Ensure that the dimension you want to use is loaded into memory.
2. From the drop-down list in the 'Name' property for the dimension which the Data Manager has automatically created (this usually appears as the last dimension under the structure), select the dimension you want to use.
3. Click on the button on the dimension object and select the fields into which you want to load the data.
4. Rearrange the order of the column objects below the table object to match the order of the fields you have specified in the dimension.

If you make this change **before** you execute the create script for the first time, the Data Manager will **not** create its own dimension.

How do I specify a class of TIME or ITEMS for a dimension?

If you need to do this, select the dimension in the Data Manager and choose TIME or ITEMS in the drop-down list in the Details sheet.