

Registered Tables in Documentum

Registered tables are the tables that are present database which are registered in Documentum. So that it can be accessed using DQL.

Basically Registered tables are used when a system needs to access data from the database within the Documentum. The Scenarios I mostly used registered tables are for providing value assistance for Object attributes. I am not getting into too much of details about Value Assistance here but **Value assistance is a list of values that a client program (such as Webtop or a Custom WDK Application) displays at runtime for an object attribute. A user can select a value from this list (or, if allowed, add a new one to it).**

You can set the Value assistance for an Attribute using DAB (Documentum Application Builder).

You can register a Table or a view as a Registered Table,

The Registered tables are stored as **dm_registered** objects in repositories. This extends **dm_sybject**. And the r_object_id of this type always starts with **19**. The following table lists the attributes of dm_registered

Name	Info	Description
column_count	Integer - Single	Number of columns in the table.
column_datatype	string(64) - Repeating	List of the datatypes of the columns.
column_length.	integer R	Lengths of the columns that have a string data type
column_name.	string(64) - Repeating	List of the names of the columns in the table
group_table_permit	integer - Single	Defines the RDBMS table permit level assigned to the registered table's group.
is_key.	Boolean Repeating	Indicates if an index is built on the column
owner_table_permit	integer - Single	Defines the RDBMS table permit level assigned to the registered table's owner
synonym_for	string(254) - Repeating	Name of the table in the underlying RDBMS (can be an Oracle table synonym, or an MS SQL Server or Sybase table alias)
table_name	string(64) Single	Name of the table.
table_owner	string(64) Single	Name of the owner of the RDBMS table (the person who created the RDBMS table).
world_table_permit	integer - Single	Defines the RDBMS table permit level assigned to the world

You should either own the table or have super user privileges to register a table. And since this object is linked with /system cabinet you should have write permission on /system cabinet. This is applicable only if the folder security is enabled in Repository

-

You cannot version a dm_registered object. And also the changes made to the table are not automatically updated in dm_registered object. So if any changes has been made to the structure of the table or view you should unregister it first and register the table again with changes.

How to Register a Table?

Use the following DQL to register a table.

```
REGISTER TABLE [owner_name.]table_name (column_def {,column_def})
[[WITH] KEY (column_list)]
[SYNONYM [FOR] 'table_identification']
```

This DQL will return the r_object_id of the newly created dm_registered object.

In this

owner_name is the name of the table owner.

table_name is the name of the RDBMS table.

column_def defines the columns in the registered table.

column_def arguments should have following syntax **column_name datatype [(length)]** the valid values for types are float, double, integer, int, char, character, string, date, time.

Length should be specified for character, char, or string data type.

column_list Identifies the columns in the table on which indexes have been built.

column_list is usually separated with commas.

table_identification is the name of the table in the Database

Example:

```
REGISTER TABLE "hr.users" ("first_name" CHAR(30), last_name (char
40), "emp_id" INT)KEY ("emp_id")
```

Granting Rights

You need to give the permission to the users to access the registered tables. The values for various permission levels are as follows

0 (None): No access

1 (Select): The user can retrieve data from the registered table

2 (Update): The user can update existing data in the registered table

4 (Insert): The user can insert new data into the registered table

8 (Delete): The user can delete rows from the registered table

If a user wants update and insert permissions the value should be $2+4 = 6$, The repository owner also should have the same level of permission in the underlying database to grand those permission to those users.

Granting Rights full permission to users in the above example

```
update dm_registered object set world_table_permit = 15 where
object_name = 'users';
update dm_registered object set owner_table_permit = 15 where
object_name = 'users';
```

```
update dm_registered object set group_table_permit = 15 where  
object_name = 'users';
```

How to Unregister a Table?

Use the following DQL to Unregister a Table.

```
UNREGISTER [TABLE] [owner_name.]table_name
```

In this

owner_name is the name of the table owner.

table_name is the name of the RDBMS table.

You should be the owner of table or super user to do this

Accessing Data from Registered Table

Just like in RDBMS you can access registered table using the following syntax

```
Select [ATTRIBUTES] from dm_dbo.[REGISTERED_TABLE_NAME] where [CLAUSE]
```

The Operations such as update/ delete also has the same RDBMS syntax that's used for a ordinary SQL, Only difference is prefixing dm_dbo to the table name

Example:

```
Select first_name, last_name, emp_id from dm_dbo.users ;
```