

Few Important Object Types in Documentum

Documentum is an Object Oriented Content Management Systems. Everything in Documentum is considered as Objects, This includes all the things that user manipulates, or the content server saves. From the User to Document, everything in Documentum is different type of Object. The whole Object types in Documentum has been structured in a Hierarchical model. Means All the Attributes from the super types are extended to the sub types. In other sense the attributes of Super types are visible and accessible from the Subtypes.

Lets see some of the important and most commonly used object types here.

Note: This is just to give an insight of Documentum Object model. Please read Object Reference Manual for very detailed information on All Object types and its attributes.

Persistence Object

This type is the super type of all the object types that are saved in Documentum. Each time user creates an object instance of Persistence object type they are objects stored in the repository, this can be retrieved at a later point of time from the Content server.

This Object type is an internal type and you cannot create an instance of it. There are only 3 attributes for this object type they are. These attributes are extended to all the object types across the Documentum.

1) **r_object_id**

A Unique ID of an object. Content server generates this ID when you create an object of any type. There are some interesting aspects about r_object_id. r_object_id is 16 characters long and its alpha numeric.

The characters from position **01 to 02** indicate the object type tag (09 = document, 0b = folder, etc.).

The characters from position **03 to 08** Repository id (Same for every object in a Repository, but different for each Repository)

The characters from position **09 -to 16** is unique identifier for this object

Consider the following r_object_id **09012a5b80075dc2** in this 09 is the Object type Tag, 012a5b is the repository ID and 80075dc2 is the unique id that represents this object.

2) **i_vstamp**

This property is basically used for versioning, each time you save changes to the object the value of this property increases by 1 and this also helps to check the concurrent modification of object.

3) **i_is_replica**

This property of object that indicates whether that object is replica of an object in a remote repository

Sys Object (extended from Persistence Object) (dm_user)

Most of the commonly used object types are extended from this object type. Most common Subtypes include Document, Folder, and Cabinet etc.

There are 4 important characteristics for this Object type. They are

- 1) Only dm_sysobject and its subtypes can be defined as **shareable**
- 2) Only sys object support property bag
- 3) They can have permissions associated (Attach ACL).
- 4) Sys object can belongs to a folder (Exception Cabinet - Cabinet cannot belong to a folder)
- 5) Sys object can have content attached to it.

ACL (extended from Persistence Object) (dm_acl)

This object type plays a very vital role in implementing security to the Documentum server if the security model of Content Server is set to ACL. All ACL objects r_object_id starts with letters **45**

User (extended from Persistence Object)(dm_user)

This stores all the information about user in a Documentum repository, Only a Super user or Sys Admin can create/activate/delete/deactivate a user. All User Object's r_object_id starts with letters **11**

Group (extended from Persistence Object)(dm_group)

A group is the group of users and it can include another groups also. This object stores information about a specific group, which includes r_object_id of all the member users, groups. This has identifier which determines whether it's a group or a role. All the group objects has r_object_id starting with **12**

Document (extended from Sys Object)(dm_document)

Documents represent a real document in Repository; it can be associated with 0 or more content objects also. A Document may be a real document or virtual document. All objects of this type has a r_object_id starting with **09**

Folder (extended from Sys Object)(dm_folder)

Folders are basically used to organize contents. All the sys objects that are created should be linked with at least one folder or a Cabinet. An Object can be linked with multiple folders also. All objects of this type has a r_object_id starting with **0b**

Cabinet (extended from Sys Object)(dm_cabinet)

Cabinet is a special type of folders and its used to organize sys object in a repository. Cabinets are the highest in the Folder Hierarchy in Documentum. A Cabinet cannot be placed inside a Cabinet or a Folder and that makes it special. All objects of this type has a r_object_id starting with **0c**

Registered (extended from Sys Object)(dm_registered)

Represents a Registered table in a RDBMS in Documentum. It has the table, name, column names and it s data type saved. All objects of this type has a r_object_id starting with **19**

Object types that are not saved in Repository

Documentum is Object oriented all the interactions with Documentum are Object based. There are few Object types in Documentum, which are not to be saved in the repository. These are some object types that are created on runtime and destroyed as its use is over. Here is the list of those object types.

Client config (New in D6)

A Client config object is created at every time when DFC is initialized. This has all the properties in the dfc.properties file.

Connection Config

This type describes a session's connection to a specific repository

Docbroker Locator

A docbroker locator object contains information about each connection broker that the client DMCL can access

Docbase Locator

This has information all repositories registered with a connection broker

Server Locator

A server locator object has information about all the servers registered to a connection broker

Session Config

Contains information about an open repository session