What is runtime object? Where are the runtime objects stored?

What is runtime object?

The runtime objects are contained in some database tables (DDNTT, DDNTF) which contain the basic technical definitions of a dictionary object (domain, data element, table, structure, ...). The ABAP interpreter needs the runtime objects to execute an ABAP program. Each time a dictionary object is created or modified, the corresponding runtime object is updated.

After you created and activated the ABAP dictionary object, the active runtime object will be created automatically. You can display the runtime object via Utilities -> Runtime Object -> Display

Where are the runtime objects stored?

There are several database tables which contain different types of runtime objects:

- **DDNTT**: contains the header data of all active runtime objects for all ABAP dictionary objects.
- **DDNTF**: contains the field information of all active runtime objects of the ABAP dictionary objects (tables/structures).
- **DDXTT**: contains corresponding data for inactive runtime objects against DDNTT.
- **DDXTF**: contains corresponding data for inactive runtime objects against DDNTF.

In a running system, only the table DDNTT and DDNTF which contains active runtime objects are used. The active runtime objects must be completely and correctly filled in table DDNTT and DDNTF at system startup. If one ABAP dictionary object is modified in transport mode, the corresponding inactive runtime object will be stored in table DDXTT and DDXTF to contain the changes temporarily. If the ABAP dictionary object is activated, the corresponding entry is deleted from table DDXTT and DDXTF, and updated in the table DDNTT and DDNTF.

Table DDNTT and DDNTF exist on database level only. They are not defined in the ABAP Dictionary. The content within them can be displayed with the function module ‘DD_SHOW_NAMETAB’. When executing this function module in transaction SE37, You can specify parameter “Status”, to display the active (Status = A) or inactive (Status = I) runtime object.

Table DDXTT and DDXTF are defined in ABAP Dictionary. Therefore they can be accessed easily via transaction SE16.

How to check whether the runtime object is consistent or not?

Sometimes, you need to check whether the active runtime object is consistent with the ABAP dictionary definition via Utilities -> Runtime Object -> Check:
You can also display the runtime object for an dictionary object using function module DD_SHOW_NAMETAB in SE37. Execute specify the object name in parameter TABNAME, execute display output list of function module.

You can use function module DD_NAMETAB_DELETE to delete the runtime object for an dictionary object in case any necessary. After that, the runtime object can be newly generated automatically during activation of the dictionary object in SE11.

Generally the runtime object should be consistent with the ABAP dictionary definition.

But in some situation, you may find that the runtime object is inconsistent with ABAP dictionary. For example:

As above, the length of field EMPLOYE_NAME in the runtime object is CHAR(20). But according to the dictionary definition, the length of field EMPLOYE_NAME should be Char(40).

In most cases, the inconsistency between the runtime object and the dictionary definition would lead to problems during system operation. If the runtime object is inconsistent with dictionary definition, jobs, programs or transactions which access the dictionary object will terminate with runtime errors, such as DDIC_TYPE_INCONSISTENCY, DDIC_TYPELENG_INCONSISTENT, DDIC_TYPE_REF_ACCESS_ERROR, TYPELOAD_NEW_VERSION etc.

So, generally it is necessary for us to make the runtime object and the dictionary definition to be consistent with each other.

Note that the inconsistencies between the dictionary definition and the runtime object are normal during transports, Support Packages implementation or system upgrade in some particular phases.