NOTE FEBRUARY 2008

ABAP/4® Developments

Development in general is about the creation of new objects or the modification of existing objects.

One language for programming in the SAP® environment is ABAP/4® [Advanced Business Application Programming].

Developments are supposed to be performed in the development system only, and from there they are technically transported with the help of the Transport Management System [transaction **STMS**] through the Quality-system into the Production system.

The overview of tools for the Change and transport system can be gained with transaction **SE03**.

Change control

Whether and how developments may occur is controlled, aside from the corresponding necessary authorizations, by technical system settings and additional development keys.

The system changeability [transaction **STCS_RSWBO004**] and the client specific settings in **SCC4**. For more detailed information, please read System and client change options.

Developer keys are stored in the table **DEVACCESS**. These keys allow general developments. For changes to SAP® standard objects, additional object keys are required. The object keys are listed in the table **ADIRACEESS**.

Request / Task

Developments are internally organized. They are recorded as requests. One request may consist of one ore more tasks as part of this development. A task is always related to one developer and may include several objects.

There are four different request types:

Customizing requests contain the corresponding changes, and are controlled by the client specific settings as part of transaction **SCC4**.

There are two types of *Workbench* requests, local [not to be transported] and transportable.

Copies allow to transport objects from a list to another system, and three types of *Relocations* allow to move objects from the original location.

Some relevant evaluation reports [access via SE38 / SA38] for searches concerning requests / tasks are:

RSWBOSDR – search for requests / tasks

RSWBO040 – search for objects

RSWBO050 – analyze objects

RDDPROTT – transport logs

To review objects in the customer name space the transaction **CUSTMON1** can be utilized.

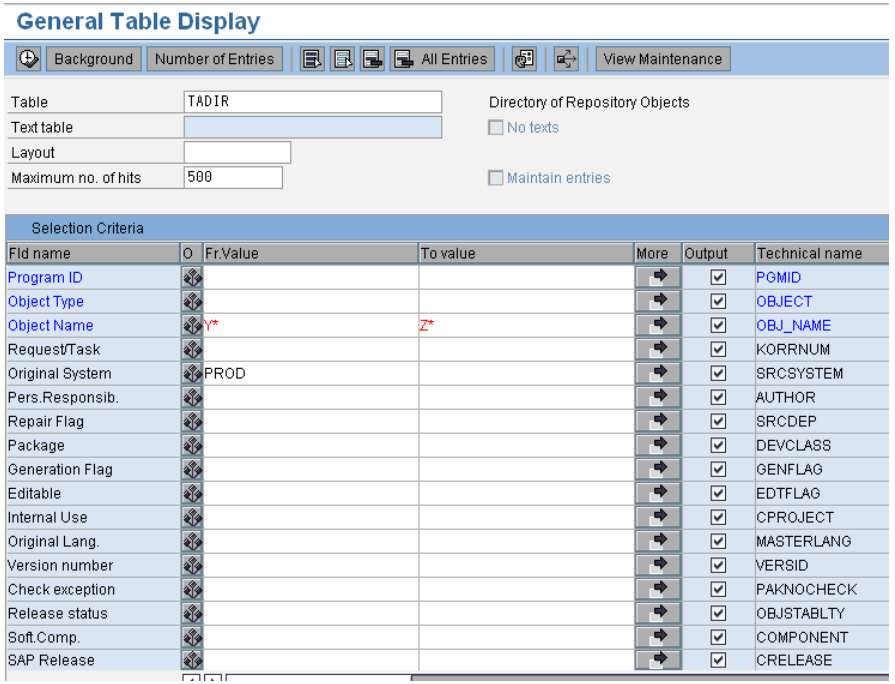
The table **E070** stores the header information of requests / tasks. Table **E070A** request attributes, table **E070C** source / target client information. The creation dates are stored in table **E070CREATE**.

In table **E071** the object entries are listed, in table **E071K** the corresponding key entries can be found.

Report Analysis

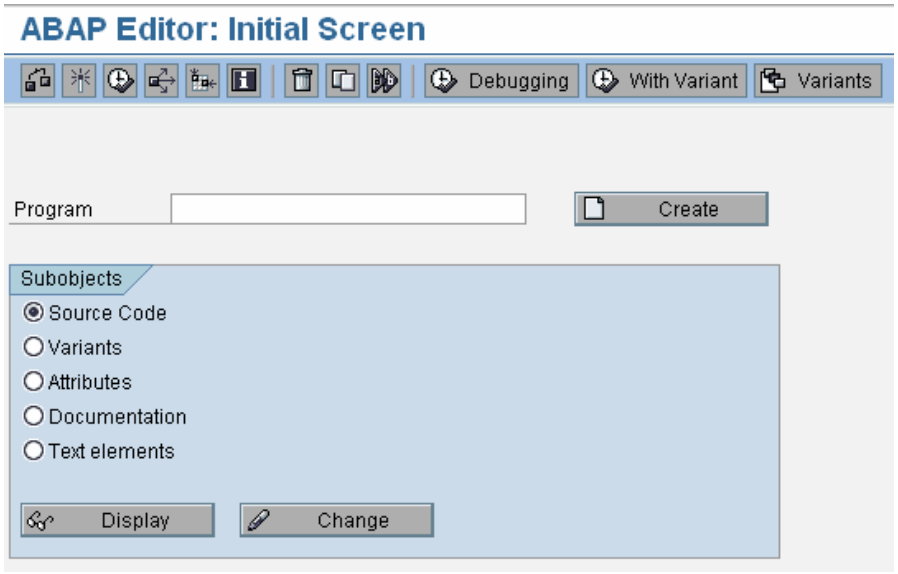
All objects of the SAP® Repository are stored in the directory table **TADIR**.

To identify objects that were developed in the production system and not in the development system, you can execute a table inquiry for **TADIR** [via transaction SE16N for e.g.] . In the field *Object name* you narrow your selection according to the “Customer namespace” [Y*, Z*], and in the field *Original system*, you enter the name of your production system. To execute the inquiry, push key *F8*.



SAP® offers several options to analyze programs.

Transaction **SE38** allows to call the APAP/4® Editor for review of the source code and corresponding documentation.

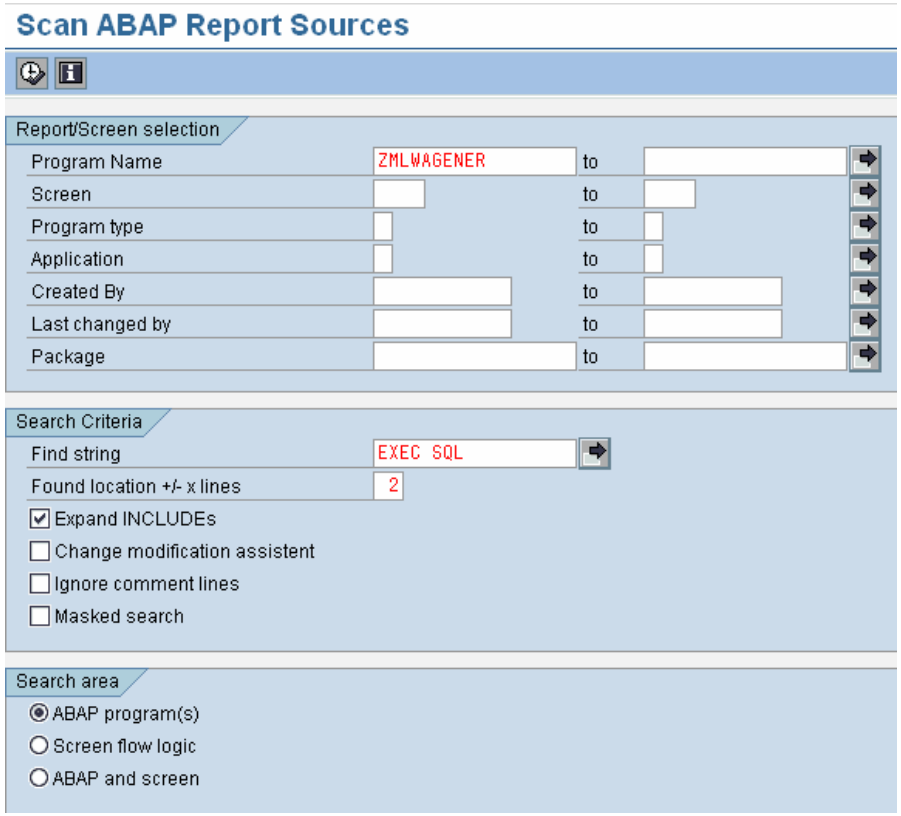


Enter the name of the report you want to analyze, activate the radio button for the desired information and push the button *Display*.

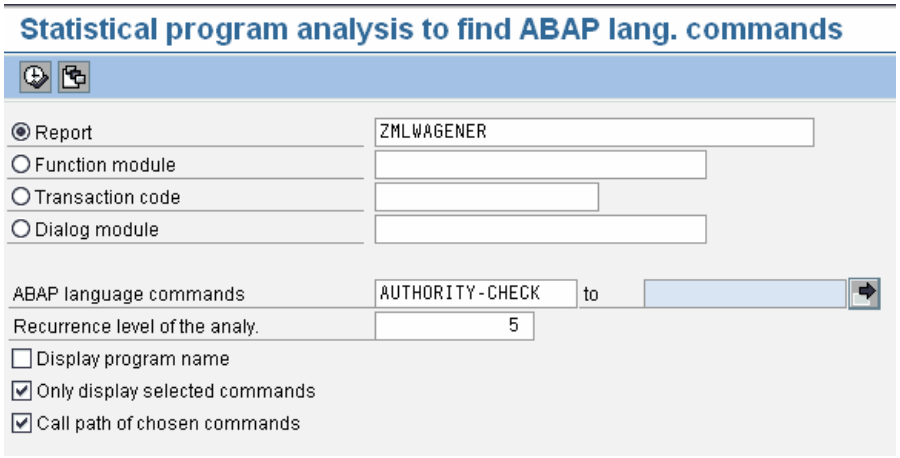
The report **RSANAL00** [can be called via transaction **SE38** or **SA38**] helps to analyze ABAP/4® programs.

Other reports that are available for the source code analysis are **RPR_ABAP_SOURCE_SCAN** and **RSABAPSC** [both can be called via **SE38**/**SA38**]. These reports allow to search for specific strings, like e.g. “authority-check”, or “EXEC SQL”.

RPR_ABAP_SOURCE_SCAN checks through all recurrence levels.



In **RSABABSC** you can enter the desired depth for the search.



Version History

The version history of objects can be reviewed with the help of the report **RSVCDI00.**

The version management directory is stored in table **VRSD**.

In the table **SMODISRC** the log information for active customer modifications are contained, while table **SMODISRCI** stores the information for inactive objects.

The table **SMODILOG** contains the log information of active customer modifications, the table **SMODILOGI** inactive with date, author and transport request /task ID for example.