Standardized dialogs (popup function modules, etc.)

- Introduction
  - Standardized dialogs
    - Standardized function modules made obsolete!
    - Other popup function modules (non standardized)
  - File dialogs

Introduction

Standardized dialogs are frequently used popup dialogs that SAP allows customer to be used in their custom programs. This article also presents other dialogs (files) though SAP does not classify them as "standardized" (but they are supported).

You may refer to:

Standardized dialogs

The introduction is copied from the SAP Library - Standardized Dialogs, and hyperlinks have been added to each function module which has its own wiki.

Since SAP NetWeaver 7.0 (ECC 6 for example), SAP simplified the function modules by merging those of function group SPO1 into POPUP_TO_CONFIRM. Merged function modules are flagged obsolete since then. Some wikis discuss about that, they are linked in next chapter

Some dialog steps and functions are required frequently during the realization of application development dialogs. These are available as function modules in self-contained modules. Their use standardizes application dialogs, which in turn simplifies use.

- Confirmation prompt dialogs - Function group SPO1
  - With this function module you create a dialog box in which you ask the user during an action, whether he or she wishes to perform the step. You pass the title and a two-line question. The user must choose Yes, No or Cancel. The possible responses are provided by the function module. In the interface, the response "Yes" is pre-selected, but "No" can also be pre-selected via a parameter. The user response (Yes, No or Cancel) is returned in a parameter.

- With this function module you create a dialog box in which you inform the user about a specific decision point during an action. You pass a title, a three-line diagnosis text and a two-line question, which he or she must answer with Yes, No or Cancel. The possible responses are provided by the function module. In the interface the response "Yes" is pre-selected, but "No" can also be pre-selected via a parameter. The user response (Yes, No or Cancel) is returned in a parameter.

- With this function module you create a dialog box in which you ask the user, during an action, whether he or she wishes to perform a processing step with a particular object. You pass a title, a two-line decision question and the object, which is inserted between the two parts of the question. The user must choose Yes, No or Cancel. The possible responses are provided by the function module. In the interface the response "Yes" is pre-selected, but "No" can also be pre-selected via a parameter. The user response (Yes, No or Cancel) is returned in a parameter.

- Dialogs for choosing from among alternatives - Function group SPO2
  - With this function module you create a dialog box in which you ask the user, during an action, whether he or she wishes to perform a processing step with loss of data. You pass the title and the two-line question. The warning that data are lost and the possible responses are provided by the function module. The user must answer "Yes", "No" or "Cancel". In the interface the response "No" is pre-selected and can not be changed. The user response (Yes, No or Cancel) is returned in a parameter.
With this function module you create a dialog box in which you require the user to choose between the two further processing alternatives offered, or to cancel the action. The action, the question and the alternative actions are passed as parameters. In the interface the alternative 1 is pre-selected, but alternative 2 can also be pre-selected via a parameter. The user action (Alternative 1, Alternative 2, or Cancel) is returned in a parameter.

With this function module you create a dialog box in which you inform the user about a specific decision point via a diagnosis text, during an action. He or she can choose one of two alternative actions offered or cancel the action. The action, the diagnosis text, the question and the alternative actions are passed as parameters. The user action (Alternative 1, Alternative 2, or Cancel) is returned in a parameter.

### Dialogs for displaying, inputting and checking data - Function group SPO4

#### POPUP_GET_VALUES

This function module sends a dialog box for data display and input. The input fields are passed in a structure and must be defined in the Dictionary. You can also specify individual field display attributes and a field text, if the key word from the Dictionary is not to be displayed as field text in the dialog box, in the structure. The standard help functionality (F1, F4) is supported.

#### POPUP_GET_VALUES_DB_CHECKED

This function module sends a dialog box for data to be input und checked against the database. The input fields are passed in a structure and must be defined in the Dictionary. You can also specify individual field display attributes and a field text in the structure, if the key word from the Dictionary is not to be displayed as field text in the dialog box. A comparison operator for checking the input data in the database is passed. You can specify whether the check is for the existence or absence of an object. A foreign key relationship check is supported. The standard help functionality (F1, F4) is supported. The user action is returned in a parameter.

#### POPUP_GET_VALUES_USER_CHECKED

This function module sends a dialog box for data to be input and checked in an external sub-routine (user exit). The input fields are passed in a structure and must be defined in the Dictionary. You can also specify individual field display attributes and a field text in the structure, if the key word from the Dictionary is not to be displayed as field text in the dialog box. The Data input by the user in the dialog box are passed to the sub-routine specified in the interface for checking. Errors found by the check are entered in an error structure and are evaluated on return from the sub-routine by the function module. The standard help functionality (F1, F4) is supported. The user action (Continue or Cancel) is returned in a parameter.

#### POPUP_GET_VALUES_USER_HELP

This function module sends a dialog box for data to be input with the possibility of a check in an external sub-routine (user exit) and branching in a user F1 or F4 help. The input fields are passed in a structure and must be defined in the Dictionary. You can also specify individual field display attributes and a field text in the structure, if the key word from the Dictionary is not to be displayed as field text in the dialog box. You can pass the data which are entered by the user in a dialog box to a sub-routine which must be specified in the interface for checking. Errors occurring in the check are stored in an error structure and are analyzed by the function module upon return from the sub-routine. The data, and an error message, if appropriate, are displayed again. The standard help functionality (F1, F4) is supported. User exits for a user F1 or F4 help can also be specified. The user action (Continue or Cancel) is returned in a parameter.

#### POPUP_GET_VALUES_USER_BUTTONS

This function module is like the previous function module POPUP_GET_VALUES_USER_HELP, with the additional possibility of passing one or two additional pushbuttons and a standard pushbutton, which the user can name.

#### POPUP_GET_VALUES_SET_MAX_FIELD

With this function module you can specify the maximum number of fields which can be displayed in dialog boxes for this function group (SPO4). The specified value is stored in the function group local memory and applies for the rest of the application. Dialog boxes which display more then this number of fields are displayed with a scroll bar.

### Text display dialogs - Function group SPO6

#### POPUP_DISPLAY_TEXT

With this function module you display a text which exists in the system in a dialog box.

#### POPUP_DISPLAY_TEXT_WITH_PARAMS

With this function module you display a text which exists in the system with parameters in a dialog box. The parameter values are passed in a table. The use of numbered texts is recommended, to make the parameter values translatable. The parameter names must be passed in upper-case letters.
• Scrolling in tabular structures - Function group STAB
  • SCROLLING_IN_TABLE

  With this function module you enable the user to scroll in a list which you have created, e.g. as a logical part of an internal table. You can enable either page-wise scrolling or positioning on individual records.

**Standardized function modules made obsolete!**

The following explain how to migrate code to replace

- Standardized dialogs obsolete since NetWeaver 7.0 (solutions based on NW 7.0: ECC 6)
  - [ ] (replaced by POPUP_TO_CONFIRM)
  - POPUP_TO_CONFIRM_STEP (replaced by POPUP_TO_CONFIRM)
  - POPUP_TO_CONFIRM_WITH_MESSAGE (replaced by POPUP_TO_CONFIRM)
  - POPUP_TO_CONFIRM_WITH_VALUE (replaced by POPUP_TO_CONFIRM)
  - POPUP_TO_DECIDE (replaced by POPUP_TO_CONFIRM)
  - POPUP_TO_DECIDE_WITH_MESSAGE (replaced by POPUP_TO_CONFIRM)

- Standardized dialogs removed since NetWeaver 7.0
  - * Data print dialogs - Function group STRP
    - TABLE_PRINT_STRUCTURE_KNOWN
      You pass data from tables whose structure is known in the program to this function module.
    - TABLE_PRINT_STRUCTURE_UNKNOWN
      You pass data from tables whose structure is not known in the program to this function module. These structure data are fetched independently by the function module.

**Other popup function modules (non standardized)**

Non standardized means SAP has not released them to be used by customers. Unfortunately, customers often bypass this rule.

- [ ]
- [ ]

- REUSE_ALV_POPUP_TO_SELECT : list display
- POPUP_WITH_TABLE_DISPLAY or POPUP_WITH_TABLE_DISPLAY_OK

**File dialogs**

- CL_GUI_FRONTEND_SERVICES=>FILE_OPEN_DIALOG method (Basis, 4.6 and after): displays list as a popup with title "Open", and returns the name of selected file
- CL_GUI_FRONTEND_SERVICES=>FILE_SAVE_DIALOG method (Basis, 4.6 and after): displays list as a popup with title "Save", and returns the name of selected file

Links:

- Working with files