Template Function Modules of the SAP Application Interface Framework

Applies to:

SAP Application Interface Framework

Summary

In customizing of the SAP Application Interface Framework several places exist where function modules can be maintained. Those function modules need to have the correct parameters. Therefore, template function modules are delivered with the SAP Application Interface Framework. This document provides you with information about those template function modules.

Author: Verena Wörner

Company: SAP AG

Created on: 03 December 2012

Author Bio

Verena Wörner is a developer for SAP Custom Development. She is a member of the development team for the SAP Application Interface Framework.

Table of Contents

Introduction

In the SAP Application Interface Framework, several places exist where function modules can be used. These function modules are usually called dynamically by the SAP Application Interface Framework and consequently need to implement a specific, predefined signature. To support the interface developer in creating the function modules, template function modules are delivered with the SAP Application Interface Framework. The template function modules can be found in function group /AIF/AIF_TEMPLATES. The interface developer can copy the templates to a new function module and implement it.

Alternatively you have the possibility to directly create your function module from the corresponding input field in customizing. You have to enter the name of the function module you would like to create. If the function module does not exist the system will ask you if you want to create it. The template function module will be copied to your new function module and you can implement it.

Note that the templates have some parameters, where no type is specified, for example, for the raw or SAP data structure. You might want to change the types according to your needs, for example, to benefit from code inspector checks. In this document, an overview of existing templates is given. The document explains where the templates are used and gives details about the function module's parameters.

Function Modules in Define Interfaces

/AIF/FILE_TEMPL_CHECK_MAPPING

Type	Parameter	Description
Importing	RAW_STRUCT	Contains the data of the source structure
	RETURN_TAB_MAPPING	Contains the messages that occurred during mapping. In the function module, you can evaluate the errors that occurred during mapping.
	SENDING_SYSTEM	Optional parameter. A field of the source structure needs to be maintained in *System Field* in the interface definition. If the function module should behave differently for different senders, you can use this parameter to differentiate the behavior.
Tables	RETURN_TAB	Insert messages, for example, if an error occurred, into this table (for example, with helper function /AIF/UTIL_ADD_MSG). The messages are written to the application log.
Changing	OUT_STRUCT	Contains the data of the destination structure. With this function module, you can change the mapped data. Using importing parameter RETURN_TAB_MAPPING, you may be able to correct errors that occurred during mapping.

The function module needs to be maintained in *Check Function Module.* The function module is executed after the mapping and before the execution of the actions. You can use the function module to facilitate a certain behavior depending on the result of the mapping. For example, you could correct mapping errors in the function module.

/AIF/FILE_TEMPL_INIT_MAPPING

Type	Parameter	Description
Importing	CONTEXT	Defines in which context the function module is called (for example, CONTEXT = TRANSFORM)
	FINF	Contains the data of the interface definition of the current interface you are using (for example, raw data structure, sap data structure, and proxy class).
	SENDING_SYSTEM	Optional parameter. A field of the source structure needs to be maintained in *System Field* in the interface definition. If the function module should behave differently for different senders, you can use this parameter to differentiate the behavior.
Tables	RETURN_TAB	Insert messages, for example, if an error occurred, into this table (for example, with helper function /AIF/UTIL_ADD_MSG). The messages are written to the application log.
Changing	DATA	Contains the destination structure. The destination structure should be empty since the function module is executed before the mapping. However, you may enrich the structure with data in the function module.
	RAW_STRUCT	Contains the data of the source structure. You can change the data in the function module.
Exceptions	CANCEL	Exception that can be raised by the function module in case of errors. However, it is recommended to add errors to return tab RETURN_TAB.

The function module is executed before mapping in the SAP Application Interface Framework. You can use the function module to manipulate data in the source or destination structure before the mapping is started. The template has to be used in *Init Function Before Mapping* in Customizing activity *Define Interfaces*.

/AIF/FILE_TEMPL_INIT_PROCESS

Type	Parameter	Description
Importing	CONTEXT	Defines in which context the function module is called (for example, CONTEXT = PROCESS)
	FINF	Contains the data of the interface definition of the current interface you are using (for example, raw data structure, SAP data structure, and proxy class).
	SENDING_SYSTEM	Optional parameter. A field of the source structure needs to be maintained in *System Field* in the interface definition. If the function module should behave differently for different senders, you can use this parameter to differentiate the behavior.
Tables	RETURN_TAB	Insert messages, for example, if an error occurred, into this table (for example, with helper function /AIF/UTIL_ADD_MSG). The messages are written to the application log.
Changing	DATA	Contains the data of the destination structure after mapping. You may change the data in this function module.
Exceptions	CANCEL	Exception that can be raised by the function module in case of errors. However, it is recommended to add errors to return tab RETURN_TAB.

The function module needs to be maintained in *Init Function Before Processing.* This function module is called after mappings were executed but before the processing of the action starts. You can use this function module to manipulate the data in the destination structure before the processing of the action(s) starts.

Value Mappings

/AIF/FILE_TEMPL_VALMAPPING

Type	Parameter	Description
Importing	VALUE_IN	The fields defined in the field mapping (*Fieldname 1* to *Fieldname 5*) are used as importing parameters.
	VALUE_IN2	
	VALUE_IN3	
	VALUE_IN4	
	VALUE_IN5	
	SENDING_SYSTEM	Optional parameter. A field of the source structure needs to be maintained in *System Field* in the interface definition. You can enter different values, based on the system the data comes from, into parameter VALUE_OUT.
Tables	RETURN_TAB	Insert messages, for example, if an error occurred, into this table (for example, with helper function /AIF/UTIL_ADD_MSG). The messages are written to the application log.
Changing	VALUE_OUT	Value that the field in the destination structure should have after the mapping function is executed.
Exceptions	NO_VALUE_FOUND	Exception that can be raised if the correct value could not be mapped.

The function module is executed during mapping. There are several places where the template function module is used:

- Customizing activity *Define Value Mappings*
- Customizing activity *Define Field Mappings*
- Customizing activity *Interface Variants* → *Define Variant Mappings* → *Assign/Define Field Mappings*

Checks

/AIF/FILE_TEMPL_CHECK

Type	Parameter	Description
Importing	DATA_STRUCT	Contains the data of the current structure. In case the check is executed during structure mapping, the current structure is either the SAP or raw data structure. This is dependent on the *Check raw data* indicator in the check assignment. In case the check is executed in an action, the structure is the destination structure. If the check is executed for a condition, the current structure is the source structure.
	DATA_LINE	Contains the current line of the structure that is mapped
	DATA_FIELD	If the check function is executed in the structure mapping or as condition and at least *Fieldname 1* is defined, this field contains the value of the field defined in *Fieldname 1*.
	MSGTY	Message type
	VALUE1	Contain the values of the fields defined when the check is assigned (*Fieldname 1* to *Fieldname 5*).
	VALUE2	
	VALUE3	
	VALUE4	

	VALUE4	
	T_IFCHECK	Contains the Customizing data of the check assignment. This parameter is filled if the check function is executed during structure mapping.
	T_IFACT	Contains the Customizing data of the assigned action for which the check function is executed. This parameter is only filled if the check is assigned to an action in the Customizing activity *Define Actions*.
	T_ACCHECK	Contains the Customizing data of the check assigned to the current action. This parameter is only filled if the check is assigned to an action in the Customizing activity *Define Actions*.
	T_FUNC	Contains the Customizing data of the function, whose execution depends on the result of the check. This parameter is only filled if the check is assigned to an action in the Customizing activity *Define Actions*.
	T_FMAPCOND	Contains the Customizing data of the *Conditional Mapping*. This parameter is only filled if the check is assigned to a conditional field mapping.
	T_CHECK	Contains the Customizing data of the assigned check. This field is always filled.
	T_TABCHK	Contains the Customizing data of the single check that is currently executed.
	SENDING_SYSTEM	Optional parameter. A field of the source structure needs to be maintained in *System Field* in the interface definition. If the function module should behave differently for different senders, you can use this parameter to differentiate the behavior.
Tables	RETURN_TAB	Insert messages, for example, if an error occurred, into this table (for example, with helper function /AIF/UTIL_ADD_MSG). The messages are written to the application log.
	DATA_TABLE	Table that contains all data of the current structure that is mapped. You can change this data in your check function.
Changing	ERROR	Indicator for indicating if the check was successful. Needs to be set by a developer. If ERROR = abap_true, the check is treated as unsuccessful.

You can use this function module to execute any type of check and return the result to the SAP Application Interface Framework. There are the following places within Customizing where this template is used as follows:

- Customizing activity *Define Structure Mapping → Assign Checks*
- Customizing activity *Define Structure Mapping → Define Conditions*
- Customizing activity *Define Actions*
- Customizing activity *Interface Variants → Define Variant Mapping*

Define Structure Mapping

/AIF/FILE_TEMPL_MAP

Type	Parameter	Description
Importing	RAW_STRUCT	Contains the source structure of the interface.
	RAW_LINE	Contains the row of the table/structure of the source structure for which the after mapping function is currently executed.
	SMAP	Contains the Customizing data of the current structure mapping.
	INTREC	Contains the record type of the current structure
	SENDING_SYSTEM	Optional parameter. A field of the source structure needs to be maintained in *System Field* in the interface definition. If the function module should behave differently for different senders, you can use this parameter to differentiate the behavior.
Tables	RETURN_TAB	Insert messages, for example, if an error occurred, into this table (for example, with helper function /AIF/UTIL_ADD_MSG). The messages are written to the application log.
Changing	OUT_STRUCT	Contains the destination structure
	DEST_LINE	Contains the data of the current line in the destination structure. You might change the data of this parameter in this function module.
	DEST_TABLE	Contains the data of the current structure in a table
	APPEND_INDICATOR	Indicator for indicating if a line should be added to the destination table. Usually, if an error occurs in a value mapping, the line is not added to the destination structure and the indicator is blank. If you want to add the line anyway, you have to set this indicator.

The function module is processed after or before the mapping of the current structure. This template is used in the Customizing activity *Define Structure Mapping* for function after mapping and function before mapping in *Assign Destination Structure*.

Define Actions

/AIF/FILE_TEMPL_INIT_ACTION

Type	Parameter	Description
Importing	CONTEXT	Defines in which context the function module is called (for example, CONTEXT = TRANSFORM)
	FINF	Contains the Customizing data of the current interface
	ACTION	Contains the name of the current action
	SENDING_SYSTEM	Optional parameter. A field of the source structure needs to be maintained in *System Field* in the interface definition. If the function module should behave differently for different senders, you can use this parameter to differentiate the behavior.
Tables	RETURN_TAB	Insert messages, for example, if an error occurred, into this table (for example, with helper function /AIF/UTIL_ADD_MSG). The messages are written to the application log.
Changing	DATA	Contains the data of the destination structure after mapping. You can adapt the data in your function module.
Exceptions	CANCEL	Exception that can be raised by the function module in the case of errors. However, it is recommended to add errors to return tab RETURN_TAB.

The template can be used in the Customizing activity *Define Actions*. The function module is executed at the beginning of each action call before the action function modules are called. How often this processing function is executed is finally determined during runtime, based on the content of the message that is mapped and the main component type the action handles. If the component type is a structure, the function is executed only once. In the case of the main component type being a table, the function module is executed for every dataset in the table.

/AIF/FILE_TEMPL_PROCESS

Type	Parameter	Description
Importing	TESTRUN	Indicator for indicating if the action is executed in test mode. You can set this indicator, for example, to prevent database updates if the action is executed in test mode. Furthermore, you can use it to pass it to BAPIs supporting test calls.
	SENDING_SYSTEM	Optional parameter. A field of the source structure needs to be maintained in *System Field* in the interface definition. If the function module should behave differently for different senders, you can use this parameter to differentiate the behavior.
Tables	RETURN_TAB	Insert messages, for example, if an error occurred, into this table, (for example, with helper function /AIF/UTIL_ADD_MSG). The messages are written to the application log.
Changing	DATA	Contains the data of the destination structure after mapping
	CURR_LINE	Contains the data of the current line. When copying the template you should at least change the type of CURR_LINE to the main component type.
	SUCCESS	Indicator for indicating if the action was executed successfully.
	OLD_MESSAGES	Contains messages that might have occurred in past processing of the action

This template should be used for every function of the Customizing activity *Define Actions → Define Functions*. **Note** that you should not raise an error. Instead, you should fill return table RETURN_TAB with your messages. If there is a message of type *E* or *A* in the return table, the SAP Application Interface Framework implies that the function call was not successful. However, you may overwrite this behavior. You can set the SUCCESS indicator to "Y" explicitly to indicate success even though the RETURN_TAB contains error messages. You can set the SUCCESS indicator to *N* explicitly to indicate failure even though RETURN_TAB does not contain error messages.

Related Content

Cookbook for the SAP Application Interface Framework

Copyright