User Guide | PUBLIC
SAP Application Interface Framework 4.0
2026-01-13

Application Help for SAP Application Interface Framework

Content

1 SAP Application Interface Framework

Product Information

Product	SAP Application Interface Framework
Release	4.0
Based On	SAP NetWeaver 7.3 EHP1 or higher
Last Documentation Update	February 2025 (Support Package 06)

Use

SAP Application Interface Framework enables you to develop and monitor interfaces as well as execute error handling in a single framework residing in your SAP back end system.

Possible sources of demand for SAP Application Interface Framework are:

- You have a complex, heterogeneous system landscape using the SAP NetWeaver PI
- You want to decouple technical and business aspects of your interfaces, thus enabling business users to perform error handling
- You experience difficulties in enforcing interface implementation guidelines

To achieve the goal of reducing the time and costs for implementing and maintaining interfaces, SAP Application Interface Framework enables you to:

- Do functional instead of technical monitoring
- Enhance data quality as a result of business users' ability to correct errors within related applications
- Restrict interface data and error monitor access by flexible authorization rules
- Enforce interface implementation guidelines
- Perform mass error handling

Features

SAP Application Interface Framework provides you with the following functions:

- For **business users**:
  - Automatic alerts that notify business users to errors
  - User-friendly transactions for interface monitoring, error handling, and the correction of errors directly from within the application system

- Both SAP GUI and Web-based user interfaces
- Use of a single tool for monitoring and handling of messages regardless of the interface technology involved

For more information about the functions of SAP Application Interface Framework, for example, interface monitoring and error handling, see SAP Application Interface Framework for Business Users [page 27].

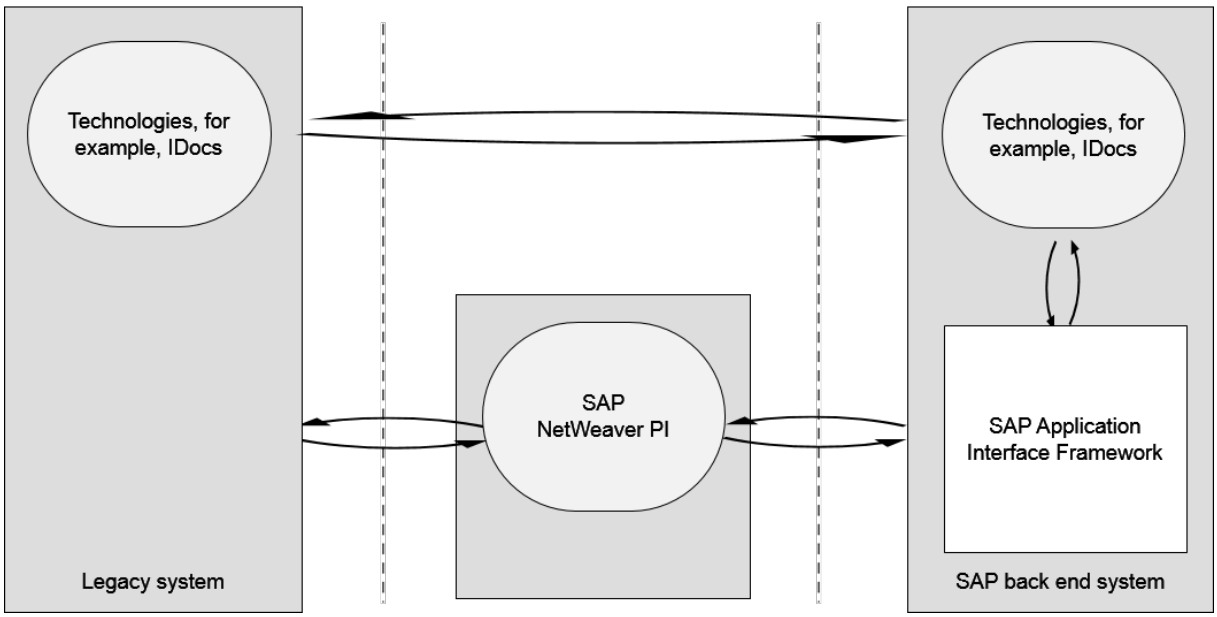
- For **IT personnel**:
  - A powerful framework for the implementation of interfaces
  - A customizable interface framework for the generation and reuse of interface building blocks over multiple interfaces
  - Flexible authorization rules to restrict access to data and to monitoring and error handling
  - Tools for system configuration and operation
  - Monitors interfaces with different interface technologies, for example, ABAP proxies and IDocs

  For more information, see SAP Application Interface Framework for IT Personnel [page 59].

Implementation Considerations

SAP Application Interface Framework is based on SAP NetWeaver technology. You can implement SAP Application Interface Framework as an add-on based on SAP NetWeaver.

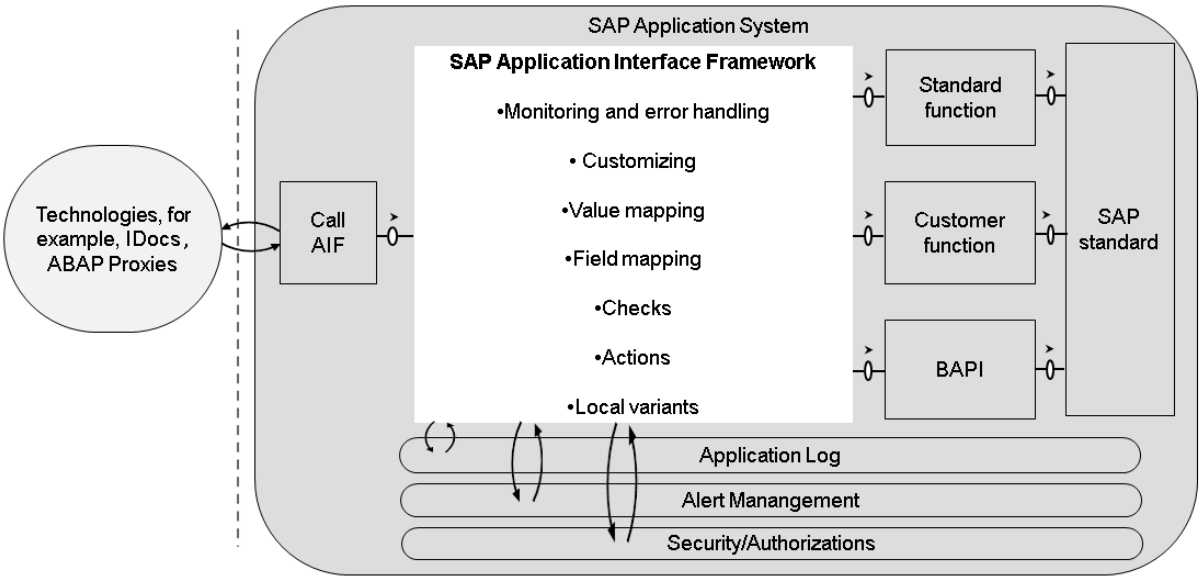
Integration



System Landscape

As is shown in the system landscape graphic above, the integration scenario for SAP Application Interface Framework involves at least a legacy system and an SAP back end system that contains SAP Application Interface Framework.

In your business processes, the legacy system can act as a sender or receiver of data. SAP Application Interface Framework supports business users in monitoring interfaces with different interface technologies, for example, ABAP proxies and IDocs. In one scenario, for example, the legacy system exchanges data first with the NetWeaver PI, which acts as a data broker. In another scenario, the legacy system exchanges data in the form of IDocs with the SAP back end system.



System Architecture

The system architecture graphic above shows SAP Application Interface Framework in detail in the SAP application system. Data is exchanged, for example, with IDocs or between the SAP NetWeaver PI and the Proxy Framework in the SAP application system. SAP Application Interface Framework uses structure mapping and value mapping to map the data from the source structure to the destination structure. Checks are used to verify the validity of the data. Actions handle the execution of your business processes in the SAP application system by calling standard functions, customer functions, or BAPIs. These processing blocks contain the actual business transaction logic and can access the data in the SAP standard.

Errors that occur while processing a message in SAP Application Interface Framework, for example, if a check does not succeed or if a value mapping fails, are written to the application log. Application log messages are loaded into the *Monitoring and Error Handling* transaction and support the authorized business user in solving errors. Additionally, you can configure the system to send alerts to users, if certain errors occur during message processing.

Activities

There are the following ways to access the menu of SAP Application Interface Framework:

- SAP GUI
  - Use transaction code `/AIF/`
    Only the SAP Application Interface Framework menu is displayed.
  - In the SAP Easy Access menu, choose *Cross Application Components* > *SAP Application Interface Framework*.
- Web-based UI
  In the SAP Easy Access menu, choose *Cross Application Components* > *SAP Application Interface Framework* > *Monitoring and Error Handling (Web)*.

There are the following ways to access Customizing for SAP Application Interface Framework:

- Use transaction code `/AIF/CUST`
  Only Customizing for *SAP Application Interface Framework* is displayed.
- In the SAP Easy Access menu, choose *Cross Application Components* > *SAP Application Interface Framework* > *Administration* > *Configuration* > *Customizing*.

More Information

For more information about the technologies supported by SAP Application Interface Framework, see Technology Support [page 125].

For more information about individual Customizing activities, for example, *Define Interfaces* and *Define Actions*, see the corresponding documentation in Customizing for *SAP Application Interface Framework*.

2 What's New

Learn more about new and changed features that are provided with the releases and support packages of SAP Application Interface Framework.

2.1 What's New in Release 4.0

Learn more about new and changed features that are provided with Release 4.0.

2.1.1 What's New in 4.0 SP06

No new features were added in support package 06.

2.1.2 What's New in 4.0 SP05

Learn about the new features available in SAP Application Interface Framework 4.0 SP05.

Log Synchronous Service Calls

SAP Application Interface Framework provides an API with which applications can log synchronous service calls.

See Interface Creation for Logging Synchronous Service Calls [page 176].

Configure Message Deletion

While SAP Application Interface Framework archives messages by default, for some interfaces, archiving data isn't necessary. In these cases, you can delete interface-specific data by configuring message deletion. All messages flagged for deletion can then be deleted regularly by a technical job.

See Message Deletion [page 124].

Content Extraction

You can extract transported content using the transaction *AIF Content Transport - Deploy* (`/AIF/CONTENT_EXTRACT`). See Transporting AIF Content [page 105].

XML Restart with Status Selection

Use the transaction /AIF/XML_RESTART to restart messages from the structured persistence or the XML persistence. This method is an alternative for when you can't restart using the Error Handling and Monitoring of SAP Application Interface Framework. See Persistence Layer and Runtime [page 145].

2.1.3 What's New in 4.0 SP04

Postprocessing Using Customer Exits (BAdIs)

You can implement a custom postprocessing procedure and integrate your implementation with SAP Application Interface Framework using a customer exit, particularly, a BAdI (Business Add-In). This means, you can trigger additional steps (for example, sending an acknowledgment) after the status of a message has changed. For example, you can trigger the sending of an information message in case a message has been canceled.

For more information, see Postprocessing Using Customer Exits (BAdIs) [page 171].

Monitoring Synchronous Outbound Messages

SAP Application Interface Framework's improved integration into Proxy/Web service technology makes it possible to monitor synchronous messages within one interface.

For more information, see Configuring AIF Interfaces for Monitoring Synchronous Outbound Messages [page 85].

Displaying Errors from SAP Cloud Integration

SAP Application Interface Framework now supports an integration with SAP Cloud Integration that enables you to display errors from SAP Cloud Integration in SAP Application Interface Framework.

For more information, see Displaying Errors from SAP Cloud Integration with SAP Application Interface Framework [page 175].

2.1.4 What's New in 4.0 SP03

Integration with SAP Focused Run and SAP Cloud ALM for operations

SAP Application Interface Framework supports integration with SAP Focused Run and with SAP Cloud ALM for operations to help you streamline monitoring and error handling.

More information:

- SAP Focused Run
- SAP Cloud ALM

Integration with SAP Workflow Service

SAP Application Framework is well integrated with SAP Workflow service to enable you to trigger a workflow in case of an error in the message status. You can configure a specific message interface to trigger a workflow instance in SAP Workflow service in case any of the messages related to that interface is in status Error. This will give you an option to configure a specific set of actions in your workflow instance in case of messages in error state, like trigger an email.

Check out this document for an overview: SAP Workflow Service Integration with SAP Application Interface Framework

Check out this document for configuring the add-on to trigger workflow instances: Configuring Application Interface Framework Add-On to Trigger Workflow Instances

2.1.5 What's New: Enhanced Data Protection

The SAP Application Interface Framework supports data protection compliance by providing some security features and specific data protection-relevant functions. With release 4.0, these features and functions were enhanced significantly.

Feature	Type of Change	Description
Master Guide	Enhanced	New chapter *Data Protection and Privacy*. Provides you with an overview of the personal data in SAP Application Interface Framework and with details about how to protect these data.
External Contact	Enhanced	New autonomous object replacing the external address of a recipient. Enables you to independently manage external contacts and to protect their personal data in the best possible way.

Feature	Type of Change	Description
Read Access Log	New	Read access logging for the content of data messages and uploaded files. Enables you to answer questions about who accessed particular data within a specified time frame.
Change Log	Enhanced	Change logging for the content of data messages, value mappings, and external contacts. Enables you to check when a change was made, which employee made which change, the previous value, and the current value.
Data Archiving	Enhanced	New archiving objects and archiving programs for data messages, file adapter, and all related data. Enable you to use SAP Information Lifecycle Management (SAP ILM) for a more sophisticated and automated information lifecycle.
Blocking and Deletion	New	New SAP ILM objects for data messages, value mappings, and external contacts. Enable you to manage the blocking and deletion of personal data according to your legal obligations.

Related Information

Assigning Users to Recipients [page 95]
Read Log and Change Log [page 119]
Data Archiving [page 111]
Data Destruction [page 113]

2.2 What's New in Release 3.0

Learn more about new and changed features that are provided with Release 3.0 und its Support Packages.

2.2.1 Interface Monitor

This section of SAP Library provides an overview of changes and new features that have been introduced in the SAP Application Interface Framework 3.0 since SAP Application Interface Framework 2.0. The following table provides an overview of new, enhanced, and removed functions that are related to the Interface Monitor.

Function	Type of Change	Required Support Package (SP)	Description	More Information
E-mail notification	New		For both a SAP GUI and a Web-based user interface, you can manage e-mail notification for alerts.	Interface Monitor [page 29]
Status alert management	New		For both a SAP GUI and a Web-based user interface, you can manage the status of alerts.	Interface Monitor [page 29]
Details and Processor Assignment	New	SP05	From the *Message Summary*, you can go into more message details, assign processors and track the progress of error solving.	Details and Processor Assignment [page 33]
Last data transfer	New	SP05	You can get summary or detailed information about the last data transfer job for interfaces.	• Interface Monitor [page 29] • Main Screen [page 37]
Calendar highlighting	Changed	SP06	For technical users, also past days are red for which there are messages in process. The current day is yellow, if there are messages in process.	Interface Monitor [page 29]
My messages	New	SP06	From the *Interface Monitor*, you can directly jump into the *Details and Processor Assignment* with only those messages displayed for which you are assigned as processor.	Interface Monitor [page 29] Details and Processor Assignment [page 33]

Function	Type of Change	Required Support Package (SP)	Description	More Information
Details and Processor Assignment - mass re-start and cancel	New	SP07	A new transaction enables you to mass-re-start or cancel data messages which have the processing status *OK to restart* and *OK to cancel*.	Details and Processor Assignment [page 33]

2.2.2 Error Handling

This section of SAP Library provides an overview of changes and new features that have been introduced in SAP Application Interface Framework 3.0 since SAP Application Interface Framework 2.0. or that have been made available with a Support Package for SAP Application Interface Framework 3.0. The following table provides an overview of new, enhanced, and removed functions that are related to error handling.

Function	Type of Change	Required Support Package (SP)	Description	More Information
Debug function	New		The t/qRFC application engines enable the debugging of the LUW.	• Data Messages View [page 40] • tRFC and qRFC [page 153]
Custom functions: offset and length	Enhanced		Parameters for transactions or reports can be filled using the offset and length of the value of a message variable.	• Log Messages View [page 43] • Custom Functions Maintenance [page 51]
Custom message texts	New		A text can be created to replace the original text of an error message.	• Log Messages View [page 43] • Custom Objects Maintenance [page 48]
Structure labels	Enhanced		Structure labels can be defined in Customizing and displayed in the Data Structure view.	Data Structure View [page 44]

Function	Type of Change	Required Support Package (SP)	Description	More Information
Field labels	Enhanced		Field labels can be defined in Customizing and displayed in the Data Content view.	Data Content View [page 46]
Hidden fields	Enhanced		Single fields can be hidden and their content is not displayed in the Data Content view.	Data Content View [page 46]
Custom functions: value mapping	Enhanced	SP01	Parameters for transactions or reports can be filled using the value mapping functionality following an authorization check.	• Log Messages View [page 43] • Custom Functions Maintenance [page 51]
Custom data link	New	SP01	A message can be enhanced with a custom data link.	• Log Messages View [page 43] • Data Content View [page 46] • Custom Objects Maintenance [page 48] • Custom Data Links Maintenance [page 53] • Maintaining Custom Data Links [page 54]
• Structure labels • Field labels	Enhanced	SP03	Instead of maintaining structure and field labels, you can decided to replace all technical names by the related DDIC short descriptions or data element field labels.	• Data Structure View [page 44] • Data Content View [page 46]
Recipients	Enhanced	SP04	Recipients can be assigned to messages by key field values and message categories.	Defining Recipients [page 90]

Function	Type of Change	Required Support Package (SP)	Description	More Information
Test file	New	SP05	You can create a test file from the content of a data message, for usage in the Interface Test Tool.	• Data Messages View [page 40] • Interface Test Tool [page 178]
Last data transfer	New	SP05	You can get summary or detailed information about the last data transfer job for interfaces.	• Main Screen [page 37] • Interface Monitor [page 29]
Hide empty components	New	SP06	In the Data Structure view, you can automatically hide components containing no data values.	Data Structure View [page 44]
Display content in source format	New	SP06	For OData messages, you can display the source content of the data message, that is, the response and the request, in XML format.	Data Messages View [page 40]

2.2.3 Value Mapping

This section of SAP Library provides an overview of changes and new features that have been introduced in SAP Application Interface Framework 3.0 since SAP Application Interface Framework 2.0. or that have been made available with a Support Package for SAP Application Interface Framework 3.0. The following table provides an overview of new, enhanced, and removed functions that are related to value mapping.

Function	Type of Change	Required Support Package (SP)	Description	More Information
Value mapping maintenance	Enhanced	SP01	Three new checks can be specified to check the uniqueness of external values, internal values, and their combination before saving the entries in value mapping.	Value Mapping Maintenance [page 56]

2.2.4 SAP Solution Manager

This section provides an overview of changes and new features that have been introduced in SAP Application Interface Framework 3.0 since SAP Application Interface Framework 2.0. or that have been made available with a Support Package for SAP Application Interface Framework 3.0. The following table provides an overview of new, enhanced, and removed functions that are related to the SAP Solution Manager integration.

Function	Type of Change	Required Support Package (SP)	Description	More Information
Integration with the SAP Solution Manager	New	SP07	In SAP Solution Manager version 7.2 SPS 5 or higher, you can monitor the most important satistics of SAP Application Interface Framework interfaces.	Monitoring in the SAP Solution Manager [page 103]

2.2.5 Role Templates

This section of SAP Library provides an overview of changes and new features that have been introduced in SAP Application Interface Framework 3.0 since SAP Application Interface Framework 2.0. The following table provides an overview of new, enhanced, and removed functions that are related to role templates.

Function	Type of Change	Description	More Information
Role templates	Enhanced	The SAP Application Interface Framework provides predefined role templates that you can use in order to define roles for your specific requirements.	Master Guide for the SAP Application Interface Framework

2.2.6 Business Rules Framework Plus

This section of SAP Library provides an overview of changes and new features that have been introduced in SAP Application Interface Framework 3.0 since SAP Application Interface Framework 2.0. or that have been made available with a Support Package for SAP Application Interface Framework 3.0. The following table provides an overview of new, enhanced, and removed functions that are related to Business Rules Framework plus (BRFplus).

Function	Type of Change	Required Support Package (SP)	Description	More Information
Business Rules Framework plus (BRFplus) integration	New	SP01	The SAP Application Interface Framework provides an integration with the Business Rules Framework plus (BRFplus) tool.	Business Rules Framework Plus Integration [page 82]

2.2.7 Analyzer

This section of SAP Library provides an overview of changes and new features that have been introduced in SAP Application Interface Framework 3.0 since SAP Application Interface Framework 2.0. The following table provides an overview of new, enhanced, and removed functions that are related to the *Analyzer*.

Function	Type of Change	Description	More Information
Analyzer	New	The *Analyzer* assists you with the process of analyzing the steps that are executed at runtime.	• Analyzer [page 85] • Interface Test Tool [page 178]

2.2.8 Automatic Reprocessing

This section of SAP Library provides an overview of changes and new features that have been introduced in SAP Application Interface Framework 3.0 since SAP Application Interface Framework 2.0. The following table provides an overview of new, enhanced, and removed functions that are related to automatic reprocessing.

Function	Type of Change	Description	More Information
Automatic reprocessing	New	The SAP Application Interface Framework can be configured to automatically restart messages where errors occurred during processing.	Automatic Reprocessing [page 96]

2.2.9 Process Observer

This section of SAP Library provides an overview of changes and new features that have been introduced in SAP Application Interface Framework 3.0 since SAP Application Interface Framework 2.0. The following table provides an overview of new, enhanced, and removed functions that are related to the *Process Observer*.

Function	Type of Change	Description	More Information
Process Observer	New	In the SAP Application Interface Framework, you can set up your interfaces to be monitored by the *Process Observer*.	Process Observer [page 98]

2.2.10 SAP HANA

This section of SAP Library provides an overview of changes and new features that have been introduced in SAP Application Interface Framework 3.0 since SAP Application Interface Framework 2.0. The following table provides an overview of new, enhanced, and removed functions for SAP HANA.

Function	Type of Change	Description	More Information
SAP HANA	New	If you are running on an SAP HANA database, there is an improvement in performance, for example, in the selection of data in the *Interface Monitor*. In this case, you do not need to run the Data Correction [page 121] report, as the data displayed is selected directly from the single index tables and there is no statistics table to be corrected.	

2.2.11 Serialization

This section of SAP Library provides an overview of changes and new features that have been introduced in SAP Application Interface Framework 3.0 since SAP Application Interface Framework 2.0. The following table provides an overview of new, enhanced, and removed functions that are related to serialization.

Function	Type of Change	Description	More Information
Serialization	New	Ensures that messages are processed in the correct order.	Serialization [page 105]

2.2.12 Workflow Event Trigger

This section of SAP Library provides an overview of changes and new features that have been introduced in SAP Application Interface Framework 3.0 since SAP Application Interface Framework 2.0. or that have been made available with a Support Package for SAP Application Interface Framework 3.0. The following table provides an overview of new, enhanced, and removed functions that are related to the workflow event trigger.

Function	Type of Change	Required Support Package (SP)	Description	More Information
Workflow event trigger	New	SP04	You can define message processing workflows that are triggered by events of SAP Business Workflow.	Workflow Event Trigger [page 107]

2.2.13 System Operations

This section of SAP Library provides an overview of changes and new features that have been introduced in SAP Application Interface Framework 3.0 since SAP Application Interface Framework 2.0. The following table provides an overview of new, enhanced, and removed functions that are related to system operations.

Function	Type of Change	Required Support Package (SP)	Description	More Information
Data Archiving	Enhanced	SP03	Supports the archiving of additional types of data. These are data changes, runtime objects, application logs, file adapter logs, and snapshots.	Data Archiving [page 111]
Uninstallation	New	SP07	If you do not want to use SAP Application Interface Framework any more, you can remove the software components AIF, AIFX and AIFGEN from your system.	2497991

2.2.14 Technology Support

This section of SAP Library provides an overview of changes and new features that have been introduced in SAP Application Interface Framework 3.0 since SAP Application Interface Framework 2.0. or that have been made available with a Support Package for SAP Application Interface Framework 3.0. The following table provides an overview of new, enhanced, and removed functions that are related to technology support.

Function	Type of Change	Required Support Package (SP)	Description	More Information
File adapter	New		The file adapter supports the upload of inbound files to the SAP Application Interface Framework.	File Adapter [page 150]
Data transfer	New		The *Data Transfer* report enables the transfer of qRFC/tRFC, CIF PP, and BDC data into the SAP Application Interface Framework.	Data Transfer [page 172]
tRFC	New		The SAP Application Interface Framework supports monitoring of tRFC interfaces.	• tRFC and qRFC [page 153] • tRFC [page 155] • Data Transfer [page 172]
qRFC	New		The SAP Application Interface Framework supports monitoring of qRFC interfaces.	• tRFC and qRFC [page 153] • qRFC [page 158] • Data Transfer [page 172]
t/qRFC function module generation	New		The *Function Module Generator* report supports interface developers in implementing the t/qRFC interface for the SAP Application Interface Framework.	tRFC and qRFC Function Module Generation [page 79]
Core Interface postprocessing	New		The SAP Application Interface Framework supports role-based monitoring of Core Interface (CIF) postprocessing records.	• Core Interface Postprocessing [page 161] • Data Transfer [page 172]
Batch input	New		The SAP Application Interface Framework supports monitoring and processing of batch input sessions.	• Batch Input [page 165] • Data Transfer [page 172]

Function	Type of Change	Required Support Package (SP)	Description	More Information
Batch input structure generation and interface definition	New		The *AIF Batch Input Structure Generator* report supports the creation of a structure for a batch input recording.	Batch Input Structure Generation and Interface Definition [page 77]
Preprocessing	New		In the preprocessing mode, the SAP Application Interface Framework executes the mapping logic on top of IDoc or Proxy raw message data.	Preprocessing [page 170]
Mass IDoc structure generation and interface definition	Enhanced	SP01	The *Mass IDoc Structure Generation and Interface Definition* report can be used for the mass generation of objects and Customizing entries based on a list of IDoc types.	IDoc Structure Generation and Interface Definition [page 74]
IDoc processing	Enhanced	SP02	The processing of outbound IDocs via the SAP Application Interface Framework is enabled.	• Process Outbound Docs Using AIF [page 138] • Combine Outbound IDoc to XML Message Using AIF [page 141]
File adapter	Enhanced	SP03	The file adapter supports specifying the file location using logical files. The file adapter provides a transaction to monitor the progress and success of file uploads.	File Adapter [page 150]
File adapter	Enhanced	SP04	The file adapter supports the upload of Microsoft Excel (.xlsx) files.	File Adapter [page 150]

File adapter	Enhanced	SP05	The file adapter supports the download of Microsoft Excel (.xlsx) files.	File Adapter [page 150]
Structured persistence	New	SP05	In addition to the XML persistence of AIF, there is a structured persistence available that stores messages in database tables representing the structure. The structured persistence is optimized for the SAP HANA database.	Persistence Layer and Runtime [page 145]
XML Persistence Messages Deletion	New	SP05	You can irreversibly delete messages from the XML persistence layer of SAP Application Interface Framework.	Persistence Layer and Runtime [page 145]
OData Support	New	SP06	You can monitor error messages for OData services. For erroneous data messages, you can display the error messages and the content of the data message in its source XML format.	OData Support [page 160]
AIF Persistence Messages Deletion	Enhanced	SP07	You can irreversibly delete messages also from the structured persistence layer of SAP Application Interface Framework.	Persistence Layer and Runtime [page 145]

2.2.15 Interface Test Tool

This section of SAP Library provides an overview of changes and new features that have been introduced in SAP Application Interface Framework 3.0 since SAP Application Interface Framework 2.0. or that have been made available with a Support Package for SAP Application Interface Framework 3.0. The following table provides an overview of new, enhanced, and removed functions that are related to the interface test tool.

Function	Type of Change	Required Support Package (SP)	Description	More Information
Transfer to eCATT	Enhanced	SP05	You can transfer test files and create eCATT configuration variants automatically by one click.	Interface Test Tool [page 178] Test Automation with Extended CATT [page 179]

3 SAP Application Interface Framework for Business Users

Use

If you are a business user, you can use the SAP Application Interface Framework for the following scenarios:

- To get an overview of the interfaces you are responsible for and their current status, use the *Interface Monitor* (see Interface Monitor [page 29]).
- To correct errors, restart, or cancel messages, use *Monitoring and Error Handling* (see Error Handling [page 34]).

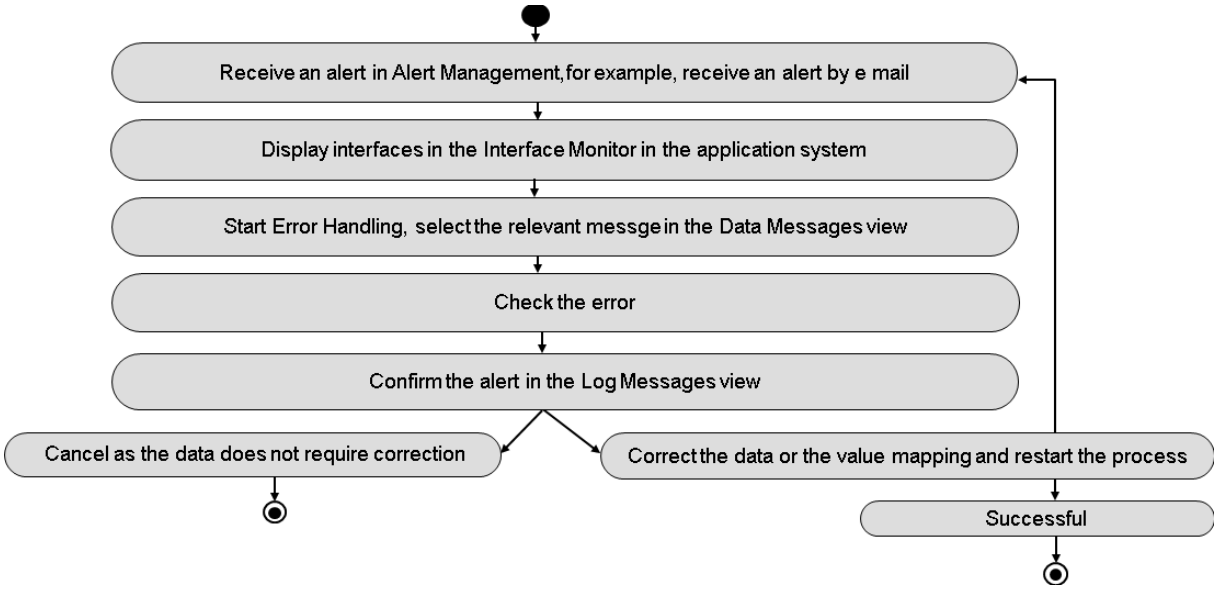
Prerequisites

You have been defined as a recipient for the alerts for the specific interfaces in the area of the business for which you are the business super user. This means that you receive the alerts for error messages relevant to these interfaces.

Process

Example of the Use of the SAP Application Interface Framework by a Business User

The following graphic shows an example of the use of the SAP Application Interface Framework scenarios by a business user handling error messages:



Example of a Business User Handling Error Messages

1. There is a new alert in your alert inbox in Alert Management and, for example, you receive an e-mail that informs you.
2. In your application system, you use one of the following options to start the Interface Monitor to see an overview of the interfaces that are assigned to you:
   - In the SAP Fiori launchpad, choose *Message Dashboard*.
   - In SAP GUI
     - Use transaction code `/AIF/IFMON`.
     - In the SAP Easy Access menu, choose *Cross Application Components* > *SAP Application Interface Framework* > *Interface Monitor*.
   - In the Web-based UI
     - Use transaction code `/AIF/ERR_WEB`.
     - In the SAP Easy Access menu, choose *Cross Application Components* > *SAP Application Interface Framework* > *Monitoring and Error Handling (Web)*.
3. In the *Interface Monitor*, you select the error icon for the interface for which the new alert is relevant and you go directly to *Monitoring and Error Handling* for the message.
4. You check the error and do one of the following:
   - Cancel
     There is no action required for this error and you can cancel it by choosing *Cancel* in the *Data Messages* view.
   - Fix the error and reprocess the message by choosing *Restart* (SAP GUI) or *Reprocess* (Web–based UI) in the *Data Messages* view
     You can fix the error in one of the following two ways:
     - Change the data in the *Data Content* view
     - Correct the value mapping, if the error resulted from a faulty or missing value mapping
       In the *Log Messages* view, you can select the application log message and choose *Maintain Value Mapping*.

Result

In this example of a business user handling error messages, in the *Data Messages* view, the message has either the status  or  .

More Information

For more information about the views in *Monitoring and Error Handling*, see Main Screen [page 37].

3.1 Interface Monitor

Use

In the *Interface Monitor*, the *Message Overview* provides you with a personalized overview of the interfaces you are responsible for.

> ⓘ Note
>
> The interface monitor is available with both a SAP GUI and a Web-based user interface (UI). In the SAP Fiori launchpad, the SAP GUI version is available under*Interface Monitor* and the Web-based UI is available under *Message Dashboard*.
>
> The features that both types of UI have in common are discussed here and the different activities you can have for each UI are also described.

For both types of UI, you can restrict the data range for the information in the message overview by selecting dates in the calendar control. You can also access the *Message Summary*, you can navigate directly to *Monitoring and Error Handling* from the *Interface Monitor*, and you can manage e-mail notification and the status of alerts.

Prerequisites

Interfaces that are to be monitored in the *Interface Monitor* have to be assigned to a recipient.

In order to display messages in the *Interface Monitor*, your user has to be assigned to a corresponding recipient and the *Include in Overview Screen* checkbox has to be selected.

For more information, see Assigning Users to Recipients [page 95].

Features

Message Overview

The following are the possible statuses of your interfaces:

- 
  All messages were processed successfully or canceled manually.
- 
  There are some messages processed with warnings.
- 
  There are some messages processed with errors.
- 
  No message was selected in the selected data range.

In addition, the number of messages for each of the following statuses is shown:

- 
  Processed successfully with warning
- 
- 
- 
- 
  Only visible if you are a technical user
- 
  Only visible if you are a technical user

The total sum of all messages  is also displayed. If an interface's key fields are used for recipient determination, those are also displayed.

> **Note**
>
> The display of interfaces depends on your assignment to recipients. If you do not see the interfaces that you expected to see, refer to your System Administrator for assistance.

Calendar

In calendar control in both UIs, the days when messages are in the system are highlighted depending on the highest error status as follows:

- Red
  There are error messages for this one day. There could also be successful, canceled, and warning messages for this day.
  For technical users, also past days are red for which there are messages in process.
- Yellow
  There are only warning messages, or both warning and successful or canceled messages, and no error messages for this one day.
  For technical users, also the current day is yellow, if there are messages in process.

- Green
  There are only successful or canceled messages for this one day.

> **Note**
>
> The tooltip tells you how many messages are available.

Activities

SAP GUI

You can access the*Interface Monitor* from the SAP Fiori lauchpad or the SAP Easy Access menu by choosing *Cross-Application Components* > *SAP Application Interface Framework* > *Interface Monitor* (transaction code `/AIF/IFMON`).

Restrict Data Range

You can restrict the selection by selecting *With date restriction* and specifying a *Date Range* or marking a date range in the calendar.

Message Summary

To access the *Message Summary* from the *Interface Monitor*, select a namespace or an interface and choose .

Last Data Transfer

To access information about tRFC, qRFC, CIF, and Batch Input data transferred in background jobs by the *Data Transfer* report, choose  .

> **Note**
>
> This pushbutton is also available to you on the *Error Handling* screen.

On the *Information About the Last Data Transfer* popup, you can get the following information about a particular data transfer job:

- Current status as well as the scheduled and actual start date and time
- Selection set name and the names of individual interfaces assigned to it
- A brief or detailed summary from the data transfer job log
  The brief summary is displayed by default. To see the detailed summary, change the layout to include the *Detailed Job Log* column and select  there.

My Messages

To jump directly into the *Details and Processor Assignment* with only those messages displayed for which you are assigned as processor, choose *My Messages*.

Switch to Error Handling

To navigate directly to *Monitoring and Error Handling*, select an interface, a namespace, or icons. If you select individual icons, only the messages for these selected icons are displayed in *Monitoring and Error Handling*. All

parameters are passed automatically. The maximum number of messages that are displayed in *Monitoring and Error Handling* depends on the *Max. Number* you define in the *Interface Monitor*.

If you choose *Display only errors* in the *Interface Monitor* and select a namespace or interface node, only those messages that have a status *Errors* are displayed in *Monitoring and Error Handling*. If you choose *Display all messages*, all messages are displayed independent of their status.

Web-based UI

You can access the Interface Monitor from the SAP Fiori lauchpad under *Message Dashboard*, or from the SAP Easy Access menu by choosing *Cross-Application Components* > *SAP Application Interface Framework* > *Monitoring and Error Handling (Web)* (transaction code `/AIFX/ERR_WEB`).

Restrict Data Range

In the *Calendar Monitor*, you select the *Current Week* or *Current Month* or you mark a date range in the calendar.

Message Summary

Select a namespace or an interface and choose *Message Summary*.

Switch to Error Handling

To navigate directly to *Monitoring and Error Handling*, select an interface or a namespace and double-click on individual icons. Only the messages for the selected icon are displayed in *Monitoring and Error Handling*. All parameters are passed automatically. You can also select an interface or a namespace and select *Interface Key Fields* to enter search criteria for selecting messages, for example, according to the status, the time of creation, or the maximum number of results, for display in *Monitoring and Error Handling*. If an interface-specific selection screen is customized, it is also possible to search using the key fields.

Both SAP GUI and Web-Based UI

User E-Mail Notification

You can activate or deactivate your own e-mail notification by choosing from the following:

- 

You do not receive e-mail notifications for the interface-recipient combination.

- 

You receive only one e-mail notification for the interface-recipient combination. You can receive another notification e-mail only after you or another user has confirmed the alert.

- 

You receive an e-mail notification for every single error of the interface-recipient combination, regardless of whether or not the alert has been confirmed.

Alert Status

You can manage the status of alerts using the following status icons:

- 

Indicates that an alert does not exist

- 

Indicates that an alert exists. You can confirm the alert by clicking this status icon. This confirmation results in the display of the  icon.

> **Note**
>
> Even if you have selected the *No e-mail* icon, the *Click to Confirm* icon can be displayed if another user receives e-mails for this interface/recipient combination and an alert exists as a consequence.

More Information

For more information, see Message Summary [page 33] and Error Handling [page 34].

3.1.1 Message Summary

Use

The *Interface Monitor* has a *Message Summary*. This summary provides you with an overview of how often a certain log message was written to the application log and how many data messages used a certain log message.

Features

In the *Message Summary*, the messages are grouped by namespace, interface name, and version. The table displays the message type, the message ID, the message number, and the message text. You can sort the messages and search or filter for specific messages.

The *Message Summary* provides you with mass reprocess and cancel functions. By selecting at least one line in the table and choosing  or  in the SAP GUI or  or  in the Web-based UI, all relevant messages can be restarted or canceled. You need the authorization to restart and to cancel messages.

You can navigate directly to *Monitoring and Error Handling* by double-clicking on a line. The number of messages displayed in *Monitoring and Error Handling* depends on the *Max. Number* you defined in the *Message Summary*.

3.1.2 Details and Processor Assignment

From the *Message Summary*, you can go into more details, assign processors and track the progress of error solving.

In addition to the known features of the message summary, like mass restart and mass cancel, the *Details and Processor Assignment* provides you with detailed log messages, grouped by concrete variable values. This enables you to identify incorrect or missing values, for example, missing document types at accounting document creation. You can also display the long texts of the messages, if available.

At this level of detail, you can assign processors to the log messages, who are responsible for solving the errors that caused the messages. To keep track of the error solving progress, the processors can set processing statuses and add comments.

Mass Restart and Cancel of Processed Messages

The transaction `/AIF/DPA_PROCESS` enables you to mass-restart or cancel data messages which have the processing status *OK to restart* and *OK to cancel*.

The additional benefits of the transaction compared to the *Details and Processor Assignment* UI are:

- You can restart and cancel across processors.
- You have a test mode to preview the numbers of messages to be restarted or cancelled.
- You can schedule the restart and cancel as a periodic task.

For more information, see the system documentation of the transaction.

3.2 Error Handling

Use

In the SAP Application Interface Framework, you can get a structured overview of the messages that are generated during data processing. You can edit the underlying data content of the data messages and get support to solve errors if the context of the error is known (which field was checked, which value could not be mapped, and so on).

Prerequisites

SAP GUI

You access *Monitoring and Error Handling* (transaction code `/AIF/ERR`) for an interface you are responsible for that has errors and use the selection screen to filter your messages. You choose  to navigate to the *Monitoring and Error Handling* main screen.

In the SAP GUI, the transaction offers a selection screen with various selection options (namespace, interface, creation date and time, status, and so on). You can access the selection screen from the SAP Easy Access menu by choosing *Cross-Application Components* > *SAP Application Interface Framework* > *Monitoring and Error Handling*.

Alternatively, you accessed the *Interface Monitor* (from the SA Fiori launchpad or using transaction code `/AIF/IFMON`) and saw that there were messages in the interfaces you are responsible for (see Interface Monitor [page 29]). If you start error handling from the *Interface Monitor*, the selection screen is not displayed. Instead, you select the messages by double-clicking on a node or icon in the *Interface Monitor* and navigating directly to the *Error Handling* main screen.

Web-based UI

You first access the *Interface Monitor* (transaction code `/AIFX/ERR_WEB`) from the SAP Easy Access menu by choosing *Cross-Application Components* > *SAP Application Interface Framework* > *Monitoring and Error Handling (Web)*. You then navigate to *Monitoring and Error Handling* by selecting messages by double-clicking on a node or an icon.

Process

The following steps describe a typical error handling process:

1. You obtain an overview about the existing messages and you can do the following:
   - If you need to reduce the amount of information displayed on the screen, you can use different layouts, filters, and selection options to show only the data you are interested in.
   - In the SAP GUI, if you need more detailed information and you have the authorization, you can switch to technical mode and, for example, go to the application log. For more information about these options, see Main Screen [page 37].
2. You analyze the errors:
   - If an error message seems to have a temporary reason, you can restart the processing and see if the error still exists.
   - If an error does not require you to take action, you can cancel the message.
     For more information, see Data Messages View [page 40].
3. You solve the underlying errors of the remaining messages. The system supports you with the following functions:
   - If there is a missing value, you can navigate directly to *Value Mapping* from the Log Messages view.
     For more information, see Log Messages View [page 43].
   - You can change the underlying data content in the Data Content view.
     For more information, see Data Content View [page 46].
   - For other reasons, the system can provide message-specific buttons to resolve the errors.
4. In the SAP GUI, you confirm the alert in the Log Messages view.

   > **Note**
   >
   > Confirming the alert is important. If you receive an alert e-mail for an interface for the first message that ended with an error and you do not confirm the alert, you will not receive any further e-mails about errors in this interface.

5. In the Data Messages view, you restart processing of the messages you have worked on as follows:
   - In the SAP GUI, choose . To see if the errors persist, choose .
   - In the Web-based UI, choose . To see if the errors persist, choose .

3.2.1 Selection Screen

Use

In the SAP Application Interface Framework, the *Monitor and Error Handing* selection screen (transaction code `/AIF/ERR`) enables you to specify the main parameters that control the range of messages and how they are displayed on the *Monitoring and Error Handling* main screen.

Features

The following are the various selection options available to you:

- *Application Selection*
  You can input an application ID. The default value is **`AIF`**.
- *Application specific selection*
  You can select the namespace, interface name, and interface version. Leaving one of the selection fields empty results in the selection of all relevant objects. The same values can also be selected by choosing *Select Interface*. A popup containing a tree view with all interfaces grouped by namespace is displayed. The selection can be restricted by entering a message class and a message number. Depending on the application ID, the application-specific selection might be different from this description.
- *More specific selection*
  This subscreen is optional. It is only visible if key fields have been assigned for an interface and a corresponding subscreen has been created by a developer. You can search for specific data messages. What you can search for depends on the key fields defined for the interface.
- *Generic selection*
  Only messages that occurred within the selected date/time range are displayed. It is possible to restrict the selection by entering a message. The date refers to the point in time when the message was processed in the SAP Application Interface Framework for the first time. It does not refer to the point in time when the message was possibly restarted.
- *Status Selection*
  Only messages are displayed that have the selected status. If no indicator is set, then all statuses are selected.
- *Additional Parameters*
  If you set the *Technical Mode* indicator, an additional fifth view is displayed in the error handling. This view displays the interface processing. If you set the *Emergency Correction* indicator, all fields can be edited, independently of what was defined in the Customizing activities for the *SAP Application Interface Framework*. This, however, depends on your authorization level (if you do not have the required authorization, then the indicators in *Additional Parameters* are not available). In *Max.Number*, you can specify the maximum number of messages that should be displayed. If this number is exceeded, the system informs you that more messages were found and that only the specified number will be displayed.

> **Note**
>
> If a transaction variant was defined and assigned to your user, the selection screen or some selection options might be skipped or there might be default entries available on the selection screen.

Activities

You can access the selection screen from the SAP Easy Access menu by choosing *Cross-Application Components* > *SAP Application Interface Framework* > *Monitoring and Error Handling*.

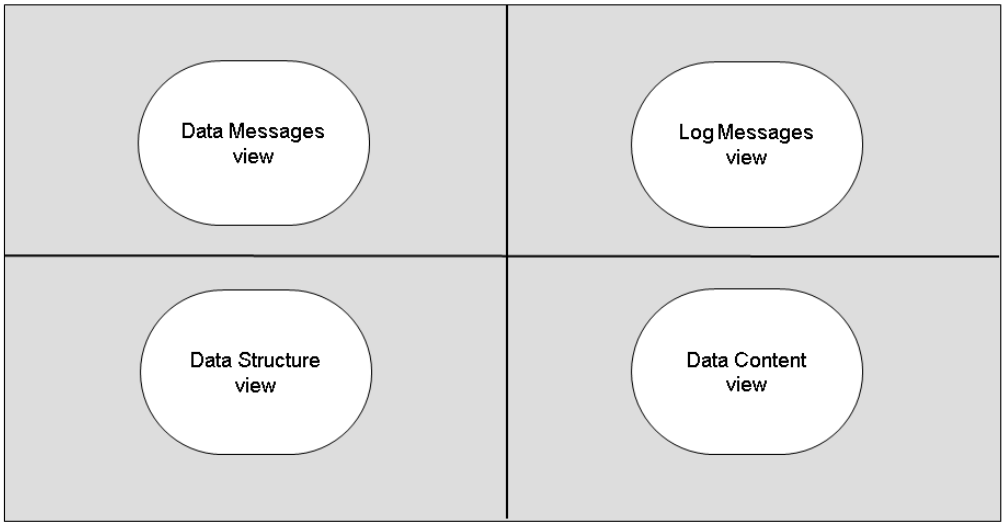
Use the selection screen to filter your messages and choose to navigate to the *Monitoring and Error Handling* main screen.

3.2.2 Main Screen

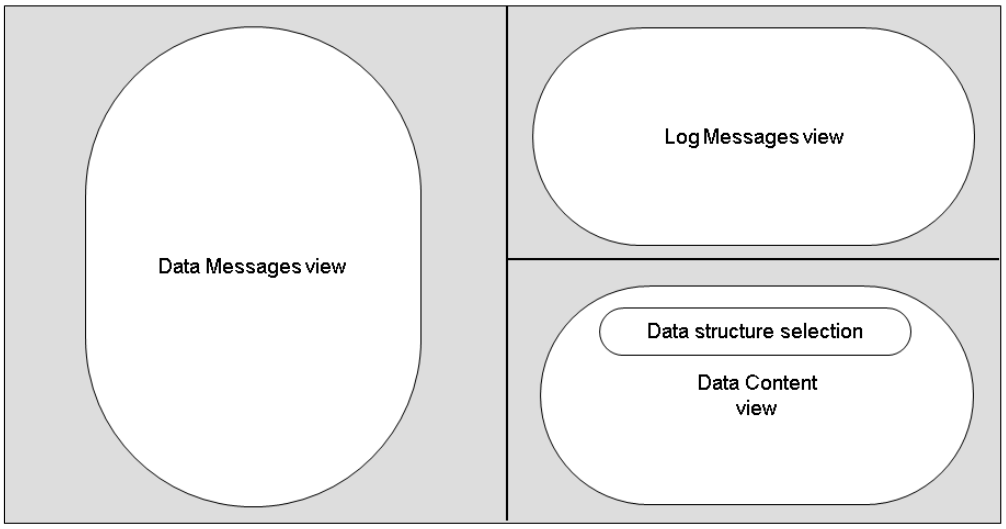
Use

The main error handling screen provides an overview of all selected messages, a detailed analysis of one or more selected messages, as well as functionality, such as, restarting and canceling messages.

Features



SAP GUI: Main Screen Views in Standard Mode



Web-based UI: Main Screen Views

As shown in the graphics above, the following views are displayed (in standard mode for the SAP GUI):

- Data Messages view [page 40]
  Displays all selected data messages in a hierarchy. The messages are grouped by namespace and interface. Optionally, the messages can be grouped by key fields, if these have been defined for an interface.
- Log Messages view [page 43]
  Displays the log messages that have been created while processing the selected message(s).
- Data Structure view [page 44]
  Displays the structure (record type) of the message data.
- Data Content view [page 46]
  Displays the data of the selected message(s).

SAP GUI

If you choose *Technical Mode*, an additional fifth view is displayed on the *Monitoring and Error Handling* main screen. This view displays interface processing and some additional buttons are added in the four standard views (for example, for sorting, printing, and transforming). The mapping steps and actions executed during interface processing are displayed. The view enables you to analyze in which processing steps errors occurred. You can choose a processing step to display the log messages related to it. This enables you to identify, for example, those messages that derived from the value mapping process.

In the main toolbar, the following functions are available:

Pushbutton	Function
 	Switches to technical mode and back to standard mode. In technical mode, an additional view is provided that displays the structure of the interface processing enabling you to filter the log messages. Also, some additional technical buttons are supplied in the other four views. This button is only displayed if you have the required authorization.
	Displays the legend that describes all possible message statuses
	Resets the content of every view. Reloads all data messages with the current selection parameters.
	Dialog box to change the date range. After you have confirmed the changed date, the system reloads all data messages with the current selection parameters within the new date range. In addition, you can change the maximum number of messages displayed.
	Navigates to the standard view for the log messages displayed (for example, transaction `/AIF/LOG`). The log messages that exist for the selected data message are displayed.
	Navigates to the alert inbox (Web browser).
	Navigates to information about the last data transfer jobs (see Interface Monitor [page 29]).

Web-Based UI

In the main toolbar, the following functions are available:

Pushbutton	Function
	Navigates to the Interface Monitor
	Saves
	Cancels
	Resets the content of every view

3.2.2.1 Data Messages View

Use

In the Data Messages view, the messages are grouped by namespace, interface name, and version, and also by key fields if these are customized.

Features

The Data Messages View Toolbar: SAP GUI

The following functions are provided:

- To expand or collapse the selected nodes, choose  or  .
- Read messages

  To read messages, choose  . Reading a message means reading the corresponding application log, loading it into the Log Messages view, and displaying the data structure in the Data Structure view. If multiple messages are selected, displaying the data structure works only if the structure is the same in all selected messages. You have the following options:
  - Double-click a message for single selection
  - Double-click a node (root node, namespace, interface, or key field) for the selection of all messages in the whole sub-tree
  - Select any number of messages/nodes and choose *Read*
- Restart

  To restart the selected message, choose  . If you have analyzed the error and corrected it, for example, by changing the value mapping, you can restart the message by choosing *Restart*. You can select single or multiple data messages or a whole node for restarting. After some seconds, choose *Read* on the main tool bar. All the views are updated and the status icon of the message indicates the new message status.
- Cancel

  To cancel the selected message, choose  . The following are 3 different cases for canceling a message:
  - A new message was sent that was successful
  - The processing steps of the message were executed manually
  - The message has become irrelevant

  You are also able to cancel single or multiple messages. When you cancel a message, the status icon is updated.

  > ⓘ Note
  >
  > If you have canceled a message, you cannot restart the processing for this message anymore.
- Debug
  - The *Debug* pushbutton works dynamically as follows:
    - tRFC
      Only outbound queues are supported

- qRFC
  Both inbound and outbound queues are supported

- Navigate to the standard monitor
  This is the monitor that is usually used to monitor messages of the interface technology that is used to process the messages of the selected SAP Application Interface Framework interface.

  Select a message and choose  . There you can find more detailed information about the message.
- Navigate to qRFC monitor

  Select a message and choose  .
- Maintain Trace Level
  To maintain the trace level for the selected message(s) or for a specific interface, select a message and choose  . If you need more information about why the processing of a message failed, you can increase the trace level for the interface and restart the message. It is important that you specify an expiry date for the trace level otherwise the message received by this interface will create more log messages than necessary.
- Display Content in Source Format

  > ⓘ Note
  >
  > Available only for OData messages and only if activated in Customizing.

  To open the content of a data message in its source XML format, select the message, choose  , and choose an option from the drop-down menu. The content will open in the program you have associated with the file type, for example, an XML editor or a Web browser.
- Create Test File

  > ⓘ Note
  >
  > Available in technical mode only

  To create a test file from the content of a data message, for usage in the *Interface Test Tool* (transaction code /AIF/IFTEST), select a message and choose the *Create Test File* button.
- Change Layout

  To change, save, and manage the layout of the hierarchical tree in the data message view, choose  . You select and manage a predefined layout or you can define a new layout that fits your needs.

The Data Messages View Toolbar: Web-Based UI

- To expand or collapse the selected nodes, choose  or  .
- Reprocess

  To restart a selected message, choose  . If you have analyzed the error and corrected it, for example, by changing the value mapping, you can restart the message by choosing  . You can select single or multiple data messages or a whole node for restarting. After some seconds, choose  on the main tool bar. The icon in the Data Message view should be updated.
- Cancel

  To cancel the selected message, choose  . The following are 3 different cases for canceling a message:
  - A new message was sent that was successful
  - The processing steps of the message were executed manually

- The message has become irrelevant

You are also able to cancel single or multiple messages. When you cancel a message and choose , the status icon is updated.

> **Note**
>
> If you have canceled a message, you cannot restart the processing for this message anymore.

- Debug
  - The *Debug* pushbutton works dynamically as follows:
    - tRFC
      Only outbound queues are supported
    - qRFC
      Both inbound and outbound queues are supported
- Refresh Selected
  Once you have reprocessed a message or selection of messages, to see whether or not errors persist, choose . If the message is reprocessed successfully, the status icon is update to indicate success.
- Trace Level
  To maintain the trace level for the selected message(s) or for a specific interface, select a message and choose *Trace Level*. If you need more information about why the processing of a message failed, you can increase the trace level for the interface and restart the message. It is important that you specify an expiry date for the trace level otherwise the message received by this interface will create more log messages than necessary.

Display of Messages in the Data Messages View: Both SAP GUI and the Web-Based UI

- The counter in the node text shows the number of messages in the corresponding sub-tree.
- The status of a message is indicated by an icon.
  The following are examples of icons:
  - 
    Message is in the queue but is not yet processed; See the qRFC Monitor.
  - 
    Restart to reprocess the changed data message is necessary.
  - 
    Errors are described in the log messages.
  - 
    Message was manually restarted, it is in the queue, but is not yet processed (see qRFC Monitor).
  - 
  - 
  - 
- The columns to the right of the tree depend on the layout variant. The following columns can be displayed:
  - Error statistics display the number of log messages that exist (also added up on each tree level). The error statistics are grouped into the number of abort, error, warning, success, and information log messages.
  - Message GUID and interface keys
- The icons used for the hierarchy nodes depend on the current interface-specific Customizing.

> ### Example
>
> 
>
> 
>
> 

- The node text of a message contains a temporary message index and the date and time. The message index is used to visualize the relation between the message in the Data Messages view and the log messages in the Log Messages view.

3.2.2.2 Log Messages View

Use

The Log Messages view shows all related log messages that were saved during the processing of a message in the SAP Application Interface Framework. These can be technical errors, application errors, warnings, success, and information messages.

Features

The Log Messages View Toolbar: SAP GUI

The following functions are provided:

- To display or hide the messages of a certain message type (*Success, Info, Warning, Technical Error*), choose the corresponding filter button.

  > ### Example
  >
  > To show only the error messages, choose  .

- If an error resulted from a faulty value mapping, you can select the application log message and choose *Value Mapping*.
- If you have filtered the display of log messages in the Data Content view with  , you can switch back to displaying all log messages by choosing  in the Log Messages view.
- If you want to create, edit, or delete a custom hint, function, message text, or data link, choose *Customize* and select the relevant option (see Custom Objects Maintenance [page 48]).
- In addition to the custom functions, buttons can be added dynamically to the toolbar by interface developers. These buttons can, for example, provide links to transactions where you can correct the reason for an error.
- If the process started with an alert e-mail in your inbox, the system provides you with the *Confirm Alert* button, which you use to confirm the alert when you have corrected the error.

The Log Messages View Toolbar: Web-Based UI

The following functions are provided:

- View
  You can filter your log messages by status by selecting one of the following:
  - All messages
  - All Errors
  - Application Error
  - Success
  - Technical Error
  - Warning
- Customize
  If you want to create, edit, or delete a custom hint, function, or message text, choose *Customize* and select the relevant option (see Custom Objects Maintenance [page 48]).
- Reset Object Filter
- If the log message is caused by a value mapping error,  is displayed in the *ValMap* (Value Mapping) column. To navigate to *Value Mapping*, choose this icon.

Message Display in the Log Messages View

The message status, for example,  , is shown in the *Type* (SAP GUI) or *Status* (Web-based UI) column.

If you have created custom hints, functions, or message texts, an icon is displayed in the corresponding *Hints*, *Functions*, or *Texts* column.

The message text or the custom message text you have created is displayed in the *Message Text* column.

If a long text exists for a message, the system displays the *Long text exists* icon in the *LTxt* (SAP GUI) or the *Long Text* (Web-based UI) column. If you click on the icon, the system displays the long text in a dialog box.

The *Index* column shows the same number as the corresponding data message in the data messages screen. This information helps you to identify which log message belongs to which data message, especially if multiple messages are displayed.

To navigate to the corresponding data line (or data field) in the Data Content view, double-click the log message.

If the message was restarted, you can only see the application log messages created during the restart. If you want to see all log messages, you have to access the application log of the message (see Main Screen [page 37]).

3.2.2.3 Data Structure View

Use

The Data Structure view shows the structure of the datasets of the selected data messages. A dataset can consist of several nested tables and structures.

Features

The data structure displays the whole tree of substructures and the subtables of the raw data structure of the selected data message(s). In the Web-based UI, select *Data Structure* to have the data structure opened in a dialog box. If you double-click on a sub-node of the record type in the data structure, the data contained in the structure is displayed in the Data Content view.

Some structures may be hidden based on the settings in Customizing for the *SAP Application Interface Framework* under *Error Handling* > *Define Namespace-Specific Features* > *Hide Structures* and under *Error Handling* > *Define Interface-Specific Features* > *Hide Structures*. If a structure is hidden, its content cannot be displayed in the Data Content view.

You can also decide to automatically hide components of structures (substructures and tables) containing no data values, in Customizing under *Error Handling* > *Define Namespace-Specific Features* > *Define Interface-Specific Features* > *Hide empty components*.

You can define structure labels in Customizing under *Error Handling* > *Define Namespace-Specific Features* > *Define Structure Labels* and under *Error Handling* > *Define Interface-Specific Features* > *Define Structure Labels*. In addition, you can replace the label of a raw structure.

Which Customizing table is evaluated depends on which of the following is selected:

- Interface Mode
  If all selected messages belong to the same interface (the namespace, interface name, and version are equal), the settings in Customizing for the *SAP Application Interface Framework* under *Error Handling* > *Define Interface-Specific Features* are relevant.
- Namespace Mode
  If the selected messages belong to more than one interface, but the interfaces belong to the same namespace, the settings in Customizing for the *SAP Application Interface Framework* under *Error Handling* > *Define Namespace-Specific Features* are relevant. This case only applies if all interfaces of this namespace share the same raw data structure, otherwise the data cannot be displayed.
- Cross-namespace Mode
  If the selected messages belong to more than one interface and these interfaces do not belong to the same namespace, all structures are displayed independent of the Customizing settings. This case only applies if all interfaces of the selected messages share the same raw data structure, otherwise the data cannot be displayed.

Instead of maintaining structure and field labels, you can decided to replace all technical names by the related DDIC short descriptions or data element field labels in Customizing for the *SAP Application Interface Framework* under *Error Handling* > *Define Namespace-Specific Features* > *Define Interface-Specific Features*.

SAP GUI

If you are in *Technical Mode*, a button is visible in the Data Structure view. If you choose , the data is transformed from the source structure to the destination structure. The Data Structure view now contains the destination structure. Double-clicking a node of the destination structure displays the transformed data in the Data Content view. Any log message that occurred during the transformation (for example, errors resulting from value mapping or checks) appears in the Log Message view. If you choose , you can switch back to the previous display of the source structure.

3.2.2.4 Data Content View

Use

In the Data Content view, the system displays the data content of the selected data message. You can edit the data content according to your authorization level. Whether you can change certain field values depends also on the settings in Customizing for the *SAP Application Interface Framework* under *Error Handling* > *Define Namespace-Specific Features* > *Define Changeable Fields* and under *Error Handling* > *Define Interface-Specific Features* > *Define Changeable Fields*.

Features

Data Display

The Data Content view displays the data of the selected record type from the data structure in a table. The values are displayed in the external format (for example, for date or numeric fields).

Some fields may be hidden based on the settings in Customizing for the *SAP Application Interface Framework* under *Error Handling* > *Define Namespace-Specific Features* > *Hide Fields* and under *Error Handling* > *Define Interface-Specific Features* > *Hide Fields*. If a field is hidden, its content cannot be displayed in the Data Content view.

You can define field labels in Customizing for the *SAP Application Interface Framework* under *Error Handling* > *Define Namespace-Specific Features* > *Define Structure Labels* and under *Error Handling* > *Define Interface-Specific Features* > *Define Structure Labels*.

Instead of maintaining structure and field labels, you can decided to replace all technical names by the related DDIC short descriptions or data element field labels in customizing for the *SAP Application Interface Framework* under *Error Handling* > *Define Namespace-Specific Features* > *Define Interface-Specific Features*. The data element field labels are displayed as column headings.

In the Data Content view, there are the following fixed columns that can never be edited:

- *Exception* (SAP GUI) or *EXP_LED* (Web-based UI)
  Indicates whether or not the data line contains an error
- *LINE_NR*
  Contains the message index with line numbers. If there are nested tables in the data structure, several line numbers can be concatenated.

  > **Example**
  >
  > For example, *1: 2.5* can have the meaning *index of data message: second financial posting, fifth item*.

SAP GUI

To switch to full screen mode, choose  . In full screen mode, it is also possible to display the error messages in an additional subscreen. Therefore, choose  . The application log messages for the selected data message are displayed. Double-clicking on the log message displays the field to which the error belongs.

To save the current layout or select a different one, choose  .

Data Editing

If a message has an error status and you have the required authorization, you can edit the content of those fields that have been defined as editable in Customizing.

To support you in correcting errors, the fields have the following colors:

- Red
  An error occurred that is related to the content of this field.

  > **Note**
  >
  > A field is highlighted in red only if it was possible for the SAP Application Interface Framework to track an error to that specific field.

- Yellow
  Changed but not yet saved
- Blue
  Not editable; System assigns value automatically

Fields are defined as editable in Customizing for *SAP Application Interface Framework* as follows:

- Under *Error Handling* > *Define Namespace-Specific Feature* > *Define Changeable Fields*
  Fields maintained here are editable in every interface that contains them and that belongs to the specified namespace.
- Under *Error Handling* > *Define Interface-Specific Feature* > *Define Changeable Fields*
  Fields maintained here are editable in the specified interface.

SAP GUI and Custom Data Links

In the SAP GUI, you can enhance a message with a custom data link. If a custom data link exists for a table/structure and field combination, the relevant column/field is highlighted in red in the Data Content view.

For more information about custom data links, see Custom Data Links Maintenance [page 53] and Maintaining Custom Data Links [page 54].

Activities

Data Editing

To edit fields, do the following:

- SAP GUI
  Double-click the field and edit the text in the dialog box that appears. To apply your changes to all rows, mark the changed row and choose *Apply*.
- Web-based UI

  The data content of all selected messages is displayed for the selected structure. If you choose  , all this data content is editable (if customized).

> **Note**
>
> For a field defined as date, time, or numeric type, you should maintain the value in the dialog text following the relevant format defined in your personal profile (use transaction SU3 to view your personal profile).

When a column or row is selected where fields are editable, you can replace values by choosing  in SAP GUI or  in the Web-based UI. In the dialog box that appears, you can enter a search string (whole cell content or string with wildcard "*") and a new value. If you confirm your entry in the dialog box, the system replaces all corresponding values in the selected column with the new value.

If you have changed data, the cells are highlighted yellow. The changes will only be saved if you choose *Save* in the Data Content view of the SAP GUI or in the main toolbar of the Web-based UI. If you select another record type or choose to leave the transaction without saving, a warning message appears.

SAP GUI

To show only the error messages, choose  . To switch back to the display of all messages, choose  .

To display the log messages belonging to a data content line in the log messages screen, mark the data content line or the concerned field in the Data Content view and choose  .

3.2.3 Custom Objects Maintenance

Use

In *Monitoring and Error Handling*, in the Log Messages view, you are able to create, edit, and delete the following:

- Custom hints
  You can, for example, write a procedure on how to correct a common error.
- Custom functions
  You can, for example, call a transaction for a selected message.
- Custom message texts
  You can write a text to replace the original text of an error message.
- Custom data links
  You can enhance a message with your own custom data link.

Integration

Business users maintain custom hints, functions, message texts, and data links in *Monitoring and Error Handling*.

System administrators can change existing custom hints, functions, message texts, and data links in the following transactions, which they can access from the SAP Easy Access menu by choosing *Cross-Application Components* > *SAP Application Interface Framework* > *Administration* > *Configuration*:

- *Define Custom Functions* (transaction code `/AIF/CUST_FUNC`)
- *Define Custom Hints* (transaction code `/AIF/CUST_HINTS`)
- *Define Custom Message Texts* (transaction code `/AIF/CUST_TEXT`)
- *Define Custom Data Links* (transaction code `/AIF/CUST_LINK`)

The activities for business users and system administrators are described separately in the relevant sections below.

Features

In *Monitoring and Error Handling*, in the Log Messages view, the following columns are displayed:

- *Hints*
  Custom hints you create are displayed with an icon.
  If you choose this icon, the hint is displayed. If you choose an empty cell in this column, the screen to create a new hint appears.
- *Functions*
  Custom functions you create are displayed with an icon.
  When you select an existing custom function, the action is executed. If you have entered values for parameters for the action when you created the custom function, the values are passed to the transaction or the report. You can also choose to use the values of the message variables in the transaction or report (see Custom Functions Maintenance [page 51]).
- *Texts*
  Custom message texts you create are displayed with an icon. If you choose this icon, the available message texts are displayed on an overview screen. If you choose an empty cell in this column, the screen to create a new message text appears.
- *Message Text*
  Message texts or custom message texts you create are displayed here.

In addition to the links between log messages and data content created during the processing of data in the SAP Application Interface Framework, you can create your own custom data link. In the Data Content view, if you have custom data links, the relevant cells or columns are highlighted in red depending on whether the custom data links are for structures or tables (see Custom Data Links Maintenance [page 53] and Maintaining Custom Data Links [page 54]).

Activities

Business Users

If you select a message and choose to create a custom hint, function, or message text, the information of the selected message is displayed in the popup.

To create a custom data link, you must select both a message in the Log Messages view and a cell in the Data Content view. If no custom data link exists, a popup is displayed and you can create a new one.

- SAP GUI
  Select a message (and a cell in the Data Content view if you want to create a custom data link), choose *Customize*, and then choose one of the following:
  - *Custom Hints*
    You can enter a text and a tooltip text for your hint in the language of your choice.
  - *Custom Functions*
    You can define functions, for example, transactions, that are specific to your user and an administrator can make them visible for certain other users or all users. Custom functions that are visible for other users are only displayed. They cannot be deleted or changed.
    For more information, see Custom Functions Maintenance [page 51].
  - *Custom Message Text*
    You can create a text, specific to your user, that replaces the original text of a message in the Log Messages view. You can write a text in the language of your choice and you can subsequently edit it in that language. Once you have created a text in a specific language, you are unable to change the language of the text.
    On the Log Messages view, if a custom text exists in the logon language, this text is displayed. The original text is displayed if there is no custom text available as shown in the following sequence:
    1. Written in the logon language
    2. Written in the second language (set in the SAP system)
    3. Written in English (the fall back language for the SAP Application Interface Framework)

    When you choose the icon for a text in the *Texts* column, the selected text is displayed in all available languages on the overview screen.
    An administrator can make your custom message text visible for other users or for all users. You can see your user-specific texts and those that are visible to all users or to a list of users that includes you. Only an administrator can change the visibility of a text. The business user can only edit or delete user-specific ones. Those that are visible for all users or a list of users are only displayed.
  - *Custom Data Links*
    Select a message in the Log Messages view and a cell in the Data Content view. Choose *Customize* and then choose *Custom Data Link*. If there is no existing custom data link, a popup appears to inform you and you can create one for the structure/table and field combination you selected.
    For more information, see Custom Data Links Maintenance [page 53] and Maintaining Custom Data Links [page 54].
- Web-based UI
  Select a message, choose *Customize*, and select one of the 3 options for the definition of custom hints, functions, or message texts.

> **Note**
>
> If you have created a custom hint, function, message text (SAP GUI and Web UI), or data link (SAP GUI only) for the selected message, you can choose to edit or delete it. You can only edit or delete user-specific ones. Those that are visible for all users or a list of users are only displayed and you cannot edit them.

System Administrators

You can determine for whom the custom hints, functions, message texts, and data links are displayed, for example, you can determine that a custom hint is only visible for a list of users. You can also determine whether they are displayed for one or for all interfaces and for a selection of messages or all messages.

In the SAP Easy Access menu, in *Define Custom Functions*, *Define Custom Hints*, *Define Custom Message Texts*, and *Define Custom Data Links*, there are *Visibility* and *Scope* fields, which you can use as follows:

- *Visibility*
  You can determine for whom the created custom hint, function, message text, or data link is displayed as follows:
  - Just for current user
  - For a list of users
    You can define users and recipients for visibility under the following:
    - *Define Custom Functions*
      In the activities *Assign Users* and *Assign Recipients*
    - *Define Custom Message Texts*
      In the activity *Assign Users for Custom Message Texts*
    - *Define Custom Data Link*
      In the activity *Assign Users for Custom Data Link*
  - For all

  > **Note**
  >
  > If you change the assignment of a message from user-specific to all users, the new assignment replaces the previous assignment to a specific user.

- *Scope*
  For the created custom hint, function, message text, or data link, you can determine whether it is displayed for one or for all interfaces and for one or for all messages in the following combinations:
  - Visible for selected message in this interface
  - Visible for all messages in this interface
  - Visible for selected message in all interfaces
  - Visible for all messages in all interfaces

3.2.3.1 Custom Functions Maintenance

Use

In *Monitoring and Error Handling*, in the Log Messages view, you are able to create, edit, and delete custom functions (see Log Messages View [page 43]).

Features

Create Custom Functions

To create a custom function, you can select from the following actions:

- SAP GUI
  - Transaction

    - Reports
- Web-based UI
    - Transactions

You can test the custom function you choose.

> ### Example
>
> You choose to create a custom function that calls the transaction to display flight bookings that has the name `BC_GLOBAL_SCUST_DISP`. Choose *Test Function*, which in this case navigates you to the transaction to display flight customers.

To specify the function attributes, you can enter a short text and a tooltip for the function in the language of your choice. You can select an icon, which is displayed for your custom function in the *Functions* columns for messages in the Log Messages view.

You can also select to open a function that is a transaction or a report in a new session and you can decide whether or not the first screen is to be skipped.

Fill Method for Parameters

The parameters for the selected transaction or report are automatically imported. In the *Parameter* group box, in the *Fill Method* field, you can select from the following methods of filling the parameters of a report or transaction:

- **`Manual Value`**
  You can enter a value manually in the *Value* field.
- **`Message Variable`**
  Use the *Value from Message Variable* field to select a value from 4 message variables.
- **`Offset Message Variable`**
  See the *Define Offset and Length from Message Variable* section below.
- **`Value Mapping`**
  See the *Value Mapping* section below.

Define Offset and Length from Message Variable

You can fill parameters for transactions or reports by using the offset and length of the value of a message variable. If you select **`Offset Message Variable`** as the fill method, in the *Maintain* column, an icon appears that you can choose to enable you to add details. The *Details Offset in Message Variable* popup is displayed and you can enter values for using the offset and length of the value of a message variable.

> ### Note
>
> To assign the first character, use offset **`0`**; to assign the second character, use offset **`1`** (see the *Example* below).

> ### Example
>
> The following is an example of the use of offset and length from a flight booking:
>
> Message: “Flight LH 1234 20140714 does not exist”
>
> Message variable 1: LH 1234 20140714

Parameter Description	Value	Value from Message Variable	Offset	Length
Airline Carrier ID Number	LH	Message variable 1	0	2
Connection ID	1234	Message variable 1	3	4
Date	20140714	Message variable 1	8	8

Value Mapping

You can fill parameters for transactions or reports by using the value mapping functionality. If you select **`Value Mapping`** as the fill method, in the *Maintain* column, an icon appears that you can choose to enable you to add an assignment. On the *Details Value Mapping* popup that is displayed, enter the namespace and value mapping name. In the *Parameters* group box, for parameters 1 to 5 in the *Parameter VMAP* field, the type of fill method must be selected from the following options:

- Manual Value
  Enter a value in the *Value* field.
- Message Variable
  Select from 4 message variables fields.
- Offset Message Variable
  See the *Define Offset and Length from Message Variable* section above.

> ⓘ Note
>
> If you select **`Value Mapping`** as the fill method, the system uses the authorization object `/AIF/VMAP` to check your authorization before you can execute a function, test a function, or simulate a value mapping.
>
> The authorization check is done for the assigned namespace, value mapping name, and for the activity 03 (Display).
>
> For more information about authorization objects and `/AIF/VMAP`, see the Master Guide for the SAP Application Interface Framework.

3.2.3.2 Custom Data Links Maintenance

Use

In *Monitoring and Error Handling*, you are able to create, overwrite, and delete custom data links.

For more information, see Maintaining Custom Data Links [page 54].

Features

If you select (double click) a message in the Log Messages view and a custom data link exists, the following is displayed in the Data Content view:

- For a table and field combination, the relevant column is highlighted in red
- For a structure and field combination, the relevant field is highlighted in red

If you select multiple data messages and only one has a custom data link, the data is loaded in the Data Structure view. You can select (double click) a structure or table and the content of all the data messages is displayed in the Data Content view. In the column that contains the relevant field for the custom data link, the only cells highlighted in red are those for the message with the custom data link.

3.2.3.2.1 Maintaining Custom Data Links

Prerequisites

In *Monitoring and Error Handling*, to create, overwrite, or delete custom data links, select both a message in the Log Message view and a cell in the Data Content view.

> ⓘ Note
>
> In the Data Content view, if a custom data link already exists, the relevant column or field is highlighted in red as follows:
>
> - For a **table** and field combination
>   In this case, the relevant line number is not known and may differ from the line number with which the link was created. Therefore, the relevant column is highlighted in red.
> - For a **structure** and field combination
>   The relevant field is highlighted in red.

Procedure

Creating a new Custom Data Link

1. In the Log Message view of *Monitoring and Error Handling*, choose *Customize* and then *Custom Data Link*.
2. If a custom data link does not exist for the table/structure and field combination, a pop up appears to inform you.
3. On this *Custom Data Link* popup, choose *Create* to create a new custom data link.
4. You get a confirmation that the new custom data link has been created. In the Data Content view, the relevant column/field is highlighted in red.

Overwriting an Existing Custom Data Link

1. In the Log Message view of *Monitoring and Error Handling*, choose *Customize* and then *Custom Data Link*.
2. A pop up appears to inform you of the existence of the custom data link and you can decide whether to overwrite with the new combination of table/structure and a field or to delete it.

3. On this *Custom Data Link* popup, choose *Overwrite* to overwrite the existing custom data link.
4. You get a confirmation that the custom data link has been modified. In the Data Content view, the relevant column/field is highlighted in red.

Deleting an Existing Custom Data Link

1. In the Log Message view of *Monitoring and Error Handling*, choose *Customize* and then *Custom Data Link*.
   - If a custom data link already exists for the table/structure and field combination, the column/field is highlighted in red. A pop up appears to inform you of the existence of the custom data link and you can decide whether to overwrite it with the new combination of table/structure and field or to delete it.
   - If a custom data link already exists with the **exact same detailed** combination of table/structure and field, the column/field is highlighted in red. A pop up appears to inform you of the existence of the custom data link and you can delete it.
2. On either of these two *Custom Data Link* popups, choose *Delete* to delete an existing custom data link.
3. You get a confirmation that the existing custom data link has been deleted. In the Data Content view, the relevant column/field is no longer highlighted in red.

Result

You have created a new custom data link or you have modified or deleted an existing custom data link. For new or modified custom data links, the relevant column/field is highlighted in red in the Data Content view. Furthermore, when you double click on the Log Message view, the corresponding structure/table is displayed.

3.2.4 Technology Support

The SAP Application Interface Framework supports business users in monitoring interfaces with different interface technologies, for example, ABAP proxies, IDocs, OData, batch input, and tRFC/qRFC.

Some functionalities are not available for certain supported technologies:

- In the SAP Application Interface Framework, messages from the following supported technologies are only visible in *Monitoring and Error Handling* when they are in an error state:
  - tRFC and qRFC
  - OData
  - Batch Input

    > **Note**
    >
    > If you select the *Keep Session* checkbox, the batch input session is not deleted and is still visible after it has been successfully processed.
- The editing of data is not supported for the following:
  - tRFC and qRFC
  - OData
  - CIF Postprocessing

> **Note**
>
> The editing of data is supported for batch input; however, in this case, a new batch input session is created and the original one is deleted.

3.3 Value Mapping Maintenance

Use

In the SAP Application Interface Framework, you can maintain a value mapping table.

Integration

Before you can maintain a value mapping, it is necessary that a value mapping has been defined by the interface developer when creating the value mapping during Customizing. This setting is made in Customizing for the *SAP Application Interface Framework* (transaction code `/AIF/CUST`) under *Interface Development* > *Define Value Mapping*.

You have to have the corresponding authorizations to display and maintain value mapping for a given namespace.

To define a date range for a validity period number for a value mapping, you can access validity period maintenance (transaction `/AIF/VPN`) from the SAP Easy Access menu by choosing *Cross-Application Components* > *SAP Application Interface Framework* > *Interface Development* > *Maintain Validity Periods*.

SAP GUI

You can access the maintenance of value mapping from the SAP Easy Access menu by choosing *Cross-Application Components* > *SAP Application Interface Framework* > *Value Mapping*.

You can navigate directly to *Maintenance of Value Mapping* from the Log Messages view of the main screen of *Monitoring and Error Handling*. If an error resulted from a faulty value mapping, you can select the log message and choose  to maintain the value mapping.

Web-Based UI

You can navigate directly to *Value Mapping* from the Log Messages view of *Monitoring and Error Handling*. If an error resulted from a faulty value mapping, you can select the log message and choose  to maintain the value mapping.

Features

SAP GUI

In *Database Type*, whether or not the values can be changed in the production system is displayed. This setting is made by interface developers in Customizing for the *SAP Application Interface Framework* under *Define Value Mapping*.

In *Value Mapping Type*, which one of the following 2 value mapping types has been specified in Customizing is displayed:

- Single value mapping
  If the value mapping is defined as a single value mapping, one external value is mapped to one internal value.
- Multiple value mapping
  If the value mapping is defined as a multiple value mapping, up to five external values can be mapped to one internal value. The number of external values is maintained by the interface developer in Customizing for the *SAP Application Interface Framework* under *Interface Development* > *Define Value Mappings*.

In *Sending System*, you can specify the name of a particular business system that is to be used for the value mapping. The selected value mapping is only valid for the specified sending system.

Missing Values is only displayed if you access *Value Mapping* from *Monitoring and Error Handling*. *Missing Values* shows the values missing according to the error message in *Monitoring and Error Handling*. You can enter the values into the value mapping table by choosing *Insert Values*.

The value mapping itself is displayed in a table. The first column is always an index, which you cannot edit. In the following column(s), the external values are defined. The last column where values can be entered is for the internal values.

> **Note**
>
> If an interface developer has maintained data elements for external values and the internal value in the value mapping definition, the column headings are replaced by the corresponding labels.

Web-Based UI

In *Value Mapping*, the first column in the table displays the value mapping number, which you cannot edit. In the following column(s), the external values are defined. In the *IntValue* (Internal Value) column, you can enter the internal values.

Activities

SAP GUI

To maintain the value mapping, choose  . The transaction also provides you with a  function that enables you to check where the value mapping is used. If you have the authorization, double-clicking on an entry in the provided list opens a new window and the Customizing where the mapping is used is displayed.

To enter new empty lines into the table, choose  and select the number of lines you want to enter.

To delete single or multiple lines from the table, select the lines and choose  .

If the value mapping was already defined for at least one sending system, you can enter those values into the current value mapping. To do this, choose  , select the business systems of the value mapping you want to use, and choose *OK*. The values are automatically entered into the current value mapping.

Insert Values is only displayed if you access *Value Mapping* from *Monitoring and Error Handling*. To insert a new value mapping using the missing values from *Missing Values*, choose *Insert Values*.

You are able to export data, for example, to a spreadsheet, by choosing  .

You are also able to copy data from Microsoft Office Excel cells. *Copy* the data you want to insert to the clipboard. Select the first cell the data is to be inserted into in the value mapping table and *Paste*. The lines are inserted into the value mapping table. If the lines inserted exceed the number of available lines, then additional lines are appended automatically.

Web-Based UI

If you have the authorization, in the *ValMap* (Value Mapping) column of the line with an error status, double-clicking on  navigates to *Value Mapping*. Once you have maintained the value mapping, choose  to save your correction and navigate back to *Monitoring and Error Handling*.

Validity Period

You can define the validity period type that should be used, that is, either the validity period number and/or the validity dates, depending on the settings in Customizing for the *SAP Application Interface Framework* under *Interface Development* > *Define Value Mappings*, where the following are available to define a validity period:

- The *Use Validity Period Number* checkbox
  Select to make the *Validity Period Number* field visible
- The *Enter Validity Period Dates Directly* checkbox
  Select to make the *Date From* and *Date To* fields visible

For a value mapping, you can maintain either the validity period number or the dates or both of these.

Uniqueness of Internal and External Values

In Customizing for the *SAP Application Interface Framework* under *Interface Development* > *Define Value Mappings*, you can specify whether or not the system checks the uniqueness of the external values, the internal values, or a combination of both before saving the entries in value mapping. You have the option of specifying whether you are alerted by an error or a warning if duplicate values are detected.

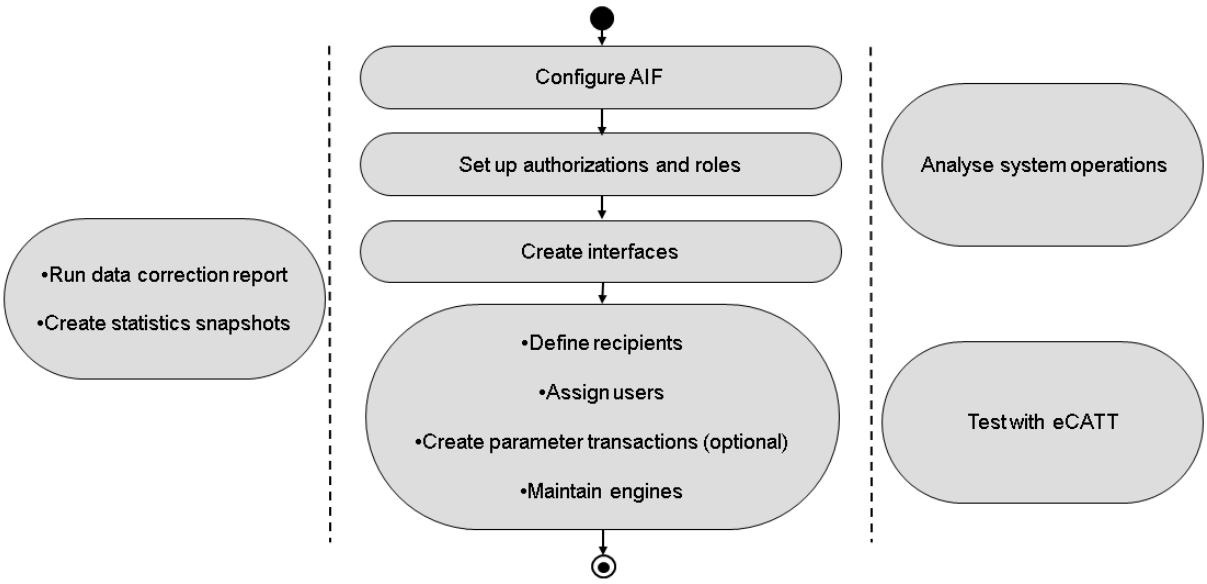
4 SAP Application Interface Framework for IT Personnel

Use

The SAP Application Interface Framework presents IT personnel with various options for configuring the runtime and checking the system state.

The SAP Application Interface Framework provides IT personnel with a menu of Customizing activities to facilitate interface set up. It enables you to reuse interface building blocks (structure mappings, value mappings, checks, and actions), which you and your colleagues have created, over multiple interfaces. You can set up flexible authorization rules and assign the relevant business users to monitor interfaces and to handle errors. You have reports and options for analyzing the operation of the system. Integration with eCATT enables automatic testing.

Process



Application Interface Framework for IT Personnel

The above graphic shows the typical activities for IT Personnel when using the SAP Application Interface Framework:

1. The system is initially configured by defining the error handling application and the trace levels (see Initial System Configuration [page 61]).
2. Security is configured by the Customizing of authorizations, for example, to limit access to monitoring and error handling by business users. For more information, see the Master Guide.

3. Interfaces are created and documented to enable the accurate and reliable transfer of data (see Interface Creation [page 68], Interface Builder [page 71], Service Implementation Workbench Support [page 73], IDoc Structure Generation and Interface Definition [page 74], tRFC and qRFC Function Module Generation [page 79], Interface Documentation Maintenance [page 80], and Status Handling [page 81]).
4. Monitoring and error handling is set up by:
   - Defining recipients
   - Assigning users
   - Optionally creating parameter transactions
   - Maintaining engines depending on the interface technology you use

   For more information, see Defining Recipients [page 90] Assigning Users to Recipients [page 95], Parameter Transactions [page 103], and Technology Support [page 125].
5. System operations can be analyzed continuously (see Operations for SAP Application Interface Framework [page 110]).
6. Reports are provided for the following:
   - To detect and correct data inconsistencies (see Data Correction [page 121])
   - To create and display error statistics snapshots (see Generation and Display of Snapshots [page 122])
7. The SAP Application Interface Framework enables you to do automatic testing using eCATT (see Test Automation with Extended CATT [page 179]).

Integration with SAP Workflow Service

SAP Application Framework is well integrated with the SAP Workflow service to enable you to trigger a workflow in case of an error in the message status. You can configure a specific message interface to trigger a workflow instance in the SAP Workflow service in case any of the messages related to that interface is in status Error. This will give you an option to configure a specific set of actions in your workflow instance in case of messages in error state, like trigger an email.

Check out this document for an overview: SAP Workflow Service Integration with SAP Application Interface Framework

Check out this document for configuring the add-on to trigger workflow instances: Configuring Application Interface Framework Add-On to Trigger Workflow Instances

More Information

To access Customizing for the *SAP Application Interface Framework*, in the SAP Easy Access menu, choose *Cross Application Components* > *SAP Application Interface Framework* > *Administration* > *Configuration* > *Customizing* or use transaction code `/AIF/CUST`. For more information about individual Customizing activities, see the Customizing documentation.

4.1 Initial System Configuration

Use

The initial system configuration describes the steps necessary for you to configure the system for productive use. You make the settings in Customizing for *SAP Application Interface Framework* (transaction code `/AIF/CUST`) under *Error Handling*.

Process

1. To activate error handling, define an error handling application in Customizing for *SAP Application Interface Framework* under *Error Handling* > *Define Applications*. For more information, see Defining Error Handling Applications [page 61].
2. Define trace level information in Customizing for *SAP Application Interface Framework* under *Error Handling* > *Define Global Features*. For more information, see Defining Trace Levels [page 64].

4.1.1 Defining Error Handling Applications

Context

To activate error handling in the *Monitoring and Error Handling* transaction, a standard error handling application is required. The standard error handling application delivered with the SAP Application Interface Framework is called *AIF*.

> → Recommendation
>
> As an initial step, it is recommended that you check if the *AIF* application is correctly defined and, if it does not exist, create and maintain it.

Procedure

1. Maintain error handling applications in Customizing for the *SAP Application Interface Framework* (transaction code `/AIF/CUST`) under *Error Handling* > *Define Applications*.
2. Choose *Maintain Application* and check that there is an entry for application ID *AIF*. If the application ID *AIF* does not exist, run the `/AIF/SETUP` transaction code. There should be the following entries:

Field	Value
Short description	**AIF Application**
Table of maintaining Appl. key fields	**/AIF/T_INF_TBL**
Entry Data Facade	**/AIF/CL_AIF_ENTRY_DATA_FACADE**
Module	**/AIF/SAPLAIF_SP_SSC**
Screen Number	**9101**
Appl. Log Context Structure Name	**/AIF/BAL_CONTEXT**
Action Handler	**/AIF/CL_AIF_ACTION_HANDLER**
Program Name	(blank)
GUI Status	(blank)
URL for Interface Monitor	(blank)
Image Object Name for Interface Monitor	**/AIF/SAP_NETWEAVER_MONITOR**

3. Make sure you select the *AIF* application and choose *Maintain Application Specific Key Fields*. There should be the following three entries:

Position Sequence No.	Field	Value
1	*Work Field*	**P_NS**
	Field Name	**NS**
	Data Element	**/AIF/NS**
	Field Is Select-Option	(blank)
2	*Work Field*	**S_IFNAME**
	Field Name	**IFNAME**
	Data Element	**/AIF/IFNAME**
	Field Is Select-Option	**X**
3	*Work Field*	**P_IFVERS**
	Field Name	**IFVER**
	Data Element	**/AIF/IFVERSION**

Position Sequence No.	Field	Value
	Field Is Select-Option	(blank)

4. Make sure that you select *AIF* and choose *Register Functions*. There should be the following six entries:

Function	View Code	Field	Value
CANCEL	1	*Sequence Number*	**2**
		Button Description	**Cancel**
		Button tool tip	**Cancel**
		Icon	**@3J@**
MONI	1	*Sequence Number*	**3**
		Button Description	(blank)
		Button tool tip	**Integration Engine: Monitor**
		Icon	**@16@**
QRFC	1	*Sequence Number*	**4**
		Button Description	(blank)
		Button tool tip	**QRFC Monitor**
		Icon	**@6P@**
RESTART	1	*Sequence Number*	**1**
		Button Description	**Restart**
		Button tool tip	**Restart**
		Icon	**@15@**
SLOGS	3	*Sequence Number*	**1**
		Button Description	(blank)
		Button tool tip	**Display Error Message(s)**
		Icon	**@DR@**

Function	View Code	Field	Value
TRACELEV	1	*Sequence Number*	**5**
		Button Description	(blank)
		Button tool tip	**Maintain Trace Level**
		Icon	**@4H@**

5. Make sure that you select *AIF* and choose *Exclude Function from Appl. Toolbar*. There should be no entries.

4.1.2 Defining Trace Levels

Use

Trace levels define the level of detail for the log messages that are saved in the application log. There are four trace levels in the delivery of the SAP Application Interface Framework: 0, 1, 2, and 3. The default trace level that is used for message processing is 0.

Procedure

1. Maintain trace level information in Customizing for the *SAP Application Interface Framework* (transaction code `/AIF/CUST`) under *Error Handling* > *Define Global Features*.
2. Choose *Define Trace Level* and check that there is an entry for each trace level.

> → Recommendation
>
> It is recommended that you maintain trace level information starting from trace level 0 because the settings of a lower trace level are automatically applied to all higher trace levels.

3. To maintain trace level information, select the trace level and choose . You specify which message types are to be included in the trace level. You can make separate entries depending on whether the message source is *from framework* or *from individual interface*. To include a specific message type from the following for a specific message category, select the corresponding checkbox:

Message Type	Description
A	Technical error
E	Application error

Message Type	Description
I	Information
W	Warning
S	Success

> **Note**
>
> The desired trace level for a specific message or all messages of an interface can be set in the Error Handling [page 34] transaction or directly in the database table `/AIF/FINF_TL`.

4.2 Configuration of Security and Authorizations

Use

SAP Application Interface Framework uses flexible authorization rules to allow you to restrict access to data and to monitoring and error handling. These security features enforce compliance by following the need-to-know principle when restricting access to interface data, and by tracking the accesses to security-relevant data.

Features

The configuration of security and authorizations in SAP Application Interface Framework includes the following objects, roles, and data:

- Standard authorization objects
- Predefined template roles
- The integration of custom-defined authorization objects
- Some additional security features and specific data protection-relevant functions.

More Information

For detailed information about security issues, see the Master Guide for SAP Application Interface Framework.

4.3 Set Up Monitoring and Error Handling

Use

In the SAP Application Interface Framework, IT personnel set up monitoring and error handling as appropriate to the various authorized business users.

Process

IT personnel are involved in setting up monitoring and error handling as follows:

1. Interfaces Developers create interfaces to enable the accurate and reliable transfer of data.
   For more information, see Interface [page 66] and Interface Determination [page 88].
2. System Administrators define recipients.
   For more information, see Defining Recipients [page 90]
3. System Administrators assign users.
   For more information, see Assigning Users to Recipients [page 95].
4. System Administrators optionally create parameter transactions.
   For more information, see Parameter Transactions [page 103].

4.3.1 Interface

Definition

A connecting point between the application and the data in the external format. At the most basic level, every interface needs data structures, data, and business logic.

Use

Building an interface can be complicated and the effort needed to build, maintain, and monitor interfaces can be underestimated. SAP Application Interface Framework provides a structured framework that enables users to easily implement interface building blocks, providing Customizing menus that allow an IT project team to start generically and then tailor the interface to business processes and the team's specific needs. This simplifies and standardizes the building and implementation process reducing development costs and time. After the key interface components are built, they can easily be reused in other interfaces. Interface components are assigned to namespaces, that is, grouped by functional area so users can easily access the components. Every time a new interface is required, interface developers working with SAP Application Interface Framework can leverage their previous work, as well as the work of their colleagues, to streamline the process, reduce errors, and contribute to cost savings.

> ### Example
>
> An interface requires that a validation be completed for a material number. SAP Application Interface Framework not only provides the steps to implement that validation as part of the interface, but also retains that implementation so the validation can be reused in other interfaces.

Structure

In SAP Application Interface Framework, there are the following four key interface building blocks:

- Structure mappings
  The sending system and receiving system often do not have the same data structures for their business objects or functions. Structure mappings are needed to map interface data from the source structure to the destination structure.
- Value mappings
  The sending system and receiving system might have different values for an object or business value. A value mapping from a source value to a destination value solves this.
- Checks
  To ensure that your interface data is correct, checks are needed to verify that the interface data is in the right format, that it is valid from a business perspective, and that it can successfully be interpreted by the application system.
- Actions
  Actions need to be triggered to execute the interface's business logic, for example, an action triggers a call to a Business Application Programming Interface (BAPI) to create a purchase order.

Integration

Use transaction code `/AIF/CUST` to access Customizing for *SAP Application Interface Framework*. In the following Customizing activities, you can define interface building blocks:

- Define Structure Mappings
- Define Value Mappings
- Define Fix values
- Define Checks
- Define Actions

The Interface Builder provides you with a quick and easy graphical way to create structure mappings for interfaces in the SAP Application Interface Framework (see Interface Builder [page 71]).

You can generate SAP Application Interface Framework relevant interface objects via the Service Implementation Workbench (see Service Implementation Workbench Support [page 73]).

The *Generate IDoc Structure and Interface Definition* report (transaction code `/AIF/IDOC_GEN`) enables you to create a structure from an IDoc basic type and, optionally, to create a corresponding SAP Application Interface Framework interface definition for the newly created structure (see IDoc Structure Generation and Interface Definition) [page 74].

The *AIF Batch Input Structure Generator* report (transaction code `/AIF/BDC_GEN`) enables you to create a structure for a batch input recording and, optionally, to create a corresponding SAP Application Interface Framework interface definition for the newly created structure..

The *Function Module Generator* report (transaction code `/AIF/RFC_FUNC_GEN`) supports interface developers in implementing the t/qRFC interface for the SAP Application Interface Framework by generating the SAP structure, an interface for the SAP Application Interface Framework, and the playback function module (see tRFC and qRFC [page 153]).

The SAP Application Interface Framework can use preprocessing to execute some logic on top of IDoc or Proxy raw message data with limited maintenance effort. To activate preprocessing, you select the *Preprocessing* checkbox in Customizing for *SAP Application Interface Framework* under *Interface Development* > *Define Interfaces* (see Preprocessing) [page 170].

More Information

For more information about individual Customizing activities, for example, *Define Interfaces* and *Define Structure Mappings*, see the corresponding Customizing documentation.

4.3.1.1 Interface Creation

Use

In SAP Application Interface Framework, interface developers create and customize interfaces and assign recipients to them to enable business users to monitor the interfaces they are responsible for.

Prerequisites

Before an interface can be developed using SAP Application Interface Framework, some steps have to be performed. However, the actual steps to be performed depend on the interface technology you want to use.

> Example
>
> - For proxy interfaces, it is necessary that the proxy class and the related proxy structure are generated.
> - For IDocs, some ALE settings have to be made, for example, create message type and maintain process code.

Depending on the interface technology and scenario, the following steps have to be executed before the interface can be customized for SAP Application Interface Framework:

- Implement the call of SAP Application Interface Framework
- Find or create BAPI(s) and function module(s)
  Usually, the business logic in the backend is executed by methods (for example, of business objects), BAPIs, or function modules that can be predefined by SAP or implemented specifically for your company.

In order to call these objects in an action of SAP Application Interface Framework, you have to build your SAP data structure to accommodate the required data and their expected format.

- Create SAP data structure
  You need to create an SAP data structure to exchange values during the processing of the function modules. The structure can consist of a hierarchy of substructures and tables. For an outbound interface, the SAP data structure constitutes the source structure. Its components are usually defined by the format of the data stored in the system. For an inbound interface, the SAP data structure constitutes the destination structure. Its components are usually defined by the class methods or function modules you want to execute in the system.

Process

> **Note**
>
> Depending on your scenario and interface technology, the steps described below might vary.

1. You define a namespace in Customizing for *SAP Application Interface Framework* (transaction code `/AIF/CUST`) under *Interface Development* > *Define Namespace*.
2. You define an interface in Customizing for *SAP Application Interface Framework* under *Interface Development* > *Define Interfaces*.
   You have to define an interface name, a version, the name of the inbound or outbound proxy class, the raw data structure, and the SAP data structure.

   > **Note**
   >
   > If your interface is a proxy interface, when you define the name of the inbound or outbound proxy class and choose *Enter*, the raw data structure and record type (which are derived from the proxy class) are filled automatically in the raw data structure.
3. Depending on the interface technology and scenario, you have to maintain the engines used to handle messages of the interface. You maintain engines in Customizing for *SAP Application Interface Framework* under *Interface Development* > *Additional Interface Properties* > *Specify Interface Engines*.
4. You can define your interface building blocks in Customizing for *SAP Application Interface Framework* (transaction code `/AIF/CUST`) under *Interface Development* as follows:
   - Create structure mappings
     In structure mapping, you map the fields of the source structure to the fields of the destination structure. You also assign value mappings, fix values, checks, and actions. You do this under *Define Structure Mapping*.

     > **Note**
     >
     > You can define the following objects in the list either by forward navigation from *Define Structure Mapping* or you can access Customizing activities to create them individually. If you want to use forward navigation from *Define Structure Mapping*, enter the namespace and the name of the object you want to create and choose *Enter*. If the object does not exist, you can choose to create it. If you want to access Customizing activities to create the objects individually, it is recommended that you create the objects (value mappings, fix values, checks, actions) before you create the structure mapping.

- If necessary, define the following:
  - Value mappings
    Value mappings are used to derive one value in the destination structure from up to five values in the source structure that can be optionally enriched with further data from the back end system. You do this under *Define Value Mappings*.
  - Fix values
    You can use fix values if you want a certain field in the destination structure to always have the same value. You can either enter a fix value directly into the structure mapping or you can create a named fix value, which can be assigned to a field in the structure mapping. You define fix values under *Define Fix Values*.

    > → Recommendation
    >
    > It is recommended to use a named fix value so that you can reuse it in different structure mappings.
  - Checks
    You can use checks to verify if the content of the specified field or fields match a certain condition. Checks can be used in several places in SAP Application Interface Framework, for example you can assign them in structure mapping or you can assign them to a function in an action. You define checks under *Define Checks*.
  - Actions
    Actions contain the business logic of your interface. You can assign multiple action function modules. Within these function modules, you can call BAPIs, SAP standard function modules, or customer function modules that process the data of the SAP data structure. You define actions under *Define Actions*.

5. Depending on your scenario, you have to set up interface determination to configure which interface in SAP Application Interface Framework should be used to process a message. You do this in Customizing for *SAP Application Interface Framework* under *Interface Determination* (see Interface Determination [page 88]).

   > ✤ Example
   >
   > Scenario: You have multiple interfaces in SAP Application Interface Framework for the same ABAP Proxy.
   >
   > Scenario: You use the generic process function delivered with SAP Application Interface Framework to process an IDoc in SAP Application Interface Framework.

6. You define a recipient, which is needed for alert configuration, and you can subsequently assign users to it. You define the recipient in Customizing for *SAP Application Interface Framework* under *Error Handling* > *Namespace-Specific Features*. Enter the namespace you want to create the recipient for and choose *Define Recipients*. Define a name and a description for your recipient.

   > ⓘ Note
   >
   > You can reuse existing recipients.

7. You assign users to your recipient in Customizing for *SAP Application Interface Framework* under *System Configuration* > *Recipients* > *Assign Recipients*. Enter the namespace and the name of your recipient. You can assign users, roles, or external addresses as follows:
   - Assign users/roles

Select the message type that is to be included in an alert message. If you set the *Include on Overview Screen* indicator, the users or roles that are assigned to the recipient see the messages in the *Interface Monitor* of SAP Application Interface Framework. If you set the *Technical User* indicator, this identifies technical users or roles who can also see messages that have the status *In Process* or *Technical Error* in the *Interface Monitor.*

- Assign external addresses
  This enables you to define external addresses (for example, e-mail and fax) that are notified if messages with the specified type occur.

> **Note**
>
> While assigning users to recipients provides you with a recipient-specific view, a user-centric view exists as well. You can use transaction `/AIF/RECIPIENTS` to display and maintain the recipients a specific user belongs to. Alternatively, you can use transaction `/AIF/MYRECIPIENTS` to see the recipients your user is assigned to.

More Information

For more information about the maintenance of users and recipients, see Defining Recipients [page 90] and Assigning Users to Recipients [page 95].

4.3.1.2 Interface Builder

Use

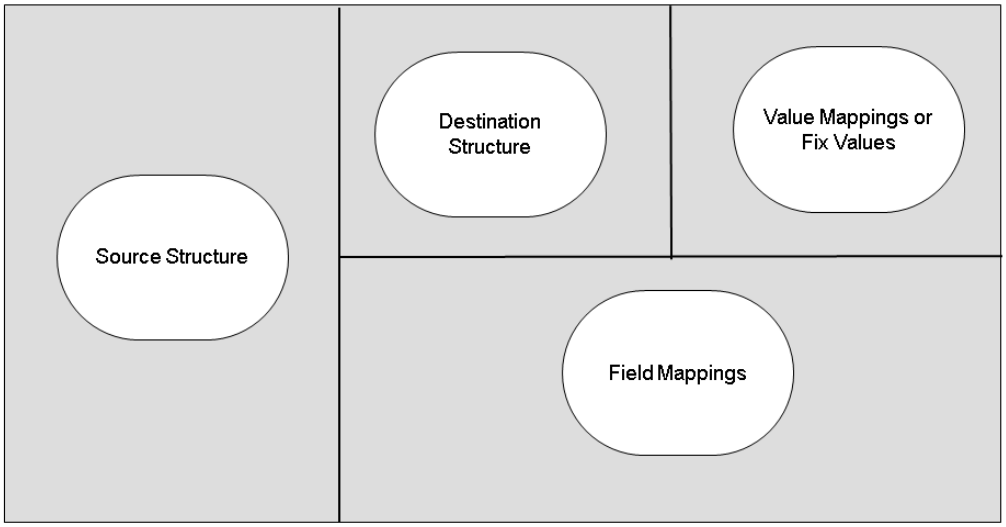
The *Interface Builder* (transaction code `/AIF/IFB`) provides you with a quick and easy graphical way to create your structure mappings for interfaces in the SAP Application Interface Framework. You can create field mappings and use drag and drop to assign value mappings and fix values.

Prerequisites

You have defined an interface in Customizing for the *SAP Application Interface Framework* (transaction code / AIF/CUST) under *Interface Development* > *Define Interfaces*.

You require authorization to use Customizing for the *SAP Application Interface Framework* under *Interface Development* > *Define Structure Mappings*.

Features



Interface Builder Views

As shown in the graphic above, the *Interface Builder* screen consists of the following 4 views:

- The first view contains the source structure in a hierarchical tree.
- The second view contains the destination structure in a hierarchical tree.
- The third view allows you to toggle between value mappings and fix values, which are also displayed in a hierarchical tree.
- The fourth view displays the field mappings of the currently selected structure mapping in a table. There are both display and change modes available to you.

Activities

You can access the *Interface Builder* report from the SAP Easy Access menu by choosing *Cross-Application Components* > *SAP Application Interface Framework* > *Interface Development* > *Interface Builder*.

To create a structure mapping, you drag and drop a structure or table from the source structure to a structure or table in the destination structure. The fields of the destination structure are then displayed in a table in the fourth view containing the field mappings.

To create a field mapping, you drag and drop a field of the source structure to a corresponding field in the field mapping table. You can also map fields of the current destination structure by dragging a cell of the column containing the destination column and dropping it on a cell for the source field.

If you want to create a value mapping or fix value, drag and drop the entries from the third view to the corresponding field in the field mapping table.

4.3.1.3 Service Implementation Workbench Support

Use

The Service Implementation Workbench (SIW) offers a single point of entry for you to create, implement, and recreate enterprise services quickly and easily in a semi-automated and guided process to ensure reusability of business functionality.

The SAP Application Interface Framework provides you with an SIW configuration and template. You can generate SAP Application Interface Framework relevant objects and data entries via SIW in different system landscapes. In the SAP Application Interface Framework, you can use the delivered SIW configuration to integrate with ECH (see Error and Conflict Handling Integration [page 143]).

Prerequisites

You require authorization for the following 3 authorization objects:

- `S_CTS_ADMI`
- `S_CTS_SADM`
- `S_SIW_CFG`

You must define the SIW system landscape and configuration.

Features

The SIW template delivered with the SAP Application Interface Framework supports you in generating many of the objects needed for an SAP Application Interface Framework interface. For example, the following objects can be generated with the template:

- Proxy class implementation
- Interface definition
- Action
- Action function
- SAP data structure

In order to guarantee flexibility and scalability, the custom SIW configuration does the following:

- Supports both bulk and single service
- Supports both synchronous and asynchronous
- Provides all the needed context variables
- Provides one template for above mentioned functions
- Is able to inject Customizing entries into transport request

Activities

Variables Maintenance and Object Generation in SIW

In SIW, a project is created based on a configuration defined for the proxy-based creation type. On the SIW *Project Creation Screen*, there is an SAP Application Interface Framework-specific creation sub-screen for proxy-based services where an interface developer can enter the following key variable values:

- Prefix
  Used for all repository objects generated with the SIW
- Package
  The package storing generated objects
- Function group
  Contains the generated action function module
- Namespace / interface name / interface version
  The AIF interface keys. The namespace is not generated by SIW, but must be defined in advance.
- Interface description
- Action
  The AIF action used by the AIF interface. The generated action function module is assigned as a function of this action.
- Action description

After the project is created, the *Project Implementation Screen* is displayed. Here, you enter the project-specific context data, for example, the SAP data structure name used for `DDIC` structure generation. You specify the function module or BAPI that should be called in the action function. The SAP data structure is generated to fit this function module's or BAPI's parameters. Interface developers can define their SAP Application Interface Framework interface via the context variables.

Once all the required context variables are correctly maintained, a service developer can generate the following:

- An implementation based on the existing proxy class
- `DDIC` objects, for example, table type and structure
- An SAP Application Interface Framework-specific action function module, as well as generate code snippets for the action function module
- Basic Customizing data for the SAP Application Interface Framework, for example, interface definition

4.3.1.4 IDoc Structure Generation and Interface Definition

Use

The *IDoc Structure Generator* report (transaction code `/AIF/IDOC_GEN`) enables you to create a structure for an IDoc basic type or a structure for a combination of IDoc basic type and IDoc extension. The report provides optional functionality to create a corresponding SAP Application Interface Framework interface that uses the newly created structure.

The structure generated by the report consists of the control record and the basic type's segments. If the structure was created for an IDoc basic type together with a specific extension or all extensions, the generated

structure also contains the extension segments. It can be used in the SAP Application Interface Framework interface definition as a raw data structure for the following:

- Displaying existing IDocs in the *Monitoring and Error Handling* transaction
- IDoc processing using the runtime of the SAP Application Interface Framework, calling the IDoc standard process function module in an action
- IDoc processing with ALE runtime, writing index tables with the AIF enabler

In the above scenarios, the generated structure is used as raw data structure and SAP data structure.

The structure generated by the report can also be used as the raw data structure for IDoc processing using the runtime of the SAP Application Interface Framework, calling any other function module or BAPI in an action. In this scenario, the SAP structure has to be defined in a way that it accommodates the structure of the function modules or BAPIs you want to call.

Mass IDoc Structure Generator

The *Mass IDoc Structure Generator* report (transaction code `/AIF/IDOC_MASS_GEN`) provides the same functions and features as the *IDoc Structure Generator* report but can be used for the mass generation of structures and interfaces based on a list of IDoc types.

Integration

The report can create an interface if the corresponding data is maintained on the screen. In those cases where the raw data structure created by the report and the SAP data structure are identical structures, the *Move Corresponding* checkbox is selected automatically by the report in Customizing for the *SAP Application Interface Framework* under *Interface Development* > *Define Interfaces* to map the source to the destination structure during the runtime of the SAP Application Interface Framework.

If the raw and SAP data structures are not the same, you must enter the SAP data structure in Customizing for the *SAP Application Interface Framework* under *Interface Development* > *Define Interfaces* and you have to define a structure mapping under *Interface Development* > *Define Structure Mappings*.

The engines for the interface are set automatically in Customizing for the *SAP Application Interface Framework* under *Interface Development* > *Additional Interface Properties* > *Specify Interface Engines*.

Furthermore, the basic type and message type are automatically maintained in Customizing for the *SAP Application Interface Framework* under *Interface Development* > *Additional Interface Properties* > *Assign IDoc Types*.

Mass IDoc Structure Generator

If you execute the selection screen of the *Mass IDoc Structure Generator* report, the system calls the *IDoc Structure Generator* report in a second step.

You can define an IDoc collection that contains a list of IDocs types for use with the *Mass IDoc Structure Generator* report. You do this in Customizing for *SAP Application Interface Framework* under *Interface Development* > *Additional Interface Properties* > *Define IDoc Type Collection*.

Features

IDoc Structure Generator

The report enables you to create one structure per basic type/extension combination. If you created a structure once for a specific basic type, it is possible to recreate the structure. However, the report does not allow you to create a structure with the same basic type/extension combination but a different structure name.

A proposal for the IDoc data structure is automatically created from the basic type and the prefix for the structure. The proposed name can be changed by the user.

The report gives you the option of creating a corresponding SAP Application Interface Framework interface definition together with the structure. The report allows you to create a new interface for a structure that is already created.

The report also enables you to transport the created objects.

After you enter your selection on the *IDoc Structure Generator* screen and choose  , the report generates the required structures. It also creates the interface if the corresponding information has been provided.

Mass IDoc Structure Generator

On the first screen of the *Mass IDoc Structure Generator*, you can select the IDoc type collection. You can maintain an IDoc type collection in Customizing for *SAP Application Interface Framework* under *Interface Development* > *Additional Interface Properties* > *Define IDoc Type Collection* .

After you enter your selection on the *Mass IDoc Structure Generator* screen and choose  , the report selects the corresponding IDoc types from the IDoc type collection and displays them in a table on the next screen. On this screen, there are pushbuttons and a *Generation Status* column that use traffic lights to show the current status of IDoc generation.

Furthermore, the data maintained in the IDoc type collection is displayed in the list. You can change some of the data, for example, the message type and interface name.

From this list, you select IDoc types and choose *Execute*. You can enter the namespace for the interface, the prefix for the structure, the package, and transport requests. Choose *Execute* to trigger jobs to generate the IDoc structures and interface.

> **Note**
>
> If the interface already exists for the namespace entered, the existing interface is not overwritten by the *Mass IDoc Structure Generator* report. Instead, an information message is added to the log.

Activities

SAP Easy Access

You can access these 2 reports from the SAP Easy Access menu by choosing *Cross-Application Components* > *SAP Application Interface Framework* > *Interface Development* > under the following:

- *IDoc Structure Generator*
- *Mass IDoc Structure Generator*

More Information

For more information about monitoring and processing IDocs in the SAP Application Interface Framework, see IDoc Scenarios [page 130].

4.3.1.5 Batch Input Structure Generation and Interface Definition

Use

The *AIF Batch Input Structure Generator* report (transaction code `/AIF/BDC_GEN`) enables you to create a structure for a batch input recording. The report provides optional functionality to create a corresponding SAP Application Interface Framework interface that uses the newly created structure. The structure generated by the report can be used in SAP Application Interface Framework interface definition as a raw and an SAP structure for the following:

- Displaying batch input using the standard batch input runtime
- Displaying batch input using the AIF enabler

Features

The report enables you to create a structure for a batch input recording created by a specific user.

The report gives you the option of creating a corresponding SAP Application Interface Framework interface definition together with the structure. If an interface definition is generated, the report also generates custom field labels that are displayed in *Monitoring and Error Handling*.

The report also enables you to transport the created objects.

Activities

You can access the *AIF Batch Input Structure Generator* from the SAP Easy Access menu by choosing *Cross-Application Components* > *SAP Application Interface Framework* > *Interface Development* > *Batch Input Structure Generator* >.

SAP Structure

To create a structure, you specify the following data:

- The name of the batch input recording
- The name of the creator of the batch input recording
- The name of the structure to be created
- A prefix for use as a prefix for the rest of the structures to be created

Interface Definition

To create a corresponding SAP Application Interface Framework interface definition for the newly created structure, you specify the following data:

- The name of the interface namespace
- The name of the interface
- The version of the interface
- A description for the interface
- The variant ID
  You specify which scenario, of the two supported for batch input (see Batch Input) [page 165], you are using to monitor and process batch input sessions.

> **Note**
>
> The system only allows you to monitor batch input sessions with one transaction. You cannot generate a structure for a batch input session with multiple transactions. The structure that is generated contains a table type that can contain multiple transactions of the same type. In *Monitoring and Error Handling*, you can monitor batch input sessions that contain multiple transactions of the same type and screen flow.

Transport

To transport data, you specify the following data:

- A package where the created structures are to be stored
- A workbench request where the created structures are to be stored
- A Customizing request where the Customizing data is to be stored, for example, the interface definition
- The name of the RFC destination for Customizing handling in another system client

> **Note**
>
> If you enter package $TMP, the workbench objects are assigned to your local objects and it is not necessary to specify a workbench request in this case.

More Information

For more information about monitoring and processing batch input sessions in the SAP Application Interface Framework, see Batch Input [page 165].

4.3.1.6 tRFC and qRFC Function Module Generation

Use

Function Module Generator

Based on a remotely-enabled function module existing in either a local or a remote system, the *Function Module Generator* report supports interface developers in implementing the t/qRFC interface for the SAP Application Interface Framework by generating the following:

- SAP structure
- AIF interface
- Playback function module
  This is called by the *Data Transfer* report to read the data of the t/qRFC transactions.

> **Note**
>
> Authorization checks are performed by the *Function Module Generator* before objects are generated.
>
> The related authorization objects are: `S_DEVELOP` and `/AIF/CUST`.

In this way, with the support of the *Function Module Generator*, the following SAP AIF monitoring functions are available for t/qRFC integration:

- Restart, cancel, and debug of the LUWs as appropriate for tRFC and qRFC transactions
- Display the tRFC and qRFC data
- Display the logs of each tRFC and qRFC transaction

Mass Function Module Generator

The *Mass Function Module Generator* report (transaction code `/AIF/RFC_MASS_GEN`) provides the same functions and features as the *Function Module Generator* report but can be used for the mass generation of objects and Customizing entries based on a list of t/qRFC function modules.

You can define a list of t/qRFC function modules in Customizing for *SAP Application Interface Framework* under *Interface Development* > *Additional Interface Properties* > *Define RFC Function Module Collection*.

On the first screen of the *Mass Function Module Generator*, the prefix of new objects can be input to combine the newly generated SAP structure and playback function module with the short name maintained in Customizing for *SAP Application Interface Framework* under *Interface Development* > *Additional Interface Properties* > *Define RFC Function Module Collection* in the sub-activity *Assign Function to RFC Function Module Collection*.

More Information

For more information, see tRFC and qRFC [page 153].

4.3.1.7 Interface Documentation Maintenance

Use

In the SAP Application Interface Framework, the interface is implemented in a structured way. You can use this structured information to generate documentation that describes the interface structure and contents. In Customizing for the *SAP Application Interface Framework* (transaction code `/AIF/CUST`), under individual Customizing activities for interface creation, you can maintain documentation related to an interface object, for example, an interface or a value mapping.

The *Interface Documentation Tool* (transaction code `/AIF/DOCU`) enables you to write documentation for an interface or for a Customizing object in the language of your choice.

Features

Maintaining Interface Documentation

During interface implementation, interface developers can enter texts to describe an interface object. The structure of the interface and the texts are used to generate an interface documentation.

In Customizing for the *SAP Application Interface Framework*, under individual Customizing activities related to interface creation, there is a *Documentation* pushbutton with a traffic light showing one of the following statuses:

- 
  Documentation is maintained and released
- 
  No documentation is maintained
- 
  Documentation is maintained but not yet released

Choose *Documentation* to maintain and release texts.

A warning is displayed if you do not save your documentation.

Interface Documentation Tool

In the *Interface Documentation Tool*, if you select *Interface* and execute the report, you get an overview of all objects belonging to the specified interface for the specified namespace. You can see if documentation is available and if any such documentation is available in the selected target language. Selecting a line enables you to create, edit, and release documentation.

If you select *Customizing Object*, the Customizing objects that fit the selection criteria are displayed. The following is a list of the possible Customizing objects:

- Interface definition
- Structure mapping
- Fix value
- Value mapping

- Action
- Check
- Field mapping
- Condition

You can export the documentation, for example, to a Microsoft Office Excel file.

Activities

You can access the *Interface Documentation Tool* from the SAP Easy Access menu by choosing *Cross-Application Components* > *SAP Application Interface Framework* > *Interface Development* > *Interface Documentation Tool*.

4.3.1.8 Status Handling

Use

In the SAP Application Interface Framework, handling of statuses (for example, in development, test, productive, obsolete) is available for interfaces.

Integration

You can make settings for status handling in Customizing for *SAP Application Interface Framework* (transaction code `/AIF/CUST`) under *Interface Development* > *Define Interfaces*.

Features

The interface status and end date are checked and, if the interface in status handling is client dependent, are compared with the system role in Client Administration.

In Customizing for *SAP Application Interface Framework* under *Define Interfaces*, you can do the following:

- Enter the status
  For example, *Test* or *Obsolete*
- Enter the end date
- Select the *Automatic Client Control* checkbox
  Indicates that status handling is client role dependent

> **Note**
>
> In order to set up a client dependent interface, you have to maintain the system role in Client Administration (transaction code SCC4).

Example

- The status of the interface is *Obsolete*, the end date is later than the current date, and all system roles are valid
  You receive a warning message that the interface is obsolete and that it is only valid until the end date is reached.
- The status of the interface is *Obsolete,* the end date is earlier than the current date, and all system roles are valid
  You receive an error message and the processing of the current message is stopped.

4.3.1.9 Business Rules Framework Plus Integration

Use

The SAP Application Interface Framework provides an integration with the Business Rules Framework plus (BRFplus), which you can access using the BRFplus transaction code. BRFplus is a rule tool that provides a comprehensive application programming interface (API) and user interface (UI) so that you can define and implement technical and business process-oriented sets of rules.

Integration

A function is the rule interface in BRFplus and it acts as a link between the application code and the BRFplus code. A function carries a context and a result. It imports the context and returns a result after processing.

In the SAP Application Interface Framework, the BRFplus integration is done with a function that you define for value mapping, structure mapping, and checks. In Customizing for the SAP Application Interface Framework (transaction code /AIF/CUST) under *Interface Development*, you specify the name of the BRFplus application (in the *BRFplus Application Name* field) and the name of the BRFplus function (in the *BRFplus Function Name* field) under the following:

- *Define Value Mapping*
- *Define Structure Mapping* > *Assign Destination Structure*
- *Checks* > *Define Single Checks*

Features

During data processing in the SAP Application Interface Framework, a BRFplus function, which is assigned to a value mapping, structure mapping, or check, is called. The data from the SAP Application Interface Framework is transferred to the BRFplus function. The BRFplus function is executed and the result is returned to the value mapping, structure mapping, or check.

Activities

In order to use a BRFplus function in the SAP Application Interface Framework, the function needs to be set up in the following specific ways:

Structure Mapping

A function for structure mapping within an application can have the following context:

Component Name in BRFplus Function	Description
`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 BRFplus mapping function is currently executed
`SENDING_SYSTEM`	A field of the source structure needs to be maintained in *Field for the Sending System* in the interface definition. If the BRFplus function should behave differently for different senders, you can use this parameter to differentiate the behavior.
`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

You can define the following objects as result data objects:

- `OUT_STRUCT`
- `DEST_LINE`
- `DEST_TABLE`

Value Mapping

A function for a value mapping within an application needs to be defined with the following context:

Component Name in BRFplus Function	Description
• VALUE_IN • VALUE_IN2 • VALUE_IN3 • VALUE_IN4 • VALUE_IN5	The fields defined in the field mapping (*Fieldname 1* to *Fieldname 5*) are used as importing parameters.
SENDING_SYSTEM	A field of the source structure needs to be maintained in *Field for the Sending System* in the interface definition. If the BRFplus function should derive different values for different senders, you can use this parameter to differentiate the behavior.

Furthermore, one result data object, with a name of your choice, is required. The resulting value of the BRFplus function is passed back to the SAP Application Interface Framework. The value is passed to the corresponding field in the destination structure.

Check

A function for a single check within an application can have the following context:

Component Name in BRFplus Function	Description
DATA_STRUCT	Contains the data of the current structure. If 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* checkbox in the check assignment. If 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 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*.
• VALUE1 • VALUE2 • VALUE3 • VALUE4 • VALUE5	Contains the values of the fields defined when the check is assigned (*Fieldname 1* to *Fieldname 5*).

Component Name in BRFplus Function	Description
SENDING_SYSTEM	A field of the source structure needs to be maintained in *Field for the Sending System* in the interface definition. If the BRFplus function should behave differently for different senders, you can use this parameter to differentiate the behavior.

One result data object can be added. This result data object should simply indicate if the result of the function is successful or if it has an error.

4.3.1.10 Configuring AIF Interfaces for Monitoring Synchronous Outbound Messages

SAP Application Interface Framework's improved integration into Proxy/Web service technology makes it possible to monitor synchronous messages within one interface.

Setting up an SAP Application Interface Framework interface to monitor synchronous outbound Web service or PI messages includes the following steps:

1. Generating the synchronous proxy service
2. Checking the message type in the consumer proxy
3. Defining the metadata repository namespace
4. Creating a local metadata repository data type for the output message type
5. Creating a local metadata repository data type for the input message type
6. Creating a local metadata repository data type for monitoring
7. Creating an SAP Application Interface Framework interface
8. Creating a test data report for outbound
9. Generating test data
10. Monitoring data in AIF Error Handling */AIF/ERR*

For a step-by-step example using a sychronous PI interface to demonstrate the monitoring of SAP Application Interface Framework, see the following blog on the SAP Community: Monitoring Synchronous Outbound Messages with SAP Application Interface Framework.

4.3.2 Analyzer

Use

In the SAP Application Interface Framework, the *Analyzer* assists you with the process of analyzing the steps that are executed at runtime. It shows you the order in which Customizing settings are executed and helps you locate errors and the origin of incorrect mapping.

The *Analyzer* supports you in finding the root cause of incorrect values. It allows you to analyze how the values have changed during runtime. Furthermore, it helps you to understand how the runtime works.

Integration

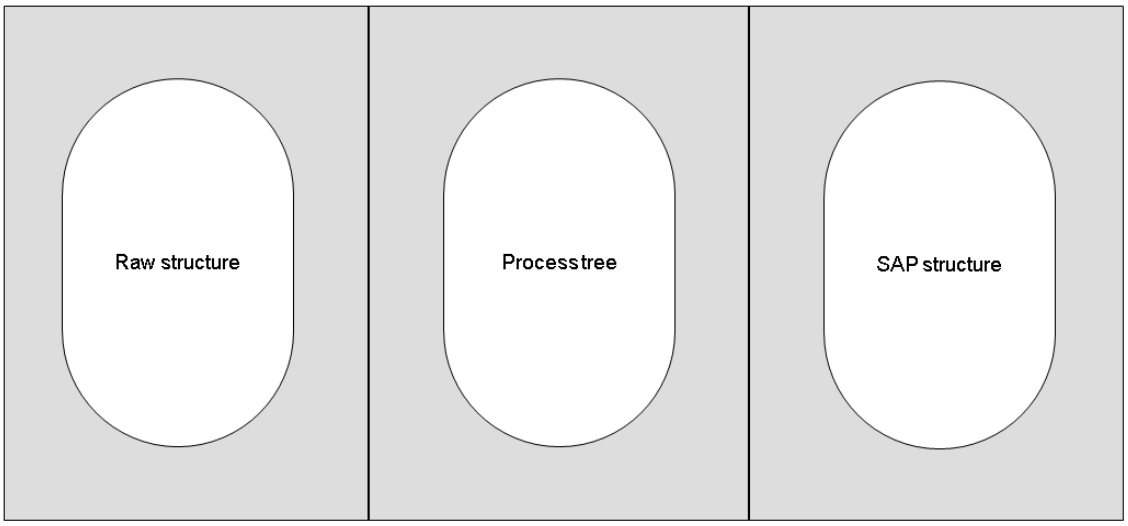
You access the *Analyzer* in the *Interface Test Tool* (from the SAP Fiori launchpad or transaction code /AIF/IFTEST) where you can create test data or reuse existing test data for the interface you have selected for analysis (see Interface Test Tool [page 178]).

In the tool bar of the *Test File Overview* screen, select *Read Data* to trigger the runtime with the transformation of the data and processing of the defined actions. When the runtime is finished, the *Process Test Files* screen of the *Analyzer* is displayed.

> **Note**
>
> When opening the Analyzer by choosing *Analyze*, the last executed step is selected showing the corresponding raw and SAP structure values in the left and right hand views respectively. Thus, you see all values at their state at the end of the process.

Features

Screen Areas



The Process Test Files Screen of the Analyzer

The following three areas are displayed on the *Process Test Files* screen of the *Analyzer*:

- Raw structure on the left
  - *RAW Structure*
    Shows the data of the raw structure at the selected process step

- *Value*
  Shows the value before the selected process step was executed
- Process tree in the center
  - *Process Step*
    Shows the sequential process steps of the specified interface, displayed in the order they had been executed in the runtime
    Each process step corresponds to a Customizing setting. The process steps are grouped according to their hierarchy level and are displayed as a tree. Each step shows how the runtime processed the interface by transforming the data and performing actions.
  - *Result*
    Shows the result of a specific process step, for example, whether or not a check or a value mapping was successful, or a fix value that was added.
- SAP structure on the right
  - *SAP Structure*
    Shows the data of the SAP structure at the selected process step
  - *Value*
    Shows the value after the selected process step is executed

Highlighted Nodes

The selected process step and the corresponding pre- and post-runtime values are highlighted in the following colors:

Yellow	The selected process step.
Orange	If available, the fields and values used for the mapping or the check. Depending on your settings the fields can be in both, the SAP and the RAW structure.
Blue	If available, the fields and values that were changed or set by the mapping, check, or fixed value.

Activities

You can start at any node in the process or the structure areas to analyze different aspects of the interface processing, for example:

- **Retrace the process steps**
  In the *Process Step* area, you can jump back to any step by double-clicking on the node. This updates both the raw and SAP structure trees and highlights the fields and values involved in this step.
  You can make transparent what happened in which sequence and correct, for example, the order of your process steps.
- **Show where a field is used**

  In the *RAW Structure* and *SAP Structure* areas, you can select a field and choose  . This highlights the process steps that will transform data from this field, and the field that will receive the transformation result.

You use this function to take a look forward in the history of a value. For example, to find out why a value that you see in the raw structure did not find its way into the right field in the SAP structure.

- **Explain the changes made to a field**

  In the *RAW Structure* and *SAP Structure* areas, you can select a field and choose  . This highlights the process steps that transformed the data in this field and the fields that delivered the input values for the transformation.
  You use this function to take a look backwards in the history of a value. For example, to find out how it came to an incorrect value that you see in the SAP structure.

4.3.3 Interface Determination

Use

In SAP Application Interface Framework, interface determination enables you to configure which interface should be used to process a data message. Interface determination is available for inbound messages for different interface technologies, for example, coming from the SAP NetWeaver PI proxy or sent via IDocs or XML. You can also develop a customer-specific interface determination.

You can use interface determination to decide which interface should be used for message processing. Different *Field Categories* exist, for example, the content of the data message, IDoc control records, or fields from the XI header data. Which *Field Categories* are available depends on the interface technology used. Interface determination also determines if the message is a restart message (the SAP Application Interface Framework interface used while restarting the message should be the same as the one used for the initial processing of the message) or if the message is new.

SAP Application Interface Framework provides version handling for the Customizing activity *Interface Determination*.

Integration

In Customizing for *SAP Application Interface Framework* (transaction code `/AIF/CUST`), you can make settings for interface determination in the following activities under ▶ *System Configuration* ▶ *Interface Determination* ▶ as appropriate:

- *Define Interface Determination for Proxy Interfaces*
- *Define Interface Determination for IDoc Interfaces*
- *Define Interface Determination for XML Interfaces*
- *Define Interface Determination for Customer-Specific Engines*

Prerequisites

Interfaces are maintained in Customizing for *SAP Application Interface Framework* (transaction code `/AIF/CUST`) under *Interface Development* > *Define Interfaces*.

Features

In SAP Application Interface Framework, interface determination is based on the following rules:

- Standard
  - Proxy class
    The name of the proxy class that handles the data message is compared to the Customizing settings for the interface. If there is an interface that has the proxy class name as its inbound proxy class name, then the highest version of this interface is used to process the message.
    If the inbound/outbound proxy class has multiple operations, you need to maintain an appropriate entry in the *Proxy Method* field in Customizing for SAP Application Interface Framework under *Interface Development* > *Define Interfaces*. In addition, you have to maintain the *Proxy Method* field in *Interface Determination for Proxy Interfaces* if more than one interface is assigned to the proxy method. If the inbound/outbound proxy class has only one operation, you can leave the *Proxy Method* field empty.
  - IDoc
    If a data message is sent via IDoc, the basic type and the message type is compared to the settings for the interface in Customizing for *SAP Application Interface Framework* (transaction code `/AIF/CUST`) under *Interface Development* > *Additional Interface Properties* > *Assign IDoc Types*. If there is an interface that has the same basic type and message type, then the highest version of this interface is used for processing the messages.
  - XML
    If a data message that is sent as XML, the name of the input structure is compared to the raw data structure in the Customizing settings for the interface. If there is an interface found, then the highest version of this structure is used.
- Non standard
  Instead of using the standard behavior, you can explicitly customize interface determination.

Activities

Version Handling

You can define a validity period for the interfaces and additional checks for verifying the right interface definition.

In Customizing for *SAP Application Interface Framework* under *Interface Determination*, to maintain the validity period, you can enter the *Date From*, the *Date To*, both of these dates, or the validity period. You can only maintain the dates or the validity period but not both.

More Information

For more information about Customizing for interface determination and version handling, see the corresponding documentation for the individual activities in Customizing for *SAP Application Interface Framework*.

4.3.4 Defining Recipients

Use

In SAP Application Interface Framework, you use **recipients** to define which users are responsible for which data messages. The recipient assignment of a user defines which data messages he will see in the Interface Monitor and for which data messages he will receive alert notifications, for example, emails. Using recipients, you can make sure the data messages reach the right persons in your organization, and only the right persons.

To enable the routing of data messages, you create recipients, assign interfaces to the recipients, and assign the recipients to users or user groups. The creation of recipients and their assignment to interfaces is described here. For information about assigning recipients to users or user groups, see Assigning Users to Recipients [page 95].

Features

When you create a recipient, it is not assigned to an interface. For assigning interfaces to your recipients, you can apply the following strategies:

> → Recommendation
>
> We recommend that you first make yourself familiar with the concepts, assignment options, and recipient types described here. For complex scenarios, you should outline a recipient matrix before you start creating and assigning recipients in the system.

By interface

As a basic strategy, to make sure that no message slips through without anybody being notified, you can set one recipient per interface as the **default recipient**. If the system cannot find any other recipient that receives a notification, a notification is always sent to the default recipient regardless of the alert settings.

In a simple business scenario, you can assign **specific recipients** to one or several interfaces. These recipients will receive all messages of the interface(s) according to their alert and monitoring settings.

> ⚠ Caution
>
> Other than the default recipients, you cannot combine interface-specific recipients (“Recipients Without Key Fields”) with message category or key field recipients.

By Message Categories

You can group log messages in message categories and assign the categories to recipients using an assignment table. You can assign **specific recipients** who will be notified about a data message if a log message in their category was thrown for this data message. Usually, you map such recipients to users or groups that are responsible for handling specific errors in the business content of the data messages.

For central administration users, who are responsible for monitoring a whole business process and need to have an overview of the entirety of (error) messages, you can add a **standard recipient**. This recipient will get all messages of the interface regardless of their category. In addition, to track which log messages are thrown but not assigned to any specific recipient, you can create a **fallback recipient**. The fallback recipient will get only those messages that are not covered by one of the specific recipients.

Example

During the processing of an interface Financial Accounting (FI) and Controlling (CO) errors might occur. Depending on the error type (FI or CO), different users are responsible for handling these errors. To dispatch the error messages, you create two message categories (FI and CO) and two corresponding recipients (FI_RECIPIENT and CO_RECIPIENT). You assign the users responsible for the FI-related errors to recipient FI_RECIPIENT. These users will only be notified if there is an FI-related error and in the Interface Monitor they will only see the FI-related messages. Do the same for the CO_RECIPIENT.

To collect the log messages that go past the **specific recipients** FI_RECIPIENT and CO_RECIPIENT, you create a **fallback recipient** (FALLBACK). This recipient will get all messages with error types other than FI or CO. In addition, you create a **standard recipient** (STANDARD). This recipient gets all log messages of the interface, independent of their message category.

The following table shows how incoming messages are dispatched to the different types of recipients at runtime:

Message GUID	Message Category	Specific Recipients	Fallback Recipient	Standard Recipient
3433	FI	FI_RECIPIENT		STANDARD
3983	CO	CO_RECIPIENT		STANDARD
3457			FALLBACK	STANDARD
3953	CO	CO_RECIPIENT		STANDARD
3567			FALLBACK	STANDARD

By Key Field Values

You can route the log messages based on the content of the data messages using an assignment table. You can define the key fields of a single index table as recipient-relevant and create **specific recipients** for certain values of the key field. If an error message is thrown for a data message containing this key field value the corresponding recipient gets notified. Usually, you assign these recipients to users or groups that are responsible for handling specific errors in the business content of the data messages.

For central administration users, who are responsible for monitoring a whole business process and need to have an overview of the entirety of (error) messages, you can add a **standard recipient**. This recipient will get all messages of the interface regardless of the key field values. In addition, to track which messages are not assigned to any specific recipient, you can create a **fallback recipient**. The fallback recipient will get only those messages that are not covered by one of the specific recipients. This way, you can track, for example, if

someone entered an incorrect value in your key field or someone introduced a new correct value, for which you need to create a new recipient.

Example

In an interface, data messages from different subsidiaries of your company are processed. Dependent on the company code, different business users are responsible for the log messages in case of errors. One group is responsible for company code 1000 and the other group for company code 2000. You define the company code as a single key field and mark it as relevant for recipient determination. You create the **specific recipients** COMPANY_CODE_1000 and COMPANY_CODE_2000 and assign them to the responsible user groups.

To collect the log messages that go past the **specific recipients** COMPANY_CODE_1000 and COMPANY_CODE_2000, you create a **fallback recipient** (FALLBACK). This recipient will get all messages with company codes other than 1000 and 2000. In addition, you create a **standard recipient** (STANDARD). This recipient gets all messages of the interface, independent of the company code value.

The following table shows how incoming messages are dispatched to the different types of recipients at runtime:

Message GUID	Company Code	Specific Recipients	Fallback Recipient	Standard Recipient
2443	1000	COMPANY_CODE_1000		STANDARD
2986	2000	COMPANY_CODE_2000		STANDARD
2417	4000		FALLBACK	STANDARD
2993	2000	COMPANY_CODE_2000		STANDARD
2547			FALLBACK	STANDARD

By Message Categories and Key Field Values

You can combine these assignment strategies to implement a complex recipient matrix. In a single assignment table, you can assign message categories and key field values to recipients. You can have **specific recipients** and **fallback recipients** on different levels, which is, per category, per field value, and per category/value combination. And you can have **standard recipients** per category, per field value, and for all messages of the interface.

Example

To dispatch your messages, you want to use both, the company code key field values and the FI/CO message categories. For example, you want to use **specific recipients** to target users responsible for the controlling-related log messages of subsidiary 1000 only, users responsible for financial accounting messages of subsidiary 2000 only, and so on. To track all messages, you need **standard recipients** and **fallback recipients** on the different levels.

Your recipient assignment table could look as follows (simplified):

Recipient	Message Category	Company Code	Fallback
FI_1000	FI	1000	
FI_2000	FI	2000	
FI_STANDARD	FI		
FI_FALLBACK	FI		X
CO_1000	CO	1000	
CO_2000	CO	2000	
CO_STANDARD	CO		
CO_FALLBACK	CO		X
STANDARD			
FALLBACK			X

The following table shows how incoming messages are dispatched to the different types of recipients at runtime:

Message GUID	Message Category	Company Code	Specific Recipients	Fallback Recipients	Standard Recipients
4812	FI	1000	FI_1000		FI_STANDARD STANDARD
4815	FI	5000		FI_FALLBACK	FI_STANDARD STANDARD
4815		5000		FALLBACK	STANDARD
4564	CO	2000	CO_2000		CO_STANDARD STANDARD
4431	CO			CO_FALLBACK	CO_STANDARD STANDARD

Activities

Creating Recipients and Specifying Default Recipients

> **Note**
>
> Perform the following steps for every strategy.

1. In Customizing of SAP Application Interface Framework, create all required recipients under *Error Handling* > *Define Namespace-specific Features* > *Define Recipients*.
2. Specify one default recipient per interface under *Error Handling* > *Define Namespace-specific Features* > *Configure Alerts* > *Default Recipient*.

Assigning Recipients to Interfaces

> **Note**
>
> Perform the following step only if you want to have interface-specific recipients but no message category or key field recipients.

Assign recipients to interfaces under *Error Handling* > *Define Interface-specific Features* > *Assign Recipients Without Key Fields*.

Make sure that there is no entry in *Error Handling* > *Define Namespace-specific Features* > *Configure Alerts* > *Recipient Assignment Table*. If there is an entry, the system uses the assignment table and ignores the setting in *Assign Recipients Without Key Fields*.

Assigning Recipients to Message Categories

1. In the ABAP dictionary, create a recipient assignment table by copying the template table `/AIF/RECA_TMPL`.
2. Assign the table to your interface under *Error Handling* > *Define Namespace-specific Features* > *Configure Alerts* > *Recipient Assignment Table*.
3. In transaction `/AIF/MSG_CAT_DEF`, create your message categories.
4. In transaction `/AIF/MSG_CAT_ASGN`, assign messages to your message categories.
5. In your recipient assignment table, enter the message categories, map specific recipients to the categories, and enter a fallback recipient.

Assigning Recipients to Key Fields

1. In the ABAP dictionary, create a single index table by copying the template table `/AIF/STD_IDX_TBL` and add the required key fields.
2. Assign the single index table to your interface under *Error Handling* > *Define Namespace-specific Features* > *Define Interface-Specific Features* > *Message Idx Table*
3. Add your key fields under *Error Handling* > *Define Interface-specific Features* > *Define Key Fields for Multi. Search* and select *Relevant for Recipient Determination*.
4. If you want to have a key field name in your recipient assignment table that is different from the name in the single index table, enter the different key field name in *Field Name in Alert Recipient Assignment Table*. If the field names are the same, leave this field empty.
5. In the ABAP dictionary, create a recipient assignment table by copying the template table `/AIF/T_ALRT_DEF`.
6. Assign the recipient table to your interface under *Error Handling* > *Define Namespace-specific Features* > *Configure Alerts* > *Recipient Assignment Table*.

7. In your recipient assignment table, add the key fields, map specific recipients to values of the key fields, and enter a fallback recipient.

Assigning Recipients to Message Categories and Key Fields

To combine both strategies, you need to do all the steps described in both sections above. For the combined recipient assignment table, use template `/AIF/RECA_TMPL` and add your key fields to the table.

4.3.5 Assigning Users to Recipients

Use

In the SAP Application Interface Framework, there are two key approaches that you can take to assign users and recipients, namely, a user-centric and a recipient-specific approach.

In Customizing for the *SAP Application Interface Framework* (transaction code `AIF/CUST`), you can assign interface-specific authorizations to restrict or expand users' activities.

Features

User-Centric Approach

For a user-centric approach for assigning users to recipients, use *Assign Recipients* (from the SAP Fiori launchpad or transaction code `/AIF/RECIPIENTS`). In the SAP GUI, you can also access it from the SAP Easy Access menu by choosing *Cross-Application Components* > *SAP Application Interface Framework* > *Administration* > *Configuration* > *Recipients of a User*.

This gives you an overview of the assignments of a user to recipients and enables you to make new assignments.

For each recipient, you can specify a message type that defines which kind of message, for example, *Application Error* or *Warning*, is included in an alert.

Set the *Include in Overview Screen* indicator to specify that the user can see the message of this recipient in the *Interface Monitor*.

Set the *Technical User* indicator to specify that the user can monitor messages with the status *In process* and messages with technical errors on the *Interface Monitor* screen.

You can also view the recipients for the current user from the SAP Fiori launchpad (*My Recipients* or, in the SAP GUI, from SAP Easy Access menu by choosing *Cross-Application Components* > *SAP Application Interface Framework* > *Recipients of Current User*.

Recipient-Specific Approach

The second approach offers a recipient-specific view for assigning a user to recipients and you access it in Customizing for the *SAP Application Interface Framework* under *Error Handling* > *System Configuration* > *Recipients* > *Assign Recipients*. You can see which users are assigned to which recipient and you can assign users, roles, and external contacts to a recipient.

For each assignment, you can specify a message type and set the *Include in Overview Screen* and *Technical User* indicators.

You can manage external contacts in Customizing for the *SAP Application Interface Framework* under *Error Handling* > *System Configuration* > *Recipients* > *Define External Contacts*.

4.3.6 Automatic Reprocessing

Use

In the SAP Application Interface Framework, the error handling process can perform automatic reprocessing. When triggered by error messages you have registered, the SAP Application Interface Framework automatically generates a background job to reprocess them.

Prerequisites

In the *SAP Easy Access* menu under *SAP Application Interface Framework* > *Administration* > *Configuration* > *Runtime Configuration Group* (transaction code `/AIF/PERS_CGR`), you have set up the runtime configuration group to define the processing of a set of messages by the SAP AIF runtime environment. You have selected the *Runtime Configuration Group Active* and *Run Scheduled* checkboxes here.

Features

In the SAP Application Interface Framework, a reprocessing action is the technical representation of a function module that is called by a batch job. A reprocessing action contains the appropriate AIF runtime configuration group and the function module that is called by the reprocessing job.

The AIF runtime schedules a reprocessing action for registered error messages according to your settings in the following configuration tables:

- *AIF Automatic Reprocessing: Define Reprocessing Action* (transaction code `/AIF/REP_AC_DEF`)
- *AIF Automatic Reprocessing: Assign Reprocessing Action* (transaction code `/AIF/REP_AC_ASGN`)

The SAP Application Interface Framework delivers the `/AIF/RESTART_MSG` function module to automatically restart registered error messages.

You can implement your own automatic reprocessing actions depending on your requirements. You can use the delivered `/AIF/TEMPL_RESTART_AUTO_REPR` template function module to call custom-specific automatic reprocessing actions.

Activities

Define Reprocessing Action

In the SAP Easy Access menu under *Cross-Application Components* > *SAP Application Interface Framework* > *Administration* > *Configuration* > *Define Reprocessing Action*, you set up a reprocessing action by defining the AIF runtime configuration group and the function module that is called by the reprocessing job.

Assign Reprocessing Action

In the SAP Easy Access menu under *Cross-Application Components* > *SAP Application Interface Framework* > *Administration* > *Configuration* > *Assign Reprocessing Action*, you register the AIF interface and the corresponding single error messages by message class and number for automatic reprocessing. You also assign a reprocessing action and you define the following:

- A minimum and a maximum waiting time

  > **Note**
  >
  > - If there is an existing job scheduled between the minimum waiting time and the maximum waiting time, the message is assigned to this job.
  > - If there is no suitable job available, a new job is created with the maximum waiting time as starting point.

- A maximum number of reprocessing trials
- Status
  You can select one of the following intermediate statuses that the message should reach while it is waiting for the next automatic reprocessing:
  - *I*
    In process
  - *E*
    Application error
  - *A*
    Technical error

  > **Note**
  >
  > This is the status that the message has in the SAP Application Interface Framework; the status in the standard monitor can differ.

Example

For automatic reprocessing, you registered an AIF interface and the corresponding single error message by message class (`R1`) and number (`084`), for example, `R1/084` = *Business Partner &1 is Currently Being Processed By &2*. You assigned an action, a waiting time of 5 minutes, a maximum of 3 reprocessing trials, and a status *I* (in process).

The business partner is locked and a job is triggered to restart this data message after 5 minutes, up to 3 times, and with the status *I* (in process) while waiting for the next reprocessing.

4.3.7 Process Observer

Use

The *Process Observer* is a standard application that enables you to manage and optimize your business processes across different applications and along the entire process management lifecycle. To enable a cross-system monitoring of message flow for business users, the *Process Observer* is used to store all messages in the context of the corresponding business process.

In the SAP Application Interface Framework, you can set up your interfaces to be monitored by the *Process Observer*. If the processing of a message in an interface is part of a business process defined in the *Process Observer*, you can make Customizing settings to be able to monitor the interface in the *Process Observer*, which provides you with an overview of the current status of a business process.

Integration

To be able to monitor messages in the *Process Observer*, you make the relevant configuration settings in transactions `POC_FACADE` (business objects and tasks definition) and `POC_MODEL` (business process definition).

To be able to use the *Process Observer* with the SAP Application Interface Framework, you make the relevant settings in Customizing for *SAP Application Interface Framework* under *System Configuration* > *Configure Interfaces for Process Observer*.

Prerequisites

You have at least one current or remote system with the *Process Observer* in your system landscape. The *Process Observer* must be active and configured correctly.

> **Note**
>
> Monitoring with the *Process Observer* is only available with *NetWeaver Business Suite Foundation* `731`, `702 SP06`, `701 SP11`, and higher releases and support packages.

You have a defined a corresponding RFC destination (the SM59 transaction code) if the *Process Observer* is on the remote system.

To be able to monitor messages in the *Process Observer*, you have made the relevant configuration settings in transactions `POC_FACADE` (business objects and tasks definition) and `POC_MODEL` (business process definition) as follows:

- In transaction `POC_FACADE`, you have defined the business object types that are used to monitor interfaces.
  A business object type is a representation of a uniquely identifiable business entity, for example, a sales order or a purchase order. You must define a business object type for the message of the SAP Application

Interface Framework. In order to pass the namespace, interface name, and interface version to the *Process Observer* you can also define a business object type for the interface.

> **→ Recommendation**
>
> We recommend that you create the `AIFME` (*AIF Message*) and `AIFIF` (*AIF Interface*) business object types as these are the default values used in the SAP Application Interface Framework.

> **ⓘ Note**
>
> You can also create business object types of your choice.

- In transaction `POC_FACADE`, you also have defined the task types, which represent the business object activity of the implementation layer in the facade of the *Process Observer*.

  > **Example**
  >
  > Task type `AIF1` (*Received in AIF (Start Event)*) specifies that this message is from the first step or "start event" at the start of the business process defined in the *Process Observer*. It also tells you that there is no preceding step in the business process, no preceding event has been raised in the *Process Observer*, and no preceding message has been received in the SAP Application Interface Framework.

  > **→ Recommendation**
  >
  > We recommend that you create the task types that are used as default values in the SAP Application Interface Framework (see the section on *Configuration of the Process Observer Using the POC_FACADE Transaction* below).

  > **ⓘ Note**
  >
  > You can also create task types of your choice.

- In transaction `POC_MODEL`, you have defined a business process.
  You assign tasks to the activities of your process. Each task is assigned to a business object type. Each activity should have a task for the business object type "message".
  The message is the main task. If you want to be able to search for the interface and a business key (for example, business partner ID) in the *Process Observer*, you can also create tasks for those business object types.

  > **→ Recommendation**
  >
  > In Customizing for *SAP Application Interface Framework* under *System Configuration* > *Configure Interfaces for Process Observer* , indicate that an interface or any other business key is an *Additional Task*.

  > **ⓘ Note**
  >
  > For business keys, you can also use an existing business object type and task from `POC_FACADE`.

Activities

Configuration of the Process Observer Using the POC_FACADE Transaction

In the *Process Observer*, in transaction POC_FACADE, you define the following business object types, task types, and tasks for use with the SAP Application Interface Framework:

Business Object Types

Business Object Type	Business Object
AIFME	*AIF Message*
AIFIF	AIF Interface

Task Types

Task Type ID	Task Type
AIF1	*Received in AIF (Start Event)*
AIF2	*Received in AIF (Previous Step)*
AIF3	*Restart in AIF*
AIF4	*Processed Successfully in AIF (Not Finished)*
AIF5	*Processed Successfully in AIF (Finish Event)*
AIF6	*Processed with Errors in AIF*
AIF7	*Canceled in AIF*
AIF8	*Assigned to AIF Interface*

Tasks

Business Object Type	Task Type ID	Task
AIFIF	AIF8	*Assign Interface*
AIFME	AIF1	*Processed in AIF (Initial)*
AIFME	AIF2	*Received in AIF*
AIFME	AIF3	*Restarted in AIF*
AIFME	AIF4	*Forwarded by AIF*
AIFME	AIF5	*Finished in AIF*

Business Object Type	Task Type ID	Task
AIFME	AIF6	*Error in AIF*
AIFME	AIF7	*Canceled in AIF*

> **Note**
>
> You can also create your own business object types and task types.

Customizing for the SAP Application Interface Framework

In order to use the *Process Observer* with the SAP Application Interface Framework, you can make the following settings in Customizing for *SAP Application Interface Framework* under ▶ *System Configuration* ▶ *Configure Interfaces for Process Observer* ▶:

- *Configure Process Observer*

  > **Note**
  >
  > If you maintain entries for the *Configure Process Observer* sub-activity and the interface keys (namespace, interface name, and version) are empty, these entries are regarded as the default. These settings are used when there is no interface-specific setting.
  >
  > Furthermore, if you leave the fields for event types and business object types empty, the system tries to use the defaults described above.

  For an interface you want to be monitored by the *Process Observer*, specify the following:

  - The different event types, for example, *Initial Event Type* and *Finished Event Type*, which are forwarded to the *Process Observer* while a message is being processed
    If you leave these fields empty, default event types are used.
  - The business object type to be used for the interface of the SAP Application Interface Framework
    If you leave this field empty, the default type `AIFIF` (*AIF Interface*) is used.
  - The business object type that is used to monitor a message in the *Process Observer*
    The message is the main object to be monitored by the *Process Observer*. The initial, start, restart, continue, finished, error, and cancel event types are raised to the *Process Observer* with this business object type.
    If you leave this field empty, the default business object type `AIFME` (*AIF Message*) is used.
  - The RFC destination, if the *Process Observer* is on a remote system

- *Configure Interfaces for the Process Observer*
  For an interface you want to be monitored by the *Process Observer*, specify the following:
  - The key field from the single index table that contains information about the preceding business object ID, for example, the message GUID or the IDoc number.
    For a message of an interface that is not the first step in a process defined in the *Process Observer*, the ID of the preceding business object has to be passed to the *Process Observer* to enable it to link the current message to the correct process instance.
  - Whether or not the interface is the last step within a business process
    If you select the *Process Finished* checkbox, you ensure that the process instance is marked as finished in the *Process Observer* (transaction `POC_MONITOR`). If a message in this interface is the last step in

a business process and it has been processed successfully, the process is completed and the finished event is raised.

- Whether or not the events are raised when a message is processed in the runtime
  If you select the *Raise POC Events in Runtime* checkbox, events for the *Process Observer* are forwarded while the message is processed in the runtime of the SAP Application Interface Framework.
- Whether or not the events are raised when a message is processed by the enabler
  If you select the *Raise POC Events in Enabler* checkbox, events for the *Process Observer* are raised while the message is processed in the enabler of the runtime of the SAP Application Interface Framework. The enabler is called during message processing to monitor a message in the SAP Application Interface Framework.

  > **Example**
  >
  > - You are using IDocs and you select both the *Raise POC Events in Runtime* and the *Raise POC Events in Enabler* checkboxes.
  > - You are using outbound proxies and you only select the *Raise POC Events in Runtime* checkbox.

- *Assign Key Fields to Additional Tasks*
  You can assign single key fields of your interface (for example, business partner ID) to be passed to the *Process Observer* as an additional task. You are then able to search for those key field values in the *Process Observer*.

  > **Note**
  >
  > Only single key fields can be used.

4.3.8 Custom Objects Administration

Use

System administrators can change existing custom hints, functions, message texts, and data links in the following transactions, which they can access from the SAP Easy Access menu by choosing *Cross-Application Components* > *SAP Application Interface Framework* > *Administration* > *Configuration*:

- *Define Custom Functions* (transaction code `/AIF/CUST_FUNC`)
- *Define Custom Hints* (transaction code `/AIF/CUST_HINTS`)
- *Define Custom Message Texts* (transaction code `/AIF/CUST_TEXT`)
- *Define Custom Data Links* (transaction code `/AIF/CUST_LINK`)

More Information

The relevant activities for custom objects maintenance are described in the *System Administrators* section of Custom Objects Maintenance [page 48].

4.3.9 Parameter Transactions

Use

You are able to access a base transaction for both the error handling and the value mapping transactions. Both transactions can be used to create a parameter transaction. Using a parameter transaction, the base transaction can be called with a certain number of reserved parameters.

Features

Error Handling Base

If you create a parameter transaction for transaction `/AIF/ERR_BASE`, the *Error Handling* transaction can be triggered directly using the created parameter transaction code.

By using the parameter selection, you are able to decide whether or not the selection screen *Monitoring and Error Handling* is shown. It is also possible, for example, to select the message status that is to be displayed to the user. Nine additional selection parameters are also available.

Value Mapping Base

If you create a parameter transaction for transaction `/AIF/VMAP_BASE`, you are able to trigger *Value Mapping* directly using the created parameter transaction code.

By using the parameter selection, you are able to decide whether or not the selection screen of *Value Mapping* is displayed and whether the transaction is to start in display or in change mode.

Activities

You can access the base transactions from the SAP Easy Access menu by choosing *Cross-Application Components* > *SAP Application Interface Framework* > *Administration* > *Base Transactions* and *Error Handling Base* or *Base Transaction for Value Mapping*.

4.4 Monitoring in the SAP Solution Manager

You can monitor SAP Application Interface Framework interfaces using the SAP Solution Manager. As of SAP Solution Manager version 7.2 SPS 5, an integration scenario for SAP Application Interface Framework is available.

In the SAP Solution Manager, you can display the most important staistics for all messages of a specific SAP Application Interface Framework namespace or interface, independently from the interface technology.

Prerequisites

- In the SAP Solution Manager, the SAP Application Interface Framework system(s) and the other source or target systems of SAP Application Interface Framework interfaces are defined.
  The SAP Application Interface Framework system(s) have the SAP Solution Manager agents installed.
  For more information, see http://help.sap.com/solutionmanager.
- The users that SAP Solution Manager uses to access the SAP Application Interface Framework system(s) have the *ALL* recipient assigned.
  For more information, see Assigning Users to Recipients [page 95].

Configuring the Integration Scenario in SAP Solution Manager

1. Open transaction `SOLMAN_SETUP`.
2. In the *Scenarios* pane, select *Application Operations* > *Integration Monitoring* > *Interfaces and Connections*.
3. In the roadmap, select step *4 Define Scope* and choose *Edit*.
4. In the *Scenario Selection* pane, create a new scenario.
5. In *Define Technical Elements*, select an SAP Application Interface Framework system and the source or target systems of your SAP Application Interface Framework interfaces.
6. In the roadmap, select step *5.1 Preparation* and execute all available automatic preparation steps.
7. In step *5.2 Configuration*, in the *Interface channels* pane, create a new channel of type *SAP AIF*.
   Depending on the SAP Application Interface Framework scenario (inbound or outbound) select the SAP Application Interface Framework system and the other system as source or target system.

   > **Note**
   >
   > Always set the measuring point to the SAP Application Interface Framework system.

8. In *Details of interface channel...* > *Interfaces*, add a new interface.
   Enter an SAP Application Interface Framework namespace, and, optionally, a specific interface and version.
   In *Metric configuration*, define for which message statuses you want to see the total numbers.
9. In step *5.3 Activation*, go to the *Interface Channel Activation* pane .
10. In the *Managed Object Type* column, for each object type, select the *Metric Group* icon and define a data collection interval and a threshold.
11. In the roadmap choose *Save*.
12. In the *Interface Channel Activation* pane, choose *Apply and Activate* > *Selected Interface Channel Only*.
13. In step *6 Complete*, choose *Finish*.

Monitoring the SAP Application Interface Framework Interfaces in SAP Solution Manager

1. Open transaction `SM_WORKCENTER`.

2. Choose *System and Application Monitoring* > *Interface Monitoring*.
3. In the *Technical Scearios* pop-up, select your SAP Application Interface Framework scenario.
   You can see a status overview for the messages of your SAP Application Interface Framework interfaces.
4. To see more details, choose your scenario in the *Scenarios* pane and drill down further.

4.5 Transporting AIF Content

Unpack content that has been delivered packed using transaction *AIF Content Transport - Deploy* (`/AIF/CONTENT_EXTRACT`).

Some applications deliver packed AIF content, which can be identified by the *Deployment Scenario*. You can unpack this content using the transaction *AIF Content Transport - Deploy* (`/AIF/CONTENT_EXTRACT`).

During the extraction process, you can assign the unpacked customizing entries to a transport request.

4.6 Serialization

Use

In SAP Application Interface Framework, for some processes, it is required that messages are processed in the correct order. Serialization ensures that SAP Application Interface Framework does not start to process a message until the preceding message, which is related to the same business object, is fully processed.

Messages that are related to the same business object can be handled by different interfaces. If the order in which messages are processed is important, you can set up serialization. Messages related to different business objects do not have to wait for each other to be processed, even though they may be handled by the same interfaces.

In SAP Application Interface Framework, for some processes, the order in which the messages are processed is not important. However, it may be important that the messages are not processed in parallel to avoid locking errors.

Messages that edit business objects that are already locked for editing by another process should wait until the other process releases the lock.

Prerequisites

Key fields must be defined for the interface in Customizing for *SAP Application Interface Framework* under *Error Handling* > *Define Interface-Specific Features*.

Features

There are four options for serializing messages:

- Internal time stamp of the creation of the message in SAP Application Interface Framework
- External time stamp of the sending system
- External index
  If messages are not received in the receiving system, or if messages overtake each other, the time stamp does not indicate that a message is missing. In this case, the sending system has to determine the order in which the messages should be processed and passes this data in a field in the payload of the message to SAP Application Interface Framework. This external index is then used to process the messages in the correct order.

  > → Recommendation
  >
  > If you want to make corrections to the last processed index for a specific business object/key combination, use the *AIF Serialization: Manual Change Of External Index* report (transaction code `/AIF/SERIAL_INDEX`).

- No serialization
  At the beginning of message processing, SAP Application Interface Framework tries to lock the object. If the object is already locked by another message, the current message has to wait until the object is unlocked (you can configure the waiting time and number of retries). Once the message is processed, the lock is released. You can define multiple objects that can be locked per interface of SAP Application Interface Framework.

Activities

In Customizing for *SAP Application Interface Framework* under *Interface Development* > *Additional Interface Properties* > *Define Serialization Settings*, you make the following settings to enable serialization:

- You define a serialization object with the serialization type, serialization table, and other related settings. The serialization type can be one of the following:
  - External index
  - External time stamp
  - Internal time stamp
  - No serialization
    Select *No serialization* when you want to define that certain objects/messages should not be processed in parallel.

  > ⓘ Note
  >
  > Templates for the serialization table are provided.

  > → Recommendation
  >
  > For the serialization table, if a high volume of data can be expected, you should consider the sequence of the table key fields or the use of a dedicated secondary table index.

- You assign the serialization object to one or more interfaces.
- You assign the interface key fields to the serialization object for serialization and locking.

> **Note**
>
> If a serialization object been assigned to more than one interface, note the following:
>
> - These interfaces share the same serialization table and the same index sequence
> - Make sure their serialization key fields are the same (field names in serialization table and total number of serialization key fields)

4.7 Workflow Event Trigger

Use

In SAP Application Interface Framework, you can define message processing workflows that are triggered by events of SAP Business Workflow. You can, for example, have flight booking SAP Business Workflow events in system ABC automatically trigger outbound proxy calls with messages containing the flight details to system XYZ.

The workflow event trigger functions of SAP Application Interface Framework support you in setting up and filling the required structures and interfaces, and setting up the required actions for such a scenario. You do not need to write a single line of code. Using the monitoring and error handling functions of SAP Application Interface Framework, you can control the process flow.

Features

To enable a workflow event trigger scenario, you need to create the following interfaces including mappings and actions:

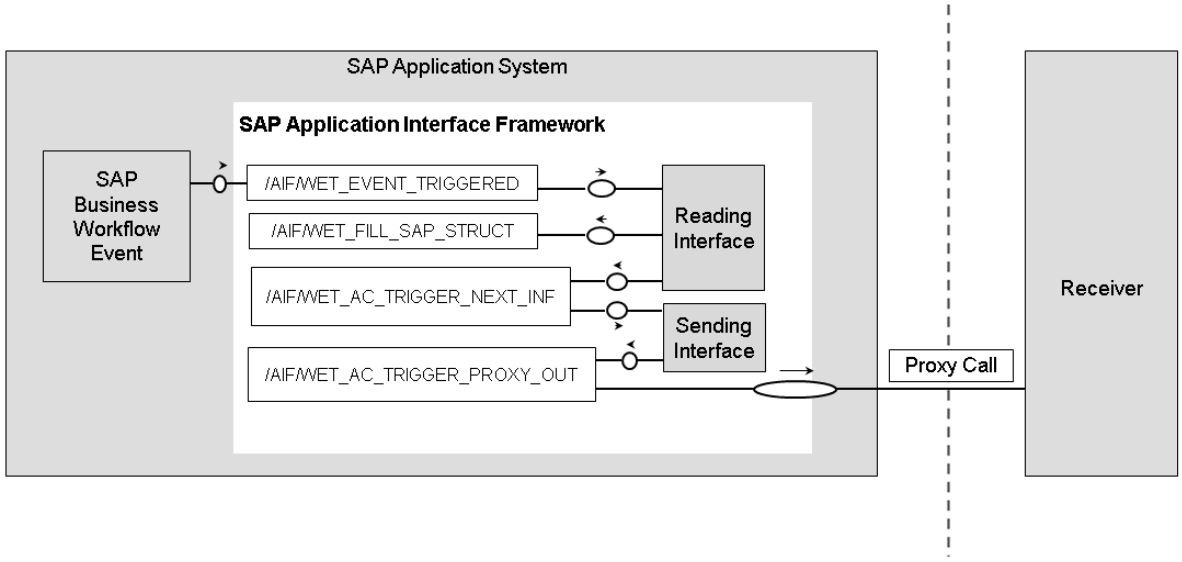
- A **reading interface** for extracting the event details.
- A **sending interface** for feeding the outbound proxy.

We ship the following system objects:

- Several Customizing activities and configuration transactions for setting up the scenario.
- Function modules:
  - `/AIF/WET_EVENT_TRIGGERED`: Event function for receiving the SAP Business Workflow event and triggering the reading interface.
  - `/AIF/WET_FILL_SAP_STRUCT`: Mapping function for filling the SAP structure of the reading interface.
  - `/AIF/WET_AC_TRIGGER_NEXT_INF`: Action function for triggering the sending interface.
  - `/AIF/WET_AC_TRIGGER_PROXY_OUT`: Action function for triggering the outbound proxy call.
- Report `/AIF/WET_GEN` for generating the SAP structure for your reading interface from the function module (BAPI) of your business object.

- Generic DDIC structure `/AIF/WET_EVENT_DETAILS_PROXY`.

The following graphic shows how the created interfaces and the shipped function modules interact, once the scenario is set up:



Workflow Event Trigger

Example

You want flight bookings created by local travel agents in system ABC to be automatically sent to your central system XYZ. If everything is set up correctly, this is what happens at runtime:

1. A local travel agent enters the flight details in system ABC and saves them. This booking activity triggers the SAP Business Workflow event `FlightCreated` of the business object `SBOOK`.
2. The system calls and executes the `/AIF/WET_EVENT_TRIGGERED` function module that is registered for this particular event. The function triggers the processing of the reading interface `WET_FLIGHTBOOK_REC` which has a generic raw structure and is registered for the particular event details of the flight booking event.
3. The reading interface maps the event details, like the event name `FlightCreated` and the object type `SBOOK`, and the event key values, like airline ID and booking number, to the SAP structure, which you have generated based on a BAPI reading flight booking details. In addition, the reading interface reads the flight booking details, like flight date and travel agency number, using this flight booking BAPI, and fills the rest of the raw structure.
4. The reading interface starts an action that triggers the processing of the sending interface `WET_FLIGHTBOOK_SEND` (still in system ABC). This sending interface maps the flight booking details from the generated structure to the output structure of the outbound proxy class method `POST_BOOKINGS01`.
5. The sending interface starts an action that triggers the `POST_BOOKINGS01` proxy method to send out the flight booking details to system XYZ.

Activities

To set up a triggered workflow for the “Read Function and Outbound Proxy” scenario, proceed as follows:

Preparation

In transaction `/AIF/WET_GEN`, from the function module (BAPI) of your business object, generate the SAP structure for your reading interface.

> **Note**
>
> Predefined data types that are not supported in structures used for SAP Application Interface Framework.

Setting up the Event

1. In transaction `SWE2`, link the SAP Business Workflow event type, enter `/AIF/WET_EVENT_TRIGGERED` as the receiver function module, and select *Linkage Activated*.

   > **Note**
   >
   > You can only use events with the object category *BOR Object Type*.

2. In transaction `/AIF/WET_CONFIG`, specify an SAP Application Interface Framework runtime configuration group suitable for filling your reading interface.
3. In Customizing of SAP Application Interface Framework, specify the type of event trigger scenario you want to use for this event under *Interface Development* > *Event Trigger Settings* > *Define Event Trigger*

Setting up the Reading Interface

1. Create the reading interface with the generated BAPI structure as SAP structure and the generic structure `/AIF/WET_EVENT_DETAILS_PROXY` as the raw structure under *Interface Development* > *Define Interfaces*
   Make sure that, for this interface, the *Application Engine* and the *Persistency Engine* are set to *XML* under *Interface Development* > *Additional Interface Properties* > *Specify Interface Engines*.
2. If you use the generic structure `/AIF/WET_EVENT_DETAILS_PROXY` in more than one interface, set up an XML interface determination for each use with the object type, event, and receiver type as selection criteria under *System Configuration* > *Interface Determination* > *Define Interface Determination for XML Interfaces*.
3. Map the event details and the object key to the corresponding fields of the structure generated from the BAPI under *Interface Development* > *Define Structure Mappings*

   > **Note**
   >
   > The object key is a combination of all key field values of the business object. You can look up the key fields of a business object in transaction `SWO1` by selecting the object and opening the*Keyfields* node. For the length of each field, open its data object. The object key of the flight booking object SBOOK, for example, is a concatenation of the carrier ID and the booking number. Booking 45 for Lufthansa has the object key `0LH00000045`.

4. Enter `/AIF/WET_FILL_SAP_STRUCT` as the *Function Module After Mapping*.
5. Define and assign an action with the function module `/AIF/WET_AC_TRIGGER_NEXT_INF`.
6. Specify the function module (BAPI) you used for generating the SAP structure of the reading interface under *Interface Development* > *Event Trigger Settings* > *Define Interface-Specific Settings*.

7. In transaction `/AIF/WET_INF_CONFIG`, select a runtime configuration group suitable for filling your sending interface.

Setting Up the Sending Interface

1. Create the sending interface with the generated BAPI structure as raw structure and the structure of the outbound proxy method as SAP structure under *Interface Development* > *Define Interfaces*.
   Make sure that, for this interface, the *Application Engine* and the *Persistency Engine* are set to *XML* under *Interface Development* > *Additional Interface Properties* > *Specify Interface Engines*.
2. Map the structures under *Interface Development* > *Define Structure Mappings*.
3. Define and assign an action with the function module `/AIF/WET_AC_TRIGGER_PROXY_OUT`.

   > **Note**
   >
   > If you are using a synchronous proxy and you want to process its return parameters, you need to implement a custom function module for this.

4. In transaction `/AIF/WET_PRXY_CONFIG`, specify a logical port of the proxy if you want to use a port other than the default.
5. Specify the class and method of the outbound proxy under *Interface Development* > *Event Trigger Settings* > *Define Proxy-Specific Settings*.

4.8 Operations for SAP Application Interface Framework

Use

This section provides an overview of the system operations that are specific to SAP Application Interface Framework.

Features

SAP Application Interface Framework provides you with the following options for analyzing system operations:

- Archive data (see Data Archiving [page 111])
- Measure the performance of SAP Application Interface Framework (see Performance Analysis [page 115])
- Examine interface data for the number of messages and errors (see Index Table Overview [page 117])
- Get an overview of the objects that are used by a specific interface (see Interface Objects Summary [page 117])
- Investigate log content for the changes made to the data messages of the selected interface (see Application Log Content [page 118])
- Display changes made in *Error Handling* to the data messages of the selected interface (see Error Handling Changes Log [page 119])
- The *Data Correction Report* checks for inconsistencies between the Proxy Framework and SAP Application Interface Framework and corrects them (see Data Correction) [page 121]

- A snapshot takes the content in the statistics tables at a specific point of time (see Generation and Display of Snapshots [page 122])

4.8.1 Data Archiving

Use

Data Archiving – a service provided by SAP NetWeaver – removes mass data that the system no longer needs online, but which must still be accessible at a later date, if required, from the database.

In SAP Application Interface Framework, you can use the two archiving objects `/AIF/MES` and `/AIF/FILE` to archive the following types of data:

- Archiving object `/AIF/MES`
  - Data messages
  - XML persistence
  - Data changes
  - Application logs
  - Structured persistence
  - Read access logs
- Archiving object `/AIF/FILE`
  - File adapter logs
  - Application logs

Prerequisites

Data in the database can only be archived using archiving objects, which describe the data structure and context. To set up Data Archiving (transaction code SARA) for SAP Application Interface Framework, you enter the name of the SAP Application Interface Framework-specific archiving object `/AIF/MES` or `/AIF/FILE` in the *Object Name* field on the *Archive Administration: Initial Screen*.

The archiving uses the *Archive Development Kit* (ADK) to provide the archiving objects `/AIF/MES` and `/AIF/FILE`, which includes all the necessary archiving programs and defines the structure of the archive file to be written. To maintain the archiving object, use transaction AOBJ.

All archiving programs maintained in the archiving object are started using *Archive Administration* (transaction code SARA).

If *SAP Information Lifecycle Management* (SAP ILM) is installed in the system, you can use SAP ILM transactions to set up residence and retention rules. These rules give you flexibility as to what data needs to be archived and when.

Process

In SAP Application Interface Framework, the archiving process is divided into the following main steps, some of which are optional:

1. Creation of archive files
   In the write phase, the data to be archived is written sequentially into newly created archive files.
   In SAP Application Interface Framework, the *AIF Data Archiving: Write Program* (`/AIF/MES_WRI` or `/AIF/FILE_WRI`) report controls the writing process of the archive files produced by the archiving objects `/AIF/MES` or `/AIF/FILE`. You can decide which data should be archived using a variant and a selection screen.
   On the selection screen, you can select the types of data you want to archive. You can further restrict the selection using dates and users. For data messages, XML persistence, structured persistence, data changes, and application logs, you can restrict using the interfaces. In addition, there are many type-specific selection parameters. By defining the package size, you can optimize the performance and memory consumption of the archiving runs. In addition, you can tweak the layout of the resulting logs.
   You can run the write program of the archiving object in test mode, which means that the program simulates the creation of an archive, based on your selection criteria and other archiving object-specific checks that test the data's suitability for archiving. The program does not create an archive file but does generate statistics about which table entries would be written to the archive and the size and number of archive files that would be created in production mode.
   Once you schedule the write process, you can monitor the corresponding job on the *Job Overview* screen.
   If ILM residence rules are set up, only data which fulfills these rules is archived. To create rules, use the **`IRMPOL`** transaction.

   > ⓘ Note
   >
   > To create a rule for all values in conditional fields, leave these fields blank (works like a wildcard (*)).

   > → Recommendation
   >
   > We recommend that you schedule the archiving as periodic background jobs that run in off-peak hours.

2. Deletion from the database
   The delete program reads the data from the archive files and deletes the data from the database.
   In SAP Application Interface Framework, you can use the *Archiving of the AIF Persistence: Delete Program* (`/AIF/MES_DEL` or `/AIF/FILE_DEL`) report to control the deletion of persistence entries that have been written successfully to an archive file and to generate statistics about the table entries that were deleted.
   If you run the delete program of the archiving object in test mode, data is not deleted from the database as the deletion is only simulated.
   Once you schedule the deletion process, you can monitor the corresponding job on the *Job Overview* screen.
3. Optional: Display of archive files
   You can use the *Archiving of the AIF Persistence: Sequential Read Program* (`/AIF/MES_READ` or `/AIF/FILE_READ`) report to display the data stored in archive files of the archiving objects `/AIF/MES` or `/AIF/FILE`. If you double click on a line, more details about the corresponding entry are displayed.
4. Optional: Reloading of archive files
   You can use the *Archiving of the AIF Persistence: Reload Program* (`/AIF/MES_REL` or `/AIF/FILE_REL`) report to reload successfully archived data, produced by the archiving objects `/AIF/MES` or `/AIF/FILE`, to the SAP Application Interface Framework tables.

You can also run the reload program in test mode. The reload is only simulated and there are no changes made to the database.
Once you select a successfully deleted archiving session, maintain the variant, and schedule the reload process, the corresponding job is displayed on the *Job Overview* screen.

> → Recommendation
>
> We recommend that you use this feature in exceptional circumstances only.

5. Optional: Deletion of archived files
   If ILM retention rules are configured, you can use the **`ILM_DESTRUCTION`** transaction to delete archived files.
   Choose data type *Archive Files (ADK)*. In *ILM Object*, enter ILM object `AIF_MES` for archiving object `/AIF/MES` and ILM object `AIF_FILE` for archiving object `/AIF/FILE`.

More Information

For more information about archiving, see the SAP Help Portal under *Introduction to Data Archiving (CA-ARC)*.

For more information about how to configure ILM rules, see the SAP Help Portal under *Information Lifecycle Management* and *I2M*.

4.8.2 Data Destruction

Use

Data Destruction – a service provided by SAP NetWeaver – removes mass data that the system no longer needs.

In SAP Application Interface Framework, you can use the following 4 data destruction objects:

- Destruction object for external contacts `AIF_EXT_CONT_DESTRUCTION`
  - Delete entries from table `/AIF/EXT_CONTACT`
  - Corresponding ILM Object: `AIF_EXT_CONT_DESTRUCTION`
- Destruction object for runtime objects `AIF_RUN_OBJ_DESTRUCTION`
  - Delete entries from tables: `/AIF/PERS_RUN`, `/AIF/PERS_RUNPCK`, and `/AIF/PERS_QMSG`
  - Corresponding ILM Object: `AIF_RUN_OBJ_DESTRUCTION`
- Destruction object for snapshots `AIF_SNAPSHOT_DESTRUCTION`
  - Delete entries from tables: `/AIF/MSG_SNAP_HD`, `/AIF/MSG_SNAP_DI`, `/AIF/MSG_SNAP_DM`, `/AIF/MSG_SNAP_IT`, and `/AIF/MSG_SNAP_MS`
  - Corresponding ILM Object: `AIF_SNAPSHOT_DESTRUCTION`
- Destruction object for value mapping `AIF_VAL_MAP_DESTRUCTION`

- Delete entries from tables: `/AIF/T_MVMAPVAL, /AIF/T_MVMAPVAL5, /AIF/T_VMAPVAL,` and `/AIF/T_VMAPVAL5`
- Corresponding ILM Object: `AIF_VAL_MAP_DESTRUCTION`

Prerequisites

SAP Information Lifecycle Management (SAP ILM) should be installed in the system. You can use SAP ILM transactions to set up retention rules. These rules give you flexibility as to what data needs to be deleted and when.

To set up Data Destruction (transaction code `ILM_DESTRUCTION`) for SAP Application Interface Framework, you enter the name of the SAP Application Interface Framework-specific ILM object in the *ILM Object* field on the *Data Destruction: Initial Screen*.

Process

Start the `ILM_DESTRUCTION` transaction.

On the selection screen, select *Data from the Database* and enter the name of an ILM object in the *ILM Object* field. On the next screen, you should schedule a job, select *Destroy*, and specify the execution variant and schedule for the job. Each destruction program has standard options for job log output and a description for the destruction job.

You can run the destruction program of the destruction object in test mode, which means that the program simulates the deletion of data, based on your selection criteria and other destruction object-specific checks (ILM rules) that test the data's suitability for destruction. The program does not delete any data but does generate statistics about which table entries would be deleted in production mode.

Once you schedule the destruction process, you can monitor the corresponding job on the *Job Overview* screen.

If ILM retention rules are set up, only data which fulfills these rules is deleted.

To create rules, use the `IRMPOL` transaction.

> ⓘ Note
>
> To create a rule for all values in conditional fields, leave these fields blank (works like a wildcard (*)).

> → Recommendation
>
> We recommend that you schedule the destruction of data as periodic background jobs that run in off-peak hours.

More Information

For more information about data destruction, see the SAP Help Portal under*SAP NetWeaver* ▶ *Information Lifecycle Management* ▶ *Data Destruction with a Data Destruction Object* ▶.

For more information about how to configure ILM rules, see the SAP Help Portal under *Information Lifecycle Management* and *I2M*.

4.8.3 Performance Analysis

Use

You can use *Performance Tracking* (transaction code `/AIF/PERFORMANCE`) to measure the performance of the SAP Application Interface Framework. Using this performance analysis tool, you are able to detect time-consuming routines and processes within the SAP Application Interface Framework.

Using tracking variants available in this transaction, you can track the performance of specific objects, for example, particular interfaces, users, and time ranges within the SAP Application Interface Framework.

Goals

The goals of the performance optimization process and the *Performance Tracking* transaction include:

- Measuring interface performance in back end systems
- Determining the area for performance improvements (for example, interface-specific logic and mapping logic)
- Defining measures for bottlenecks with most saving potential
- Implementing performance improvement measures
- Confirming the positive impact of performance improvement measures

Features

Scope

The scope of the interface performance measuring is defined for all interfaces via SAP NetWeaver Process Integration and the SAP Application Interface Framework.

The following are measured:

- ABAP Proxy
- Interface-specific logic (for example, value mappings and validations)
- Posting logic (for example, BAPIs)

The following are not measured:

- Legacy extractors
- Transmission of data (for example, file transfer and MQSeries)

- SAP NetWeaver Process Integration

Activities

You can access the *Performance Tracking* report from the SAP Easy Access menu by choosing *Cross-Application Components* > *SAP Application Interface Framework* > *Administration* > *Tools* > *Performance Tracking*.

Set Up Tracking

In order to track performance, a variant needs to be set up. You can either select a variant that already exists or create a new variant. If you select an existing variant, the corresponding data for the variant is displayed on the screen.

If a variant has not been created, the default values are shown in the selection screen of the transaction and the variant is marked as *Unsaved*. While creating a variant on the selection screen, you can specify a message ID, namespace, interface name, interface version, and the name of the user starting the message. You can also enter the minimum duration and the time range.

You can start tracking by choosing  . You can stop tracking by choosing  .

Analyze Results

Performance issues could arise in the following areas:

- Interface-specific logic
- Posting logic (for example, SAP BAPIs)

Examples of performance issues in above areas and possible solutions for you for these issues are shown in the following table:

Examples of Performance Issues	Possible Solutions
Frequent selection of identical data, for example, function modules that are Customized on an item level	Buffering in hashed internal table
Slow select statements, for example, database index not used	• Usage of database index, if available • Creation of database index, if feasible • Database access via alternative table
Long read access to internal tables	• Usage of sorted or hashed tables • Usage of field symbols
Long runtime of SAP BAPIs	• Check for user exits with poor performance • Redesign of interface if BAPI supports multi-post

4.8.4 Index Table Overview

Use

The *Index Tables Overview* report (transaction code `/AIF/IDXTBL`) gives you an overview about the number of messages and the number of errors that exist for each interface of the SAP Application Interface Framework.

Features

On the selection screen of the *Index Table Overview* transaction, you can select the namespace, interface name, and interface version. If you leave those fields blank, all interfaces that exist are selected. In addition, you can decide if only error messages are to be counted or if all existing messages are to be counted.

Activities

You can access the *Index Table Overview* report from the SAP Easy Access menu by choosing *Cross-Application Components* > *SAP Application Interface Framework* > *Administration* > *Tools* > *Index Table Overview*.

Example

An interface administrator runs the report to check if the message number for each interface is within the expected range. The administrator identifies an unusual high number of messages for one interface and decides to do the following:

- Checks that data archiving is set up for the interface
- Ensures that performance is still good in the error handling
- Runs a performance report
- Checks technical attributes of the corresponding single index table in order to see if it is optimized for such a large number of messages

4.8.5 Interface Objects Summary

Use

The *Summary Report* (transaction code `/AIF/CUST_OVERVIEW`) provides an overview of the objects in SAP Application Interface Framework that are used by a specific interface.

SAP Application Interface Framework introduces several objects for the development of interfaces, for example, value mappings, fix values, checks, and actions. Since most of these objects are maintained in different locations in SAP Application Interface Framework, there is no aggregated view of all the objects used by a specific interface. The *Summary Report* provides you with such a consolidated view.

Activities

You can access the *Summary Report* from the SAP Easy Access menu by choosing *Cross-Application Components* > *SAP Application Interface Framework* > *Administration* > *Tools* > *Summary Report*.

4.8.6 Application Log Content

Use

Interface Logs (transaction code `/AIF/LOG`) offers you a specialized view of the transaction to analyze the application log (transaction code `SLG1`). In the specialized view, the log object is predefined as `/AIF/LOG`, so that only application log messages created by the SAP Application Interface Framework are displayed.

Features

On the *Select Interface Logs* screen, you have options to restrict the selection, for example, by using the following:

- *Subobject*
  In the SAP Application Interface Framework, this refers to the namespace and interface name.
- *External ID*
  The message GUID or a file number of the test tool
- *Time Restriction*
  You can define a date and time range.
- *Log Triggered By*
  The user, transaction code, or program that triggered the log
- *Log Class*
  All logs or logs with various degrees of importance

Output

In the upper part of the *Display Interface Log Messages* output screen, the selected application log messages are displayed grouped by the date, time, and user. An icon shows whether the messages have the message type *Information, Warnings*, *Error*, or *Termination* depending on the highest error status of the group. When you select an icon, the number of individual messages for each individual message type is displayed. When you then select an icon, for example, for those with message type *Error*, the details for the individual log messages with message type *Error* are displayed in the lower part of the output screen.

Activities

You can access the *Interface Logs* report from the SAP Easy Access menu by choosing *Cross-Application Components* > *SAP Application Interface Framework* > *Administration* > *Log* > *Interface Logs*.

4.8.7 Read Log and Change Log

Use

You can use the *Read Log Viewer* (transaction code `/AIF/READ_LOG`) to display all read accesses to the following:

- Data messages of the selected interface or interfaces in *Error Handling*, *Interface Monitor*, or *Change Log Viewer*
- Uploaded file

You can use the *Change Log Viewer* (transaction code `/AIF/CHANGE_LOG`) to display the changes made to the following:

- Data messages of the selected interface or interfaces in *Error Handling*
- Value mapping entries in Customizing or transaction `/AIF/VMAP`
- External contact entries

Features

Selection

On the selection screen of the *Read Log Viewer*, you can select the type of data to be displayed as follows and you can further restrict the selection using dates and times:

- Data message content
  You can restrict the selection by namespace, interface name, and version. You can refine the selection by entering a message ID.
- Uploaded files
  You can restrict the selection by namespace and configuration ID

On the selection screen of the *Change Log Viewer*, you can select the type of data to be displayed as follows and you can further restrict the selection using dates and times:

- Data message content
  You can restrict the selection by namespace, interface name, and version. You can refine the selection by entering a message ID.
- Value mapping
  You can restrict the selection by namespace and value mapping name.
- External contact
  You can restrict the selection by external contact ID and external contact name.

Output

On the output screen of the *Read Log Viewer*, a tree view displays all selected interfaces with versions for AIF messages, configuration namespace, and configuration ID for uploaded files. When you select a node, the following read access logs are displayed for data messages content and uploaded files as follows:

1. Data message content
    - The GUID of the data message
    - The namespace, interface, and interface version
    - The Read Access Log identifier
    - The transaction code where access was logged
    - The part of the transaction where data was displayed
    - The field path of the field that is sensitive and can be shown
    - The date and time of the access
    - The user who had access
2. File upload content
    - The GUID of the data message
    - The transaction code where access was logged
    - The part of the transaction where data was displayed
    - The field path of the field that is sensitive and can be shown
    - The date and time of the access
    - The user who had access
    - The configuration name, configuration ID, and file path

On the output screen of the *Change Log Viewer*, a tree view displays all selected interfaces with versions for message content, value mapping name and line number for value mapping, and external contact name for external contact. When you select a node, the following changes made to data messages content, value mappings, and external contacts are displayed as follows:

1. Data message content
    - The GUID of the data message
    - The namespace, interface, and interface version
    - The field path of the field that was changed by the user
    - The `LINE_NR` that contains the message index with line numbers
    - The new and old value
    - The date and time of the modification
2. Value mapping
    - The namespace and value mapping name
    - The value number that contains the entry index with line numbers
    - The field path of the field that was changed by the user
    - The new and old value
    - The date and time of the modification
3. External contact
    - The GUID of the external contact
    - The external contact name
    - The field name that was changed by the user

- The new and old value
- The date and time of the modification

Activities

In SAP GUI, you can also access the *Read Log Viewer* and *Change Log Viewer* reports from the SAP Easy Access menu by choosing *Cross-Application Components* > *SAP Application Interface Framework* > *Administration* > *Log* > :

- *Read Log Viewer*
- *Change Log Viewer*

4.8.8 Data Correction

Use

There might be inconsistencies between the Proxy Framework and SAP Application Interface Framework. The *Data Correction Report* (transaction code `/AIF/CORRECTIONS`) checks for inconsistencies and corrects them. A test mode exists that allows you to check for inconsistencies but not to actually correct them. By default, the *Testrun* checkbox is selected.

Features

The following inconsistencies can be fixed:

- There might be inconsistencies between the message status in the Proxy Framework and the single index tables in SAP Application Interface Framework. These inconsistencies can occur if a message was canceled, deleted, or archived in the Proxy Framework, but the status was not updated in the single index tables of SAP Application Interface Framework.
- There might be inconsistencies between the message index table and the SAP NetWeaver PI message status. These inconsistencies occur if a message, which ended with errors in SAP Application Interface Framework, is restarted with the Proxy Framework and processed successfully. In this case, the status in SAP Application Interface Framework might not be updated.
- If an SAP NetWeaver PI message was deleted or archived, the corresponding application log written by SAP Application Interface Framework must be deleted. Inconsistencies must be corrected.
- When there are changes made in Customizing for alert recipients, the alert index table might not be updated correctly. Therefore, the table might contain inconsistent records when compared to current Customizing for alert recipients. You can regenerate the messages' statistics to ensure that no inconsistencies exist and the messages' statistics can be compressed.

When running the *Data Correction Report*, the system checks for messages that exist in the message index tables of SAP Application Interface Framework but not in the Proxy Framework. These are regarded as

inconsistent and are deleted from the message index tables of SAP Application Interface Framework. It is recommended to also delete the corresponding application log messages. Hence, the *Data Correction Report* should be planned as a background job after execution of the deletion or archiving jobs for the Proxy Framework with the following settings:

- *Namespace*, *Interface Name*, and *Interface Version* should be blank
- *Testrun* should not be selected
- *Check/Fix Deleted or Archived Messages* should be selected
- *Check/Fix Application Log* should be selected
- *Regenerate Statistics* should be selected

Activities

You can access the *Data Correction Report* from the SAP Easy Access menu by choosing *Cross-Application Components* > *SAP Application Interface Framework* > *Administration* > *Tools* > *Correction Report*.

> → Recommendation
>
> It is recommended that the *Data Correction Report* is run regularly as a background job to ensure that no inconsistencies exist.

> → Recommendation
>
> It is further recommended that you run the *Data Correction Report* after your deletion or archiving job for messages in the Local Integration Engine was executed.

4.8.9 Generation and Display of Snapshots

Use

In the SAP Application Interface Framework you are able to generate and display snapshots. A snapshot takes the content in the statistics tables (`/AIF/MSG_STAT` and `/AIF_MSGSTATALL`) at a specific point of time and saves it in separate snapshot tables. The same applies to the message summary table (`/AIF/T_MMSG_IDX`).

Features

Generation of Snapshots

A snapshot can be taken using *Generate Snapshot* (transaction code `/AIF/GENMSGSNAP`). This transaction allows you to create snapshots of statistics and summary data for messages in the SAP Application Interface Framework and track this data over time.

On the *Generate Snapshot* selection screen, select a date for which the snapshot is to be created (this only applies to daily statistics data). You can also select a package size, that is, the size of the packages in which the single messages are processed. Changing the number of the package size could increase or decrease the performance of the report considerably. *1000* is the suggested value for most situations. However, the optimum package size depends on the system hardware and the current system load.

Display of Snapshots

A snapshot can be displayed using *Display Snapshot* (transaction code `/AIF/DISPMSGSNAP`). This transaction allows you to display snapshots of statistics and summary data for messages available in the SAP Application Interface Framework and track this data over time.

There are several ways you can filter for snapshots, for example, ID, creation data, creation user, and the date the snapshot was created.

You can select from the following 4 views:

- 
  Lists all the statistics entries for messages available in the system at the time of snapshot creation.
- 
  Lists all the message statistics for the *effective date* specified for the snapshot.
- 
  Lists all the log message entries available in the system at the time of snapshot creation. Log messages are grouped and their amount cumulated on the basis of namespace, interface name, interface version, recipient namespace, recipient name, message type, message ID, and message number. You can see the message text (with placeholders) and the key fields that were the basis for the determination of the specific recipient.
- 
  Shows all the log messages for a specific date, that is, the effective date of the snapshot.

Activities

You can access the reports to generate and display error statistics snapshots from the SAP Easy Access menu by choosing *Cross-Application Components* > *SAP Application Interface Framework* > *Administration* > *Snapshots* > and *Generate Snapshot* or *Display Snapshot*.

> → Recommendation
>
> It is recommended that *Generate Snapshot* is run daily as a background task.

4.8.10 Message Deletion

Configure deletion for messages that don't require archiving.

Context

While SAP Application Interface Framework archives messages by default, for some interfaces, archiving data isn't necessary or overly extensive. In these cases, interface-specific data can be deleted.

After flagging messages for deletion using *Configure Message Deletion of Interfaces (Default)* or *Configure Message Deletion of Interfaces*, you can delete them with a regular technical job based on report `/AIF/MESSAGE_DELETION`.

> **Note**
>
> To ensure consistency across systems, SAP Application Interface Framework can communicate with SAP Process Integration regarding any proxy messages that have been deleted by SAP Process Integration. These messages can then be removed by a deletion job in the SAP Application Interface Framework system as well, which closes message gaps between the systems. For implementation information, see SAP Notes 3214922 and 3289252.

> **Note**
>
> The message deletion reports in SAP Application Interface Framework don't consider ILM settings, for example, retention periods. If the report and the ILM settings are in conflict, the message is deleted regardless.

Procedure

1. Open *Customizing* (transaction code `/AIF/CUST`) and navigate to *Interface Development* > *Additional Interface Properties* > *Configure Message Deletion of Interfaces (Default)* (view `/AIF/V_FINF_DEL`).

   Alternatively, you can overwrite the default settings by navigating to *System Configuration* > *Custom Extensions* > *Configure Message Deletion for Interfaces* and using the view `/AIF/V_FINF_DELC`. However, the content of this view isn't transported by default and must be transported **manually** in on-premise systems.
2. Choose *Change* to enter edit mode.
3. In column *Allow deletion*, tick the checkbox for every interface for which you want to enable message deletion.
4. In column *Older than*, enter the number of days after which the messages can be deleted.
5. *Save* your changes.
6. Additionally, configure the technical job that regularly deletes the messages.
   1. Open *Define Job* (transaction code `SM36`).

2. Create a new job using report `/AIF/MESSAGE_DELETION` and variant `SAP&AIFDEL_STD`. Schedule its execution to your preference.

7. If you want the deletion report to delete all messages that have been marked as deleted by XI, perform the following actions:
   1. In the *ABAP Editor* (transaction `SE38`), execute report `/AIF/MESSAGE_DELETION`.
   2. Create a variant with *Data Selection Mode* being *XI*.
      The deletion report then includes the relevant messages from the XI integration table in its selection, which means the report deletes any of those messages that are in status *A* or *D*.

      > ⚠ Caution
      >
      > If you select *XI*, the report ignores any configurations you made in Customizing for column *older than*.

   3. Create a new job using report `/AIF/MESSAGE_DELETION` and the variant that you created. Schedule its execution to your preference.

4.9 Technology Support

Use

SAP Application Interface Framework supports business users in monitoring interfaces with different interface technologies, for example, ABAP proxies and IDocs. It is also possible for users to monitor interfaces that are not processed by SAP Application Interface Framework interfaces.

Depending on the interface technology of an interface, data displayed in *Monitoring and Error Handling* is selected from different sources. Functionalities like restart or cancel are supported for all technologies that allow these functions.

Features

To support *Monitoring and Error Handling* functions for different interface technologies, SAP Application Interface Framework has an engine factory with many different types of engines, for example, Application Engine, Persistency Engine, Selection Engine, and Logging Engine. Engines for some scenarios are provided by SAP and you can also create your own engines. For more information, see Engine Factory [page 126].

When you define an interface, you need to maintain the engines that should be used for the interface. If you do not maintain any settings for engines, the SAP Application Interface Framework default is that the interface is in an ABAP proxy interface that is processed with SAP Application Interface Framework.

More Information

For more information for business users about the technology supported by SAP Application Interface Framework , see Technology Support [page 55] under SAP Application Interface Framework for Business Users.

4.9.1 Engine Factory

Use

To support monitoring and error handling functions for different interface technologies, the SAP Application Interface Framework has an engine factory.

Integration

You define which engines are to be used by an interface in Customizing for the *SAP Application Interface Framework* (transaction code `/AIF/CUST`) under *Additional Interface Properties* > *Specify Interface Engines*. Interface developers have to define the application engine, persistence engine, selection engine, and logging engine that *Monitoring and Error Handling* uses to handle messages of the interface. The interface developer is able to maintain engines provided by SAP as well as customer-specific engines. The system default is to handle an interface as an ABAP proxy. This default setting also ensures that interfaces can still be monitored after an upgrade.

Features

The following is a list of the engines available:

- Application engine
  The application engine either handles a request from the *Monitoring and Error Handling* itself or it forwards the request to one of the other engines. The application engine is responsible for restarting and canceling messages. Depending on the interfaces' interface technology, the corresponding restart and cancel procedures are triggered. Furthermore, the application engine is responsible for the navigation to the external monitor. The external monitor is the transaction that is usually used for monitoring messages of a specific technology. The application engine enables the initial processing of a message received by the system the SAP Application Interface Framework is located on. The application engine also sends messages to a different system using the SAP Application Interface Framework.
- Persistence engine
  The persistence engine is responsible for handling the data a message transferred. The persistence engine reads the data from an interface's persistence layer. In order to display the message data in the Data Content view, the persistence engine transforms the data into the raw or SAP data structure (see Data Content View [page 46]). If data is changed by a user in *Monitoring and Error Handling*, the persistence

engine is responsible for transforming the changed data into a format that can be persisted. Afterwards, the persistence engine updates the data on the persistence layer. Furthermore, if a user edits a message's content, the persistence engine locks the message to prevent different users from editing the same message at the same time. The persistence engine unlocks the data after the user has saved the data to the persistence.

- Selection engine
  The selection engine is responsible for selecting the data displayed in the Data Message view (see Data Messages View [page 40]). If the SAP Application Interface Framework is involved in processing a message, this data can be selected from an interface's index table. For scenarios where the SAP Application Interface Framework is not involved in processing the message, the data is retrieved from the table containing the header data of the processed messages. For example, for IDocs, the data is selected from the IDoc control records. The data the selection engine selects is dependent on the entries made on the selection screen. Furthermore, the selection engine has to format the data so that it can be displayed in the Data Message view.
- Logging engine
  The logging engines retrieves the data displayed in the Log Message view (see Log Messages View [page 43]). The log messages are retrieved and arranged in the SAP Application Interface Framework application context structure that can be displayed in the Log Messages view. Furthermore, if log messages exist, which are related to a specific field/line of the data message, context data is prepared to allow the user to navigate to the error in the Data Content view (see Data Content View [page 46]).
- Custom engines
  You can define your own engines in Customizing for the *SAP Application Interface Framework* under *Additional Interface Properties* > *Define Custom Engines*. You must define a namespace and a custom engine ID. For each namespace / custom engine ID combination, you can maintain an application engine, persistence engine, selection engine, and/or logging engine. Once you define a custom engine, you can maintain it in Customizing for the *SAP Application Interface Framework* under *Additional Interface Properties* > *Specify Interface Engines*.
- The following are engines supported when processing data with the SAP Application Interface Framework:
  - Selection engine for index tables
    For messages that are processed within the SAP Application Interface Framework, index table entries are written during processing. If the selection engine for index tables is selected in interface definition, data message are selected from an interface's index tables. Which data messages are selected depends on the selection criteria entered on the selection screen. If an interface-specific sub-selection screen is customized, selection criteria inserted there are considered when the messages are selected.
  - Logging engine for application log
    For data messages processed within the SAP Application Interface Framework, an application log is written. The application log engine is responsible for selecting log messages from the application log.
- The following are engines to support test files:
  The file engines are used to display test file data processed with the Interface Test Tool (transaction code `/AIF/Test`) in *Monitoring and Error Handling*. The file engines are used if this was specifically Customized during interface definition, or if the file was processed with the Interface Test Tool of the SAP Application Interface Framework.
  - Application engine for test files
    The application engine for files is responsible for test files processed with the Interface Test Tool. Independent from the engine customized in the interface definition, the application engine for test files is always called if the message is a test file.
  - Persistence engine for test files

The persistence engine retrieves and updates test files processed with the Interface Test Tool in *Monitoring and Error Handling*. Furthermore, it locks and unlocks the test file if a user edits a file in *Monitoring and Error Handling*.

- The following are the default engines used:
  - Application engine for ABAP proxies
  - Persistence engine for ABAP proxies
  - Selection engine for index tables
  - Logging engine for application log

More Information

Fore more information about the individual views on the main screen of *Monitoring and Error Handling*, see Main Screen [page 37].

4.9.2 Proxies

The proxy engines of SAP Application Interface Framework are responsible for messages received from the SAP NetWeaver PI or sent to the SAP NetWeaver PI using an ABAP proxy.

Furthermore, the engines for proxies are responsible for web service messages.

> **Note**
>
> From SAP NetWeaver 7.40, there is a new WebService infrastructure available. For new web service messages, SAP Application Interface Framework provides a dedicated application engine and a persistence engine. The Proxy engines are still functional for all web service messages, but the new web services engines provide better performance.

In detail, there are the following engines available:

Application Engines

Proxy and *New WebService* can be used in inbound scenarios and in outbound scenarios.

Proxy is used for messages that are processed in the local integration engine of SAP NetWeaver PI and is responsible for web service messages. *New WebService* is used for web service messages of the new WebService infrastructure (from SAP NetWeaver 7.40).

Persistence Engines

Proxy and *New WebService* can be used in inbound scenarios. They can also be used in outbound scenarios where the messages are only monitored in SAP Application Interface Framework, but no mapping is performed in AIF.

Proxy and *New WebService* are responsible for retrieving the SAP NetWeaver PI or web service messages from the persistence layer. The payload is transformed to the source data structure to display it in the *Data Content* view of *Monitoring and Error Handling*.

In outbound scenarios where the mapping is performed in AIF, you need to use the *XML* persistency engine.

> ⓘ Note
>
> In case of a processing issue, SOAP messages using MTOM (Message Transmission Optimization Mechanism) cannot change payload or be reprocessed. To enable SOAP messages with attachments to be processed by SAP Application Interface Framework, unselect *Enable MTOM* in the Properties section of your SOAP tool.

Selection Engine and Logging Engine

Use *AIF Index Tables* and *AIF Application Log* in all proxy scenarios.

4.9.3 IDoc Support

Use

In the SAP Application Interface Framework, you can process and monitor IDocs.

Features

The following IDoc engines are available:

- Application engine for IDocs
  In order to display IDocs in the SAP Application Interface Framework, the application engine for IDocs is used. The application engine for IDocs enables the users to restart and cancel all IDocs displayed in *Monitoring and Error Handling*.
- Persistence engine for IDocs
  The persistence engine for IDocs retrieves an IDoc's data records. Furthermore, the persistence engine locks and unlocks the IDoc. After an IDoc is edited, the persistence engine updates the IDoc's data records.
- Logging engine for IDoc status records
  If the IDoc was not processed with the SAP Application Interface Framework, the status records written by the IDoc runtime have to be read from EDIDS. In order to display the status records in the Log Message

view of *Monitoring and Error Handling*, the status records are added to a temporary application log. If a segment number and field name exists in the status record, it is possible to navigate from the Log Message view to the erroneous field in the Data Content view. If the IDoc has written its own application log, the log content is also displayed in the Log Message view.

- Selection engine for IDoc control records
  For IDocs processed without the SAP Application Interface Framework, no index table entries exist and the data has to be selected from the IDoc control records. In order to determine which IDoc belongs to which interface, you need to maintain the basic type and message type. Optionally, you can maintain a logical message variant and logical message function. You can maintain these in Customizing for *SAP Application Interface Framework* under *Interface Development* > *Assign IDocs Types*. In the *Selection Screen* of *Monitoring and Error Handling*, you can select by namespace, interface name, and version. You can also select by status, creation date, and IDoc number.

4.9.3.1 IDoc Scenarios

Use

In the SAP Application Interface Framework, you can monitor and process IDocs.

Integration

In Customizing for *SAP Application Interface Framework* under *Interface Development* > *Additional Interface Properties* > *Specify Interface Engines*, you need to make settings for the engines for each of the scenarios where IDoc technology is supported by the SAP Application Interface Framework.

Features

In the SAP Application Interface Framework, there are the following different scenarios in which you can process and monitor IDocs:

- IDoc Scenario 1: See Monitoring of existing IDocs in Monitoring and Error Handling [page 131]
- IDoc Scenario 2: See Process IDoc Using AIF; Call IDoc Function Module in Action [page 132]
- IDoc Scenario 3: See Process IDoc Using AIF; Call BAPI in Action [page 134]
- IDoc Scenario 4: See Process IDoc Using ALE; Write Index Tables with AIF Enabler [page 136]
- IDoc Scenario 5: See Process Outbound Docs Using AIF [page 138]
- IDoc Scenario 6: See Combine Outbound IDoc to XML Message Using AIF [page 141]

4.9.3.1.1 Monitor Existing IDocs in Monitoring and Error Handling

Use

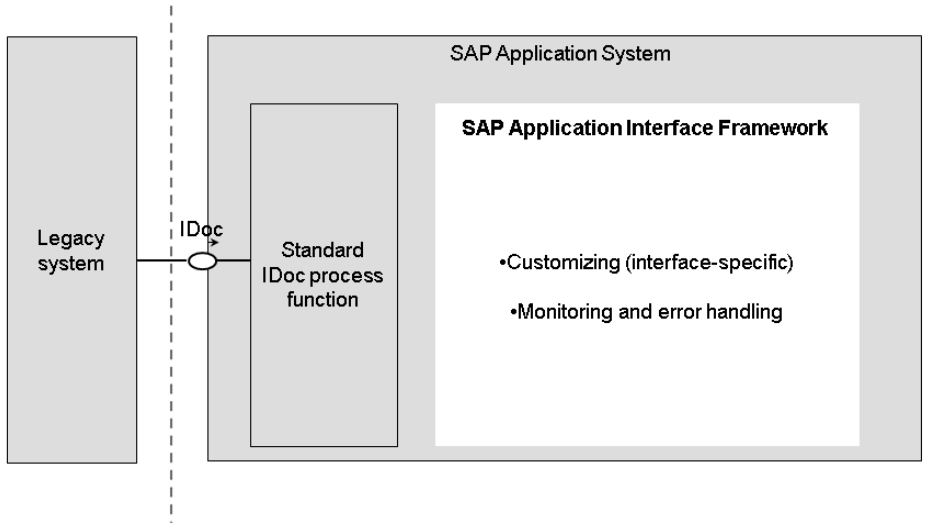
In this scenario, existing IDocs are monitored in *Monitoring and Error Handling* in the SAP Application Interface Framework. IDocs are processed in the ALE runtime. This scenario is applicable for inbound and outbound IDocs.

You should use this scenario if you have an existing IDoc, for which you do not want to change the configuration but which you want to monitor with the SAP Application Interface Framework.

Integration

In order to display an existing IDoc in *Monitoring and Error Handling*, you have to create a raw and an SAP data structure for an IDoc's basic type and you need to define an SAP Application Interface Framework interface. To do this, use the *Generate IDoc Structure and Interface Definition* report (transaction code `/AIF/IDOC_GEN`, see IDoc Structure Generation and Interface Definition [page 74]).

The raw and SAP data structures are the same.



Monitoring of Existing IDocs in Monitoring and Error Handling

In the scenario for processing IDocs shown in the graphic above, the SAP Application Interface Framework is not involved in processing the IDoc and the IDoc from the legacy system is, as always, processed with ALE runtime.

The standard process function, process code, and message type is used. During processing, the data is written to the standard IDoc tables, for example, `EDIDC`. In the SAP Application Interface Framework, there is no data written to the AIF tables and data is selected from the IDoc tables in order to display the IDocs in *Monitoring and Error Handling*.

The raw data structure containing the data from the IDoc tables is displayed in the Data Structure view of *Monitoring and Error Handling*.

The status records written during IDoc processing are displayed in the Log Messages view. If an application log for an IDoc was written, the application log containing the data from the IDoc tables is displayed in the Log Messages view too.

Features

In this scenario, the following are **not** possible in the SAP Application Interface Framework:

- Displaying messages in the Interface Monitor
  There are no entries in the AIF index and statistic tables as the IDoc is not processed by SAP Application Interface Framework
- Creating alerts using and sending e-mail notifications
- Defining an interface-specific selection screen
- Using a key fields based search in *Monitoring and Error Handling*
  There are no index table entries

Activities

In order to display an existing IDoc in *Monitoring and Error Handling*, engines have to be assigned to interfaces in Customizing for *SAP Application Interface Framework* under *Interface Development* > *Additional Interface Properties* > *Specify Interface Engines* as follows:

- Application engine for IDocs
- Persistence engine for IDocs
- Selection engine for IDoc control records
- Logging engine for IDoc status records

4.9.3.1.2 Process IDoc Using AIF, Call IDoc Function Module in Action

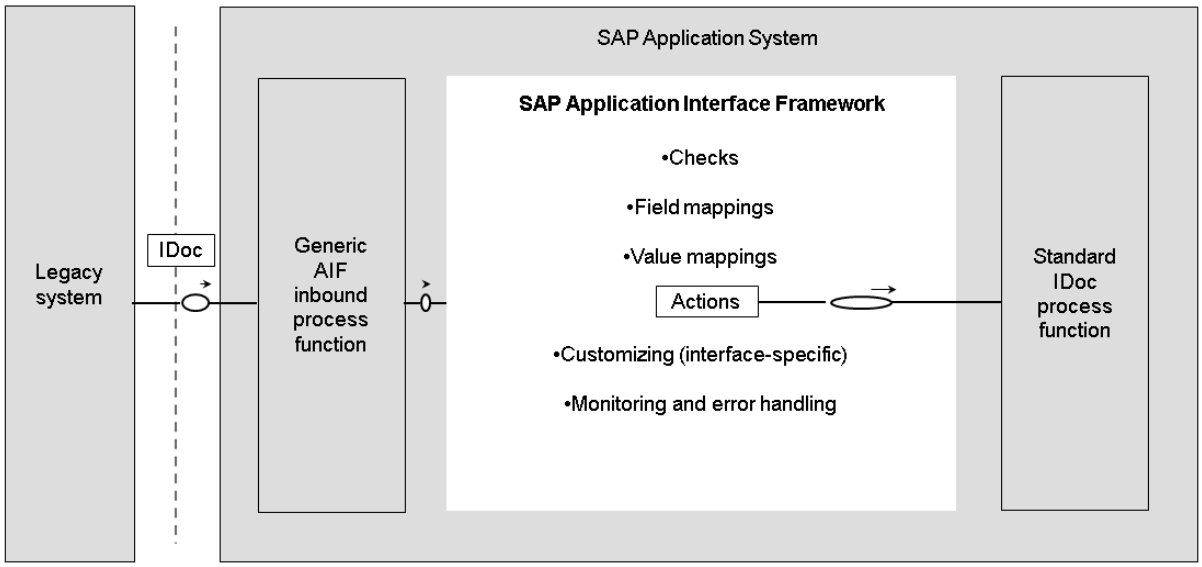
Use

In this scenario, IDocs are processed using the SAP Application Interface Framework and an IDoc function module is called in an action.

Integration

In order to display the IDoc in *Monitoring and Error Handling*, you have to create a raw and an SAP data structure for an IDoc's basic type and you need to define an SAP Application Interface Framework interface. To do this, use the *Generate IDoc Structure and Interface Definition* report (transaction code `/AIF/IDOC_GEN`, see IDoc Structure Generation and Interface Definition [page 74]).

The raw and SAP data structures are the same. In Customizing for *SAP Application Interface Framework* under *Interface Development* > *Define Interfaces*, you can select the *Move Corresponding* checkbox. This maps the data from the raw to the SAP data structure while processing the data in the SAP Application Interface Framework. If you require further logic, for example, checks and value mappings, you can define these in Customizing for *SAP Application Interface Framework* under *Interface Development* > *Define Structure Mappings*.



Process IDocs Using AIF and Calling IDoc Function Module in Action

In the scenario for processing IDocs shown in the graphic above, the IDoc from the legacy system is processed in the SAP Application Interface Framework where index tables are written.

A generic process function is delivered with the SAP Application Interface Framework. A new process code is needed. The process function transforms the IDoc data into the raw data structure defined in Customizing for *SAP Application Interface Framework*. Afterwards the interface is determined and processing in the SAP Application Interface Framework is triggered.

In this scenario, you have to create an action to call the standard process function. To facilitate this, the `/AIF/IDOC_ACTION_FUNCTION` function module is delivered. You can call the function module in your action function. Pass the name of the standard process function you want to call and the parameter `DATA` of the action function module to `/AIF/IDOC_ACTION_FUNCTION`.

If data is to be displayed in the Data Content view, the IDoc data has to be transformed into the raw data structure.

Features

In this scenario, the following are possible in the SAP Application Interface Framework:

- Using the Interface Monitor
- Creating checks, value mappings, and field mappings
- Selecting data from index tables
- Defining key fields
- Defining an interface-specific selection screen, which can be used to facilitate a selection of messages if customized
- E-mail notification

Activities

Engines

You have to assign engines to interfaces in Customizing for *SAP Application Interface Framework* under *Interface Development* > *Additional Interface Properties* > *Specify Interface Engines* as follows:

- Application engine for IDocs
- Persistence engine for IDocs
- Selection engine for index tables
- Logging engine for application log

Preprocessing

The SAP Application Interface Framework can use preprocessing to execute some logic on top of IDoc raw message data with limited maintenance effort. If you want to activate preprocessing, select the *Preprocessing* checkbox in Customizing for *SAP Application Interface Framework* under *Interface Development* > *Define Interfaces* (see Preprocessing [page 170]).

4.9.3.1.3 Process IDoc Using AIF, Call BAPI in Action

Use

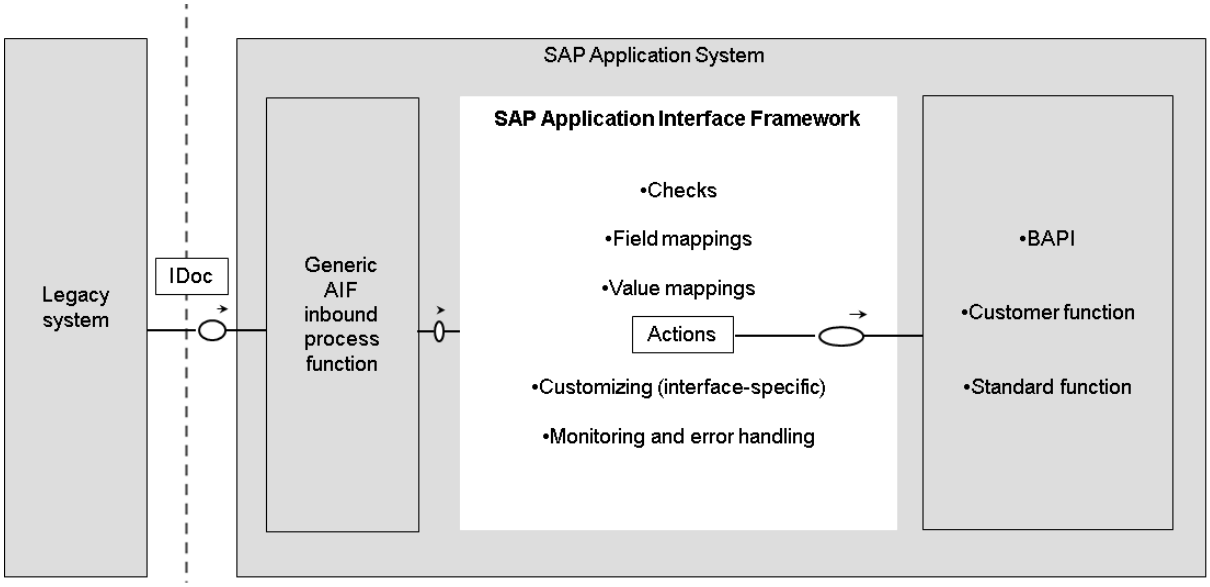
In this scenario, IDocs are processed using SAP Application Interface Framework and a BAPI is called in an action.

Integration

You need to create an SAP Application Interface Framework interface and you have to maintain a raw and an SAP data structure.

Use the *Generate IDoc Structure and Interface Definition* report (transaction code `/AIF/IDOC_GEN`) to create the raw data structure based on the IDoc basic type (see IDoc Structure Generation and Interface Definition [page 74]).

You have to create the SAP data structure manually and this structure depends on the action's functions. Since the raw and SAP data structure differs in most cases for this scenario, you have to create a structure mapping including, for example, field mappings and value mappings.



Process IDoc Using SAP Application Interface Framework and Calling BAPI in Action

In the scenario for processing IDocs shown in the graphic above, the IDoc from the legacy system is processed by SAP Application Interface Framework, where functionality such as index tables, checks, and mapping can be used.

In order to process the IDoc with SAP Application Interface Framework, a generic process function is provided. A new process code is required. The generic process function transforms the IDoc data to the raw structure of the interface. Afterwards, SAP Application Interface Framework is called to process the data.

In this scenario, you have to create an action to process the data. Within the action function, the BAPI is called to process the data.

Features

In this scenario, the following are possible in SAP Application Interface Framework:

- Displaying messages in *Monitoring and Error Handling*
- Supporting the field link from the Log Messages view to the Data Content view, if an error occurred while processing the IDoc
- Maintaining index tables
- Creating checks, value mappings, and field mappings
- Grouping messages by key fields in the Data Messages view
- Creating an interface-specific selection screen

If an interface-specific sub selection screen was customized, this screen can be used to facilitate a selection of messages.

Activities

You have to assign engines to interfaces in Customizing for *SAP Application Interface Framework* under *Interface Development* > *Additional Interface Properties* > *Specify Interface Engines* as follows:

- Application engine for IDocs
- Persistence engine for IDocs
- Selection engine for index tables
- Logging engine for application log

4.9.3.1.4 Process IDoc Using ALE, Write Index Tables with AIF Enabler

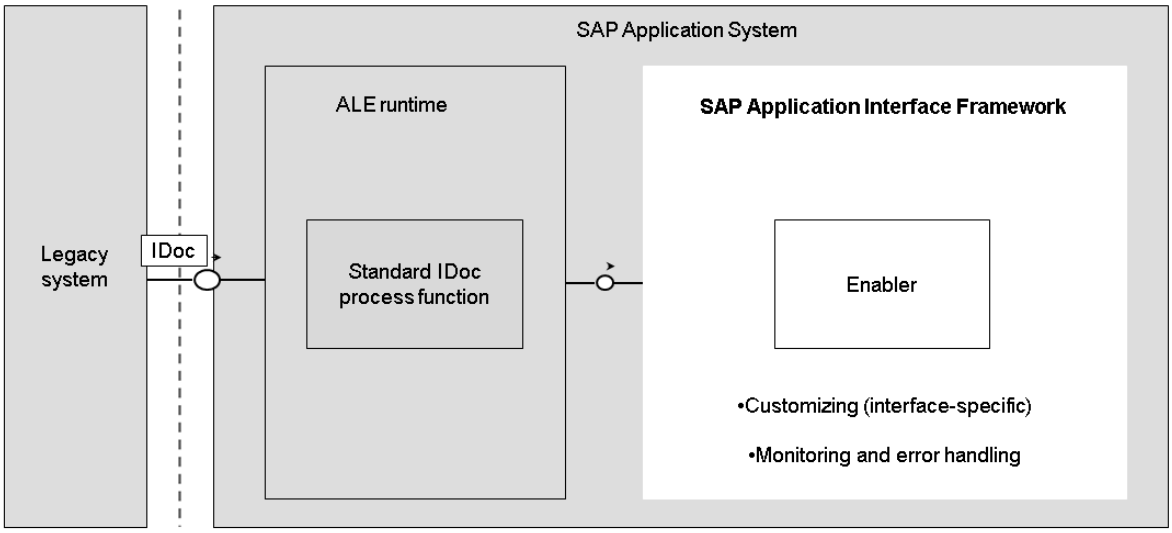
Use

In this scenario, IDocs are processed using ALE runtime and index tables are written by the SAP Application Interface Framework enabler for IDocs. This scenario is applicable for inbound and outbound IDocs.

Integration

You need to create an SAP Application Interface Framework interface and you have to maintain a raw and an SAP data structure. The raw and SAP data structure are the same.

Some settings in Customizing for *SAP Application Interface Framework*, for example, for structure mappings or actions, are not relevant since the IDoc is not processed with the SAP Application Interface Framework runtime.



IDoc Processing Using ALE Runtime and Calling AIF Enabler

In the scenario for processing IDocs shown in the graphic above, the IDoc from the legacy system is processed by standard IDoc processing.

The SAP Application Interface Framework enabler for IDocs is called implicitly during ALE runtime when the status of the IDoc changes to write SAP Application Interface Framework-specific table entries. The SAP Application Interface Framework enabler for IDocs is responsible for writing index tables and message statistics and for creating alerts.

It is also possible to write an SAP Application Interface Framework-specific application log. If a message is selected and an AIF application log is written, this application log is loaded into the application log view. If no AIF application log is written, the status records and application log the process function has written are loaded into the application log view in *Monitoring and Error Handling*. Whether an application log is written or the status records are used depends on the logging engine maintained in Customizing for *SAP Application Interface Framework* under *Interface Development* > *Additional Interface Properties* > *Specify Interface Engines*.

> **Note**
>
> For performance reasons, writing an SAP Application Interface Framework-specific application log is optional and the SAP Application Interface Framework enabler for IDocs avoids transforming the complete IDoc data into the SAP structure.

Since the IDoc was not processed with the SAP Application Interface Framework runtime, there is no data link between the Log Messages view and the Data Content view.

Prerequisites

If you want to use SAP Application Interface Framework as the monitor of your IDoc runtime, implement the *ALE: Integration of SAP Application Interface Framework* note (the 1844763 note number).

Features

In this scenario, the following are possible in SAP Application Interface Framework:

- Defining recipients for alert management
- Grouping messages by key fields in Data Messages view
- Maintaining interface-specific selection index tables
- Defining an interface-specific selection screen, which can be used to facilitate a selection of messages if customized

Activities

You have to assign engines to interfaces in Customizing for *SAP Application Interface Framework* under *Interface Development* > *Additional Interface Properties* > *Specify Interface Engines* as follows:

- Application engine for IDocs
- Persistence engine for IDocs
- Selection engine for index tables
- Logging engine for IDoc status records

4.9.3.1.5 Process Outbound IDocs Using AIF

Use

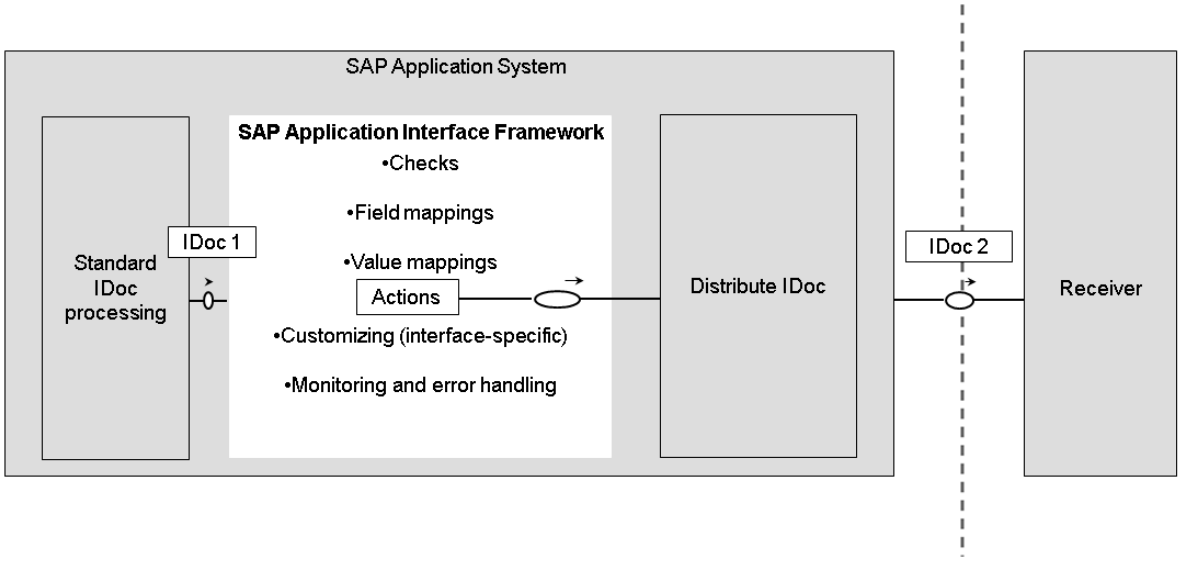
In this scenario, outbound IDocs are processed using the SAP Application Interface Framework.

Integration

For this scenario, the following are required:

- 2 SAP Application Interface Framework interfaces
- A port function
  To trigger the processing in the SAP Application Interface Framework
- An action
  To send the final IDoc to the receiver

To create the required structure and interfaces, use the *Generate IDoc Structure and Interface Definition* report (transaction code `/AIF/IDOC_GEN`, see IDoc Structure Generation and Interface Definition [page 74]). The raw and SAP data structure are identical for both interfaces.



Process Outbound IDoc Using the SAP Application Interface Framework

In the scenario for processing IDocs shown in the graphic above, the first IDoc is send to the AIF port function and is processed via the SAP Application Interface Framework. All mappings, checks, fix values, and value mappings have to be done in the first interface, which is used for the processing of the first IDoc. To finally send the IDoc to the receiver, an action is required to trigger the sending of the IDoc.

Sending the IDoc to the receiver leads to a second IDoc. This second IDoc is monitored in the second AIF interface. This interface is a pure monitoring interface that displays the data as it is sent to the final receiver.

The SAP Application Interface Framework delivers the `/AIF/SINGLE_IDOC_PORT_FUNCTION` function module. In *Ports in IDoc processing* (transaction code WE21), you need to maintain the port function in an `ABAP-PI` port. You have to assign this port in the partner profile. You can reuse the port function for different basic types. If the port function does not meet all of your requirements, you can create and use your own port function.

The Customizing for the first interface contains the processing logic of the IDoc. You must maintain some mapping for the first interface. In order to be able to send the second IDoc to the correct receiver, you need to adjust the sender/receiver data of the `EDIDC` structure (Control Record) to the values of the final sender/receiver of the IDoc.

> ⚠ Caution
>
> If you do not do this, the IDoc ends in a loop of creating one copy after another without sending the IDoc to the final receiver.

To finally send the IDoc, you need an action. As the action function module for you to use to send the IDoc, the SAP Application Interface Framework provides the `/AIF/CALL_MASTER_IDOC_DIST` function module.

For the second interface, a pure monitoring scenario is sufficient. The interface is used to monitor the IDocs that are sent to the final receiver.

> **Note**
>
> As both interfaces have the same message type/basic type combination, you need an entry in the interface determination. You do this in Customizing for *SAP Application Interface Framework* under *System Configuration* > *Interface Determination* > *Define Interface Determination for IDoc Interfaces*.

Features

In this scenario, the following are possible in the SAP Application Interface Framework:

- Using the Interface Monitor
- Creating checks, value mappings, and field mappings
- Selecting data from index tables
- Defining key fields
- Defining an interface-specific selection screen, which can be used to facilitate a selection of messages if customized
- E-mail notification

Activities

Engines

You have to assign engines to interfaces in Customizing for *SAP Application Interface Framework* under *Interface Development* > *Additional Interface Properties* > *Specify Interface Engines* as follows:

- Application engine for IDocs
- Persistence engine for IDocs
- Selection engine for index tables
- Logging engine for application log
  Or IDoc status records for the second IDoc

Assign the engines to both interfaces.

Preprocessing

The SAP Application Interface Framework can use preprocessing to execute some logic on top of IDoc raw message data with limited maintenance effort. If you want to activate preprocessing, when you are defining the first interface, select the *Preprocessing* checkbox in Customizing for *SAP Application Interface Framework* under *Interface Development* > *Define Interfaces* (see Preprocessing [page 170]).

4.9.3.1.6 Combine Outbound IDoc to XML Message Using AIF

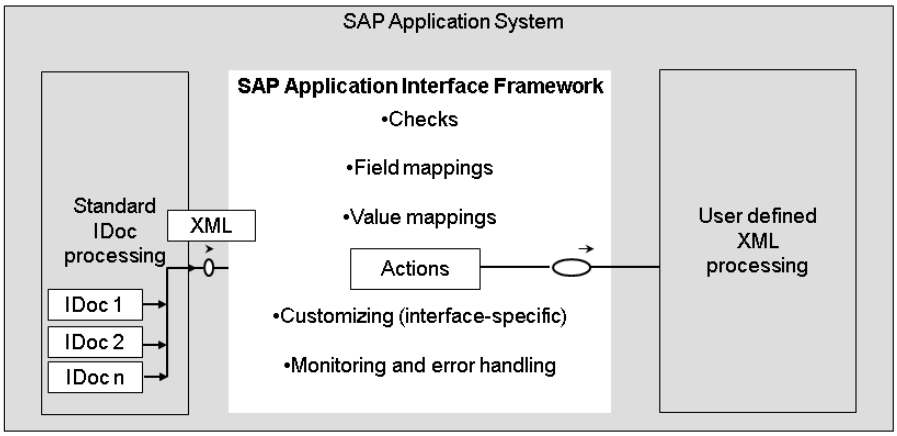
Use

In this scenario, different outbound IDocs can be combined in one XML message.

Integration

For this scenario, an SAP Application Interface Framework XML interface and a port function are required.

The required interface can be created in Customizing for *SAP Application Interface Framework* under *Interface Development -> Define Interfaces*. If required, select the *Move Corresponding* or the *Preprocessing* checkbox. For the raw and SAP data structure, use the `/AIF/MERGE_IDOCS_TO_AIF_MSG` structure. If you require further logic, for example, checks and value mappings, you can define these in Customizing for *SAP Application Interface Framework* under *Interface Development -> Define Structure Mappings*.



Combine Outbound IDoc to XML Message Using the SAP Application Interface Framework

In the scenario for processing IDocs shown in the graphic above, the port function combines the collected IDocs in one XML message and hands it over to the SAP Application Interface Framework for further processing. Follow up processing is triggered via an action in the interface.

To set up this scenario, you require a port function. The SAP Application Interface Framework delivers the `/AIF/MERGE_IDOCS_TO_AUIF_MSG` function module for you to use as function module for the port function. Besides creating the port function, you need to adjust the partner profile of the IDoc that is to be sent. You also need to maintain the new port function in the receiver port. You can reuse the port function for different basic types. If the port function does not meet all of your requirements, you can create and use your own port function.

You need to maintain all required mappings and add an action to trigger the follow up processing.

> **Note**
>
> You may require an interface determination. You do this in Customizing for *SAP Application Interface Framework* under *System Configuration* > *Interface Determination* > *Define Interface Determination for XML Interfaces*.

Features

In this scenario, the following are possible in the SAP Application Interface Framework:

- Using the Interface Monitor
- Creating checks, value mappings, and field mappings
- Selecting data from index tables
- Defining key fields
- Defining an interface-specific selection screen, which can be used to facilitate a selection of messages if customized
- E-mail notification

Activities

Engines

You have to assign engines to interfaces in Customizing for *SAP Application Interface Framework* under *Interface Development* > *Additional Interface Properties* > *Specify Interface Engines* as follows:

- Application engine for XML
- Persistence engine for XML
- Selection engine for index tables
- Logging engine for application log

Preprocessing

The SAP Application Interface Framework can use preprocessing to execute some logic on top of IDoc raw message data with limited maintenance effort. If you want to activate preprocessing, when you are defining the first interface, select the *Preprocessing* checkbox in Customizing for *SAP Application Interface Framework* under *Interface Development* > *Define Interfaces* (see Preprocessing [page 170]).

4.9.4 Error and Conflict Handler Integration

Use

SAP Application Interface Framework provides you with monitoring and error handling functionality. The *Error and Conflict Handler* (ECH) is a forward error handling mechanism provided by SAP NetWeaver. SAP Application Interface Framework supports ECH as an optional and alternative error handing functionality.

Integration

ECH messages can be monitored in SAP Application Interface Framework and error messages generated by SAP Application Interface Framework can be sent to ECH for processing.

You can define an interface assigned to an ECH component and business process in Customizing for *SAP Application Interface Framework* (transaction code `/AIF/CUST`) under *Interface Development* > *Additional Interface Properties* > *Assign ECH Component*.

Features

ECH supports forward resolution of asynchronous messages. It is integrated by default into standard Enterprise Services and enables clients to register errors and creates postprocessing orders accordingly via the postprocessing office (PPO).

In SAP Application Interface Framework, interface developers can use the SIW configuration and template delivered with SAP Application Interface Framework to create an interface supporting ECH (see Service Implementation Workbench Support [page 73]).

SAP Application Interface Framework Monitoring and Error Handling supports ECH

If the SAP NetWeaver PI message is processed by ECH, SAP Application Interface Framework's *Monitoring and Error Handling* can monitor and handle (for example, edit, save, restart, and cancel) PPO persistence in addition to the standard ECH monitoring transaction. The locking, version handing, and status controlling reuse ECH standard functions, so that they are consistent and synchronized between *Monitoring and Error Handling* in SAP Application Interface Framework and ECH monitoring. This integration scenario is independent of SAP Application Interface Framework runtime as *Monitoring and Error Handling* works on the PPO persistence layer utilized by ECH.

The interface determination function of SAP Application Interface Framework is used (see Interface Determination [page 88]).

An ECH-specific engine supports SAP Application Interface Framework *Monitoring and Error Handling* functions, for example, the following:

- You can select a PPO in *Monitoring and Error Handling* and easily navigate to the ECH monitoring tool
- You can navigate between the application log and data content (dual directions)
- You can mass edit and save

- You can mass restart and cancel

In order to be able to handle ECH messages in *Monitoring and Error Handling*, you have to maintain the following engines in Customizing for *SAP Application Interface Framework* under *Additional Interface Properties* > *Specify Interface Engines*:

- Application engine: ECH
  Handles asynchronous data messages received via a proxy from SAP NetWeaver PI but processed by the *Error and Conflict Handler* application. Each message has a corresponding postprocessing order.
- Persistence engine: ECH
  Handles the data content of messages processed by the *Error and Conflict Handler* application. Includes load data and save data as new version.
- Selection engine: ECH
  Handles message processed by the *Error and Conflict Handler* application instead of SAP Application Interface Framework. Data is selected from postprocessing order related tables.
- Logging engine: ECH
  Handles messages for a postprocessing order raised during *Error and Conflict Handler* application processing.

> **Note**
>
> Not all the functions, for example, data transformation and statistics, are supported.

ECH Supports SAP Application Interface Framework

A proxy developer can choose to use the ECH monitoring tool to monitor messages and handle errors and can choose to use SAP Application Interface Framework to process the message as follows:

- ECH collects errors generated by SAP Application Interface Framework in runtime
- ECH restarts or cancels any messages processed by the SAP Application Interface Framework runtime
- SAP Application Interface Framework *Monitoring and Error Handling* can load an original SAP NetWeaver PI message and the relevant application logs

An SAP Application Interface Framework-specific ECH action class is delivered to enable ECH runs with the SAP Application Interface Framework runtime. A proxy developer can use the SIW to switch on ECH support and can generate a proxy class implementation that calls the SAP Application Interface Framework-specific action class. This action class provides the following functions:

- Message processing is delegated to SAP Application Interface Framework
- Restart/Retry is delegated to SAP Application Interface Framework
- Cancel is handled by the SAP Application Interface Framework action class

In this integration scenario, SAP Application Interface Framework *Monitoring and Error Handling* works on top of SAP NetWeaver PI message persistence. However, the SAP NetWeaver PI message is processed by ECH and the status recorded in the SAP NetWeaver PI message persistence no longer allows you to change the SAP NetWeaver PI message data. You cannot handle (for example, edit, save, restart, and cancel) errors in *Monitoring and Error Handling* and you have to use the ECH monitoring tool (transaction code `ECH_MONI_SEL`).

In order to monitor such an SAP Application Interface Framework interface, in Customizing for *SAP Application Interface Framework* under *Additional Interface Properties* > *Specify Interface Engines*, select the following engines:

- Application Engine: Proxy

- Persistence Engine: Proxy
- Selection Engine: AIF Index Tables
- Logging Engine: AIF Application Log

4.9.5 Persistence Layer and Runtime

Use

In SAP Application Interface Framework, the following are available to provide you with fast and parallel message processing:

- **Persistence layer**
  The persistence enables the storage of interface messages. The SAP Application Interface Framework persistence layer comes in the following flavors:
  - An XML persistence that stores messages as a string representing an XML structure.
  - A structured persistence that stores messages in database tables representing the structure.

    > ⓘ Note
    >
    > The structured persistence is optimized for the SAP HANA database. This persistence profits from the compression and search capabilities on SAP HANA and eases the handling of key fields. For new interfaces, we recommend using the structured persistence instead of the XML persistence.
- **Runtime environment**
  Supports synchronous or asynchronous message processing. Its configuration is done using runtime configuration groups.

Integration

SAP Application Interface Framework supports a number of other persistence layers and runtimes that are part of SAP NetWeaver:

- PI services
- Web services (web service reliable messaging)
- `ALE/IDOC`

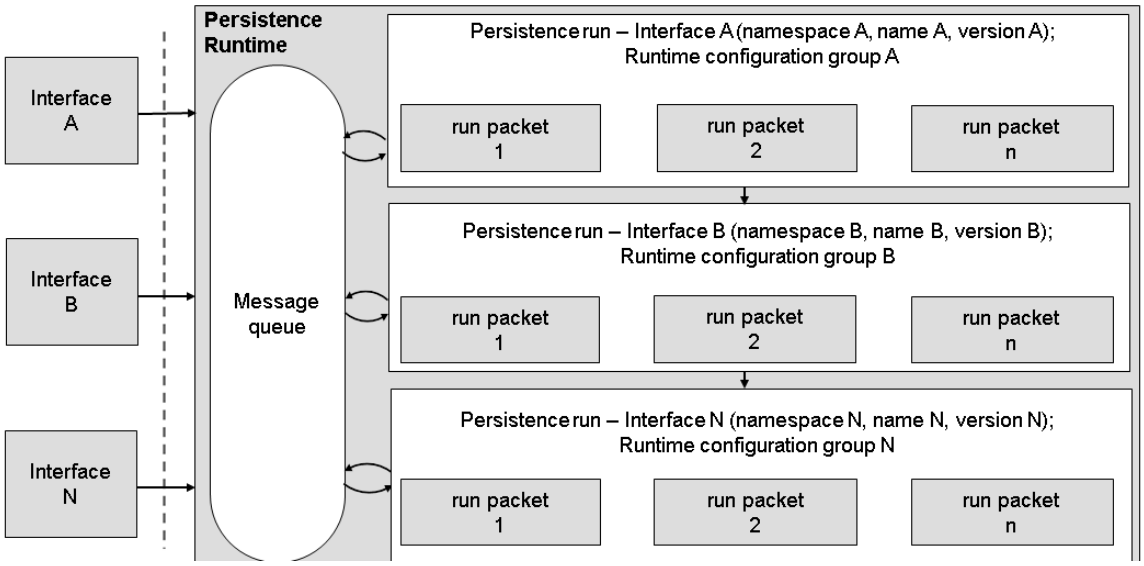
In addition, SAP Application Interface Framework has its own runtime that uses the standard job scheduling mechanisms to process data in background jobs. To monitor background jobs, use the standard transaction code `SM37`.

Features

Using the runtime environment of SAP Application Interface Framework, you can transfer messages directly from customer-specific reports. This works for both of SAP Application Interface Framework's own persistence

flavors. The customer-specific coding has to call the `TRANSFER_TO_AIF` static method (for single messages) or `TRANSFER_TO_AIF_MULT` (for multiple messages) of the `/AIF/ CL_ENABLER_XML` class.

Both methods have optional parameters to indicate which runtime configuration group to be used. According to settings in the runtime configuration group, the message is stored in a message queue to await synchronous or asynchronous processing. A runtime configuration group can be created and maintained in transaction `/AIF/PERS_CGR`.



SAP Application Interface Framework Runtime

The previous graphic shows message processing within the SAP Application Interface Framework runtime environment. Messages are stored in the message queue to be processed. The runtime environment creates persistence runs to manage message processing synchronously or asynchronously using background jobs. A run is created automatically for a specific interface, that is, for a certain combination of interface namespace, interface name, and interface version.

The messages are processed by run packets, which are also created automatically from the runs. The processing by the runs and run packets can be configured using the runtime configuration group, which you maintain with the `/AIF/PERS_CGR` transaction. In this transaction, you can define several runtime configuration groups with meaningful descriptions per namespace. To balance the load on your system over time, you can decide to schedule a run in dedicated background job and even to schedule the individual packets in dedicated background jobs.

The processing status of a message in the SAP Application Interface Framework runtime is stored in the `/AIF/PERS_QMSG` application table. The runtime processing status can be one of the following values:

- *N*: *New*
- *A*: *Assigned*
- *S*: *Started*
- *F*: *Finished*
- *R*: *Reprocess*

When a message is passed to the runtime, the processing status is set to *New*. A namespace and a name for a runtime configuration group is provided to manage message processing according to a certain configuration. The runtime environment sets the status according to the processing.

Activities

Transferring Messages to the AIF Persistence Layer

To store messages in the persistence layer of SAP Application Interface Framework, call the enabler for SAP Application Interface Framework, for example, in a report or function module. The enabler for SAP Application Interface Framework provides 2 static methods both of which save the messages in the persistence of SAP Application Interface Framework (structured or XML) and schedule the messages' processing.

Which of the following 2 methods you should call depends on whether you want to process a single message or multiple messages:

- `/AIF/CL_ENABLER_XML=>TRANSFER_TO_AIF`
  Call this method if you want to process a single message in SAP Application Interface Framework's runtime. In the following table, the method's most important input parameters are described:

Parameter Name	Description
`IS_ANY_STRUCTURE`	You can pass a single message to this importing parameter for processing. The single message should be the type of your raw data structure.
`IV_QUEUE_NS`	(Optional) Namespace of a runtime configuration group
`IV_QUEUE_TYPE`	(Optional) Runtime configuration group

- `/AIF/CL_ENABLER_XML=>TRANSFER_TO_AIF_MULT`
  Call this method if you want to process multiple messages. In the following table, the method's most important input parameters are described:

Parameter Name	Description
`IT_ANY_STRUCTURE`	You can pass a table that contains multiple messages to this parameter. The type of the table should be a table type of your raw data structure.
`IV_QUEUE_NS`	(Optional) Namespace of a runtime configuration group
`IV_QUEUE_TYPE`	(Optional) Runtime configuration group

> **Note**
>
> You can create a runtime configuration group in the `/AIF/PERS_CGR` transaction. If you don't specify a configuration group, the default runtime configuration group is used.

Creating Interfaces Utilizing AIF Persistence and Runtime

You need to create an interface in Customizing for *SAP Application Interface Framework* under *Interface Development* > *Define Interface*. The raw data structure that you maintain should be the same as the one you used when calling the enabler for AIF. If you used the `TRANSFER_TO_AIF_MULT` method, the raw data structure and the line type of your table type have to be the same.

For the structured persistence, you need to generate the database tables from the RAW structure using the *Table Generator for Structured Persistence* (transaction `/AIF/PERS_TBL_GEN`). You select the interface, a package and transport request, and define a prefix for the DDIC objects to be created. The program provides you with a preview of which RAW structure components map to which database tables and fields or columns. After execution, you get a log listing all objects that have been created.

> **Note**
>
> If you're using the structured persistence and you're defining key fields on the RAW structure, you can use the corresponding persistence table created by the *Table Generator for Structured Persistence* directly as multi index table. In this way, you don't need to create a separate index table holding redundant data and you can subsequently define key fields that also enable the selection on past messages. If you're using a non-SAP HANA database, you might consider adding an index on these columns of the used persistence tables, which are used in key fields.
>
> You define multi index tables in Customizing for *SAP Application Interface Framework* under *Error Handling* > *Define Interface-Specific Features* > *Define Key Fields for Multi.Search*.

To enable storing messages in one of SAP Application Interface Framework's own persistence flavors, in Customizing for *SAP Application Interface Framework*, you have to define the following engines for an SAP Application Interface Framework interface under *Interface Development* > *Additional Interface Properties* > *Specify Interface Engines*:

- **Structured persistence**
  - Application Engine: Structured Persistence
  - Persistence Engine: Structured Persistence
  - Selection Engine: Structured Persistence
  - Logging Engine: AIF Application Log
- **XML persistence**
  - Application Engine: XML
  - Persistence Engine: XML
  - Selection Engine: AIF Index Tables
  - Logging Engine: AIF Application Log

If you created several interfaces for one raw data structure, you have to maintain an interface determination in Customizing for *SAP Application Interface Framework* under *System Configuration* > *Interface Determination* > *Interface Determination for XML Interfaces*.

Creating a Runtime Configuration Group

You can create your own runtime configuration group. You can access the *Runtime Configuration Group* overview from the SAP Easy Access menu by choosing *Cross-Application Components* > *SAP Application Interface Framework* > *Administration* > *Customizing* > *Runtime Configuration Group* (transaction code `/AIF/PERS_CGR`). For a selected namespace, you can maintain the following:

- Runtime Configuration
  A three character identifier for a runtime configuration group
- Runtime Configuration Group Active
  If you do not select the check box, messages are saved to the persistence of SAP Application Interface Framework but they are not scheduled or processed.

> **Note**
>
> You can add new messages to the run of an inactive configuration group. Once the number defined in *Messages Per Run* is reached, the status of the run switches to *Ready* and the system creates a new run for further messages. To manually trigger the runs in status *Ready*, use the report `/AIF/PERS_RUN_EXECUTE`. You can also schedule this report as a background job.

- Run Scheduled
  If you select the checkbox, the runs are executed asynchronously in a job. Otherwise, the run is executed synchronously.
- Schedule Packages
  If you select the checkbox, the packages are processed asynchronously, each one in its own job. Otherwise, the packages are processed synchronously, which means in the same job as the run, if the run is scheduled.

Furthermore, you can specify the user under whom the job is to be processed (you can be this user if you have the correct authorization) and the user that released the scheduled batch job. You can specify the number of messages per package and the number of messages per run.

In SAP Application Interface Framework, if you do not define a runtime configuration group, the runtime processes messages based on the following defaults:

- Background job planning user is `SY-UNAME` (the user who passes the message to the AIF runtime)
- Background job execution user is `SY-UNAME` (the user who passes the message to the AIF runtime)
- 20 messages are processed per message packet
- 100 messages are managed per run

Restarting Messages

Use the transaction `/AIF/XML_RESTART` to restart messages from the structured persistence or the XML persistence. This method is an alternative for when you can't restart using the Error Handling and Monitoring of SAP Application Interface Framework.

If the enabler for SAP Application Interface Framework is terminated during the inbound processing of a package of messages, the job executing the enabler is canceled. In the Error Handling and Monitoring, you see messages in status *New*, *Started*, or *In Process*, which you can't restart from there. Instead, use `/AIF/XML_RESTART` to restart the stuck messages.

To restart your messages, perform the following steps:

1. Run transaction `/AIF/XML_RESTART`.
2. Select the relevant *Namespace*, *Interface Name*, and *Interface Version*.
3. In the next section, you can select specific information like time frame (*From* and *To*), *Message GUID*, *Create User*, and *Max. Number of Messages*.
   Select *From Index Table* if you want data entries to be retrieved from the standard index table. Alternatively, the data is retrieved from the queue table, for which you can enter a *Queue Message Status*.
4. Select one or more statuses for your message: *In Process*, *Application Errors*, or *Technological Errors*.
5. Finally, if necessary, select a *Runtime Configuration Group*.

Deleting Messages

You can delete messages from the persistence layer of SAP Application Interface Framework (XML and structured persistence) using *AIF Persistence Messages Deletion* (transaction `/AIF/PERS_DEL`).

> ⚠ Caution
>
> The messages deletion is irreversible. You cannot restore messages deleted using *AIF Persistence Messages Deletion*.

This program is to be used in exceptional cases only, for example, if you need to clean up a test system. You usually do not delete but archive finalized messages from time to time. For more information, see Data Archiving [page 111].

For executing this program, there is a two-person authorization concept in place. An administration user can authorize another user (but not himself) to execute the report for a specific interface at a specific date. The administrator needs to have the authorization object `/AIF/PERSD` assigned. He can authorize the other user using transaction `/AIF/PERS_DEL_AUTH`. In the log table `/AIF/PERS_D_AUTH`, you can see all authorizations, the users who created the authorizations, and when they created them.

4.9.6 File Adapter

Use

The file adapter supports the upload of files to the SAP Application Interface Framework, where they are processed and can be monitored by business users. You can read a file or files from different locations and map their content into raw structures of an interface.

Prerequisites

You have created interfaces in Customizing for the *SAP Application Interface Framework* under *Interface Development* > *Define Interfaces*.

Features

Customizing

In Customizing for the *SAP Application Interface Framework* under *System Configuration* > *Configure File Adapter*, you determine how the file data is mapped into raw structures.

To read a file from the file system, the minimum you need to specify is the following:

- *File Type*
  Specify one of the following:
  - Text file
  - Binary file (for example, Microsoft Excel)
- *File Content*

Specify one of the following:

- Flat structure
- Complex structure
  The file content can be mapped into a complex structure, which may contain header, substructures, and sub tables.
- XML
  Mapping is done via `XSLT` or `ST` (Simple Transformation). The transformation name must be stored in the *Transformation* field.
- User defined
  Customer-specific mapping can be done via a customer-specific function module, the name of which has to be stored in the *Mapping Function* field. The `/AIF/LFA_MAP_USER_SPEC_TEMPL` function module template is delivered.
- Excel (.xlsx)
  You can define mapping and conversions in the file adapter Customizing. The mapping can be done automatically or using a detailed mapping table.

Depending on the file content, you need to make several mandatory settings as follows:

File Content	Text Type	Length of Type Determination Value	Transformation	Mapping Function
Complex Structure	You must select a text type	You must enter a value		
Flat	You must select a text type	You must enter a value		
XML			You must enter a transformation name	
User defined				You must enter a customer-specific function module
Excel (.xlsx)				

For mapping, you need to specify at least a raw structure that describes the resulting data type (for example, a BAPI structure).

Upload Files to the SAP Application Interface Framework

Use the *Uploading Files to AIF* report (transaction `/AIF/LFA_UPLOAD_FILE`) to transfer **individual** files and to map them to a raw structure. You can access the report from the *SAP Easy Access* menu under *SAP Application Interface Framework* > *File Upload*.

The report provides the following features:

- Access files from a client PC or from the application server.
- Specify the file location either directly or using logical files (for more information, see transaction `FILE`).
- Split large data files into smaller blocks.
- Specify the use of the queues of the persistency runtime configuration.

- Have the file and the result structure displayed after upload.

Read Files from a Folder and Send to the SAP Application Interface Framework

Use the *Reading Files from a Folder and Sending Them to AIF* report (transaction `/AIF/LFA_CHECK_SEND`) to transfer and map **multiple** files into the SAP Application Interface Framework. You can access the report from the *SAP Easy Access* menu under *SAP Application Interface Framework* > *Administration* > *Tools* > *Mass File Upload from Server*.

The report provides the following special features:

- Access files from the application server only.
- After a successfully transfer, automatically move the files to another folder to avoid multiple selections and transfers of the same files.

Download Excel Files from the SAP Application Interface Framework

In the SAP Application Interface Framework, you can use the file adapter to upload Excel files, to merge data into Excel files you have uploaded, and to download these enhanced Excel files.

> **Note**
>
> The file adapter only supports the *.xlsx Excel format.

This function has the following features:

- Use the file adapter reports to upload existing Excel spreadsheets
- An SAP Application Interface Framework interface processes the data from an uploaded Excel file
- An action is called at the end of processing and triggers the download of the data via a second interface and the XML enabler
- In the second interface, an action is called at the end of processing that generates the output Excel file. To generate the Excel file, you can use the `/AIF/LFA_EXCEL_DOWN_MERGE` function module as an action function.

In Customizing for the *SAP Application Interface Framework* under *System Configuration* > *Configure File Adapter*, you specify the raw data structure that contains the output settings for downloading files.

Monitoring

Use the *Log of File Adapter* report (transaction `/AIF/SHOW_BLOCK`) to monitor the progress and success of file uploads done by the SAP Application Interface file adapter.

The report provides the following features:

- Select files by ID, standard interface file name, user, date, or configuration namespace and ID.
- Display logs and performance details on file and block level.

4.9.7 tRFC and qRFC

Use

SAP Application Interface Framework supports role-based monitoring of tRFC and qRFC interfaces. In *Monitoring and Error Handling*, business users can monitor messages and handle errors related to tRFC and qRFC interfaces.

Integration

SAP Application Interface Framework enables you to monitor tRFCs and qRFC transactions that have a "pending" (in error) status. It works on top of standard t/qRFC monitoring, functionalities, and persistence. Existing t/qRFC interfaces and function modules are not modified for monitoring in SAP Application Interface Framework.

In order to monitor tRFC and qRFC interfaces with SAP Application Interface Framework, you have to define SAP Application Interface Framework interfaces. For each t/qRFC function module, you have to create an SAP Application Interface Framework interface with the corresponding SAP structure to display all the required parameter values.

For both tRFC and qRFC, central monitoring (for example, in SAP ECC) and also local monitoring (for example, in SAP APO) are supported. You can configure from which system(s) the data is read and in which system the monitoring is done.

Prerequisites

Before you can make t/qRFC integration settings, you have used the `SM59` transaction code to define RFC destinations that link the external systems with SAP Application Interface Framework.

Features

The following are the major components of the tRFC/qRFC integration into SAP Application Interface Framework and each component is described in detail in the following sections or linked topics:

- A *Function Module Generator* report (transaction code `/AIF/RFC_FUNC_GEN` for single generation and transaction code `/AIF/RFC_MASS_GEN` for mass generation)
  Fore more information, see tRFC and qRFC Function Module Generator [page 79].
- A read function
- A *Data Transfer* report
- Customizing
- Engines
  See tRFC [page 155] and qRFC [page 158] for lists of the appropriate engines.

Read Function

The read function reads the following data that is required in SAP Application Interface Framework:

- Logical units of work (LUW)
- Parameters of the called function modules
- Additional application logs written by the functions

As the read function gets the list of the already known LUWs (the transaction IDs with timestamp), it determines which LUWs are new, changed, or deleted; the parameters and additional data is only read for new or changed data.

The read function writes the values of the parameters into the specific SAP structures that are generated by the *Function Module Generator* (see tRFC and qRFC Function Module Generation [page 79]).

Data Transfer

The read function is called by the *Data Transfer* report, which saves the data in the persistency of SAP Application Interface Framework and converts it into the required data format for *Monitoring and Error Handling*.

The *Data Transfer* report is used to support several technologies including tRFC, qRFC, CIF postprocessing, and batch input (see Data Transfer [page 172]).

Customizing

- In Customizing for *SAP Application Interface Framework* under *System Configuration* > *Configure Data Transfer*, you make settings to enable the *Data Transfer* report to transfer t/qRFC data into the persistence of SAP Application Interface Framework.
- You have to assign engines to interfaces in Customizing for *SAP Application Interface Framework* under *Interface Development* > *Additional Interface Properties* > *Specify Interface Engines*.
- Use Customizing for *SAP Application Interface Framework* to make detailed settings for the generated interface.

Engines

See tRFC [page 155] and qRFC [page 158] for the appropriate engine settings.

The required functions for the application engine for the t/qRFC integration are:

- Restart
- Cancel
- Debug
  The *Debug* pushbutton works dynamically as follows:
  - tRFC
    Only outbound queues are supported
  - qRFC
    Both inbound and outbound queues are supported

More Information

For more information about transferring t/qRFC related data to the persistence of SAP Application Interface Framework, see Data Transfer [page 172].

4.9.7.1 tRFC

Use

SAP Application Interface Framework supports monitoring of tRFC interfaces with the following two scenarios:

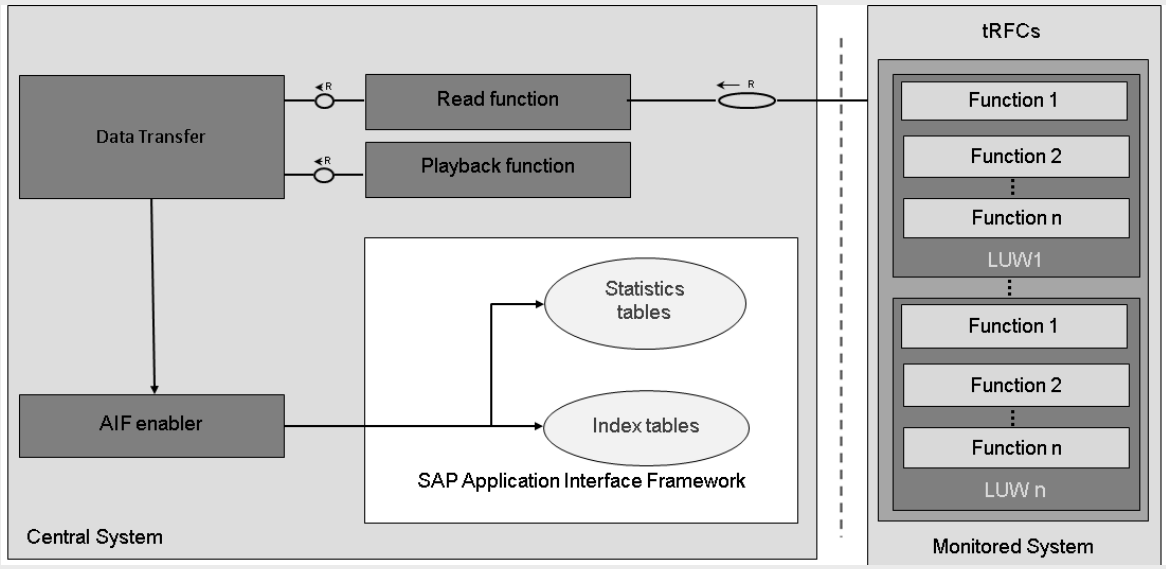
- Monitor tRFCs with SAP Application Interface Framework index table and statistics table
  All the SAP Application Interface Framework standard functionalities, for example, defining key fields-based authorizations and roles and delegating monitoring to the corresponding responsible person, are fully supported.
- Monitor tRFCs using `ARFCSSTATE` standard table
  Business users can monitor tRFC data in *Monitoring and Error Handling* without complex Customizing.

Integration

The solutions consist of the following parts:

- Monitor tRFCs by index table and statistics table
  - tRFC engine (for restart/cancel/debug functions)
  - *Function Module Generator* report, which generates the following:
    - Playback function modules for reading tRFC data
    - SAP structures for the supported tRFC function modules
    - Corresponding SAP Application Interface Framework interface
  - A read function that can be executed remotely on every system
  - A *Data Transfer* report that can be scheduled
- Monitor tRFCs by standard table `ARFCSSTATE`
  - tRFC engine (for restart/cancel/debug functions)
  - *Function Module Generator* report (transaction code `/AIF/RFC_FUN_GEN`), which generates the following:
    - Playback function modules for reading tRFC data
    - SAP structures for the supported tRFC function modules
    - Corresponding SAP Application Interface Framework interface

Example



tRFC Integration with Monitoring by Index Table and Statistics Table

The graphic shows an example of transferring tRFC data to SAP Application Interface Framework for monitoring tRFCs by index table and statistics table.

In order to use SAP Application Interface Framework as a central tRFC monitoring platform, the data of the sent messages (LUWs, status / error message, function module payload) in the tRFC monitored system has to be transferred into the central monitoring system. The *Data Transfer* report needs to be triggered to get the current tRFC status. The report passes data and the log messages to the SAP Application Interface Framework enabler (AIF enabler). For every LUW, the AIF enabler creates a message consisting of an entry in the AIF persistency or the AIF RFC local persistency, some table entries in the corresponding index tables, and an application log. When the LUW is deleted, these entries are also deleted.

Note

If monitoring tRFCs by status table, the SAP Application Interface Framework messages come from the standard **ARFCSSTATE** table and the index table and statistics table are not used. In this case, the tRFC persistency calls the playback function module directly to get the data of the SAP structure or raw structure.

Features

Monitoring tRFCs by index table and statistics table supports the following requirements:

- *Interface Monitor*
  - Role-based overview
  - Mass restart/cancel in *Message Summary*
- *Monitoring and Error Handling*
  - Select by key fields

- Display data content
- Display error log
- Key field-based authorization check
- Restart/Cancel by LUW
- Debug possibility

Monitoring tRFCs by standard ARFCSSTATE table supports the following requirements in *Monitoring and Error Handling*:

- Display data content
- Display error log
- Restart/Cancel by LUW
- Debug possibility

Activities

To generate objects and create related Customizing for monitoring tRFCs, in the *Function Module Generator* report, select *tRFC (AIF Index Table)* or *tRFC (tRFC Status Table)* depending on the monitoring scenario you want to use.

In Customizing for *SAP Application Interface Framework* under *System Configuration* > *Configure Data Transfer*, you can make settings to enable the *Data Transfer* report to transfer tRFC data into the persistence of SAP Application Interface Framework and thus support the integration with tRFC. When you run the *Data Transfer* report with a specific selection set you have defined in this activity, only the logical systems assigned to the selection set are reached and only the tRFC function modules defined in the specific selection set are transferred.

Engines

You have to assign engines to interfaces in Customizing for *SAP Application Interface Framework* under *Interface Development* > *Additional Interface Properties* > *Specify Interface Engines* as follows:

- tRFC with index table and statistic table
  - Application Engine: tRFC
  - Persistence Engine: qRFC/tRFC
  - Selection Engine: AIF Index Table
  - Logging Engine: AIF Application Log
- tRFC with status table
  - Application Engine: tRFC Status Table
  - Persistence Engine: tRFC Status Table
  - Selection Engine: tRFC Status Table
  - Logging Engine: tRFC Status Table

More Information

For more information about transferring tRFC data to SAP Application Interface Framework persistence, see Data Transfer [page 172].

For more information about the *Function Module Generator* report, see, tRFC and qRFC Function Module Generation [page 79].

4.9.7.2 qRFC

Use

SAP Application Interface Framework supports role-based monitoring of qRFC interfaces.

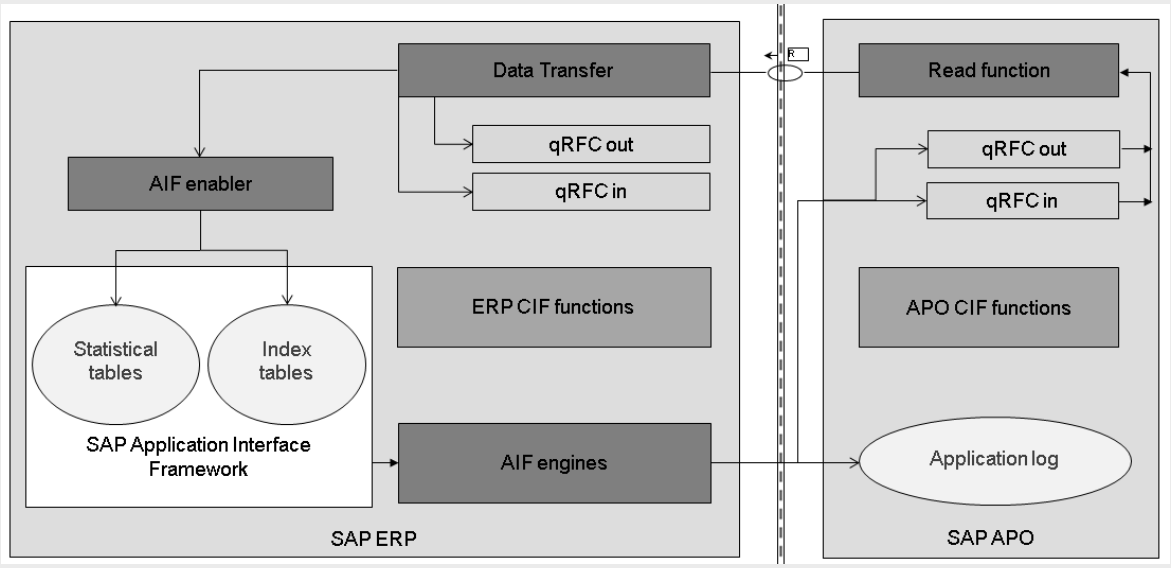
Integration

The solution consists of the following parts:

- qRFC engine (for restart/cancel/debug functions)
- *Function Module Generator* report (transaction code `/AIF/RFC_FUNC_GEN`), which generates the following:
  - Playback function modules for reading qRFC data
  - SAP structures for the supported qRFC function modules
  - Corresponding AIF interface

  For more information, see tRFC and qRFC Function Module Generation [page 79].
- Customizing
  - To configure qRFC integration in SAP Application Interface Framework
  - To specify the qRFC engines
- A read function that can be executed remotely on every system
  For more information, see the “Read Function” section in tRFC and qRFC [page 153].
- A *Data Transfer* report that can be scheduled

Example



System Architecture for a qRFC Call from SAP ERP to SAP APO

The graphic shows the overall system architecture using, as an example, a qRFC call from SAP ERP to SAP APO with monitoring in SAP ERP, which is defined in this case as a central qRFC monitoring system.

In order to use SAP Application Interface Framework as a central qRFC monitoring platform, data about the sent messages available in the qRFC monitors (LUWs, status/error message, parameters) has to be transferred to SAP ERP where SAP Application Interface Framework is located.

When the *Data Transfer* report is triggered, the read function is executed and selects the queues in the connected SAP APO system. For every logical unit of work (LUW), the *Data Transfer* report uses the AIF enabler to create an SAP Application Interface Framework message consisting of an entry in the XML persistency or local AIF RFC persistency, some table entries in the corresponding index tables, and an application log.

Features

SAP Application Interface Framework supports the following for qRFC integration:

- The display of a role-based overview in the *Interface Monitor*
- The selection by key fields in *Monitoring and Error Handling*
- The definition of authorizations by special key fields

Activities

In Customizing for *SAP Application Interface Framework* under *System Configuration* > *Configure Data Transfer*, you can make settings to enable the *Data Transfer* report to transfer qRFC data into the persistence

of SAP Application Interface Framework and thus support the integration with qRFC. When you run the *Data Transfer* report with a specific selection set you have defined in this activity, only the logical systems assigned to the selection set are reached and only the qRFC queues defined in the specific selection set are transferred.

Engines

You have to assign engines to interfaces in Customizing for *SAP Application Interface Framework* under *Interface Development* > *Additional Interface Properties* > *Specify Interface Engines* as follows:

- Application Engine: qRFC
- Persistence Engine: qRFC/tRFC
- Selection Engine: AIF Index Table
- Logging Engine: AIF Application Log

More Information

For more information about transferring qRFC data into SAP Application Interface Framework persistence, see Data Transfer [page 172].

4.9.8 OData Support

Use

SAP Application Interface Framework supports the monitoring of error messages for OData services. For erroneous data messages, you can display the error messages and the content of the data message in the source format. Correcting the contents of the data messages, restarting, or cancelling data messages of the OData service using the SAP Application Interface Framework is not supported.

> **Note**
>
> SAP Gateway provides a pure error log, that contains no warnings or success messages.
>
> The visibility of the error messages and of data message contents depends on the global settings of the SAP Gateway error log in transaction `/IWFND/ERROR_LOG`. If, for example, the error log level is set to *Secure* (default), all business-relevant data are filtered out and only technical errors are available.

Activities

1. To receive up-to-date error logs of OData services in SAP Application Interface Framework, select the services in the program `/AIF/ODATA_TRANSFER_JOB` and schedule the program to run periodically. You can look up the available oData services in transaction `/IWFND/MAINT_SERVICE`.
2. Create an interface with a raw structure containing at least a component of type `/AIF/ODATA_STD_RAW_FLDS_S`. This component contains the standard fields of the OData error log, for example, the service name, the user, and a timestamp. A mapping is not required for an OData interface.

You define the interface in Customizing of SAP Application Interface Framework under *Interface Development* > *Define Interfaces*.

3. For the OData interface, specify the following interface engines:
   - Application Engine: OData application engine
   - Persistence Engine: OData persistence engine
   - Selection Engine: AIF Index tables
   - Logging Engine: OData logging engine

   You specify the engines in Customizing of SAP Application Interface Framework under *Interface Development* > *Additional Interface Properties* > *Specify Interface Engines*.
4. Specify the OData services and elements, for example entity sets, whose error logs you want to monitor using this interface.

   > **Note**
   >
   > Entering an element is optional and has no effect, if the error log level is set to *Secure* in SAP Gateway.

   You specify the services in Customizing of SAP Application Interface Framework under *Interface Development* > *Additional Interface Properties* > *Define OData Services*.
5. Define a message index table that at least includes the structure `/AIF/ODATA_STD_IDX_FLDS_S`. For an example of such an index table, see the template table `/AIF/T_ODATA_IDX`.
   Add your index table to the OData interface in Customizing of SAP Application Interface Framework, under *Error Handling* > *Define Namespace-specific Features* > *Define Interface-specific Features* > *Message Idx Table*.
6. Optionally, you can decide to enable the business users to display the source contents of the OData data message, that is, the response and the request. If enabled, there is the *Display Contents in Source Format* button available in the *Data Messages* view of the *Monitoring and Error Handling*.

   > **Caution**
   >
   > There is always the complete data of the message displayed in the source XML format. You cannot hide parts of request or response to protect, for example, personal data. If you need to protect certain data, do not enable this function.

   To enable the *Display Contents in Source Format* button, in Customizing of SAP Application Interface Framework, select *Error Handling* > *Define Namespace-specific Features* > *Define Interface-specific Features* > *Displ source button*.

4.9.9 Core Interface Postprocessing

Use

SAP Application Interface Framework supports role-based monitoring of Core Interface (CIF) postprocessing records with the following two scenarios:

- An SAP Advanced Planning and Optimization (SAP APO)-centric scenario

You can monitor multiple SAP ERP systems connected to an SAP APO system in a single SAP Application Interface Framework instance in the SAP APO system.

- An SAP ERP-centric scenario
  You can monitor multiple SAP APO systems connected to an SAP ERP system in a single SAP Application Interface Framework instance in the SAP ERP system.

In *Monitoring and Error Handling* of SAP Application Interface Framework, business users can monitor messages and handle errors related to postprocessing CIF.

> **Example**
>
> You use CIF interfaces to exchange master data and transaction data between your central SAP ERP system, where SAP Application Interface Framework is located, and several SAP APO systems that are used for production planning. You authorize business users to monitor errors related to CIF postprocessing in SAP Application Interface Framework according to the individual plants for whose production planning they are responsible.

Integration

SAP Application Interface Framework enables you to monitor CIF postprocessing records that have one of the following processing statuses:

- *Still for Processing*
- *Processed*
- *Obsolete (Set Manually)*
- *Obsolete (Set Automatically by System)*
- *Retransfer Not Allowed*

> **Note**
>
> Each of these processing statuses is mapped to a message status in SAP Application Interface Framework, for example, *Still for Processing* is mapped to *Application Errors* and *Processed* is mapped to *Processed Successfully*.

Standard CIF postprocessing functionality is integrated into monitoring in SAP Application Interface Framework.

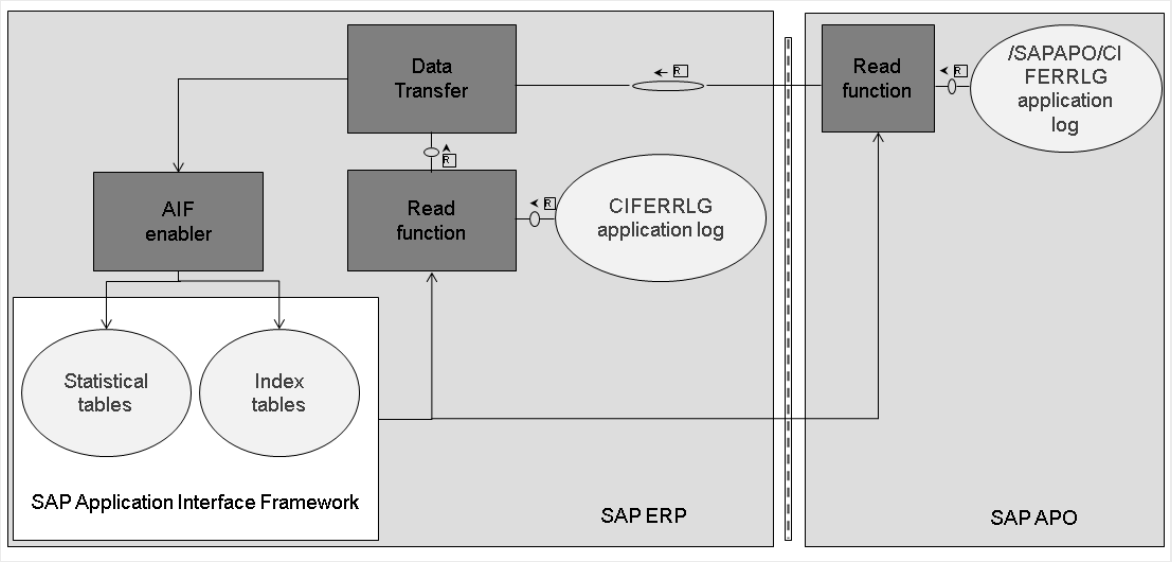
Existing CIF interfaces and function modules are not modified. To monitor CIF interfaces with SAP Application Interface Framework, you have to define SAP Application Interface Framework interfaces. A fixed raw data structure, based on the structure of the CIF postprocessing error log, is used for any number of SAP Application Interface Framework interfaces for CIF postprocessing.

The solution consists of the following major parts:

- A standard single index table for CIF postprocessing
- A raw data structure for CIF postprocessing
- A transfer engine for CIF postprocessing
- A read function for reading local and remote CIF postprocessing records and application logs
- A *Data Transfer* report (see Data Transfer [page 172])

- An enabler for CIF postprocessing
- Application and persistence engines for CIF postprocessing

Example



SAP ERP-Centric Scenario

The graphic shows an example of transferring CIF postprocessing records from SAP APO to SAP ERP for *Monitoring and Error Handling* in SAP Application Interface Framework.

SAP Application Interface Framework enables *Monitoring and Error Handling* by filtering on key attributes (for example, source system and R/3 object type) from the error log of the CIF postprocessing records. It is dependent on the settings in Customizing for *SAP Application Interface Framework* under *Error Handling* > *Define Interface-Specific Features* > *Define Key Fields for Multi. Search*.

CIF postprocessing records and the corresponding application logs are remotely selected from the connected SAP APO system.

This data is passed back to the *Data Transfer* report, which triggered the transfer. An SAP Application Interface Framework enabler is then used to update and store the CIF postprocessing data, for example, in the index table, for *Monitoring and Error Handling*.

Features

SAP Application Interface Framework supports the following:

- *Interface Monitor*
  - The display of a role-based overview
  - Mass restart/cancel in *Message Summary*
- *Monitoring and Error Handling*
  - Selection by key fields

- Error log display in Data Content View
- Application log display
- Definition of authorizations by special key fields (for example, plant)
- Restart/Cancel

> **Note**
>
> Independent of the monitoring system (SAP ERP or SAP APO), you can restart the message. Depending on the error location, the action that is executed is either *Send to APO* or *Send to R/3*.

Activities

Integration Configuration

In Customizing for *SAP Application Interface Framework* under ▶ *System Configuration* ▶ *Configure Data Transfer* ▶, you can make configuration settings for data transfer. These configuration settings define a set of parameters to be used by the *Data Transfer* report when transferring CIF postprocessing records to SAP Application Interface Framework.

> **Example**
>
> You make configuration settings so that the *Data Transfer* report only transfers CIF postprocessing records that are located on the SAP ERP system (*Read ERP* checkbox selected) and that only relate to production orders (*R/3 Object Type*).

For more information about this activity, read the relevant Customizing documentation.

Define Interfaces

When creating your interface, you have set structure `/AIF/S_CIFPP_RAW` as the raw and SAP structure. Furthermore, it is important that you assign the CIF postprocessing-specific single index (`/AIF/T_CIFPP_IDX`) table to your interface. You do this in Customizing for *SAP Application Interface Framework* under ▶ *Error Handling* ▶ *Interface Specific Features* ▶.

Engines

You have to assign engines to interfaces in Customizing for *SAP Application Interface Framework* under ▶ *Interface Development* ▶ *Additional Interface Properties* ▶ *Specify Interface Engines* ▶ as follows:

- Application Engine: CIF Postprocessing
- Persistence Engine: CIF Postprocessing
- Selection Engine: AIF Index Table
- Logging Engine: AIF Application Log

More Information

For more information about transferring CIF postprocessing records to the persistence of SAP Application Interface Framework, see Data Transfer [page 172].

4.9.10 Batch Input

Use

In SAP Application Interface Framework, you can monitor and process batch input sessions, which contain all the information and data necessary for data transfer using batch input.

> ⓘ Note
>
> Batch input is also known as batch data communication (BDC).

In SAP Application Interface Framework, there are the following 2 scenarios in which you can monitor and process batch input sessions:

- Batch input session scenario 1: See Monitor Existing Batch Input Sessions in Monitoring and Error Handling [page 166]
- Batch input session scenario 2: See Process BI Data with Data Transfer; Write AIF Tables with Enable [page 168]

> ⓘ Note
>
> In SAP Application Interface Framework, batch input sessions are only visible in *Monitoring and Error Handling* when they are in an error state. Once they are processed successfully, they are deleted and are not visible there. If the *Keep Session* checkbox is selected, the session is visible after it has been processed successfully.
>
> In SAP Application Interface Framework, you can only restart those batch input sessions in *Monitoring and Error Handling* that you created in the Batch Input.

Integration

In order to display batch input sessions in *Monitoring and Error Handling*, some Customizing settings are required, an interface needs to be created in SAP Application Interface Framework, and a raw and an SAP data structure have to be maintained.

Customizing

In Customizing for *SAP Application Interface Framework* under ▶ *Interface Development* ▶ *Additional Interface Properties* ▶ *Specify Interface Engines* ▶, you need to make settings for the engines for each of the scenarios where batch input technology is supported by SAP Application Interface Framework.

See the individual scenarios for lists of the appropriate engines and for more information about other Customizing settings.

More Information

For more information about the *Batch Input Structure Generation and Interface Definition* report, see Batch Input Structure Generation and Interface Definition [page 77].

For more information about transferring batch input related data to the persistence of SAP Application Interface Framework, see Data Transfer [page 172].

4.9.10.1 Monitor Existing Batch Input Sessions in Monitoring and Error Handling

Use

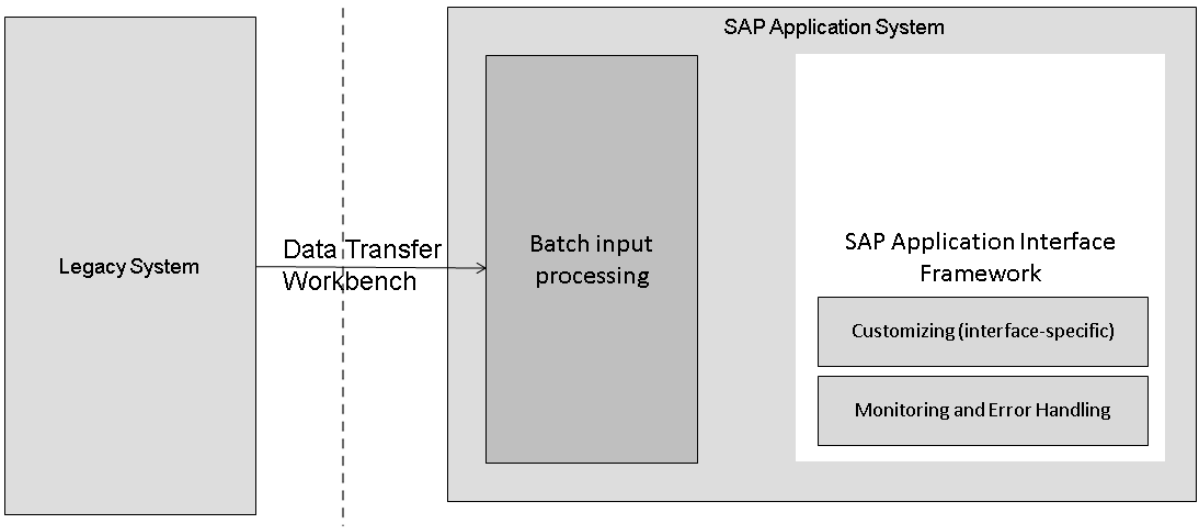
In this scenario, SAP Application Interface Framework is not involved in processing batch input data in batch input sessions.

Integration

In order to display batch input sessions in *Monitoring and Error Handling*, you have to create a raw and an SAP data structure and you need to define an SAP Application Interface Framework interface.

The raw and SAP data structures are the same for a batch input recording. In order to create the structure and create an SAP Application Interface Framework interface, an *AIF Batch Input Structure Generator* report (transaction code `/AIF/BDC_GEN`) is provided (see Batch Input Structure Generation and Interface Definition) [page 77].

In order to display the batch input session data belonging to the corresponding SAP Application Interface Framework interface, make settings in Customizing for *SAP Application Interface Framework* under *System Configuration* > *Interface Determination* > *Assign Batch Input Session and Creator*. The batch input session belonging to one SAP Application Interface Framework interface is determined by the batch input session name and creator.



Monitoring of Batch Input Sessions in Monitoring and Error Handling

In the scenario for processing batch input data shown in the graphic above, SAP Application Interface Framework is not involved in processing the batch input data from the legacy system.

With the batch input method, an ABAP program reads the legacy system data that is to be entered in the SAP system and stores the data in a batch input session (transaction code `SM35`).

You can process batch input sessions in the background processing system. During processing, the batch input data is written into the standard batch input tables (for example, the `APQI` and `APQD` tables). Since no entries are written to the SAP Application Interface Framework tables in this scenario, the batch input data is selected from the batch input tables in order to display the batch input sessions in *Monitor and Error Handling*.

Batch input sessions are maintained by the system in the batch input queue. The queue information needs to be selected from the `APQI` table. The batch input data is selected from the `BDC_OBJECT_READ` function module. It is necessary to transform the data from the `APQI` table and the queue data into the structure defined as the raw structure in SAP Application Interface Framework. In *Monitoring and Error Handling*, this structure is displayed in the Data Structure view. The data for the Log Message view is selected from the `BDC_OBJECT_READ` function module.

Features

In this scenario, the following are **not** possible in SAP Application Interface Framework:

- Displaying messages in the Interface Monitor
  There are no entries in the AIF index and statistic tables as batch input data is not processed by SAP Application Interface Framework.
- Creating alerts using and sending E-mail notifications
- Defining an interface-specific selection screen
- Using a key fields based search in *Monitoring and Error Handling*
  There are no index table entries.

Activities

In Customizing for *SAP Application Interface Framework* under ▶ *System Configuration* ▶ *Interface Determination* ▶ *Assign Batch Input Session and Creator* ▶, assign the batch input session and creator for batch input interfaces for which you want to use this monitoring scenario.

In order to display batch input sessions in *Monitoring and Error Handling*, you have to assign engines to interfaces in Customizing for SAP Application Interface Framework under ▶ *Interface Development* ▶ *Additional Interface Properties* ▶ *Specify Interface Engines* ▶ as follows:

- BDC application engine
- BDC persistency engine
- Logging engine for BDC message header
- Selection engine for BDC error segments

4.9.10.2 Process BI Data with Data Transfer; Write AIF Tables with Enabler for BI

Use

In this scenario, batch input sessions are processed by standard processing. SAP Application Interface Framework is not involved in processing batch input sessions

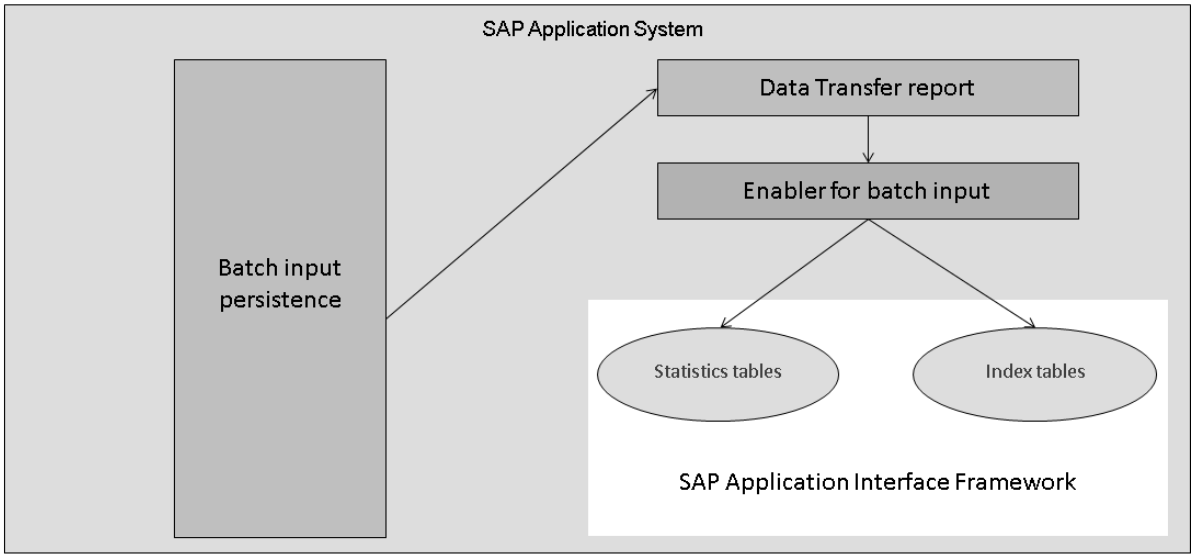
The data of existing batch input sessions can be passed to SAP Application Interface Framework via the *Data Transfer* report. The report calls the enabler for batch input to write SAP Application Interface Framework-specific table entries.

Integration

Use the *AIF Batch Input Structure Generator* report (transaction code `/AIF/BDC_GEN`) to create a raw and an SAP data structure and to define an SAP Application Interface Framework interface (see Batch Input Structure Generation and Interface Definition [page 77]).

The raw and SAP data structures are the same and are created for a batch input recording.

Some settings in Customizing for *SAP Application Interface Framework*, for example, for structure mappings or actions, are not relevant since the batch input session is not processed with the SAP Application Interface Framework runtime.



Process Batch Input Sessions with Data Transfer, Write AIF Tables with Enabler

In the scenario for processing batch input data shown in the graphic above, the batch input sessions are processed as normal.

To get the batch input data into SAP Application Interface Framework, the *Data Transfer* report can be configured to read the data from the batch input persistence to update the SAP Application Interface Framework index and statistic tables and to create alerts. This is done by the AIF enabler for batch input that is called by the *Data Transfer* report.

Interface determination for batch input sessions is performed for a combination of the batch input session name and the name of the creator of the batch input session. The raw structure is determined based on the combination of the batch input session and creator. With the help of the selected raw structure, interface determination is executed (see Interface Determination [page 88]).

If an interface-specific selection screen was defined in Customizing, in *Monitoring and Error Handling*, the user can select messages. The messages can be grouped by key fields in the Data Messages view (see Data Messages View [page 40]). If a message is selected and an AIF application log is written, this application log is loaded into the Log Messages view (see Log Messages View [page 43]). Since the batch input session was not processed with the AIF runtime, the data link between the Log Messages view and the Data Content view does not work.

Features

In this scenario, the following are possible in SAP Application Interface Framework:

- Displaying messages in the Interface Monitor
- Creating alerts using and sending e-mail notifications
  Recipients can be defined for alert management
- Defining an interface-specific selection screen
- Defining interface-specific index tables

Activities

In Customizing for *SAP Application Interface Framework* under *System Configuration* > *Interface Determination* > *Define Pre-Interface Determination for Batch Input*, you can define an SAP data structure to be used for a specific batch input session name and creator combination for interface determination.

You have to assign engines to interfaces in Customizing for SAP Application Interface Framework under *Interface Development* > *Additional Interface Properties* > *Specify Interface Engines* as follows:

- BDC application engine
- BDC persistency engine
- Logging engine for application logs
- Selection engine for AIF index tables

4.9.11 Cross Technology

There are the following cross technology features in the area of technology support:

- Preprocessing [page 170]
  This applies to the support of the following technologies:
  - ABAP Proxies [page 128]
  - IDoc Support [page 129]
- Postprocessing Using Customer Exits (BAdIs) [page 171]
- The Data Transfer [page 172] report
  This applies to the support of the following technologies:
  - tRFC and qRFC [page 153]
  - Core Interface Postprocessing [page 161]

4.9.11.1 Preprocessing

Use

In scenarios where SAP Application Interface Framework supports IDoc and Proxy technology, preprocessing can be required, for example, in the following cases:

- You have a standard inbound proxy or existing custom inbound proxy where you want to be able to use the mapping functionalities of SAP Application Interface Framework but you do not want to change the existing proxy class implementation.
- You have a standard or existing custom-specific IDoc where you want to use some mapping functionalities of SAP Application Interface Framework.
  Additionally, you only want to modify or check some data before the existing function is executed.

In the preprocessing mode, SAP Application Interface Framework executes the mapping logic on top of IDoc or Proxy raw message data. Therefore, there is limited maintenance effort in Customizing.

Prerequisites

You have defined interfaces in Customizing for *SAP Application Interface Framework* under *Interface Development* > *Define Interfaces*.

In order to be able to use preprocessing for proxy interfaces, you have to implement note 1828776.

Features

If you have an existing interface (IDoc or proxy), you often only have some fields that you want to modify or check in SAP Application Interface Framework before the standard logic is called. The mappings and checks defined in SAP Application Interface Framework are executed on top of the raw structure. Therefore, you only need to define, for example, value mappings, checks, conditions, and fix values for those fields that you want to change. Simple 1 to 1 field mappings, for example, are not needed.

> **Note**
>
> If you have nested tables, you still have to create indirect mappings.

Proxies

SAP Application Interface Framework is called implicitly before the proxy method is called. The only thing that is needed here is an AIF interface and the required mappings and checks. Once the mapping in SAP Application Interface Framework is successful, the proxy method is automatically called.

IDocs

You have to configure the ALE runtime to call the generic inbound process function of SAP Application Interface Framework. You need to create the mappings and checks you want to perform. You also need an action that calls the standard IDoc process function.

Activities

To activate preprocessing, select the *Preprocessing* checkbox in Customizing for *SAP Application Interface Framework* under *Interface Development* > *Define Interfaces*.

4.9.11.2 Postprocessing Using Customer Exits (BAdIs)

You can trigger additional steps (for example, sending an acknowledgment) after the status of a message has changed. For example, you can trigger the sending of an information message in case a message has been canceled.

You can implement a custom postprocessing procedure and integrate your implementation with SAP Application Interface Framework using a customer exit, particularly, a BAdI (Business Add-In).

Creating BAdI Implementation

Perform the required steps to implement your BAdI (using transaction *SE19*). For *Enhancement Spot*, select */AIF/GENERAL*.

Specify the settings according to the implementation of your postprocessing logic.

Make sure that as *BAdI Definition* you choose */AIF/POST_MESSAGE_PROCESS*.

Adding Filter Values and Activating BAdI Implementation

After you've implemented the postprocessing logic, edit the filter values (also using transaction *SE19*).

For example, you want to trigger postprocessing for all interfaces of namespace `ZAIF` when messages are canceled. In this case, specify the following filter value combination:

You can use the wildcard character (*****) for filter values in case you want to activate the BAdI implementation for multiple interfaces.

To trigger postprocessing when a message goes into successful status, for filter **STATUS** specify value **S**.

To trigger postprocessing when a message is canceled, for filter **STATUS** specify value **C**.

Filter Name	Comparator	Filter Value
NS	**=**	**ZAIF**
STATUS	**=**	**C**

> ⓘ Note
>
> Postprocessing is working for all AIF interfaces with *Selection Engine* set to *AIF Index Tables*. You can find this setting using transaction */AIF/CUST*. Select *Interface Development* > *Additional Interface Properties* >. For your interface settings for the Selection Engine, specify *Interface Engines*.

4.9.11.3 Data Transfer

Use

The *Data Transfer* report reads data in the persistence of a supported technology, locally or remotely, and saves the data to the persistence of SAP Application Interface Framework. It then uses an enabler in SAP Application Interface Framework to create entries in index tables and statistics tables and to create alerts.

You can use the *Data Transfer* report to support integration with the following technologies:

- tRFC

- qRFC
- CIF Postprocessing
- Batch Input

Integration

Customizing

In Customizing for *SAP Application Interface Framework* under *System Configuration* > *Configure Data Transfer*, you make the following settings to enable the *Data Transfer* report to transfer data from supported technologies to the persistence of SAP Application Interface Framework:

- Define logical systems that point to external/internal local systems that do the monitoring by using the monitoring functions of SAP Application Interface Framework
- Define a name for the selection set, which is composed of a set of external or local systems, to be used by the transfer report.
  For each selection set, you define a set of selection options per selected logical system. When the *Data Transfer* report runs using a specific selection set, only the logical systems assigned to the selection set is reached.
  You can specify how a job is activated (for example, *Asynchronous - Batch Job*) and the name of the user who released a scheduled batch job.
  You can also specify how much time should elapse before the *Data Transfer* report can be run again for a specific selection set.
- Assign an interface to a selection set
  You can only assign one interface to one selection set but you can assign a selection set to several interfaces.
- Settings specific to the supported technologies:
  - Settings specific to tRFC, qRFC, and CIF postprocessing
    - Specify maximum number of transactions to be transferred
    - Specify the persistence type you want to use for storing transaction data (not available for CIF postprocessing)
    - Assign a logical system
    - tRFC-specific setting
      - Specify a function module to define the tRFC transactions to be transferred
    - qRFC-specific settings
      - Define the queues to be transferred
      - Select to have data from the inbound queue, the outbound queue, or both transferred
    - CIF postprocessing-specific settings
      - Read SAP APO and/or SAP ERP
        Specify that error logs from SAP APO and/or SAP ERP are transferred
      - R/3 Object Type
        Specify the object types for which error logs are transferred, for example, sales order or production order
      - Processing Status
        Specify that only postprocessing records with a certain processing status are transferred, for example, *Still for Processing* status

- Settings specific to batch input
  - Specify a batch input session
  - Assign a user name

Features

The *Data Transfer* report can be scheduled regularly in the central system.

The report is also executed if an interface, which is assigned to a selection set, is selected in either *Monitoring and Error Handling* or the *Interface Monitor*. Depending on the interfaces, the report is triggered with the corresponding selection set.

The report uses a read function to read which systems should be checked for data from a configuration table, where the logical systems and RFC destinations of the connected systems are maintained.

The existing transactions of each logical system are passed to the read function, which determines if transactions are new, restarted, or deleted since the last update.

The report takes the transactions data and uses the enabler to update the data in the persistence of SAP Application Interface Framework. Also, the enabler can update index tables, which are the basis for e-mail notification, authorizations and role-based display in the *Interface Monitor*.

An interface is created (with corresponding data structures) in SAP Application Interface Framework, so that users can display data and the logs in the central monitoring system.

For batch input, the *Data Transfer* report only considers batch input sessions that are on the local system. Therefore, the data is not persisted again in the AIF persistence. The enabler only updates the index tables and statistics tables and create alerts. The data is retrieved directly from the batch input storage in *Monitoring and Error Handling*.

Activities

You can use the following transaction codes to access this report:

- `/AIF/TRANSFER` and start it manually
- `SM36` to schedule it as a batch job (it can be run periodically)
- `/AIF/IFMON` (for the *Interface Monitor*) and `/AIF/ERR` (for *Error Handling*)
  If you use one of these transactions, the report is started automatically, if at least 1 interface has been selected that is assigned to a selection set.

More Information

For more information about the supported technologies, see the following:

- tRFC and qRFC [page 153]

- Core Interface Postprocessing [page 161]

4.9.11.4 Displaying Errors from SAP Cloud Integration with SAP Application Interface Framework

SAP Application Interface Framework supports integration with SAP Cloud Integration. This integration enables you to display errors from SAP Cloud Integration in SAP Application Interface Framework and can be set up with a few actions.

> ⓘ Note
>
> For these integrations to work for SAP Application Interface Framework 4.0, your system needs to be at least SAP NetWeaver 7.40.

The connection between SAP Cloud Integration and SAP Application Interface Framework can be set up in two different ways:

Option 1: Using an exception subprocess

In the first option, you add an exception subprocess to your existing integration flow. When an error occurs, the exception subprocess directly contacts a web service, which in turn sends a message with the details of the error message to the SAP Application Interface Framework system.

To follow this option, go to Using an Exception Subprocess to Forward Errors from SAP Cloud Integration to SAP Application Interface Framework, and follow the steps described in Configuring SAP Application Interface Framework to Display Cloud Integration Errors.

> ⓘ Note
>
> The section **Activating Error Propagation Monitoring** does not apply to SAP Application Interface Framework 4.0. Instead, you have to enable your interfaces for monitoring of error propagation yourself using the Web service `Error_Propagation_Request_In`.

Afterwards, you must modify the integration flow to enable it to forward its errors to SAP Application Interface Framework as described in Modifying Integration Flow on SAP Cloud Integration.

Option 2: Using a transfer job

The second option involves collecting error messages from the SAP Cloud Integration system in regular intervals. As a requirement for reading these error messages, you need to define settings for error propagation integration, which allow you to set the relevant SAP Cloud Integration system and integration flows.

To follow this option, go to Using a Transfer Job to Pull Errors from SAP Cloud Integration to SAP Application Interface Framework and follow the steps described in Configuring SAP Application Interface Framework On-Premise to Display Cloud Integration Errors.

> ⓘ Note
>
> The section **Activating Error Propagation Monitoring** does not apply to SAP Application Interface Framework 4.0. Instead, you have to enable your interfaces for monitoring of error propagation yourself using the REST call under Message Processing Logs.

4.9.12 Interface Creation for Logging Synchronous Service Calls

SAP Application Interface Framework provides an API that can be used by applications to log synchronous service calls.

To log synchronous service calls, you must create an ABAP Dictionary structure, define a new interface, and specify the engine settings, as described in the following steps.

Creating an ABAP Dictionary Structure for the Interface

Use transaction *SE11* to create a new ABAP Dictionary structure. The structure must represent the input and output of your service call.

Developing the Interface

Go to transaction */AIF/CUST* and perform the following configuration steps.

1. Navigate to *SAP Application Interface Framework* > *Interface Development* > *Define Interfaces*.
2. Enter the name of the namespace.
3. Create a new entry. Specify the following parameters for interface development:

Parameter	Description
Interface Name	Enter the name of the interface to be monitored.
Interface Version	Enter the interface version.
Description	Enter a meaningful description. **Note** The description is visible for users.
Lifetime of Application Log	Enter the number of days messages are stored in the application log (for example, 90d).
SAP Data Structure	Select the ABAP Dictionary structure created before.
Raw Data Structure	Select the ABAP Dictionary structure created before.
Direction	Choose *Inbound*.

4. Save the interface.

After defining your interface, specify the engine settings for it in the following way:

1. Go to transaction */AIF/CUST*.
2. Navigate to *SAP Application Interface Framework* > *Interface Development* > *Additional Interface Properties* > *Specify Interface Engines*.
3. Select your interface.
4. Change *Application Engine* to *Sync log application engine*.
5. Change *Persistence Engine* to *Sync log persistency engine*.
6. *Save* your settings.

Implementation

The following coding example shows the basic implementation steps.

Sample Code

```
TRY.
* Step 1. create enabler instance via method transfer_to_aif. If the AIF
interface is known, you can use optional parameter is_aifkeys, otherwise AIF-
interface will
* be determined automatically using the datatype of ls_raw_struc.
DATA(lr_enabler) = /aif/
cl_enabler_sync_log=>transfer_to_aif( is_raw_structure = ls_raw_struc
is_aifkeys = ls_aif_key ).


IF lr_enabler IS NOT BOUND.
* do some proper error handling
ENDIF.

* Step 2. call your service.

* Step 3. depending on the outcome of your service call, provide sucess,
warning, error messages
DATA(lt_bapiret) = VALUE bapirettab( ).
* fill lt_bapiret.

* Step 4. update message status
lr_enabler->update_message(  is_raw_structure = ls_raw_struc
                iv_message_status = 'S' "other values see corresponding
domain
                it_bapiret = lt_bapiret ).


CATCH cx_root INTO DATA(lr_root).
* do some proper error handling
ENDTRY.


* call commit work if not done by framework or application.
COMMIT WORK .
```

If you can't specify the interface, it's determined at runtime using the data type of the raw data structure. If multiple interfaces are defined for the same raw data structure and the sync log application engine, you must implement BAdI `/AIF/SELECT_IF_SYNC_LOG` of enhancement spot `/AIF/ENHS_SYNC_LOG_API`. This step is required to uniquely determine the interface to be used.

Since restart isn't defined for synchronous calls, no restart feature is available by default. If you want to implement a restart feature, implement BAdI `/AIF/RESTART_SYNC_MSG` of enhancement spot `/AIF/ENHS_SYNC_LOG_API`.

When you cancel your message in the monitoring application, the status of the message is set to *Cancelled*. You can implement additional application features in BAdI `/AIF/CANCEL_SYNC_MSG` of enhancement spot `/AIF/ENHS_SYNC_LOG_API`.

4.10 Interface Test Tool

Use

In the SAP Application Interface Framework, the *Interface Test Tool* (from the SAP Fiori launchpad or transaction code `/AIF/IFTEST`) can be used to define and store test data in a central repository.

Integration

In the *Error Handling*, technical mode, you can create test files from the content of single data messages for usage in the *Interface Test Tool*.

The *Interface Test Tool* enables you to set up SAP Application Interface Framework test cases for integration with eCATT.

In the *Interface Test Tool*, you can use the *Analyzer* to get an overview of the entire SAP Application Interface Framework runtime process (see Analyzer [page 85]).

Activities

On the *Select Test Files* screen, you can select files and execute the selection to have a *Test File Overview* screen displayed.

On the *Test File Overview* screen, you can create or select test files, and set up the integration with eCATT. Choose *Read Data* to have the *Process Test Files* screen displayed.

You can maintain data in your selected file. Transform the data from the source to the destination structure to see if your mapping works as expected.

In the tool bar of the *Test File Overview* screen, select *Analyze* to see the process steps that are executed in the SAP Application Interface Framework for your interface.

4.10.1 Test Automation with Extended CATT

Use

The *extended Computer Aided Test Tool* (eCATT) is used to create and execute functional tests for software. The primary aim is the automatic testing of SAP business processes. Each test generates a detailed log that documents the test process and results.

In the SAP Application Interface Framework, the *Interface Test Tool* is used to store predefined test data in a central repository and perform manual tests. In addition, the *Interface Test Tool* provides functions to convert the test data to an eCATT-friendly XML format and to transfer these data to eCATT. In eCATT, you can then run SAP Application Interface Framework tests manually or can be scheduled them for regular execution.

Integration

To access eCATT, use transaction code `SECATT`.

You can access the *Interface Test Tool* from the SAP Easy Access menu by choosing *Cross-Application Components* > *SAP Application Interface Framework* > *Interface Development* > *Interface Test Tool*.

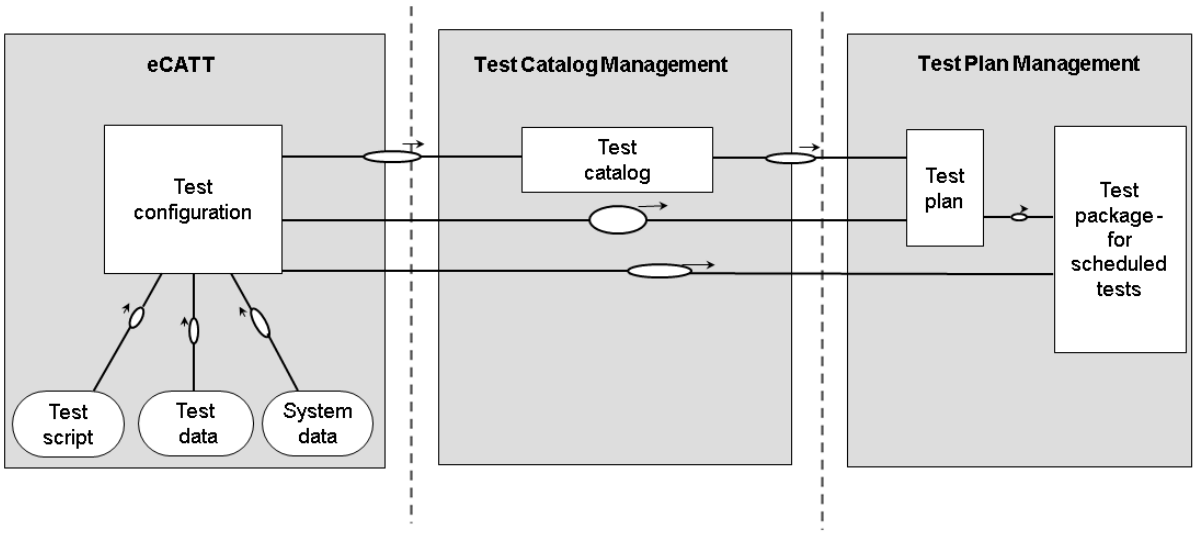
Prerequisites

The following are required to run tests for the SAP Application Interface Framework with eCATT:

- eCATT System: SAP_BASIS version 730 or more recent
- Target system: Web AS 4.6 or more recent
- CATT and eCATT need to be allowed in transaction `SCC4`
- RFC destinations need to be created in transaction `SM59` for target systems
- Function modules: `/AIF/ECATT_TESTS_PROCESS` on the target system
- The `/AIF/TEST_ECATT_SCRIPT_PROCESS` test script for a description of the course of the test

Features

Extended CATT



Test Landscape with eCATT

The testing landscape is shown in the graphic above.

In eCATT, the *Test Configuration* combines the information from the test script, test data, and system data for a test run and selects which test cases are used in the test run.

In *Test Catalog Management*, the *Test Catalog* is a collection of several *Test Configurations* that allows you to bundle test cases with different target systems.

In *Test Plan Management*, a *Test Plan* consists of at least one *Test Catalog* and is a prerequisite for creating *Test Packages*, which are in turn prerequisites for the creation of scheduled test runs. For the creation of automated, scheduled test runs, a *Test Plans* and at least one *Test Packages* are necessary. If the runs are marked as periodic, they are run periodically in the background.

> → Recommendation
>
> Starting test runs via *Test Configuration* is sufficient for tests whose compilation changes often and that are run occasionally. However, for tests that do not vary and are run regularly, for example, regression tests, we recommend that you create *Test Plans* and *Test Packages*. This allows you to organize tests in separate configurations, bundle them up, and run them automatically.

The Interface Test Tool

The following test tool tables are displayed on the *Process Test Files* screen:

- *Source Data Structure*
  In the first table, the source data structure is displayed and you can fill the structure with data. Choose *Transform* to trigger the mapping of the source to the destination structure in the SAP Application Interface Framework.
- *Destination Data Structure*
  This table is automatically filled after transformation. How the table is filled depends on the mapping you define for the interface. If you double click on a field of the destination structure, the value of this field and the structure path for this field is added into the *Expected Values* table.
- *Display Logs*

Displays the application log of the current Interface. If you select a message from the application log and choose *Add to exp. messages* (Add to Expected Messages), the *Expected Message* table is filled.

- *Expected Values* table
  The *Expected Values* table displays the path of the structure where the value is found in the source structure, the expected value, and also a status icon. If the light is green in the *Status* field, the value in the *Expected Messages* table is same as the value in the source structure; Otherwise the status light is red.
- *Expected Messages* table
  The *Expected Messages* table displays the corresponding message type, message ID, message number, message variables, and an *Incl. Msg. Var.* (Include Message Variable) checkbox. If you select the checkbox, the corresponding values of message variables in the message are considered.

The following buttons on the *Process Test Files* screen enable you to transfer your test data to eCATT

- *Parameter for eCATT*
  Along with information about the interface (namespace, interface name, interface version, and expected status), the expected values and test data are shown as XML string. For a manual transfer, you can copy and paste the data displayed in to the parameters of the eCATT test data container.
- *Transfer to eCATT*
  This function triggers the creation of a variant for the eCATT configuration assigned to the test file and an automatic transfer of the test data you see under *Parameter for eCATT*.

Activities

To create a test data container, copy the `/AIF/TEST_ECATT_DATA_TMPL` template and specify, for example, the target system and its system data container.

For entering interface test data, use the SAP Application Interface Framework *Interface Test Tool* (transaction code `/AIF/IFTEST`).

To create a new test configuration, copy the `/AIF/TEST_ECATT_CONFIG_TMPL` template and specify the application component, system data container, test script, target system, and test data container.

To create a new test catalog, use transaction code `STWB_1`.

To create a new test plan, use transaction code `STWB_2`.

Extended CATT

To test the functionality of interfaces, the test script requires you to enter the test data parameters, shown in the following table with both mandatory and optional entries described:

Input Parameter	Description	Mandatory or Optional
`IV_NS`	Interface namespace	Mandatory
`IV_NAME`	Interface name	Mandatory
`IV_VERSION`	Interface version	Mandatory
`IV_XML_TEST_DATA`	XML input test data as string	Mandatory

Input Parameter	Description	Mandatory or Optional
IV_XML_EXPECTED_VALUES	XML input expected values	Optional
IV_XML_EXPECTED_MESSAGES	XML input MSG log as string	Optional
IV_EXPECTED_STATUS	Expected status	Optional
IV_CHECK_FM	Function module for custom checks	Optional

To create a valid test case, you have provide the interface's namespace, name, and version and also a raw data structure that can be processed by the interface. You also need to provide a string containing XML data that describes the structure.

You can provide a table converted to an XML string with expected values and the path where the fields are in the source structure. You can also provide an *Expected Message* table with fields such as message ID, message number, message type, and message variables if required.

> **Note**
>
> The following data can be transferred using *Transfer to eCATT* or copied using *Parameter for eCATT* in the *Interface Test Tool* (transaction code `/AIF/IFTEST`):
>
> - `IV_XML_TEST_DATA`
>   Can be copied from *Test Data as XML String*
> - `IV_XML_EXPECTED_VALUES`
>   Can be copied from *Expected Values as XML String*
> - `IV_XML_EXPECTED_MESSAGES`
>   Can be copied from *Expected Messages as XML String*
> - `IV_EXPECTED_STATUS`
>   Can be copied from *Expected Status*
>
> In addition, when you choose *Transfer to eCATT*, parameters `IV_NS, IV_NAME,` and `IV_VERSION` are transferred to the variant. Which values are transferred depends on the interface for which you created the test file. `IV_CHECK_FM` is transferred from the *Check Funct. Module*.

Additionally, you can enter the name of a custom function module that overwrites the implementation logic of the standard AIF test script. To do this, it imports all of the test case's parameters plus the destination structure. The function module returns the status of the interface processing and the tables for the failed expected messages and values.

Since the result of a test run often depends on the messages that are raised while the input data is being processed, the test script allows you to check if a certain status has been set, a certain message has been raised, or a certain field mapping has been done while processing the data. When a test run execution is complete, the results are displayed in the eCATT log. Green lines signify a successful result. Red lines indicate failure, which can mean that either an error message in the `/AIF/ECATT_TESTS_PROCESS` function module was raised, the checks of expected values or messages have failed, or the expected status does not fit. The failed checks are displayed in the eCATT log.

More Information

For more information on eCATT, see SAP Library for eCATT: extended Computer Aided Test Tool (`BC-TWB-TST-ECA`) on SAP Help Portal at http://help.sap.com.

4.11 Glossary

Term	Definition
action	An interface building block that handles the execution of the interface logic in the SAP application system by calling standard functions, customer functions, or BAPIs.
AIF runtime	The runtime environment of the SAP Application Interface Framework that enables the scheduling of runs and packets for message processing. A runtime configuration group allows you to configure the processing of a set of messages.
application engine	A class that is responsible for handling the application-specific parts of a data message which differs for different interface technologies. For example, it is responsible for restarting and canceling data messages. In addition, it connects the Monitoring and Error Handling transaction with the other engines.
custom function	A hotspot to call a transaction or a report for one or more log messages.
data message	A message that is transferred between systems and that carries the actual business content.
default recipient	A type of recipient who receives alerts for an interface, regardless of the alert settings, if the system cannot find any other recipient to notify. **Note** Superordinate term: recipient
enabler	A class that lets you use your standard processing and not have to use the SAP Application Interface Framework for processing; Use the enabler to write SAP Application Interface Framework-specific data for a message to the data storage of the SAP Application Interface Framework so that the SAP Application Interface Framework can monitor and display data.
fallback recipient	A type of recipient who receives alerts when the alert settings mean that the specific recipient is not notified. **Note** Superordinate term: recipient
field mapping	The process of mapping the value or values of up to five fields of a source structure to the value of a single field of a destination structure.

Term	Definition
file adapter	A support for the upload of files to the SAP Application Interface Framework. The content of the file is then processed in the AIF runtime and can be monitored in *Monitoring and Error Handling*.
fix value	An attribute used in structure mappings to assign a constant value to a field in the destination structure. Fix values that are assigned to a namespace can be reused in different structure mappings.
hierarchy mapping	A structure mapping that is nested over multiple levels of a structure. An example is a table that includes a table: The mapping for the first table is done on the first level, while the mapping for the second table is done on a subordinated level.
interface	The connecting point between the application data and data in an external format. An interface consists of one or more of the following reusable building blocks: • Value mappings • Fix values • Checks • Actions
interface determination engine	A class that is responsible for determining the correct SAP Application Interface Framework interface. Interface determination depends on the interface technology that is used.
Interface Monitor	A transaction that provides personalized access to interface monitoring and error handling.
interface variant	An option used to change the behavior of an interface. Interface variants are used for specific data messages where some adaptations to the interface processing logic are required.
log message	A message that is displayed in the application log. Log messages can be error, warning, success, or information messages that are saved during data message processing.
logging engine	A class that is responsible for retrieving the log messages data for monitoring and error handling.
message category	A group of log messages. A message category can be created to group log messages of interest to a certain user group. During recipient determination, the message category can be used to notify the user group when log messages of interest to them occur during processing.
namespace	A concept used for the logical structuring of objects, for example, interfaces and interface building blocks.
payload	The content (business data) of a message.
persistence engine	A class that is used to handle the content a data message transfers. It is responsible for everything that is related to the persistence layer of a data message, for example, retrieving and updating message content.

Term	Definition
playback function	A function module that is generated based on an RFC enabled function module. The playback function is used during data transfer to retrieve the data from the LUW/queue for tRFC/qRFC calls that are struck.
processor	A user responsible for following up and solving a certain error message. Users can assign processors directly to detailed error messages and track the error-solving progress using statuses and comments.
raw data structure	The structure used for communication with external systems. In structure mapping, the raw data structure is the source structure for inbound interfaces and the destination structure for outbound interfaces.
recipient	An entity that defines which messages a user or a group of users is responsible for. The recipient defines in which cases the users assigned to it receive an alert and see messages in the Interface Monitor. The messages are assigned to recipients by interfaces, by message categories, by key field values, or a combination of message categories and key field values.
record type	The characterization of information in the record type screen of the error handling transaction of the SAP Application Interface Framework. The record types are used to display the multilevel object structure of the interface data to select the data level to be displayed in the source data screen.
reprocessing action	A configuration name for automatic reprocessing. The configuration contains a function module and a runtime configuration group and defines how a message should be reprocessed automatically.
runtime configuration group	A configuration that defines the processing of a set of messages by the SAP AIF runtime environment.
runtime queue	The set of messages to be processed by the SAP AIF runtime environment.
SAP Application Interface Framework	A tool for the implementation of interfaces between SAP systems and external systems. It provides business users with the functionality of interface monitoring and error handling.
SAP data structure	The structure used to process data in the application system. In structure mapping, the SAP data structure is the destination structure for inbound interfaces and the source structure for outbound interfaces.
selection engine	A class that is responsible for selecting the data messages for Monitoring and Error Handling.
selection set	The definition of which data is selected during data transfer for tRFC, qRFC, CIF postprocessing, and batch input.
serialization object	A technical representation of a business object, where the messages have to be processed in the correct order. A serialization object consists of a namespace and a serial object name.

Term	Definition
serialization type	A setting that specifies how messages are serialized. The following serialization types are available: • External index • Timestamp • Internal timestamp • No serialization
specific recipient	A type of recipient who receives alerts for messages for one or more specific interfaces, message categories, key field values, or a combination of message categories and key field values. **Note** Superordinate term: recipient
standard recipient	A type of recipient who receives alerts for all messages for one or more specific interfaces independent of message categories or key field values. **Note** Superordinate term: recipient
structure mapping	The process of mapping the data of a source structure to a destination structure.
structured persistence	A persistence layer in the SAP Application Interface Framework that enables the storage of interface messages for different purposes. The structured persistence stores messages in database tables representing the structure of the interface.
trace level	An option for defining the level of detail for tracing the steps that the system performs when processing data messages.
tracking variant	A template for analyzing the performance of SAP Application Interface Framework interfaces that is used for selected interfaces, users, or time ranges.
value mapping	The process of mapping at field value level between up to five source values and a particular destination value.
variant mapping	The process of defining alternative value mappings, actions, structure mappings, and fix values that are used in an interface variant.
XML persistence	A persistence layer in the SAP Application Interface Framework that enables the storage of interface messages for different purposes. The XML persistence stores messages as strings representing the XML structure of the interface.
batch data communication (BDC)	Batch data communication (BDC) is also referred to as batch input.
batch input	Data transfer technology that allows data sets to be passed in an automated way to dynpros called by transactions, and thus to AS ABAP. Batch input is controlled by batch input sessions.

Term	Definition
batch input session	Contains all necessary information and data for data transfer using batch input, namely summary of a sequence of transaction calls including the input data and user actions. A batch input session can be used to execute dialog transactions in batch input, where some or all of the screens are processed by the session. Batch input sessions are stored in the database as database tables and can be used within a program as internal tables when calling transactions. A batch input session (BI session) is created using a BI program. This BI session must then be processed using the batch input monitor.
Core Interface (CIF)	Interface for data transfer between an ERP system (SAP R/3 or SAP ERP) and a connected SCM system (such as SAP Advanced Planning and Optimization (SAP APO) or SAP Supply Network Collaboration (SAP SNC)). The basis for the data transfer is the integration model on the ERP side.
eCATT	eCATT is the abbreviation for "Extended Computer Aided Test Tool". It is a tool to perform automatic test cases.
Error and Conflict Handler	The Error and Conflict Handler supports forward resolution of asynchronous messages. It enables clients to register errors and creates postprocessing orders accordingly.
Logical Unit of Work	Inseparable sequence of database operations that must be executed either in its entirety by a database commit, or not at all.
qRFC	An enhancement of transaction remote function call that lets you define the order of the calls.
Service Implementation Workbench	A tool for automatic or semi-automatic generation of the implementation of Enterprise Services. It facilitates and accelerates the creation of uniform and standardized service implementations by usage of code generation techniques.
tRFC	A remote function call that flags the remote function to be executed and starts it with a COMMIT WORK.

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

www.sap.com/contactsap