zenon manual

SAP interface

v.7.20

COPADATA
do it your way

Contents

1. Welcome to COPA-DATA help

GENERAL HELP

If you cannot find any information you require in this help chapter or can think of anything that you would like added, please send an email to documentation@copadata.com (mailto:documentation@copadata.com).

PROJECT SUPPORT

You can receive support for any real project you may have from our Support Team, who you can contact via email at support@copadata.com (mailto:support@copadata.com).

LICENSES AND MODULES

If you find that you need other modules or licenses, our staff will be happy to help you. Email sales@copadata.com (mailto:sales@copadata.com).

2. SAP interface

To connect the SAP ERP level (Enterprise Resource Planning level) to the process level of zenon, both programs communicate using a two-way interface. This gives SAP users direct access to the process level of zenon.

The control system can be configured to send messages about any type of events to the SAP application. Companies can establish a direct link between the process level and the ERP level via a closed loop, getting a complete overview over all processes. This allows them, for example, to monitor warehousing in real-time, as the control system in the SCADA level provides exact data about real resource consumption. With this, companies get important basic data for just-in-time production but also for long-term planning.

zenon provides an integrated communication module, which makes it possible to transfer process data directly to the SAP ERP application and to call up zenon functions directly from the SAP application. In an SAP environment, this data is used, amongst other things, for the following tasks:

- ▶ Transmission of the status of process orders
- ▶ Information about consumption and production of materials
- ▶ Information about the status of resources
- ▶ Transmission of selected process events
- ▶ Transfer of order data to zenon

Process messages in SAP can be used for tasks such as creating electronic lot and operation logs or updating a process order or the stock of materials.

> **License information**
>
> *Must be licensed for Editor and Runtime (single-user, server, standby).*

INTERACTIONS

- ▶ Maintenance message: (on page 21) Maintenance messages can be created from zenon variables and forwarded to the SAP system. These reports can then trigger other actions in the SAP system, depending on the workflows defined there.
  For example, a maintenance message may automatically trigger the creation of a maintenance task for the affected piece of equipment.

- ▶ Measurement documents: (on page 23) In zenon, it is possible to measure the values of process variables according to time schedules and then transfer them to the SAP system as a measurement reading. These measurement documents can be used in the SAP system for planning and creating maintenance tasks or for statistical purposes.

- ▶ Function calls: (on page 29) SAP RFC-Functionblocks can be called up from zenon and executed in Runtime. The zenon SAP function contains a list of objects that display the parameters and tables of the SAP RFC-Functionblock.

- ▶ Server programs: Call-up of zenon functions from an SAP application. For example, in order to transfer order data to zenon.

3. Basics and requirements

In order to use the direct integration into the SAP world, the computer on which zenon is running must also have, in addition to zenon, the `SAP GUI` installed. The program libraries included in the `SAP GUI` are absolutely required for communication with a SAP system. This `SAP GUI` is supplied with SAP.

No further configurations on the SAP side are required. Installing the `SAP GUI` is sufficient for enabling data exchange. The rest of the parameters can be entered directly in the development environment of the zenon Editor.

> **Information**
>
> *Integration of SAP into zenon based on SAP ECC 6.0 for the PM-PCS 4.0 scenario*

REQUIREMENTS IN ZENON

In zenon there must be a license for the `SAP Interface` *module.*

FUNCTION BLOCKS

Using the RFC interface, actions in the SAP system are executed by calling up function blocks (known as SAP RFC-Functionblocks). This standard function blocks are supplied with the SAP system. User-defined function blocks can also be programmed using the ABAP Workbench. ABAP is a programming language developed by SAP, which is used for all function blocks in the SAP system. The integration in zenon is currently limited to the standard function blocks delivered by SAP. However, a flexible interface was created in the control system, which allows integration of customer-specific function blocks.

The complete descriptions of function blocks with their parameters, internal tables and structures used can be read from the SAP system via RFC API functions or by means of different zenon functions.

CLIENT PROGRAMS AND SERVER PROGRAMS

SAP applications make a distinction between Client programs and Server programs when calling up function modules:

▶ Client program: Is called up by an external system (such as another SAP application) or an external application (such as zenon) and executed in the SAP application.

▶ Server program: Is called up by the SAP application on a different system on an external system. The external system executes the function.

3.1 Maintenance reports

The planning system must at all times be informed about malfunctions, downtimes, etc. Productivity indicators can only be calculated correctly when the according uptimes and downtimes are known. The SAP system uses maintenance messages for this.

Every maintenance message defined in zenon for a process variable can be configured as a malfunction report. Relevant messages are usually documented in SAP. "Relevant" means that the messages require maintenance services in order to be resolved.

Maintenance messages are created from zenon alarms. When there is an alarm for one of the variables linked to the element, a maintenance message will be created in the maintenance module of the SAP system. If there is a `Cleared event` for an alarm, the maintenance message will be acquired via the document number, and the time stamp of the `Cleared` event will be recorded.

For each type of message from SAP, you own group of message types can be created in zenon. The most significant property of a message in the SAP system is the "notification type". It defines how the message will be processed in the workflow of the SAP system. There are predefined notification types in the SAP system. If you do not specify one of the predefined notification types when trying to create a maintenance message, the SAP system will not create a maintenance message but will return an error message instead.

> 💡 **Information**
>
> *You can also choose to define a maintenance plant and a reporter for every maintenance message. If you do not specify a plant or a reporter, the SAP system will insert default values.*

Every maintenance message refers to an installation part that is predefined in the SAP system. Such an installation part can be a "functional location" or "equipment". Every functional location can in turn contain functional locations and/or equipments.

Every functional location and every item of equipment have a unique identification that must be specified when creating a malfunction report. The identification for functional locations or equipments is taken from the **resources label** of the linked variables. Via the property "**Functional location/Equipment**", you can configure whether the resources label shall be interpreted as the identification of a functional location or an equipment.

You can now link variables to the maintenance messages you just defined. If there is an alarm for one of the linked variables, a maintenance message will be created in the SAP system using the defined parameters and the value and the limit text of the variable.

The maintenance messages receive a unique document number in the SAP system, which will be stored as a comment in the alarm entry of the zenon variable. This allows you to map the alarm event to the document number of the maintenance message in the SAP system.

> ⚠ **Attention**
>
> If an alarm has a dynamic limit text, the comment field of the entry is not available; instead, an entry to the Chronological Event List will be created, which allows to map the alarm to the document number.

3.1.1   Technical background

Maintenance messages in the SAP system are created by executing the function blocks ALM_PM_MAINTNOT_CREATE and ALM_PM_MAINTNOT_SAVE. If the alarm is cleared, the "Cleared time" will be entered into the maintenance message created before by executing the function blocks ALM_PM_MAINTNOT_MODIFY_DATA   and ALM_PM_MAINTNOT_SAVE.

These function blocks are created in the SAP system during the standard installation of the maintenance module and are then available for external applications.

> ⚠ **Attention**
>
> *Due to individual adaptations of the SAP system to specific customer requirements, these function blocks may no longer be available! In this case, the SAP system will return the error message* FU_NOT_FOUND.

If the event `Alarm cleared` occurs in zenon for one of the linked variables, the maintenance message belonging to the alarm will be searched via the document number and the time of clearance of the malfunction will be inserted. This happens by executing the ALM_PM_MAINTNOT_MODIFY_DATA function block. The document number of the maintenance message is handed over to the function block with the NUMBER   parameter. The values of the fields that need to be changed are specified in the parameter NOTIFHEADER ; the fields that need to be changed are specified with flags in the parameter NOTIF .

The modified maintenance message will then be stored in the SAP database by executing the function blocks ALM_   and BAPI_TRANSACTION_COMMIT.

3.2   Measurement readings

In SAP applications, measured values describe a certain status of a production system at a certain time in the process flow. Any variable existing in zenon can be defined as a measurement document. This allows handing over protocol-independent data from the real time process to the superordinate management level.

In many cases, it is better to aggregate process data already on the process level and then pass it on in a condensed format to the SAP system for analysis. For example, zenon delivers average values and other statistical values calculated from value series to the SAP system. This avoids overloading the management system with raw data.

Measured values and counter readings are recorded cyclically and stored as measured documents for a measuring point in the SAP system. The properties of the measuring point in the SAP system determine whether it is a measured value or a counter. The current variable value is used as the value of the measurement document.

Several measuring points can be defined for every technical location or every piece of equipment in the SAP system. During the definition of the measuring points, you can configure whether the measuring

point shall continue a measured value or a counter reading. Counter readings must have monotonously rising values, i.e. every new value you enter must be equal to or higher than the previous value.

Measured values and counter readings of measuring points are passed on to the SAP system by means of what are known as "measurement documents".

In the control system, measured values are recorded according to **schedules**. For every schedule, you can define the weekdays and the times at which measured values shall be recorded. A schedule can be executed either daily or only on specific weekdays. Furthermore, you can define a starting time and an ending time as well as an interval for every schedule. Between the starting time and the ending time, the measured values of the linked variables in the specified interval will be recorded.

You can link a number of values with every schedule. Every time the schedule is executed, the current values of the variables will be read out and a measurement document for every variable value will be created in the SAP system. Every measuring point has a unique number that must be put down in the measurement document. The **resource label** of the linked variables is used to determine the number of the measuring point.

> ### 💡 Information
>
> *The control system variables and the SAP measuring points are mapped via the ID of the measuring point in the SAP system. This unique consecutive number must be entered in the **resource label** of the variables in the control system.*

3.2.1   Technical background

Measurement documents for measuring points and counters are created in the SAP system with the function block "MEASUREM_*" . The measurement document is allocated to a measuring point or counter. Whether it is a measuring point or a counter, will be defined during the creation of the measuring point. The measuring point/counter is identified via a unique number in the SAP system.

The measuring document contains the measuring date (date, time), a measured value and its unit, a reporter (creator of the document) and a short text. If the measuring date is not specified, the date of the creation of the document will be used. In case the unit of the measured value is left out, the unit specified at the measuring point/counter will be used.

Setting the COMMIT_WORK parameter causes the document to be written to the SAP database immediately. Measurement documents are created in the SAP system by executing the function block MEASUREM_DOCUM_RFC_SINGLE_001.

> 💡 **Information**
>
> *The MEASUREM_DOCUM_RFC_SINGLE_001 function block is created during the standard installation of the maintenance module in the SAP system.*

> ⚠ **Attention**
>
> *Due to individual adaptations of the SAP system to specific customer requirements, these function blocks may no longer be available! In this case, the SAP system will return the error message **FU_NOT_FOUND**.*

3.3 Function calls

Desired functions in the SAP system can be called up from zenon in the programming interface node with a very small amount of configuration work. To do this, an image of the interface of the function block called up is displayed in the editor with the required parameters, tables and structures and linked to zenon variables or functions. The actual function call is made by executing a zenon function (on page 36), which is given this image as a parameter.

FUNCTION BLOCKS

Actions in the SAP system and the transfer of data to the SAP system are carried out by calling up the function blocks. These function blocks are supplied with the SAP system; however user-defined function blocks can also be created via the ABAP Workbench.

PARAMETERS

Data is transferred to and from function blocks via the import and export parameters. Input data is "imported" from the point of view of the function blocks in the SAP system; result data is "exported". The documentation of the function blocks in the SAP system is based on this point of view.

In addition to import and export parameters, there are also changing parameters, which serve for both the transfer of data and the return of results.

INTERNAL TABLES

Data in table form or lists can be transferred in what are known as internal or temporary tables and can be transferred by function blocks.

DATA TYPES, STRUCTURES

In addition to the other simple data types (numerical, text, date, time, etc.), data can also be transferred in structures. Data in tables is usually transferred as structures, but parameters can also take on structured data.

SUPPORT FROM THE SAP SYSTEM

The complete interface descriptions of function blocks including their parameters, internal tables and structures used can be read from the SAP system.

The zenon SAP interface makes use of this possibility in that it offers an assistant (on page 29) to select function blocks and to read in the interface description.

The configured objects are also compared with the interface description of the function block in the SAP system called up when an SAP function is called up.

3.4   Server programs

In order for an SAP application to be able to call up functions in an external system, RFC destinations are managed (transaction `SM59`). This is used to stipulate how an external system and an SAP application communicate. There are two possibilities in principle:

▶   `Started`: The SAP application starts the external application.

▶   `Registered`: The external system logs into the SAP application using a unique program ID.

zenon uses the `registered` method.

The external system decides how function calls that are received via the RFC interface are handled. There are two possibilities available for this:

▶   Callback-Functions:
    At the RFC interface, Callback functions that are automatically executed by the interfaces are registered. This possibility is primarily suited to when the function calls that are expected via the RFC interface have already been established during implementation of the external application.

▶   Name-based decision:
    The name of the function module called up is queried at the interface. A decision is made, using the name, on which action is to be executed. With this possibility, it is possible to react more flexibly and generally to function calls.
    zenon uses this method because the user decides which function calls should be available.

Some standard functions, such as `RFC_PING` or `RFC_SYSTEM_INFO` are automatically registered and executed by the RFC interface.

4. Engineering in the Editor

Measuring points/maintenance messages and function calls can only be configured in the development environment if there is a licensed SAP interface. If this is licensed there are   Settings for the SAP connection in the project tree of the `SAP interface` node. Details about the configuration of the parameters are described in the following chapters.
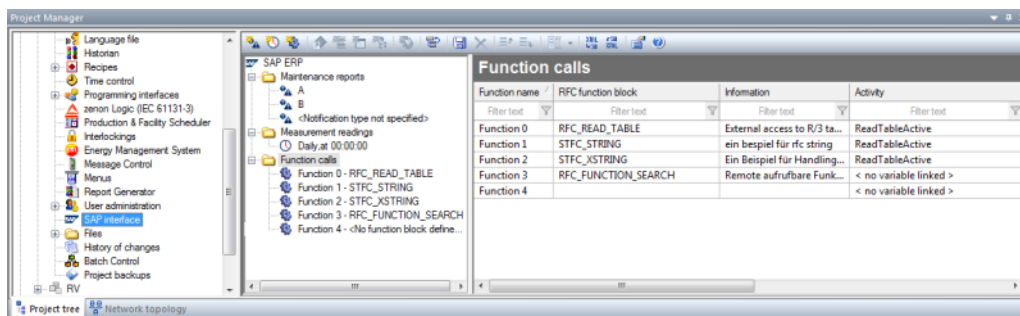
CONTEXT MENU PROJECT MANAGER NODE SAP ERP

Menu item	Action
`Save`	Saves changes in the SAP ERP configuration.
`Export XML all...`	Exports all entries as an XML file.
`Import XML`	Imports entries from an XML file.
`Editor profile`	Opens the drop-down list with predefined editor profiles.
`Help`	Opens online help.

4.1 Toolbar and context menus

Symbol	Action
`New message type`	Inserts a new empty message into the maintenance messages list.
`New time plan`	Inserts a new empty time plan into the measurement readings list.
`New SAP function`	Opens the assistant to select a new SAP function block (on page 29).
`Add variable`	Opens the dialog for selecting variables.
`New parameter`	Creates new parameters for the highlighted function.
`New list`	Creates new table for the highlighted function.
`New structure field`	Creates a new structure field for highlighted parameters or highlighted table.
`Create standard function`	Creates a zenon "Execute SAP function" (on page 36) for the highlighted SAP function.
`Jump back to starting element`	If you entered the list via function `linked elements`, the symbol leads back to the start element. Only available in the context menu when all linked elements are opened.
`Save`	Saves changes in the SAP configuration.
`Delete`	Deletes the selected entry after confirming a confirmation message.
`Move up`	Moves highlighted element in the detail view up by a position.
`Move down`	Moves highlighted element in the detail view down by a position.
`Expand all`	Expands all closed structures. Clicking on the arrow next to the symbol opens the drop-down list to select from: ▸ Expand all ▸ Reduce all ▸ Expand selected ▸ Reduce selected
`Export XML all...`	Exports all entries as an XML file.
`Import XML`	Imports entries from an XML file.
`Properties`	Opens the property window.
`Help`	Opens online help.

4.1.1 Context menu Project manager

Menu item	Action
Save	Saves changes in the SAP ERP configuration.
Export XML all...	Exports all entries as an XML file.
Import XML	Imports entries from an XML file.
Editor profile	Opens the drop-down list with predefined editor profiles.
Help	Opens online help.

4.1.2 Context menu node SAP ERP

Menu item	Action
Save	Saves changes in the SAP ERP configuration.
Export XML all...	Exports all entries as an XML file.
Import XML	Imports entries from an XML file.
Help	Opens online help.

4.1.3 Maintenance messages context menu

Menu item	Action
New message type	Inserts a new empty message into the list.
Export all XML maintenance messages	Exports all entries as an XML file.
Import XML	Imports entries from an XML file.
Help	Opens online help.

4.1.4 Maintenance message context menu

Menu item	Action
Add variable	Opens the dialog for selecting variables.
Linked elements	Shows elements linked to the maintenance message in a drop-down list and makes it possible to jump to these elements.
Delete	Deletes the selected entry after confirming a confirmation message.
Export selected XML	Exports selected entries as an XML file.
Properties	Opens the properties window.
Help	Opens online help.

4.1.5 Measurement documents nodes context menu

Menu item	Action
New time plan	Inserts a new empty time plan into the list.
Export all XML measurement documents	Exports all entries as an XML file.
Import XML	Imports entries from an XML file.
Help	Opens online help.

4.1.6 Time plan context menu

Menu item	Action
Add variable	Opens the dialog for selecting variables.
Linked elements	Shows elements linked to the time plan in a drop-down list and makes it possible to jump to these elements.
Delete	Deletes the selected entry after confirming a confirmation message.
Export selected XML	Exports selected entries as an XML file.
Properties	Opens the property window.
Help	Opens online help.

4.1.7 Maintenance message and measurement document variable list context menu

Menu item	Action
`Add variable`	Opens the dialog for selecting variables.
`Linked elements`	Shows elements linked to the variable in a drop-down list and makes it possible to jump to these elements.
`Remove selected variable`	Deletes selected variables from the list.
`Help`	Opens online help.

4.1.8   Function calls context menu

Function call nodes context menu

Menu item	Action
New SAP function	Opens the assistant to select an SAP function block.
Export all SAP functions	Exports all entries as an XML file.
Import XML	Imports entries from an XML file.
Help	Opens online help.

FUNCTION CALL TABLE CONTEXT MENU

Parameters	Description
New SAP function	Create new SAP function
Create standard function	Creates a new zenon function to call up the selected SAP function.
Linked elements	Shows elements linked to the SAP function in a drop-down list and makes it possible to jump to these elements.
Delete	Deletes the selected entry after confirming a confirmation message.
Export selected XML	Exports selected entries as an XML file.
Import XML	Imports entries from an XML file.
Read RFC function block from SAP system	Reads in the interface description of the function block from the SAP system. If no SAP name is given, the assistant to select the function block is displayed.
Display documentaton of the RFC function block	Reads in the documentation of the function block from the SAP system and displays it as an HTML file using the web browser.
Help	Opens online help.

4.1.9   Individual SAP function context menu

SAP FUNCTION CONTEXT MENU

Parameters	Description
New parameter	Create new parameter with the SAP function.
New report	Create new table with the SAP function.
Create standard function	Creates a new zenon function to call up the selected SAP function.
Linked elements	Shows elements linked to the SAP function in a drop-down list and makes it possible to jump to these elements.
Delete	Deletes the selected entry after confirming a confirmation message.
Export selected XML	Exports selected entries as an XML file.
Properties	Opens the property window.
Read RFC function block from SAP system	Reads in the interface description of the function block from the SAP system. If no SAP name is given, the assistant to select the function block is displayed.
Display documentaton of the RFC function block	Reads in the documentation of the function block from the SAP system and displays it as an HTML file using the web browser.
Help	Opens online help.

CONTEXT MENU OF THE PARAMETER LISTS OF AN SAP FUNCTION

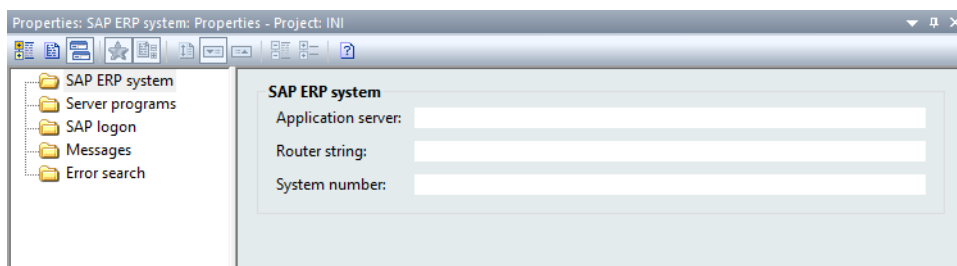| TAGs	Description
New parameter	Create new parameter with the SAP function.
New report	Create new table with the SAP function.
New structure field	Create new structure field for parameter or table.
Linked elements	Shows elements linked to the SAP function in a drop-down list and makes it possible to jump to these elements.
Delete	Deletes the selected entry after confirming a confirmation message.
Expand/reduce nodes	Display or hide structure fields of parameters and tables.
▸ Expand all	Display structure fields of all parameters and tables.

▸ Reduce all	Hide structure fields of all parameters and tables.
▸ Expand selected	Display structure fields of all selected parameters and tables.
▸ Reduce selected	Hide structure fields of all selected parameters and tables.
Help	Opens online help.

CONTEXT MENU PARAMETER LIST OF AN SAP FUNCTION WITH STRUCTURE FIELD SELECTED

TAGs	Description
New structure field	Create new structure field for parameter or table.
Linked elements	Shows elements linked to the SAP function in a drop-down list and makes it possible to jump to these elements.
Delete	Deletes selected structure fields.
Move up	Push selected structure field up by one position.
Move down	Push selected structure field down by one position.
Help	Opens online help.

4.2   Connection parameters to an SAP system

In order to establish a connection to an SAP system, some basic parameters must be specified. These are configured in the SAP ERP properties in the project manager detail window.   The following groups are available:
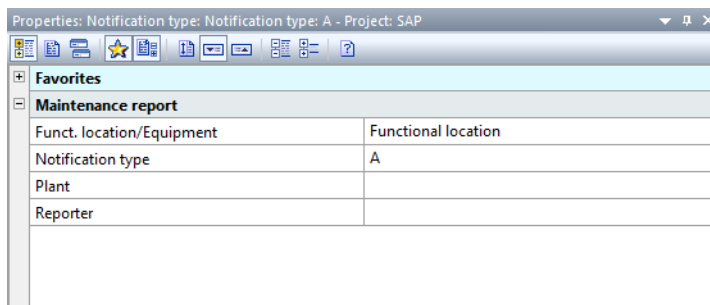
Parameters	Description
SAP ERP system	Settings for the connected SAP ERP system.
Server programs	Settings for Server programs.
SAP logon	Settings for the registration on the SAP system.
Messages	Settings for messages.
Error search	Settings for troubleshooting.

You can find details on the individual properties in the properties help for the respective property.

4.3   Messages

The planning system must at all times be informed about malfunctions, downtimes, etc. The SAP system uses maintenance messages for this. Every maintenance message defined in zenon for a process variable can be configured as a malfunction report.

> **Information**
>
> *If there is an alarm for one of the linked variables, a maintenance message will be created in the SAP system using the defined notification type and the limit text of the alarm. The number of the installation part (functional location or equipment) will be acquired from the resources label of the variable.*
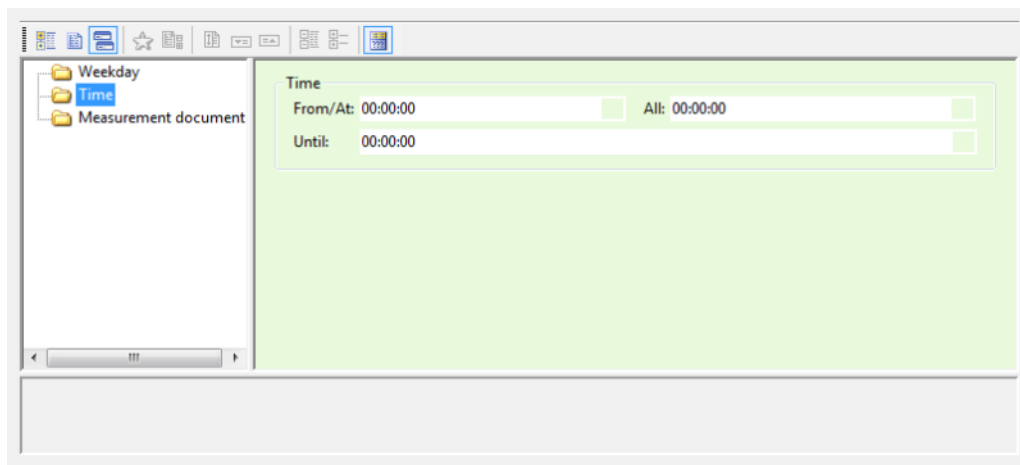
MAINTENANCE MESSAGE CONTEXT MENU (GROUP)

Menu item	Action
New message type	Inserts a new empty message into the list.
Export all XML maintenance messages	Exports all entries as an XML file.
Import XML	Imports entries from an XML file.
Help	Opens online help.

MAINTENANCE MESSAGE CONTEXT MENU

Menu item	Action
Add variable	Opens the dialog for selecting variables.
Linked elements	Shows elements linked to the maintenance message in a drop-down list and makes it possible to jump to these elements.
Delete	Deletes the selected entry after confirming a confirmation message.
Export selected XML	Exports selected entries as an XML file.
Properties	Opens the properties window.
Help	Opens online help.

4.4   Measurement readings

In SAP applications, measured values describe a certain status of a production system at a certain time in the process flow. Any variable existing in zenon can be defined as a measurement document. This allows handing over protocol-independent data from the real time process to the superordinate management level. Measurement documents are passed on to an SAP system based on a fixed time schedule.

> ### Information
>
> *At the times defined in the schedule, the current values of the linked variables are read out. Then, a measurement document in the SAP system is created for every variable value.*

MEASUREMENT DOCUMENTS CONTEXT MENU (GROUP)

Menu item	Action
`New time plan`	Inserts a new empty time plan into the list.
`Export all XML measurement documents`	Exports all entries as an XML file.
`Import XML`	Imports entries from an XML file.
`Help`	Opens online help.

MEASUREMENT DOCUMENTS CONTEXT MENU

Menu item	Action
`Add variable`	Opens the dialog for selecting variables.
`Linked elements`	Shows elements linked to the time plan in a drop-down list and makes it possible to jump to these elements.
`Delete`	Deletes the selected entry after confirming a confirmation message.
`Export selected XML`	Exports selected entries as an XML file.
Properties	Opens the property window.
`Help`	Opens online help.

4.5 Server programs

To use Server programs:

1. Go to the property group **SAP ERP system/Server programs**.

2. Enter, for the **Program ID** property, the program ID of the RFC destination via which the connection to the SAP application is to be established.

3. Define, for each desired SAP function call, whether it is to be executed as a Server program. To do this, configure the properties in the nodes (on page 33) **Server program** of the function calls.

The call can also be made via the programming interface (on page 26).

LIMITATIONS

The following limitations are applicable for Server programs:

▶ Server programs can only be registered in zenon on standalone computers or **Server 1**, but not on Clients or **Server 2**. When switching between **Server 1** and **Server 2**, the server programs on the old server are de-registered and registered on the new server.
Background information: SAP applications allow the registration of several external systems on the same RFC destination. However the system on which the Server program is executed is not defined.

▶ In principle, an external system can register with the destination system more than once. Because with zenon, the evaluation of the Server program called up is only carried out by means of the name, this is prevented. If the same function module (same FB name) is configured in several SAP functions as a Server program, then the registration of a pre-existing registration of the same name is de-registered again. Thus the last-registered SAP function is active.

▶ Only zenon functions that can be executed on the local computer can be linked. To do this, the **Current computer** property in the **Execution** group must be activated for the zenon function. If, in the selection dialog, a different function was selected, this is not applied. Filtering for functions that meet this condition is not possible.

▶ Calls of SAP functions as a Client program are always asynchronous.
If the call of an SAP function as a Client is linked to an SAP function to be called up as a Server program, this is only executed once the Server program has completed execution. Any response values that may be required thus do not achieve the Server program call!

ERROR MESSAGE

If an error occurs when calling up a Server program, a corresponding `Exception` is returned to the SAP application:

▶ If the SAP application attempts to call a Server program that was not registered by zenon, the `Exception FUNCTION_NOT_IMPLEMENTED` is returned.

▶ If an error occurs when the linked zenon function is called up, the `Exception INTERNAL_ERROR` is returned with a description of the error.

▶ If the call via the API (`ServerCall` event) returns an error that is not `RFC_OK` (**tpSapErrorOk** = 0), the `Exception INTERNAL_ERROR` is also returned.
Recommendation: Assign, in the implementation of the Events, the `LastError` property to the SAP function with a meaningful text.

ENTRIES INTO CEL

The writing of log entries in the CEL can be controlled using the **LOG messages in the CEL** property. It is activated by default.

TRACE

The SAP interface writes logs as `Traces` in the current working folder. In zenon, this is either the program folder or the project folder. The `RFC_TRACE_DIR` environment variable can be used to stipulate a fixed folder for `Trace` files.

4.5.1   Call via the programming interface

Server programs can also be executed using the zenon programming interface. To do this, the interface of the `SAPFunction` object was supplemented with the `StartServer` and `StopServer` methods and the `ServerCall` event.

▶ `StartServer`: This method registers `SAPFunction` as a Server program. The settings of the zenon project are used for the connection to the SAP application.

▶ `StopServer`: De-registers `SAPFunction`.

▶ `ServerCall:` This event is triggered when a function from an SAP application is called up. Within events, the values of the import parameters can be evaluated and the export parameters can be set.

EXAMPLE OF A SERVER PROGRAM IN VBA

The example shows, using the `Z_RFC_ADD2` function module, how the `SAPFunction` object is initialized and registered as a Server program:

```
Public WithEvents obFunRfcAdd As SapFunction

Public Sub FbAdd()
    ' Collection of the SAP functions in the project
    Dim obFunctions As SapFunctions

    ' Parameter that is transferred to the SAP function
    Dim obParOper1 As SapParameter  ' Parameter 'I_OPER1'
    Dim obParOper2 As SapParameter  ' Parameter 'I_OPER2'
    ' Parameter that is returned by the SAP function
    Dim obParResult As SapParameter ' Parameter 'E_RESULT'

    ' === Get collection of the SAP functions
    Set obFunctions = thisProject.SapFunctions

    ' === Create SAP function "Z_RFC_ADD2" if not present
    Set obFunRfcAdd = obFunctions.Item("Z_RFC_ADD2")
```

```
    If obFunRfcAdd Is Nothing Then
        Set obFunRfcAdd = obFunctions.CreateSapFunction("Z_RFC_ADD2")
        If Not obFunRfcAdd Is Nothing Then

            ' === Create parameter if not present
            Set obParameters = obFunRfcAdd.Parameters
            If Not obParameters Is Nothing Then

                ' First operand
                Set obParOper1 = obParameters.Item("I_OPER1")
                If obParOper1 Is Nothing Then
                    Set obParOper1 = obParameters.CreateSapParameter("I_OPER1")
                End If

                ' Second operand
                Set obParOper2 = obParameters.Item("I_OPER2")
                If obParOper2 Is Nothing Then
                    Set obParOper2 = obParameters.CreateSapParameter("I_OPER2")
                End If

                ' Result
                Set obParResult = obParameters.Item("E_RESULT")
                If obParResult Is Nothing Then
                    Set obParResult = obParameters.CreateSapParameter("E_RESULT")
                End If
            End If
        End If
    End If

=== Start Server program
    If Not obFunRfcAdd Is Nothing Then
        bOk = obFunRfcAdd.StartServer()
    End If
End Sub


=== Incoming function call
Private Sub obFunRfcAdd_ServerCall(ByVal obFunction As ISapFunction, ByRef Error As
tpSapError)
    If obFunction.Name = "Z_RFC_ADD2" Then
```

```
        Set obParameters = obFunction.Parameters
        If Not obParameters Is Nothing Then
            Dim obParOper1 As SapParameter  ' Parameter 'I_OPER1'
            Dim obParOper2 As SapParameter  ' Parameter 'I_OPER2'
            Dim obParResult As SapParameter ' Parameter 'E_RESULT'
            Dim Val1 As Integer
            Dim Val2 As Integer


            ' === Get parameter objects
            Set obParOper1 = obParameters.Item("I_OPER1")
            Set obParOper2 = obParameters.Item("I_OPER2")
            Set obParResult = obParameters.Item("E_RESULT")


            ' === Read input parameter
            Val1 = obParOper1.Value
            Val2 = obParOper2.Value


            ' === Set output parameter
            obParResult.Value = Val1 + Val2
            Error = tpSapError_Ok
        Else
            Error = tpSapError_Failure
        End If
    Else
        Error = tpSapError_Failure
    End If
End Sub


Private Sub obFunRfcAdd_ActivityChanged(ByVal obFunction As ISapFunction, ByVal Activity
As tpSapActivity)
    ' If the activity status of the SAP function changes,
    ' the "ActivityChanged" event is triggered


    thisProject.Variables.Item("Active").Value = obFunction.Activity
End Sub


Private Sub Project_Inactive()
    ' === End Server program
    If Not obFunRfcAdd Is Nothing Then obFunRfcAdd.StopServer
```

```
    ' When Runtime is ended at the latest, the object of the function module
    ' must be deleted because otherwise there will be memory leaks.
    thisProject.SapFunctions.DeleteSapFunction ("Z_RFC_ADD2")
End Sub
```

4.6 Function calls

Actions in the SAP System and/or in zenon, as well as the transfer of data between zenon and the SAP system are initiated and executed by the call (on page 31) from the RFC function modules.   Standard RFC Function blocks are supplied with the SAP system. However, user-defined function blocks can also be programmed via the ABAP Workbench.

Desired SAP function blocks can be called up directly from zenon. To do this, an image of the interface of the function block to be called up is displayed in the Editor with the required parameters, tables and structures (on page 35) and linked to zenon variables. The actual function call is made by executing a zenon function (on page 36), which is given this image as a parameter.

Note: The SAP RFC interface only permits one function call at a time. To avoid overlapping function calls, the activity can be monitored (on page 32) using a variable.

SAP NAMING CONVENTION

When creating new user-generated objects in an SAP application, the following naming convention should be adhered to:

▶ The names of self-created objects should start with **y** or **z**. This prevents these objects being overwritten during an update of the SAP application. Original SAP objects never start with **y** or **z**.

4.6.1 Select function block with assistant

To select a function block from SAP:

1. In the SAP interface detail view, select the **Function calls** node
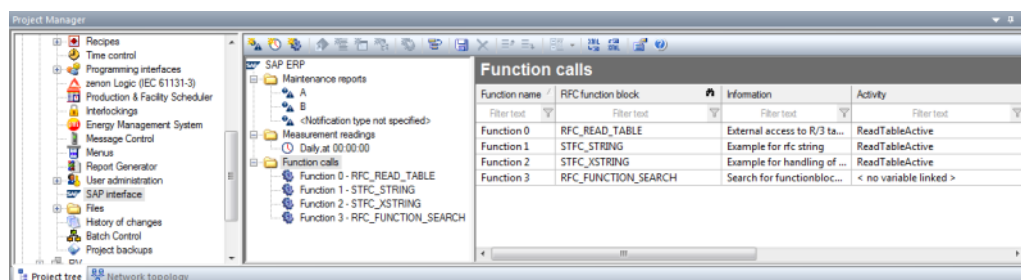
2. Select **New SAP function...** in the context menu or in the tool bar

3. The assistant for the selection of the SAP function module is opened.
   Note: To do this, the **Use assistants** setting must be activated in the options of the zenon project.
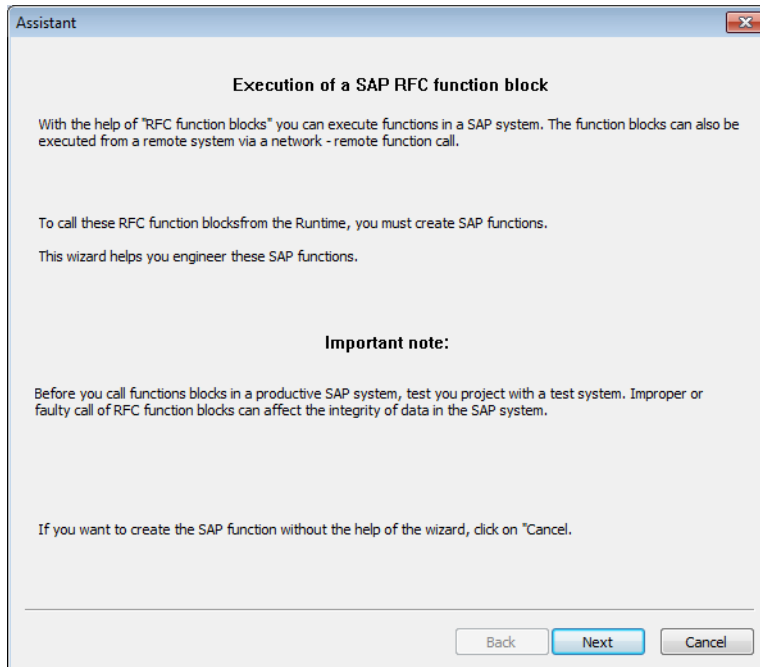
Attention: Test your project in a test system before you call up SAP function blocks in a productive system. Errors when calling up can impair the integrity of the data in the SAP system.

4. Select the desired function block.

SELECT FUNCTION MODULE DIALOG
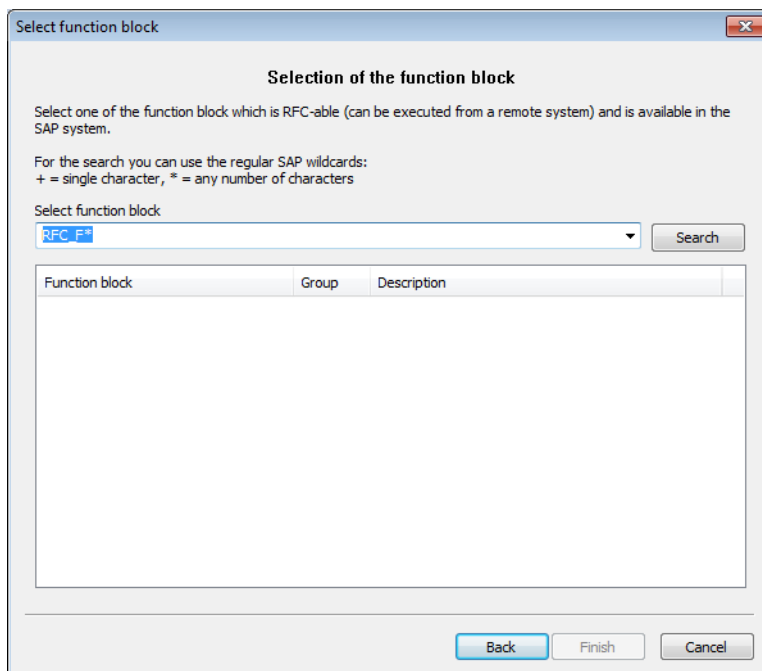
Parameters	Description
Select function block	Combobox for selection of SAP function blocks.
	The search term can be entered freely or selected from a drop-down list.
`List of function blocks`	List of function blocks found that correspond to the search term.
`Back`	Switches back to the assistant start screen.
`Finish`	Saves function block with all attendant SAP elements in a zenon SAP function.
`Cancel`	Creates an empty SAP function in zenon.

4.6.2   SAP functions

An SAP function displays a function block from the SAP system in zenon. It contains a list of objects that display the parameters and tables of the SAP function block. With an SAP function, variables can be linked that provide information on the call status and result of the last call.

CREATE SAP FUNCTION

SAP functions are created via the `New SAP function` menu item in the context menu of the `Function calls` node or the list of SAP functions.

If the `use assistants` property was activated in the project options, the assistant (on page 29) starts for the selection of an SAP function block. The assistant reads in the interface description of the selected function block from the SAP and creates all objects in the editor.

The SAP functions can also be edited manually:

▶ To compare the SAP function with the interface description in the SAP system in the process, select the `Read in RFC function block from SAP system` command. If no function block name has been given, the assistant (on page 29) is opened to select function blocks. Settings that have already been defined, in particular linked variables, are not changed in the process.

▶ When creating an SAP function, the **Function name** property receives an automatically-created name for the SAP function; this can be changed at will.

▶ The **RFC function block** property contains the name of the function block in the SAP system. The given name must correspond to the name in the SAP system, otherwise the `FU_NOT_FOUND` Exception occurs when the SAP function is called up.

Activity monitoring

A (numerical) variable linked with the **Activity** property can assume three different values depending on the status of the SAP function:

▶ `0`: The SAP function is inactive and is not currently being executed.

▶ `1`: The SAP function is queued in the call queue (the SAP RFC interface only permits one function call at a time).

▶ `2`: The function call is sent to the SAP system and a response from the SAP system is awaited. After this, the variable value goes back to `0` (`inactive`).

This variable can be used for an interlocking, for instance, to avoid overlapping function calls.

Error messages

Variables give information on the result of the last call of the SAP function. To do this, the variables are linked to the respective properties:

▶ **Error code**: contains a numerical code that corresponds to the result of the last call.

▶ **Error text**: a short description of the last error code that occurred.

▶ **Error description**: extensive description of the last error that is generated by the SAP RFC interface.

ERROR CODES OF THE RFC INTERFACE

Error code	Error text	Meaning
−1	No connection to SAP system	SAP system cannot be reached (network error, registration etc.).
0	OK	No error occurred.
1	Error occurred	Unknown error, details in the error description property.
2	Exception raised	Exception occurred in the function block (invalid parameter etc.).
3	System exception raised, connection closed	Exception occurred in the system.
4	Call received	Other function call still active.

Note:

▶ The language of the error text depends on the system language of Runtime

▶ Not all codes occur when the function blocks are called up.

Server program

To execute a function in zenon with a function call as a Server program:

1. Configure an SAP function.

2. Select a zenon function for the **Function** property.

3. Ensure that the **Current computer** property is activated for the zenon function.

4. Activate the **Server program** property for the SAP function.

5. If necessary, stipulate a waiting period is waited before and after calling the linked function, so that variables can be written or read with certainty. You set these using the **Waiting period** property.
   A waiting period is most of all recommended for variables on controllers with slow communication and/or long read cycles.

6. If necessary, link the **Activity** property to a variable (on page 32). This is set to `active` (2) when the server program is called up.

The procedure when going into the function call from the SAP application:

▶ The zenon function stated in the **Function** property is executed.

▶ The parameter values for import and export are replaced with their linked variables.

If no zenon function is linked to the SAP function, only the parameter values for import and export are replaced with their linked values. Thus the values of export parameters can be calculated from the

values of the import parameters using math variables for example, without a function having to be executed.

RFC function block documentation

The documentation for an SAP function block can be called up via the `Display documentation of the RFC function block` in the context menu.

The documentation

- ▸ must be present in the SAP system for this
- ▸ is read in in the language of the registered SAP user
- ▸ Is displayed in the web browser in HTML format.

Documentation that has already been read in is stored in the "Temp" folder of the current Windows user and called up from there next time it is displayed.

INDIVIDUAL DESIGN OF THE DISPLAY

To design the display of the documentation individually:

1. Create a stylesheet with the name `sapdoc.css`
2. Store the stylesheet in the "Temp" folder of the user

EXAMPLE FOR A STYLESHEET

```
BODY,H1,H2,H3,H4,H5,H6,P,TD,TH,UL,DL,DIV {font-family: Geneva, Arial, Helvetica,
sans-serif;}
BODY,TD {font-size: 90%;}
H1 {font-size: 120%; color: 1c1e41;}
H2 {font-size: 110%; color: 1c1e41;}
H3 {font-size: 100%; color: 1c1e41;}
H4 {font-size: 90%; font-style: italic; color: 1c1e41;}
H5 {font-size: 90%; font-style: normal; font-weight: lighter; color: 1c1e41;}
TH {font-size: 90%; font-weight: bold; text-align: left; color: 1c1e41;}
```

4.6.3 Parameters, tables and structure fields

When calling up an SAP function (on page 31), data is transferred via parameters and tables. The values of the parameters, tables and their structure fields are taken from the linked variables (incoming parameters) or allocated to the linked variables after the call (return parameters).

CREATE PARAMETERS, TABLES AND STRUCTURE FIELDS

Parameters, tables and structure fields are created via the context menu in the parameter list of an SAP function.

▶ Naming

- When being created via the context menu, the objects receive automatically-created names (**Name in SAP system** property). These must be changed in such a way that they correspond to the related objects in the SAP system.

- When creating an SAP function with the help of the assistant (on page 29), the names are automatically issued according to the interface description in the SAP system.

▶ Parameters and tables

- When the SAP function is called up, only those parameters and tables are used that are also actually in the interface description of the function block in the SAP system.

- If non-optional parameters or tables are not configured or not linked to variables, they are populated with initial values when the SAP function is called up. This generally leads to an Exception and thus to an error message when a function is called up.

▶ Informative properties

- Some properties of parameters, tables and structure fields (type, optional and info) are only for information. These are populated with the information from the interface description in the SAP system by the assistant when an SAP function is created, but can be changed at any time.

- They are not evaluated when the SAP function is called up. Instead, the actual inputs of the interface description in the SAP system are used.

4.6.4 Link variables

The **Variable** property is used to link variables of an SAP function with parameters, tables and structure fields (on page 35).

To link variables:

▶ Use either the **Variable** property in the properties window: Click on the . . . button to open the dialog for selecting variables
or

▶   Drag & drop the desired variable onto an object in the parameter list with the mouse

If structure variables are linked to structured parameters and tables, the variables of the structure fields are also populated with the structure element of the variable if possible.

Attention: This currently only works when linking via drag & drop.

CALLING UP AN SAP FUNCTION

When calling up an SAP function (on page 36), the values of the variables are read, which are linked with parameters and tables for the transfer of data to the function block and are allocated to the parameters and tables.

And vice versa, the parameters and table values that are returned from a function call are allocated to the linked variables.

ROW NUMBER FOR TABLES

Objects for tables can be linked via a variable with the **Number of lines** property.

Accepting values:

▶   Before a function call: The value of the variable determines the number of table lines processed in zenon and provided by (`Number of Lines`).

▶   After the function call: the value provides the number of lines in the table returned from the function call.


4.6.5   Call up SAP functions with zenon

SAP functions are called up using the zenon function **Execute SAP function**. The function has the internal ID of the SAP function to be executed as a parameter.

To call up an SAP function of zenon in Runtime:

1.   Create a new zenon function via Functions -> Context menu -> New functions

2.   Select, in the **Application** node, the **Execute SAP function** entry.

3.   The dialog for configuring functions is opened.

4.   Select the desired SAP function

5.   Confirm the configuration by clicking on the oĸ button

EXECUTE SAP FUNCTION DIALOG
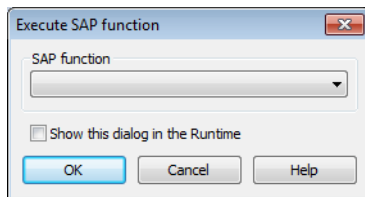
Parameters	Description
SAP function	Selection of an SAP function from drop-down list. The function must already be created in the project.
Show this dialog in Runtime	Opens this dialog in Runtime for the execution of the function.
`OK`	Creates function with the selected parameters.
`Cancel`	Creates function without parameter.
`Help`	Opens online help.

Hint: In the editor, the `Create standard function` item in the context menu can be used to automatically create a function for a selected SAP function, which executes the selected SAP function in Runtime.