ABAP Web Service Monitors

The SAP ABAP Web Service Infrastructure provides a set of monitors to display information on the different entities involved in Web Service communication. The monitors are connected to the support utilities for Web Services.

Figure 1 Navigation Graph

Figure 1 shows the possible navigation between the different monitoring views and the connected support utilities. The diagram is a state transition diagram where the round corner nodes represent the states — in this case views/screens. The white on black numbers in the round corner nodes should make it easier to relate the text and the screenshots to the navigation graph. The sharp corner nodes represent activities triggered from screens. Such activities also exist between two screens — like execution of a query when going from a selection screen to a list screen — but in most of the cases these are not shown to keep the figure simple. The bold arrows represent forward and back (F3) navigation. The thin lined arrows represent navigation paths where one cannot navigate back.

The central entity in Web Service communication is a message. Therefore one usually enters the monitors via the message monitor. The corresponding transaction name is SRT_MONI. This entry point is shown at the very top of Figure 1. As only asynchronously processed
messages are persisted the monitor displays only these. You enter the message monitor via an initial message selection screen (Figure 2) where you can do your selection via a time range and/or via values for different Web Service entities e.g. message ID or sequence ID.

![Web Service Utilities: Message Monitor](image)

**Figure 2** Initial message selection screen - Standard selection -

On the selection screen you can decide if you want to first display a summary of the messages matching the selection criteria on the bottom part or if you want to directly display the list of messages. In addition you can switch between displaying a restricted or an extended set of message attributes in the message list – basic view vs. technical view.

The selection criteria are partitioned into two sets, standard and advanced selection. In case of standard selection (Figure 2) you have two selection modes. You can either select based on an explicit set of IDs – messages or all messages belonging to a list of sequences. The second mode is based on a time range with optional addition of message attribute values. If you do not enter any additional selection criteria the prefilled time range is used for the summary display as well as for the detail message list display.

The attributes for time range based selection are either direct message attributes like adapter type, user name or sender/receiver party and interface name or the derived attribute processing status group.

The values for adapter type are:

- Plain SOAP – non-WSRM based SAP reliable messaging
- Groupware scenario – not available in NW releases below 8.0
- Mapping - not available in NW releases below 8.0
• Eventing – publish/subscribe mechanism which is based on WS messages
• Standard Webservice – native Web Services
• Shortcut – local Web Services that are called in a performance optimized way
• Integration Server – in an XI environment messages form integration server

The processing status is a derived message attribute that combines the native message state, the sequence state, the asynchronous processing queue (aka bgRFC queue) state and for on-failure (eg. Tentative Update and Confirm/Compensate – TUC/C) messages the respective state. The processing state tells if there are problems in the processing of a message or not. All states listed above need to be considered to make such a statement.

In order to provide an easier management of the processing states these are grouped into the processing state groups. The values for the processing state groups are:

- **Application error** – an error occurred in the application that implements a service provider.
- **System error** – an error in the processing infrastructure occurred.
- **Erroneous** – this state group combines “Application error” and “System Error”.
- **Terminated with dump** – message processing led to a short dump.
- **Wait for processing** – the message is waiting for processing. The reason can be that the message is currently not the first in the processing queue. In that case all predecessors need to be processed first. It could also indicate an overload situation preventing a fast processing of the message.
- **Incomplete processing** – this state group combines “Application error”, “System error” and “Wait for processing”.
- **Incomplete (no “pulling”-messages)** – in NW releases below 8.0 the same result than “Incomplete processing” (pulling is a synonym for groupware).
- **Cancelled** – the message was manually cancelled or it is an automatically annulled compensate message in TUC/C communication.
- **Isolated** – the logical port of the message is isolated. Therefore the message will not be sent to the service provider system.
- **Cancelled or isolated** – this state group combines “Cancelled” and “Isolated”
- **Finished** – the message was successfully processed.
- **Incomplete (no “pulling”-messages)**

The advanced selection tab is shown in Figure 3. It provides more technical attributes for the message selection query. You can again either select via a set of identifiers for the asynchronous processing queue or the message persistence entry. As a message ID can actually represent two messages – one at service consumer side and one at provider side, both could be stored on the same system – the message persistence ID is a unique identifier for a message.

**Figure 3** Initial message selection screen - Advanced selection -

The second method to query messages is here again based on a time range. This time the SAP passport identifiers root context ID and transaction ID can be used to further restrict the result set. The SAP passport can be used to identify communication entities in a cross system communication. It can be helpful to get an end-to-end monitoring view in a communication scenario where the message ID cannot be used as unique ID throughout all involved systems.
When you decide to first display the message summary view you’ll get a result similar to the one in Figure 4. The numbers of messages for the following status groups are displayed.

- Application Error
- System Error
- Wait for Processing
- Cancelled
- Isolated
- Finished

Via line selection and clicking on the glasses button or by double click on a line you get to the message list for the respective status group.

As already mentioned the message list (Figure 5) can be displayed in two different flavors. The basic view shows a predefined set of message attributes which does not show some more technical attributes like the unique message persistency ID which are shown in the technical view. The technical view also does not show all possible attributes for a message. The user can define a personal view where he can select a personalized set of attributes to be displayed.

From the current message list one can trigger a reselection of the messages to be displayed in two ways. Either one chooses a general reselection which displays a popup that gives similar select options than the initial selection screen. In this case the reselection has no relationship to the currently displayed messages. Alternatively one can display the set of related messages. These are related to the currently selected messages (by the user in the table control). Here one can choose:

- Sequence messages – all messages belonging to the same sequences than the selected messages are displayed. This is an extension of the current selection.
- Parent messages – all parent messages to the currently selected messages are displayed. A parent/child relationship exists for example in the event scenario, where starting from an initial event message several (or none) child messages are created that have the initial event message as parent message.
• Child messages – all child messages to the currently selected messages.

The message list can be exported in different formats which is a standard functionality of the table control. In addition the selected messages in the table control can be exported in XML format.

The following help texts are available. A description of the fields in the table control can be displayed via clicking the “i” button. A description of the processing state values can be popped up via Message menu. The PAF states can be ignored in releases below NW 8.0. A description of a specific processing state can be displayed via clicking on the respective underlined entry in the processing state column of the table three.

There are some actions that can be executed to the messages selected in the table control.

These actions are:

• Restart – only messages in error state can be restarted. The messages are added to the asynchronous processing queue and by that will be processed again. A restart makes for example sense if since the last execution trial that led to a communication error the configuration was changed.
• Restart with debug – only allowed if a single erroneous message was selected. The debugger is started in the WS runtime code not in the application code.
• Cancel – only messages in error state can be cancelled. Cancelling a message sets it to a final state without being processed. In some cases a message cannot be cancelled in the message list screen. It then needs to be cancelled form the inconsistent message list screen.
• Consistency check – only available in technical view. For the selected messages a consistency check between the message state data and the asynchronous processing data is executed. It leads to the inconsistent message list screen. For technical reasons inconsistencies are only detected 5 minutes after the activity that led to the inconsistency took place.

The consistency check leads to the inconsistent message list (Figure 6). It looks similar to the general message list with some different functionality. This view is intended for support activities as it shows some more technical detail information than the message list.

In the inconsistent message view you cannot do a general reselection but instead you can run a consistency recheck for the displayed messages. You can enforce the cancellation of the selected messages skipping some checks that are done in the general message list cancellation action. Reconciliation is a functionality that is only available in releases from NW 8.0 onwards. In lower releases ignore this button.

![Web Service Utilities: Consistency Check](image)

**Figure 6 Inconsistent message list**

From the message list and from the inconsistent message list one can display details for a single message (Figure 5). For this one can either click on the respective message ID (underlined) in the table control or double click in a field of the message list where the content is not underlined.

![Web Service Utilities: Message Properties](image)

**Figure 7 Message details - Payload view**

Figure 7 shows the payload view which is the default view that you reach when you want to display message details. The message payloads are viewed in a table on the right side. In this example one can see that in case of error the initial message version and the error version are available.
An XML view of the payload can be displayed in a popup window.

Via the message menu one can switch to a different view which shows apart from the payloads some more message information like header information (Figure 8). It also shows more detail about the actual message structure. For example you can see in the example that the payload is stored as an attachment to the actual message.

![Web Service Utilities: Message Properties](image)

**Figure 8** Message details - Extended view -

Returning to the central message list one can navigate to several other views and tools. The entity most related to the message is the sequence. It controls the reliable messaging – WSRM or SAP’s Plain Soap. In SAP’s WSRM implementation also messages with quality of service exactly once (EO) are sent in the context of a WSRM sequence – in this case with one business message entry plus a technical terminate message. That’s why for all asynchronous messages a related sequence exists. The sequences related to a certain message set are displayed in the sequence list (Figure 9).

![Web Service Utilities: Sequence Monitor](image)

**Figure 9** Sequence list

The sequence list screen is looking quite similar to the message list. You have again reselection possibilities. Some relating to selected sequences and also a general reselection popup. As actions related to the selected sequences you can trigger a soft or a hard termination. A **soft termination** closes the sequence on consumer side for new messages. If the sequence contains unfinished messages these are processed and after the last message is processed the sequence is set to state closed. In case of a **hard termination** the sequence is terminated immediately and all not yet processed messages are cancelled.

Usually one would access the sequence list via the message list as the message is the central entity. In some cases it is useful or even necessary to enter the message monitor via selecting a sequence directly. For example if there are no (more) messages existing for a persistent WSRM sequence. That sequence can only be found via the message selection entry.

A direct access to the sequence monitor is done by starting transaction SRT_MONIS. First the **initial sequence selection** (Figure 10) screen is displayed. Here one can either enter a list of explicit sequence IDs or do a selection based on a time range adding some sequence attributes. From the selection screen one gets to the sequence list screen.
From the message list, the inconsistent message list and the sequence list one can navigate too the queue entry list. In this screen either the update task monitor (Transaction SM13) is displayed or the asynchronous processing queue (bgRFC) monitor (Transaction BGRFCMON - Figure 11). The update task monitor is only displayed if there is a pending update task in relation to the selected message(s).

The asynchronous processing queue monitor shows in case of multiple message selection only the data for the first message in the selection set. The data of the processing queue is displayed in the left part. The queue ID is in case of WSRM communication composed of a prefix and the sequence ID. On the right side on can see the units in the queue. Each message is associated to one unit. On provider side this is a 1:1 relationship. Technical messages like a close sequence message are also visible as a unit. The state of the unit and the queue is displayed as traffic lights. In the example in Figure 11 there is a problem in processing.

The queue entry list is intended for specialists who have deep knowledge of the message processing infrastructure. Especially the actions that can be executed here can lead to inconsistent processing states if the user does not exactly know what he/she does.
Figure 12 Application Log

From the message list and inconsistent message list one can navigate to the application log entries (Figure 12) related to the currently selected messages. There are two flavors, message related entries and context related entries. In the first case all application log entries which contain the message ID, a sequence or a TUC/C (tentative update and confirm or compensate) compensate message related to the selected messages set are displayed. A message log entry is for example written if a message is manually cancelled or a sequence is manually terminated. Context related application log entries are related to the message processing infrastructure independent on specific processed messages.

A very important tool for error analysis in message processing is the error log. From the message list the inconsistent message list and the sequence list one can navigate to the error log list (Figure 13) for the selected messages/sequences. Already in the list there is some technical data displayed for the respective error. Most of it is cut off in the screenshot. Very important for root cause analysis is the error text. By double clicking on a line further error detail is displayed below the list. In the bottom area one can for example trigger the display of the call stack at the moment when the error was detected or the http error page sent back to the service consumer. From the error log list one can navigate to the trace view for a selected error log entry. Traces have to be switched on in advance in order to get a result.

The full functionality of the error log and trace functionality (Web Service Support Utilities – transaction SRT_UTIL) is described in a separate document Web Service Utilities. These tools are intended for support by the development teams of the Web Service infrastructure. For SAP customers the error log short text and the error log data can help to find the root cause of a problem in message processing.

The error log tool and the traces tools are accessed directly via transaction SRT_UTIL. From the entry screen one can start the error log viewer one can switch on and configure traces and display the trace results. There are three different types of traces:

- Performance trace – for the message processing infrastructure. Regarding the service provider application code only the total time is recorded.
- Payload trace – to trace details of the message payload.
- Functional trace – to see more details on the data inside the message processing infrastructure.

All this is not contained in Figure 1 to keep the picture simple.
From the message list the inconsistent message list and the sequence list one can navigate to the list of **runtime errors** (aka ABAP short dumps) relating to the selection **(Figure 14)**. Via a double click one can navigate to the **runtime error detail** view well known to ABAP programmers (ST22). In case of an error in the service provider implementation where it makes no sense to proceed usually a short dump is triggered and the error data is logged.

In the runtime error list one can do a reselect and export the list data. For error analysis it usually makes more sense to navigate to the error detail view and export the detailed error info for further use.