Welcome to the training on scenario development for the integration framework of SAP Business One. This course introduces the scenario monitoring and debugging tools provided by the integration framework.
When a scenario runs in production, you need to know how to troubleshoot if something goes wrong. After completing this course, you will know how to:

- Monitor events from SAP Business One
- Perform general monitoring of a running scenario step using the message log
- Display error messages and exceptions
- Debug and retry a scenario step from the message log
- Use the monitoring tools to monitor internal queues and processing calls
- Monitor and debug the internal process steps and processing flow of the integration framework
The integration framework provides various tools for monitoring different parts of the integration processing, including:

- The B1 Event Monitor and Event Sender for monitoring SAP Business One events
- The Message Log for filtering and viewing XML messages passing through the integration framework, with the option to debug a running scenario step
- Cockpit and Process Control for overall monitoring for errors and exceptions in the integration framework processing
- The Queue Monitor for monitoring the internal queues of the integration framework and the Call Monitor for monitoring calls
- Process Control and the Control Center for monitoring and debugging the internal processing of all running scenario steps in the integration framework
The cockpit tab was introduced in SAP Business One release 9.1 PL08 and provides a quick overview of any potential errors.

You can see instantly how many scenario packages are active and the overall number of messages processed today. Error information is shown including errors in receiver systems, blocker internal queues, and activation conflicts. System resource information includes BizStore database size and resource usage.

If you see an error in any of the widgets, you should locate and correct the error. You can click in the widget to see debugging information including the error message.

Note that there is also a cockpit available in the Control Center tool of the integration framework. This cockpit is part of the B1iP Control Center and shows status information on the internal processing of the integration framework.
SAP Business One Event Monitoring

The first topic covers specific monitoring tools for SAP Business One events.
Imagine that you have a scenario that uploads sales orders daily from SAP Business One to a SAP ERP system. The scenario has been running for several months. You get a call that no sales orders have been received from SAP Business One. Yet you know that sales orders have been created. Where can you look first?

Events from SAP Business One are held in the SEVT table of the common database.

The SAP Business One event sender, the DI Proxy and the SLD are all involved in the exchange of the event from the event table to the integration framework.

On the SAP Business One server, make sure all services are running.

Imagine that you have a scenario that uploads sales orders daily from SAP Business One to a SAP ERP system. The scenario has been running for several months. You get a call that no sales orders have been received from SAP Business One. Yet you know that sales orders have been created. Where can you look first?

Events from SAP Business One are held in the SEVT table of the common database. The SAP Business One event sender, the DI Proxy and the SLD are all involved in the event exchange of events originating in SAP Business One to the integration framework.

On the SAP Business One server make sure that the event sender and DI Proxy services are started and running. If the event sender is not running, events will be queued in the SEVT table and will not reach the integration framework.
To see events from the event sender, you can use the B1 Event Monitor in the integration framework. The B1 Event Monitor is selected from the Monitoring tab and shows events coming from SAP Business One systems. For a production system this log is normally turned off, so you need to enable it first by selecting from the Log dropdown list. For a development system this log is already enabled in the system profile (from SAP Business One release 9.1 PL09 and up).

You can display events by status: success, failure, or filtered. Choose the Refresh icon to refresh the display.

A successful event is shown here, and you can see the company database, object ID 17 (a sales order), the document number in the Key Value column, a timestamp, the target host name and the name of the internal processing step in the integration framework. Internal processing steps will be covered later in this course.
Most likely you will be looking for events that fail, but if you see an event with the filtered status, this can be caused if:

- The SAP Business One system is not defined in any active scenario sender list. Sender lists are defined during step setup and activation.
- The sender SAP Business One system is defined in a sender list, but does not meet any filtering conditions for the event. Event filters are established using the B1 Event Filter Generator tool.
- There are no active scenario packages defined for the event, or a scenario package has been deactivated.
If the event is displayed with the failure status, you will see an error message as the final field in the log entry.

For both successful and failed events the log entry includes two action icons:

- The left-hand edit icon enables the Target B1i Application field for editing. You should only edit this field to change the sender system ID if there is a mismatch between SAP Business One company databases and integration framework information.
- The right-hand icon allows you to retry or resend the event through the integration framework processing.
If there are no events shown in the B1 Event Monitor, it could be that there are errors or inconsistencies in the event sender configuration and SAP Business One notification settings. The event sender uses the SAP Business One notification mechanism.

The Event Sender troubleshooting tool can check this for you. To locate the Event Sender troubleshooting tool, navigate to the Tools menu and choose the Troubleshooting option, then select the B1 Event Sender from the dropdown list of functional groups. Choose the Submit button to open the B1 Event Sender troubleshooting tool.

The tool provides five checks that you can run individually. Choose Start to run the check for each of the listed categories. After the check completes, the Display button will be enabled to show the result of the check.
The event sender checks display:

- General system information
- Configuration information for the event sender
- Status of the notification mechanism for each SAP Business One company database. Since notification problems can prevent events from being sent, this is something to check.
- A report of events stored in the SEVT table for each company database. An example of this report is shown in the slide. If there are events for a database, select the Events button on the row to see the event details. If events remain in the SEVT table, you should investigate why they are not being processed.
- A summary of all information, with an overview for each company database. A sample of this report is shown in the slide.
Monitoring Messages using the Message Log

If events from SAP Business One are reaching the integration framework, the event sender is working properly.

You should next check the XML messages in the integration framework using the Message Log.
The Message Log shows all messages being processed through the integration framework:

- Message logging is by default switched on in both productive and development system profiles, or you can enable it by selecting the checkbox.
- Set the log level to Full message so that the full XML message is logged.
- Enable daily deletion so that log entries are removed each day.

System profiles were introduced in SAP Business One release 9.1 PL09, and message logging is by default switched on in both productive and development system profiles. If the message log is turned off in your environment you can enable it using the path shown in the slide and selecting the Message Log checkbox. The Indexed Access checkbox is enabled by default and should not be disabled.

In a productive system profile the log level should always be set to “Infoset” to keep the log database to a minimum. However if you are debugging in a productive system, set the Log Level parameter to “Full message” so that the full XML message is captured in the message log. Remember to set the log level back to Infoset after you finish the debugging in the productive system.

**Important:** SAP recommends you enable Daily Deletion. When you do this the message log entries are automatically removed each day, and you will only see entries in the message log for a maximum 24 hour period.
The message log displays information for each message entering the integration framework. The information recorded in the log includes scenario package and step, sender and receiver system, trigger, date and time.

In this screenshot you can see a successful message highlighted in the log. You can filter the displayed messages using various filters, such as status, date, sender or receiver system, scenario package, or object.

The status filters are Success, Failure, Processing or Filtered. Filtered messages are messages that no scenario step has subscribed to, or messages that the processing logic has filtered.

Note: If you have enabled daily deletion of message logs, which is recommended, you will only see messages for the current 24 hour period.

For each scenario step, you have the option to define two search keys. Search keys are defined in the scenario package. Choose Scenarios → Package Design, and select the Definitions button. You can define search keys to easily search for a specific item code or document number. The search key values are entered as XPath expressions.
If the message has the failure status, the status icon and result message are both highlighted in red.

Result message will contain error message details

You can delete or export messages from the log.

The four message log icons apply to all messages and allow you to:

- Delete one or more selected messages from the log
- Delete imported messages that were previously imported
- Export messages
- Show or hide columns in the log. This opens up a list of column names, allowing you to select just the columns you want to display instead of scrolling sideways.
The Last Exception button in the message log display shows a list of all internal process steps that have ended with an exception. The integration framework processing for a scenario consists of internal process (IPO) steps, and this report allows you to quickly scan for step processing errors.

The Last Exception button shows a report called the IPO-Step Status Panel. This report is also available to run from the B1iP Control Center.

In the slide you can see highlighted:

- The application name of the processing step. This link also contains the system ID and the step name.
- The value for the activated field below the application name. This will be false if the step has been deactivated, and you have the option to reactivate it to run again. If the step is active, you can deactivate it to prevent recursive errors.
- The TID link that opens up a panel allowing further debugging. This will be covered later in this course.
Sometimes connectivity problems or incorrect data mappings can prevent the message reaching a receiver system. If an error occurs when the integration framework tries to hand over a message to a receiver system, the integration framework puts the message into the error inbox of the receiver system. Typical errors displayed include connectivity problems to the receiver system, system availability, or incorrect data mappings.

To see the error inbox for each receiver system defined in the SLD, select the Error Inbox button in the Message Log display.

In the error inbox list, select the receiver system marked with a red cross to open the error information. From this window you can see the error message. You have the option to retry the message or delete it.
This topic shows you how to drill down from a message to debug the internal processing of the integration framework.
For each message shown in the message log, the Status column for a message shows two icons:

1. The message status (green for success or red for failure). If an exclamation mark is displayed for the status, the message has INCOMMIT status and the icon will be selectable and will show more information.
2. An icon that indicates the message is from the local integration framework system, or is imported.

The Actions column icons allow you to start debugging the internal XML message. From left to right, the actions are:

- Debug the message. This opens the transaction ID panel for debugging.
- Open (view) the XML message.
- Retry the message.

Let’s look at these actions in more detail.
The middle action icon opens the current XML message. If the message ended with the status of failure or success, this is usually the final message.
You can view the sections of the message, such as the namespace information at the top, the header section, and the body section with the sender trigger, the sender payload and the receiver payload. You should check for any error messages contained in the XML.
Note that you only get all the sections if you have set the log level for the message log to Full message.
After you have fixed an error, the *Retry* icon allows you to resend the message through the integration framework. The icon opens a dropdown list that allows you to retry or resend the message from different points of execution (new from start, step processing and outbound, or only outbound). You should only do this in a development system and for obvious reasons you should not retry a message in a production environment.

Note: The Retry icon is enabled only if you have set the log level for the message log to *Full message*. 
The Debug action icon in the message log entry opens a window where you can debug a scenario step.

The example shows an asynchronous scenario step. The integration framework uses internal process steps (IPO steps) to process a scenario step, and the sequence of these steps is shown by the arrow. You can see the OUT_FILE step has an error, since the status “C” indicates the transaction was cancelled, and the error handler has been called.

For each process step the status is shown, and you can using the various buttons to drill-down to more detailed debugging. This will be covered later in this course.

Note that to see the details shown here in the Transaction-ID Panel, you need to enable the capture of detailed debugging information. This will be covered next.
This topic shows you how to drill down to debug the internal processing of the integration framework.
To debug the internal process steps of a scenario, you need to first enable the capture of debugging information in the transaction log.

In the development system profile:
- Set xCellerator log level to CONFIG
- Select Scenario debug info checkbox

Note: In the production system profile detailed debugging information is not captured but you can dynamically enable it from the Process Control screen.

In a development system profile, the xCellerator log level is set to CONFIG which is the most detailed level of logging. The debug information is collected in a transaction log that is separate from the message log. This transaction log has an in-built garbage collection process which runs in the background.

Additionally, the Scenario Debug Info checkbox is selected so that debugging information is viewable when debugging a scenario.

In a productive system profile the xCellerator log level is set to SEVERE and the Scenario Debug Info checkbox is not selected. This means that errors are logged in the transaction log but no debugging information is captured. This is done for performance reasons. If you want to have the debugging information logged in a productive system, you can dynamically switch this on for a scenario process from the Process Control screen. This will be covered later in this course.
When you choose the *Debug* action icon in a message log entry, you can see the internal processing steps of the integration framework.

Using the *Graphical View* link in the debug panel, you can view the IPO steps in a graphical way.

- An asynchronous scenario step, as shown here, consists of multiple process steps that use the internal queue mechanism to hand over the message to the next step. The example shows a B1 event as the inbound and a file outbound. The integration framework displays the starting point of the message flow in the upper left corner. For each scenario step subscribed to the incoming message, a process step (PRC_*) is available. The integration framework processes your own scenario step inside this process step. If there is more than one step subscribed, the box is grey not orange, and you can drill-down to each scenario step. If there is only one receiver system, the integration framework leaves out the distribution step – this is indicated with a white box. The last process step is the outbound to receiver systems. If there is one receiver object, the box is orange, otherwise it is grey, and you can select the receiver system processing you want to debug further.

- A synchronous process flow consists of an inbound step and a processing step. The integration framework processes your scenario step inside this process step. There is no outbound step since the processing IPO step returns the result to the original sender.
The BizFlow processing flow window allows you to track and follow the internal atoms for a scenario step. The presence of a red arrow in an atom indicates that the atom was executed. You can follow the red arrows to check where processing stopped and drill down to sub flows. Each atom displays the duration time and the size of the inbound message.

You can open the processing flow window from the message log. Select the Debug action icon in the message entry to open the Transaction-ID Panel. Then select either the Debug button for an internal process step, or in the graphical view, click the orange box for any internal process step.

By selecting either the red chevrons or the red arrows throughout the process flow, you can:
- Check the inbound parameters to an atom
- Open the incoming XML message for an atom
- Check configuration details for an atom

**Note:** To see the flow debugging arrows, you must enable scenario debugging for the step processing. If you are using the development system profile, the scenario debug checkbox will be enabled for you. You can also enable scenario debug from the Process Control section of the Monitoring tab. Process Control will be covered later in this course.
Further debugging information is available for the scenario step using the buttons in the Transaction-ID Panel:

- **IPO-Log** button shows a log with detailed information. In case of an error/exception in the integration framework you will find the error information and call stack at the end of this log.

- **MsgFlow Inq** button is useful for complex processing flows where a message is subscribed by different scenario steps, and the processing inside the integration framework ends up as a tree. With the message flow inquiry option you can follow the integration processing tree structure from the root level through to successor transactions. For each step, you can display the flow analysis.

- **MsgProcTmng** button shows the timing information for each IPO step

**Note:** This detailed log information is available only if you have set the xCellerator log level to CONFIG in the system profile.
Continuing the buttons available for each IPO step:

- **ExecDetails** displays all processing step details in one document.
- **Debug** opens the BizFlow processing flow debugging window.
- **Profiler** allows checking potential performance issues on different levels.
- **Delete** removes the log entry.

**Note:** This detailed log information is available only if you have set the xCellerator log level to CONFIG in the system profile.
The next topic covers additional monitoring tools.
The integration framework uses internal queues when processing asynchronous scenario steps.

If messages do not appear in the message log, it could be that they are stuck in an internal queue. The *Queue Monitor* displays all internal queues that contain a message. Typically this log will be empty, since messages are processed immediately therefore you will only see a message in this log if something has gone wrong. If a message gets stuck in the queue this will block subsequent messages that use the same queue.

Each queue has one or more sub-queues called streams. The above example shows a processing of an incoming B1 event. When you click and expand the queue in the Queue column, the stream will be displayed in the Streams column next to it. By clicking and expanding a particular stream, all available messages are displayed in the third column.

For each message you can use the buttons to:
- View the message
- Show the process that has created this particular queue
- View the message flow details
- Delete a message. To remove all messages from a stream, use the *Clear* button.
The **Call Monitor** report shows a report of all calls made by transactions, regardless of scenario.

You can select the types of calls for the report, for example, email, HTTP, queue, SQL, Void, etc.

In the report you can view and optionally debug call issues including SQL errors.

The **Call Monitor** report shows a report of all calls made by scenario step transactions, regardless of scenario.

You can select the types of calls for the report, for example, email, HTTP, queue, SQL, Void, etc.

Calls happen in the inbound, processing, and outbound phase, during authentication and in the framework itself and this report can help with troubleshooting issues with calls.

The report shows the time of the call, the type, duration, and the inbound and outbound from the call. For debugging SAP Business One scenarios, choose SQL as the type to view SQL statements and database results.

Note: to see the call monitor information, you must have set the xCellerator log level to CONFIG in the system profile.
Process control provides a very powerful tool for overall monitoring of the process steps for multiple systems. To open process control, choose the Monitoring tab then the Process Control option.

The first three sections of the process control screen show a list of all systems and scenario steps categorized by processing phase (inbound, processing, and outbound). You may notice that some steps are shown multiple times, with different systems. For each section you see a count of transactions executed for each scenario step that are:
- Started or running
- Have INCOMMIT status
- Have INROLLBACK status
- Cancelled
- Completed successfully

If you see transactions stuck in INCOMMIT status, you can force the transaction to complete using the Clear INCOMMIT Transactions button.

In other sections of Process Control you can see the internal processes of the integration framework and inter-company processes (if the Intercompany Integration Solution for SAP Business is installed).
A very powerful feature of Process Control is the ability to dynamically activate or deactivate a specific process step using the *Active* checkbox on the row, or to activate or deactivate all process steps for a phase using the *Activate* and *Deactivate* buttons at the top. When deactivating a process step, use with caution as a single process step could be shared by multiple systems.

Similarly, you can dynamically switch on or off the capture of scenario debugging information at run time. Use the *Debug* checkbox on the process row to switch on or off the capture of detailed information for a single step and system, or use the *Debug On* or *Debug Off* buttons for all process steps. When you rerun the scenario, you can view the scenario debugging information in the Transaction-ID panel and drill down to the processing flow.

Note: If you are running with the development system profile, where scenario debugging is already enabled, you will see that the Debug checkbox is already selected for all step processes.
As well as providing an overview of all running step transactions, Process Control allows you to debug a process step.

If the Exception checkbox is marked for a process step, this indicates an exception occurred in the processing. An example is highlighted in the slide. You can select the Debug button to open the Transaction-ID panel. This is similar to the panel that you open from the Message Log; except that it shows a list of all the internal steps executed for the selected processing phase (Inbound, Processing or Outbound) for the selected row (scenario step and system).

You can drill-down to the IPO Log and the processing flow atoms using the buttons. These buttons were described earlier. To see this information you need to have switched on the capture of scenario debugging information.
In SAP Business One release 9.1 PL08 and higher, you can configure which sections will be shown in the Process Control report. The default display shows several sections including Scenario Processes, Framework Processes, and Individual Scenario Package Processes. You can change the sections displayed by choosing the *Configuration* icon, highlighted in the slide.
This topic covers the Control Center tools for inspecting the internal processing of the integration framework.
The **B1iP Control Center** is a tool for low-level configuration and troubleshooting of the internal process steps of the integration framework.

When you select this tool from the *Tools* menu, the Control Center opens in a separate browser window.

The Control Center has a cockpit that shows the overall status of IPO process steps and transaction execution status. Also shown are database queue status, BizStore status, initialization status and server status.

You can click in the widgets to view more details. For example, you can open the IPO-Step Status panel as shown in the slide. This report is similar to the Last Exception report you can run from the Message Log.
The Control Center also provides a Monitoring tab with a set of monitoring tools as well as various statistics such as the results of the garbage collection runs. Some of the tools can be useful for scenario debugging. The example shown here is the IPO-Step status. This is similar to the Last Exception report you can run from the message log; however you can enter search criteria for a specific step, deactivated steps or steps with an exception.
The Transaction-ID panel can also be viewed from the Control Center Monitoring tab. This is another way of viewing the scenario debugging information that you can see from the Debug action icon in the message log.

Since this is a report, selection criteria can be entered, such as timestamps, step, and execution status. Here we have selected steps that have been cancelled.
Quiz 1

Why would a message be displayed in the filtered section of the message log?

Take a moment to decide your answer then advance to the next slide.

A. One of the internal process steps ended in an error
B. No scenario step has subscribed to the message
C. The message threw an exception during IPO step processing
D. A search key was set up for the scenario step
Quiz 1 Answer

Why would a message be displayed in the filtered section of the message log?

A. One of the internal process steps ended in an error
B. No scenario step has subscribed to the message (correct)
C. The message threw an exception during IPO step processing
D. A search key was set up for the scenario step
Quiz 2

What do the three icons in the Actions column of the message log entry allow you to do? 📚🔍➡️

Take a moment to decide your answers then advance to the next slide.

A. Debug the message
B. Open the flow analysis diagram
C. Retry the message though step processing
D. Open the message XML file
E. Clear a transaction with INCOMMIT status
F. Print the log message
Quiz 2 Answer

What do the three icons in the Actions column of the message log entry allow you to do?

A. Debug the message (correct)
B. Open the flow analysis diagram
C. Retry the message though step processing (correct)
D. Open the message XML file (correct)
E. Clear a transaction with INCOMMIT status
F. Print the log message
Thank You!

For more information visit SDN at http://sdn.sap.com