This course covers the recommended settings for using the integration framework as a development environment for creating new scenario packages. This course is only relevant if you plan to use the integration framework for scenario package development.
After completing this topic, you will be able to:

- Change the integration framework environment from a productive system to a development system
- Configure the message log for use in a development environment
- Further configure the development system by:
  - Setting up notifications for blocked queues
  - Changing the default error handling for scenario processing
  - Expanding Tomcat memory
  - Starting the integration framework in safe mode

At the end of this course, you will be able to:

- Change the integration framework environment from a productive system to a development system
- Configure the message log for use in a development environment
- Optionally make additional configurations in a development system by:
  - Setting up notifications for blocked queues,
  - Changing the default error handling for scenario processing
  - Expanding Tomcat memory
  - Starting the integration framework in safe mode
The first topic describes the different system profiles for productive and development environments.
The integration framework can operate in one of two modes – either as a productive system or a development system. This is controlled by the system profile. A system profile brings together the recommended configuration settings in one place, optimized for a productive system or a development system.

Each mode has a color-coded upper line – blue for a productive system and yellow/gold for a development system.

After installation, the integration framework is set for a productive system profile. If you want to create and test new scenarios, you need to switch the system profile to a development system.
When you use the integration framework for scenario development, it is useful to know the capabilities and features available to you.

Select the *System info* function under the Maintenance tab of the integration framework user interface to see information about the installed version of the integration framework. If you need to report integration framework errors to SAP Support, you will be asked for the version number, which is shown in this screen.

By clicking the *Version Information* button, you can display a list of changes implemented in your current integration framework version.

By selecting the *Functions* button you can view the scope of functionality available in your current integration framework version for building scenarios. For example, you can see the inbound and outbound channels, atoms, and features available to you for scenario development.

Note that the function list is increased with successive releases of the integration framework.
As of SAP business One release 9.1 PL09 and later releases, the integration framework allows you to easily switch over to a development system profile so you can create and test new scenario packages.

The profile contains the recommended configuration settings for scenario development, testing and debugging. Prior to this release, in order to configure a development environment, you needed to edit and change several settings in the xCellerator configuration file. Now all these settings are gathered together in one place in the system profile.

Switching profiles is done from the Maintenance tab. Choose System Info, then select the ellipsis button to the right of the System Profile field and choose Development System. The system will prompt you to restart the SAP Business One Integration Service for the change to take effect.

Follow the same process to switch back to a productive system.

The settings specific to the development profile are discussed next.
To open the system profiles, choose the icon located to the right of the documentation icon in the System Info screen. You will see both system profiles so you can compare the settings side by side. The active system profile is the one selected in the System Profile field.

While you can edit any of the values in the development system profile, the values are optimized for scenario development:

- The session timeout value is increased for login sessions, to avoid log outs
- WebDAV support is enabled (set to full). This enables a WebDAV client, such as an XML editor, to access the BizStore, allowing you to edit the XSL documents of the transformation atoms while developing and testing scenarios. For security reasons, this parameter is disabled in a productive system.
- Remote access is enabled, allowing you to access the integration framework from a remote computer. Obviously in a productive system this setting is disabled and access is restricted to local for security reasons.

When you first use the development system profile you should enter a development prefix and description for your company. This prefix is added to all scenarios you develop and identifies your content within the integration framework. SAP recommends using a three character prefix.

Note: The number of internal threads is set to default (-1). If you use more than one processor, you should change this parameter to reflect the number of available processors. You should also change this in the productive system profile to take advantage of multiple processors.
The development profile has settings enabled for testing and debugging of new scenarios:

- The option to record test messages at runtime for later use. When this is enabled, and you trigger a scenario step for testing, the input XML message will be automatically saved in the scenario step folder of the BizStore. You can later re-use this message for further testing or debugging without the need to connect to sender systems.

- The option to activate scenarios with detailed logging. To use this feature, you must first enable the capture of detailed logging in the transaction log by setting the xCellerator log level in the development system profile. Then, when a scenario is activated the detailed information will be automatically shown when you debug a scenario step.

- The option to allow modification of an active step. This is useful for testing new scenarios, as you can edit and change an activated scenario without having to deactivate it first.

- The embedded XML editor. This XML editor is provided by SAP and does not need a license. During scenario development and testing, if you enable this, you can open and edit XML files directly in transformation atoms and for format control documents.
The development profile has settings enabled for the message log:

- The level of information written to the message log is **Full message**, meaning that logged messages include the XML message details. For a productive system this is an overhead.
- The B1 event monitor is enabled for a development system, and shows events originating in SAP Business One.
- The checkbox is enabled to enforce message log entries for simplified HTTP and Web service calls, even if the settings have been disabled in the call.
- The final message only checkbox is unchecked allowing the logged message to be updated during processing. In a productive system only the final message is logged, for performance reasons.

Note that many of these message log settings can also be set from the **Maintenance > Cfg MsgLog** screen.
The development system profile has settings to enable the capture of detailed transaction information for debugging:

- The xCellerator manages all processes and adapter connections in the integration framework, and the log level parameters specify how much information is captured during processing. The CONFIG level is the most detailed level. For a productive environment the log level should be set to SEVERE to minimize logging overhead.

- The scenario debug option enables the display of debugging arrows in the atom processing flow as you test and debug scenario steps. Note that you can also control this setting from the Process Control section of the Monitoring tab.

For more information on detailed debugging, see the companion course “Introduction to Scenario Monitoring and Debugging.”
To avoid large log files, you should set up automatic deletion of the message logs. You do this from the Cfg MsgLog option of the Maintenance tab:

- **Daily Deletion**: Enable automatic daily deletion of message logs
- **Indexed Access**: improves performance of accessing message logs (default)
- **Blocksize**: Number of messages displayed
- **User Interface Refresh Timeout**: controls how long time filters are displayed in message log view

To avoid large log files, you should set up automatic deletion of the message logs. You do this from the Cfg MsgLog option of the Maintenance tab:

- Select the *Daily Deletion* checkbox so that the integration framework automatically deletes message log entries on a regular basis.
- *Indexed Access* improves the performance of accessing message logs. This is the default setting. If you switch on indexed access, you will see additional options for viewing the message log information in the message log monitor.
- **Blocksize**. Here you can enter the number of messages displayed in the message log list on one page. The default block size is 50.
- **User Interface Refresh Timeout** value. This value controls for how long the message log screen remembers the timestamps which are used as filters in the Message Log view. The default is 10 minutes. If you enter 0, when you view the Message Log from the path Monitoring > Message Log, the user interface opens with the default time filters to display the whole day.

Note: The Message Log checkbox will appear checked here, based on the development system profile.
If you enable the *Daily Deletion* checkbox, you should set parameters for the daily deletion run by selecting the *Daily Deletion* button:

- **Deletion Time**: hour
- **Backup Buffer**: number of days to exclude from deletion
- **Delete Imported Messages**: imported messages included in daily deletion

If you check the *Delete Imported Messages* checkbox, imported messages will be included in the daily deletion run. An example of importing messages is if you are supporting a customer, and need more details about a certain situation. You can ask the customer to export message log information from the message log monitor and send it to you. You can then import this information into your message log monitor for analysis, as if it had been produced in your local integration framework. The imported log contains all detailed debug information if enabled in the customer system.

To save these settings, click the *Save* button.

Note: The *Last Deletion* field shows the timestamp of the last automatic deletion.

You also have the option to instantly delete log information one time by selecting the *Manual Deletion* button to the left of the Daily Deletion button.
As well as the system profiles, you can specify error handling options for your development environment.
For developing new scenarios, it is helpful to know how the integration framework performs error handling. Choose the path **Maintenance > Cfg Error Handling** to view the default actions for error handling. In all cases a failure message is written to the message log.

For synchronous transactions, the call fails with a fault and a notification goes to the sender system.

For asynchronous transactions, the error handling depends on the phase. The default actions are shown in the slide. For the processing phase, you have an option to either retry after one minute and stop processing (as shown in the slide) or to remove from queue and continue processing.

In addition to the default error handling shown here, the integration framework offers the options to define specific error handling steps on scenario package level. To define the scenario-specific error handling, choose **Scenarios -> Package Design**, click the **Definitions** button and select the **Error Handling** option.
The *Overall Actions* button allows you to configure a notification e-mail to be sent if a message blocks an internal queue for a long time. The notification is important since all other messages in the queue behind this message cannot be processed and they stay in the queue. To do this:

- Choose the *Overall Actions* button and enable the *QBlocker Alert* checkbox.
- Then define how many times you want to receive a notification e-mail. If you enter 1, 10, for example, you receive an e-mail when a message blocks the queue for the first time, and after the tenth attempt of the integration framework to process the message again.
- Save your settings.
- Then, choose the *Configure* button, to provide the details for the notification e-mail, such as sender and receiver e-mail address, and e-mail subject. You can choose to attach the erroneous integration framework message to the e-mail.
The **Outbound Error Actions** button allows you to define specific error actions that the integration framework performs if errors occur during the outbound phase. These error actions can apply at the scenario package, scenario step, or outbound channel level. To define outbound error actions, choose the **Outbound Error Actions** button then choose the **Add Error Action (+)** icon.
To define error actions from the Error Actions window:
1. Select the *Error Action Type* from the list to specify whether the error action applies to the scenario package, scenario step or outbound channel.
2. Select the *Error Action Event* from the list. The list shown depends on what you selected as the error action type. In this example, the SAP business One event is selected.
3. Click the *Select Actions* button then select applicable actions to respond to the error action type and event.
4. Choose the *Activate* button to activate the action.

The actions you can choose from are:
- **Keep in Inbound Queue:** The integration framework will keep the message in the inbound queue for outbound processing.
- **Automatic Retrial:** The integration framework will try to process the message again. You can provide timer settings in the *Configure* function of the *Error Actions* user interface.
- **Send E-Mail:** You can provide e-mail settings in the *Configure* function of the *Error Actions* user interface.
- **Don’t Send to Error Inbox:** The integration framework will not send the message to the error inbox. Together with the *Keep in Inbound Queue* option, you keep the message in the integration framework and define an individual error handling.
- **Wait and Retry:** The integration framework will wait for a defined period of time and try to process the message again for a defined number of times. Only after this time, the integration framework performs the error handling and all other actions you have selected.
As well as the configuration settings, there are various maintenance settings for the development environment.
If you expect your development system to run under very high load and to process a huge amount of messages, you can assign more random-access memory (RAM) to the integration framework server to improve performance.

You assign the memory at the Tomcat level.

To assign memory:

- Open the file `tomcat6w.exe` from the Tomcat folder. If the system denies access, select `tomcat6w.exe`, open the context menu and select the *Run as Administrator* option.
- Select the Java tab, and increase the maximum memory pool amount as follows:
  - In a 32-bit operating system, the maximum memory pool is 1024 MB.
  - In a 64-bit operating system, there is no maximum memory pool defined by Tomcat. The default is 2048 MB but you can increase it according to the server memory.

For more information on Tomcat memory, see the Operations chapter in the integration framework online help.
Safe Mode Configuration
xcellerator.cfg File

- You can start the integration framework in safe mode by changing the parameter in the configuration file:
  Path: ..\SAP\SAP Business One\SAP Business One Integration\Integration Server\Tomcat\webapps\B1iXcellerator\xcellerator.cfg

Safe Mode Use cases
- Locked administrator user – see SAP Note # 1556041
- An error-prone process blocks the integration framework, even after restart
- Integration framework will not start due to application deployment problems

Activation
- Set xcl.safemode=true
- Restart the integration service

Safe Mode Limitations
- All adapters are disabled
- User authentication is disabled
- Access is only possible from the local machine.

Similar to operating systems, you can start the integration framework in safe mode. Safe mode is not required to use the integration framework for development, but you should consider use safe mode in the following cases:

- You have locked the administrator user. This is probably the most common use case. For more information, see the SAP Note referenced in the slide. In safe mode the relevant administration tools still work, and you can unlock or reset the password for an administrator account.
- An error-prone process blocks the integration framework, even after a restart.
- The integration framework will not start due to application deployment problems.

To activate safe mode:
1. Edit the xcellerator.cfg file using the path shown in this slide, and change the value of xcl.safemode from 'false' to 'true'
2. Restart the SAP Business One Integration service then verify that the integration framework starts in safe mode.

In safe mode the integration framework adapters are disabled, user authentication is disabled, and access is possible only from the local machine. You should not run scenario packages in safe mode.

To reverse safe mode, edit the xcellerator.cfg file again and change the safemode parameter to false. Then restart the SAP Business One Integration service.
The integration framework provides system profiles for a development and productive environment. These profiles contain the recommended configuration settings for each environment.

After installation, the integration framework is set to a productive environment. To develop and test scenarios, you need to switch to the development system profile.

Development profile is optimized for scenario testing and debugging, including:
- WebDAV client access
- Remote access
- Modification of active scenario step information
- Full message logging to enable scenario monitoring
- Detailed transaction logging and scenario debugging

You can optionally set email notifications for blocked queues and select actions for outbound channel errors.

You have the possibility to expand Tomcat memory and start the integration framework in safe mode.

- The integration framework provides system profiles for a development and productive environment. These profiles contain the recommended configuration settings for each environment.
- After installation, the integration framework is set to a productive environment. To develop and test scenarios, you need to switch to the development system profile.
- The development profile is optimized for scenario testing and debugging:
  - WebDAV client access
  - Remote access
  - Modification of active scenario step information
  - Full message logging to enable scenario monitoring
  - Detailed transaction logging and scenario debugging enabled
- In addition to the development profile, you can optionally set email notifications for blocked queues and select actions for outbound channel errors
- You have the possibility to expand Tomcat memory and start the integration framework in safe mode.
Thank You!