Copyright © 2007 Infor Global Technology Solutions GmbH and/or its affiliates and subsidiaries

All rights reserved. The word and design marks set forth herein are trademarks and/or registered trademarks of Infor Global Solutions Technology GmbH and/or its affiliates and subsidiaries. All rights reserved. All other trademarks listed herein are the property of their respective owners.

Important Notices

The material contained in this publication (including any supplementary information) constitutes and contains confidential and proprietary information of Infor Global Solutions.

By gaining access to the attached, you acknowledge and agree that the material (including any modification, translation or adaptation of the material) and all copyright, trade secrets and all other right, title and interest therein, are the sole property of Infor Global Solutions and that you shall not gain right, title or interest in the material (including any modification, translation or adaptation of the material) by virtue of your review thereof other than the non-exclusive right to use the material solely in connection with and the furtherance of your license and use of software made available to your company from Infor Global Solutions pursuant to a separate agreement (“Purpose”).

In addition, by accessing the enclosed material, you acknowledge and agree that you are required to maintain such material in strict confidence and that your use of such material is limited to the Purpose described above.

Although Infor Global Solutions has taken due care to ensure that the material included in this publication is accurate and complete, Infor Global Solutions cannot warrant that the information contained in this publication is complete, does not contain typographical or other errors, or will meet your specific requirements. As such, Infor Global Solutions does not assume and hereby disclaims all liability, consequential or otherwise, for any loss or damage to any person or entity which is caused by or relates to errors or omissions in this publication (including any supplementary information), whether such errors or omissions result from negligence, accident or any other cause.

Trademark Acknowledgments

Microsoft, and Windows are registered trademarks of Microsoft Corporation in the United States and/or other countries.

Java is a trademark or registered trademark of Sun Microsystems, Inc. in the United States and/or other countries.

i-Flow is a trademark of Fujitsu Software Corporation in the United States and/or other countries.

All other company, product, trade or service names referenced may be registered trademarks or trademarks of their respective owners.

Publication Information

Document code : U9302A US
Release : Infor ERP LN Workflow 7.4
Publication date : August 07
Table of Contents

Chapter 1  Introduction.................................................................................................................1-1  
Start Menu Choices ....................................................................................................................1-1  
Infor Workflow Client Menu Choices ..........................................................................................1-1  
Infor Workflow Server Menu Choices.........................................................................................1-3  

Chapter 2  To model with Infor Workflow 7.4 .........................................................................2-1  
Creating a Base Workflow ...........................................................................................................2-1  
To modify an existing template .....................................................................................................2-5  
To modify template properties ......................................................................................................2-7  
To modify Infor Activity Node properties ....................................................................................2-13  
To model an activity for FTP tasks ...............................................................................................2-31  
To model using the Infor Web Service Node ..............................................................................2-39  
   To use the Infor Web Service Node ...........................................................................................2-40  
   Modeling with a web service node ............................................................................................2-42  
To model using the Infor Action Type: Infor ERP LN.................................................................2-60  
   To select the best Infor ERP LN session ..................................................................................2-61  
   To model the session ................................................................................................................2-62  
   Concept of attributes ................................................................................................................2-68  
   Concept of sub-session authorization ......................................................................................2-74  
   To run the Infor ERP LN session .............................................................................................2-76  
   Workflow limitations ................................................................................................................2-77  
To model using the Infor Action Type: Call Program .................................................................2-83
Table of Contents

To work with attachments ........................................................................................................... 2-85
  Add attachment ...................................................................................................................... 2-85
  Delete attachment .................................................................................................................. 2-87
  View Attachment .................................................................................................................... 2-88
  Edit attachment ...................................................................................................................... 2-88
  Attachments administration ................................................................................................. 2-89

Chapter 3  Workflow Triggering .............................................................................................. 3-1
  Overview ................................................................................................................................. 3-1
  Workflow Triggering Configuration ....................................................................................... 3-2
  Workflow triggering runtime ................................................................................................. 3-12
  Sample code to launch a trigger from an Infor ERP LN program script ............................. 3-13

Chapter 4  Administration ...................................................................................................... 4-1
  Workflow Configurator .......................................................................................................... 4-1
    To open the Configurator ..................................................................................................... 4-1
    Data source definition maintenance ................................................................................... 4-5
    To configure the XML File Listener .................................................................................... 4-11
  Auditing the Process Instance .............................................................................................. 4-14
    Accessing the Audit Functionality ...................................................................................... 4-15
  Client configuration ............................................................................................................... 4-24
    Infor WebTop 8.4 for Infor ERP LN ..................................................................................... 4-24
    Infor Worktop 2.4.15 ............................................................................................................. 4-24

Appendix A  Workflow-controlled Infor ERP LN session ...................................................... 4-1
  To change the Exit command ................................................................................................. 4-2
  To change the Kill command ................................................................................................. 4-2
  Change the Abort command ................................................................................................. 4-3
  Change the Execute commands places ............................................................................... 4-3
This document describes the procedure to use Infor Workflow 7.4. This product consists of two major parts: the administration of Workflows, and the use by the client users.

This document consists of the following chapters:

Chapter 1, “Introduction”, provides a brief overview of the WorkFlow7.4 for Infor ERP LN and describes the Start menu options.

Chapter 2, “To model with Infor Workflow 7.4,” describes the procedure to use the Infor Workflow product to create a base workflow model and to use the specific Infor ERP LN Workflow tools to model more-complex workflow models.

Chapter 3, “Workflow triggering,” describes the process to configure workflow triggering specific to Infor ERP LN and the Infor Workflow for ERP LN.

Chapter 4, “Administration,” describes administrative tasks such as the procedure to configure the workflow clients, and configure data source tasks.

Appendix A, “Workflow-controlled Infor ERP LN session”, explains the modifications made in the Infor ERP LN session to enable them to work with Infor WorkFlow 7.4

Infor Workflow utilizes Fujitsu’s Interstage Business Product Manager (iBPM) as its core engine. In addition to this document, refer to the following guides for information on the administration and use of Interstage BPM. These documents are available as .pdf files in the location: \<<install directory>>\docs

Infor ERP LN Workflow utilizes two workflow client applications: Infor WebTop for Infor ERP LN and Infor ERP LN Worktop. Refer to the following documentation for the installation, administration, and use of these products:

- **Infor Webtop 8. 4 for ERP LN - Installation and Configuration Guide** (U8715D US)
- **Infor Webtop – Online Help**
- **Installation Guide Worktop 2.4 for Infor ERP LN** (U8476C US)
- **Administrator’s Guide Worktop 2.4 for Infor ERP LN** (U8478F US)
- **Configuration Guide Worktop 2.4 for Infor ERP LN** (U8477C US)
- **SSA® Worktop – Online Help**

**Send us your comments**

We continually review and improve our documentation. Any remarks/requests for information concerning this document or topic are appreciated. Please e-mail your comments to mailto: documentation@infor.com.

In your e-mail, refer to the document number and title. More specific information will enable us to process feedback efficiently.
This chapter provides an overview of the features available on the workflow server.

Start Menu Choices

The Windows Start → Programs → Infor Workflow 7.4 menu displays applications and utilities of the Infor Workflow. The applications and utilities provided are broadly categorized into Client and Server.

Infor Workflow Client Menu Choices

Click Infor Workflow 7.4 → Client, the following choices are displayed:
1 **Infor Worklist**

Worklist enables the user to execute the work items using the Infor Worklist client.

You can use the browser to execute the workitems that are not modeled using the Infor Action. As a result, the user without an LN authentication can execute the workitems when ERP LN sessions are not involved.

User can also access the above functionality using the following URL:

http://<<host name>>:<<port>>/ibpm/jsp/Worklist.htm

2 **Infor Workflow 7.4 client**

This client is used by an end user as well as an administrator. This is the default client provided by the iBPM Workflow server. You require this client to create process definitions and administration tasks. Features such as Configurator, Auditing and Infor Worklist are included.

- **BPM Console**
  Use the BPM Console to create and maintain WorkFlow templates. You must have necessary user access privileges to log on to the console.

- **Administration**
  You can perform tasks such as maintaining templates which includes publishing a template and viewing the template instance history. You require administrator privileges to log on.

**Note:** To add a user with administrator privileges, refer to Chapter 3, section 3.5: User Accounts and Roles of the document: *Interstage Business Process Manager (iBPM) Installation Guide for the Advanced Edition (AEInstallation_Guide.pdf)*

- **Reports**
  Run workflow reports:

  This feature provides the following four functionalities:

  I. Run report templates.

  II. View report documents and save the documents to your computer.
III. Delete report documents created on the server.

IV. Make your own custom queries to the workflow database.

Infor Workflow Server Menu Choices

Click **Infor Workflow 7.4 Server**, the following choices are displayed:

- **Start Workflow Server**
  Starts/Restarts Infor Workflow services.

- **Stop Workflow Server**
  Stops the Infor Workflow services.

You can utilize the Stop and Start Services to:
  - Clear the connection problems with the servers.
  - Update the configuration after modifying configuration files
  - Backup the database

- **Uninstall Infor Workflow 7.4**
  Uninstalls the Infor workflow.

**Worktop \ WebTop Client**

These clients are used to execute activity nodes of **Infor ERP LN** and other nodes of workflow.

You can find more information on Worktop \ Webtop later in this document.
This chapter describes the process to create a base workflow.

Creating a Base Workflow

To create or modify a workflow, you must have an Infor Workflow account and sign on. Access the login page using the following URL:

http://<<host name>>:<<port>>/ibpm/

If you work on the machine where the workflow server is installed, you can access the page from Start menu:

Click Start -> Program Files -> Infor WorkFlow 7.4 -> Client -> Infor WorkFlow 7.4 Client

Any of these actions opens the following page:
To create a base workflow template, follow the steps:

1. In the Infor Workflow Clients page (shown in figure) click **BPM Console**.
2 The Infor Workflow Log in screen appears.
3 Enter your user name and password, and click **Login**. The workflow **worklist** process definition screen appears.
4 Click the **Process Definitions** tab. In the resulting screen, you can create or edit the process definitions. You can also view the list of process definitions created.

5 **Click Create New**, the following page opens:

6 Press SPACEBAR or ENTER to activate and use the toolbar.

7 Select a node from the tool bar and click in the work area below.

**To modify an existing template**

1 Click the edit icon that is next to the desired template as indicated in the following screen:
A graphical representation of the template is shown for editing.
To modify template properties

1. Click on the **Properties** button on the toolbar. A dialog with following tabs appears:
   - General
   - Process Owners
   - User-defined Attributes
   - Script
   - Timers
   - Action Set
   - Triggers
2 On the **General** tab, you can name the template, and enter a descriptive text that describes the purpose of the template.
3 The Process Instance Owners tab enables you to define the group of users who can own the process. The owner of a process has the rights to maintain the workflow during the normal processing cycle.

Note: The Process Instance Owners tab enables you to display templates in the workflow clients. In case processes must be started from templates by a workflow client user, this user must participate in the list of template owners.

4 The User-defined Attributes tab enables you to define variable names and their characteristics. You can use these variables during subsequent processing steps.
On the **Script** tab, you can write custom Java script actions to perform specific processing required for your organization.

**Note:** For detailed information, refer to the documentation referenced in the “About This Guide” section of this guide.
6 The **Timers** tab enables you to specify when specific actions are to be performed. The type of timer can be based on Calendar date, or Time milestones.

In addition, at this point, you can select the action or actions:

![](image1.png)

7 On the **Action Set** tab, you can define specific actions that must occur when the workflow is triggered. This procedure is described in more detail later in this chapter.

![](image2.png)
On the **Triggers** tab, you can define if and when the workflow starts. This feature is used only when you want the Process template to be instantiated depending on an event that occurred outside the workflow application. This event can be an event on a database table or arrival of an xml file in a particular workflow directory.

These templates can also be manually started by a user with suitable permissions like a Process Template without any trigger defined on it.

On the **Triggers ➔ General** tab, you can use **Enabled** check box to turn on or off the current workflow. Ensure that all the properties are properly set before you select the **Enabled** check box. In case properties are missed or not set properly, it may lead to unpredictable behavior at run time.

On this tab, you can define the conditions for a workflow to trigger.

The **Triggers ➔ Event** tab enables selection of the XML Event Type.

**Note:** For detailed information about triggers with Infor ERP LN, refer to Chapter 3, “Workflow triggering.”
To modify Infor Activity Node properties

The toolbox button 🔄 is an Infor Activity Node. When you select this element, it can be used to create Infor Activity Nodes as model elements.

1. Right-click the Infor activity node to edit. The Infor Activity Node Properties dialog box appears:
On the **General** tab, you can define the name that appears in the lower portion of the activity node. The name field is limited to 30 characters.

The **Commit Transaction** is used to ensure that the workflow data (UDA variables etc) that is modified in a Java Action is committed to the database before the next Java Action occurs. If this check box is cleared, all the Java Actions in a set are committed at once. If an exception occurs in one java action, effects of all the java actions will be rolled back.

The **Commit Transaction** is applicable in the context of Java Actions. Refer to the chapter 6, section 6.2.1: Assigning Java Actions of document *Interstage Business Process Manager (BPM) User’s Guide.pdf*.

2 Optionally, you can provide additional descriptive information about the activity node.
3 On the **Assignee** tab, you can define the role that can act on the node’s activity.

For example, you can assign an action to a role of Role. If no further individual is defined, all Role participants can act upon the activity.

Only the participant in Role that accepts the activity first can act on the activity.

The users can also be assigned using a script written for assigning users to activities. The script is:
setActivityAssignees("wfuserSales,wfuserAccounts");

This sample scripts adds the two users *wfuserSales,wfuserAccounts* as assignees to an activity. For more details refer to the Chapter 5, section 5.4.6: Assign Activities of the document *Interstage Business Process Manager (BPM) User’s Guide.pdf*
4. On the **User-defined Attributes** tab, you can enter the variable name, the data item type, and additional option attributes. You can define the same **User-defined Attributes** at the Template level, as described in “To modify template properties,” previously in this chapter, or at the Activity Node level.

**Note:** User defined attributes once defined, can be used throughout the process template. These are global to the Process Template.

**Note:** Do not use special characters, such as #, $, and so on.
5 On the **Script** tab, you can define Java script to perform any type of function or activity.

6 On the **Infor Action** tab, you can define the Infor extensions that appear if you perform the action in a work list.

   The **Action Types** that you can select are **Infor ERP LN** or **Call Program**. These actions are described in more detail in "To model using the Infor
To model Infor ERP LN" and "To model using the Infor Action Type: Call Program," later in this chapter.

On the Due Date tab, you can set a specific time and date to a task to specify when that particular task is due to be completed.

**Note:** For detailed information, refer to the documentation referenced in the "About this guide" section of this document. Click the Help button for more information about the dialog.
On the **Timers** tab, you can define a specific period of time to elapse before a subsequent action occurs.

**Note:** For detailed information, refer to the documentation referenced in the “About this guide” section of this document.

---

On the **Action Set** tab, you can define specific actions to occur for a role or prologue and epilogue actions. The java actions that are available here are the standard java actions provided by the application. Custom java actions can be built by the user and deployed in the application to be made available here. For more details about the standard java actions refer to the chapter 6, sections 6.2 and 6.3 of *Interstage Business Process Manager (iBPM) Developer’s Guide.pdf*

10 Click on the Add button in any of the Actions. Select **Assign Task to User** from the **Action List Types**
The **Assign Task to User** dialog box appears:

When clicked, this button toggles between **V** (variable), **C** (constant), and **E** (expression).
11 Click the **Assignment Selection** tab.

12 Select the **Use Workflow Variable** check box.

In this case, you assign an activity to a \( V \) variable. In this case, you must have defined the variable as a user defined attribute previously. Select the variable from the list box.

13 Click **OK**. The variable is now selected and the window enables you to select the users.

However, if you clear the **Use Workflow Variable** check box, you can select a specific user to be assigned to the task.
On the Details tab, you can name the action and add descriptive notes.
15 On the **Triggers** tab, you can define additional Infor Activity conditions that enable you to move a workflow along.

This example has no additional trigger. However, this works in the same way that the trigger works to start a workflow.

You can use a timer to remind you to perform an action. A timer is created to model a **Delay Node**.
In this example, a five-second delay precedes the action defined in the action set.
16 In the **Action** group box, you can choose to send an e-mail message or evaluate a script to determine the actions that will be carried out.

![Delay Node Properties Dialog](image1.png)

17 On the **Action Set** part of the properties dialog, you can define the description of the action set. Click **Edit** to specify the actions to be performed.

**Note:** Click **Add** to create a new action, and click **Edit** to modify the existing action.

![Action editor - Set Process Instance Name](image2.png)

This example shows that an expression is defined.
In this example, after the five-second wait period, the task reappears in the work list and the process name changes to **Credit Limit Reminder**, followed by the customer number.

18 To edit the process name expression, click **A+B**.

The **Expression Builder** dialog box appears:

The expression builder works in the same way as shown previously. However, in this case, you have the drop-down list box from which to select variables.

19 In this case, the **Current Stock Level** user-defined attribute is selected. Next, click **Add**.
This example shows how you can define the process name. The C button enables you to define a constant value for the process name.

If the C button appears, you can enter the process instance name in the corresponding box.

This example shows how you can define the process name. The V button enables you to define a variable value for the process name. Select the variable name from the list box.

The Absolute check box enables you to define a specific time when the action set will be carried out.
Notice the appearance of the screen with the Calendar-Periodic check box selected. In this example, if you select the Periodic check box, the action will be carried out every five seconds.
If you select the **Business** option without selecting the **Periodic** check box, the action set is performed after the specified elapsed time. **Business** uses a different calendar than the normal **Calendar**.

**Note:** For detailed information on business calendars, refer to the *Interstage Business Process Manager (iBPM) Administration Guide* and to the *Interstage Business Process Manager (iBPM) User's Guide*.
Notice Business Periodic performs the Action Set every time period defined.

The Advanced option enables you to define your own expression.

To model an activity for FTP tasks

This section describes the procedure to use an activity node to transfer files from the Workflow server to a remote FTP location. You can use UDA variables to specify the configuration required for the FTP connection and the file on workflow server and the remote location. These values can change at run-time, as a result a dynamic FTP functionality is provided. You can now transfer different files to different remote FTP locations in each activity of a process instance.
Before you use the FTP action agent, ensure that all the UDA variables required for FTP action are defined for the process.

The UDA variables required for the FTP action to work are described in the following table:

<table>
<thead>
<tr>
<th>UDA Name</th>
<th>Type</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTP_SERVER_NAME</td>
<td>String</td>
<td>Optional.</td>
<td>Name of the remote ftp server, where the destination location exists.</td>
</tr>
<tr>
<td>FTP_PORT</td>
<td>Integer</td>
<td>Optional.</td>
<td>The port on which the Remote FTP server is running. If value is not specified, the default port 21 is used.</td>
</tr>
<tr>
<td>FTP_USER</td>
<td>String</td>
<td>Optional.</td>
<td>FTP user name for the remote FTP server</td>
</tr>
<tr>
<td>FTP_PASSWORD</td>
<td>String</td>
<td>Optional.</td>
<td>FTP password for the remote FTP server</td>
</tr>
<tr>
<td>FILE_TRANSFER_MODE</td>
<td>Boolean</td>
<td>Optional</td>
<td>TRUE - ASCII mode</td>
</tr>
<tr>
<td>UDA Name</td>
<td>Type</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>FTP_APPEND</td>
<td>Boolean</td>
<td>Optional</td>
<td>TRUE - append to an existing file, if it exists on the FTP server</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FALSE - do not append to the existing file if already present on the FTP server (Overwrites the existing file)</td>
</tr>
<tr>
<td>REMOTE_FTP_DIRECTORY</td>
<td>String</td>
<td>Optional</td>
<td>Remote FTP directory where the files will be placed</td>
</tr>
<tr>
<td>REMOTE_FTP_FILES</td>
<td>String</td>
<td>Optional</td>
<td>The names with which the received files will be saved. There are three cases:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Empty: If this field is empty, the files are saved with the same names as the source files</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Single File name: The received file is saved with this name. If more than one file is received, the last received file is saved with the specified file name.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>List of file name: Multiple, file names can be specified as list separated by semi-</td>
</tr>
<tr>
<td>UDA Name</td>
<td>Type</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>colon</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OUTGOING_FILES</td>
<td>String</td>
<td>Optional</td>
<td>The files to be transferred to the remote FTP location. Can be specified in three modes:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- File: Complete path of the file including the file name</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- File list: List of complete file paths, including file name separated by semi-colon.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Directory: Specify the file path till the directory to transfer all the files in that directory.</td>
</tr>
<tr>
<td>ERROR_MESSAGE</td>
<td>String</td>
<td>Filled at run-time by the FTP Agent</td>
<td>If there is an error during the FTP process, the error message is assigned to this variable</td>
</tr>
<tr>
<td>AgentServiceStatus</td>
<td>String</td>
<td>Assigned at runtime by the FTP Agent</td>
<td>This is the status of the FTP agent service. Following are the possible values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Failed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Success</td>
</tr>
<tr>
<td>AgentServiceResult</td>
<td>String</td>
<td>Assigned at runtime by the FTPAgent</td>
<td>The Result of the FTPAgent action</td>
</tr>
</tbody>
</table>


The following UDA variable values must be present for the FTP action to be performed successfully.

- FTP_SERVER_NAME
- FTP_USER
- FTP_PASSWORD
- FTPPort
- FTPDirName

The above UDA’s values can be defaulted at by configuring from the ibpm.properties file located at <<install root>>\bin.

- FTPServerName=<FTP SYSTEM NAME>
- FTPUserName=<USER ID>
- FTPPassword=<PASSWORD>
- FTPPort=21
- FTPDirName=<FOLDER NAME>

If values for FTP_SERVER, FTP_USER_NAME, and FTP_PASSWORD are not provided at the execution time, the FTP fails and displays an error message.

- All the remaining variables will be given default values or will be taken as null during execution.

The following steps describe the process to create an activity for FTP

1. Add an activity node to the work area.
2 In the **Activity Node Properties** dialog, go to the **Assignee** tab

3 In the **Role** field, specify the Role name as @INFORFTPAGENT
You can also select the available roles using the **Browse** button. A list of available roles is displayed when you click the **Browse** button.

Choose the role **@INFORFTPAGENT** and click **OK**.

4. You must have two **Actions (out going arrows)** for every activity node assigned to an INFORFTPAGENT, for the process to go forward and complete.
One arrow represents a success path (Files transferred successfully). This arrow/action must be named **DONE**.

The other represents a failure path (Files not transferred due to some error). This arrow/action must be named **Failed**.

The FTP agent configuration information is present in the file – **agentsConfig.xml** at location `d:\InforWorkflow\agents`

Sample content of the file is:

```
<ActionAgent>
  <Name>@INFORFTPAGENT</Name>
  <Description>Invoke the FTP Service</Description>
  <RetryInterval>20</RetryInterval>
  <EscalationInterval>2</EscalationInterval>
</ActionAgent>
```
To model using the Infor Web Service Node

Web Services Node in the context of the workflow or process engine is used to invoke web service as part of the workflow activity.

Infor Web Service Node has the following features:

5 The ability to model web service node with the help of WSDL saved on workflow server, even when the web service is not available.

6 User does not require **XPath** knowledge to design the web service node.

7 Easier and fool-proof way to map UDA values to request and response of web service node.
8. It has the features of activity and agent of workflow. User intervention is not required to execute this task on completion of task, it automatically selects the choice, **Done** or **Failed**.

9. Provides better error handling capabilities, that is, a way to capture error messages from web services and ability to define the course of action based on the error message. (Some examples of error messages are **service not available, connection lost**).

Technical errors such as **time out, unable to contact the server** are handled with number of retries after a specified duration.

### To use the Infor Web Service Node

#### Pre-requisites to use Infor Web Service node:

On client machine where the browser is opened for designing the process definition, complete the following steps:

1. Go to **Start -> Settings -> Control Panel**

2. Open the **Java Plug-In** dialog shown in above figure.

3. Click on the **Advanced** tab.
4 Enter the Java Runtime Parameter

-Djava.security.policy=%WorkFlow Root%\bin\java.policy

5 Create a process definition with UDAs, to provide the input and to store the retrieved output from the web service.
Modeling with a web service node

1. Select the **Infor Web Service Node** icon from the toolbar.

2. Right click the **Infor Web Service Node** to open the **Properties** dialog box.
General Tab

You can specify the name of the web service node and the description.
Service Definition Tab

This tab contains the following sub-tabs:

- **WSDL**

  Use this tab to load the WSDL file corresponding to a web service.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSDL Location</td>
<td>The location of the WSDL file. User can enter the location.</td>
</tr>
<tr>
<td>Local WSDL URL</td>
<td>Location of the downloaded WSDL file on the workflow server.</td>
</tr>
<tr>
<td>Notes</td>
<td>Description of the WSDL file</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>Load</td>
<td>Downloads the WSDL file from the location specified in the <strong>Local WSDL URL</strong> field to the workflow server.</td>
</tr>
<tr>
<td>Browse</td>
<td>Enables the user to locate and browse the web service on a UDDI repository.</td>
</tr>
</tbody>
</table>
Operations tab

Operations tab displays operations provided by the web service represented by the WSDL.

To select the operation, complete the following operations:

- Click and expand the tree.
- Select the required operation from the list.
- **Input Tab**

Use this tab to map the UDA variables of the process definition to the Web service operation’s input parameters. When you click on the Input tab, you see the name of the web service and the web service operation name displayed.

The Input parameters of the selected web service method are represented in a tabular format. The table has four columns:

- **ParamName**: The Input parameters for the selected web service Method. The input parameters are displayed as a tree. To expand the tree, you must
double-click on each parent node or click the ‘+’ symbol adjacent to the parent node of the input parameters tree.

Double-click on **Request** under the **ParamName** Column to expand the tree further.

If the input parameter expects array elements to be passed, you can create new array elements by selecting the element in the ParamName column and then clicking on the **Add Array Element Button**. The element to be selected is the element that has data type as an array and also has the array index symbol ‘[]’ in its ParamName column.
After the addition of new array element

To remove the array element, select the corresponding element and click **Remove Array Element**.
After removing the array element:

The user can map discrete input parameters to UDA variables which are explained.

The user can also map a UDA to a parent node in a complex type input parameter.
For example in the above figure, **Address** is a complex type of input parameter. The **POline** is a composite of several other elements. The user can map a variable directly to the **POline** element. The input value for the parameter is the **xml** which represents structure of the **POline** object.

The value for the UDA **Address** is in xml format. The XML format is obtained by clicking the **XML** button shown in the below figure.
Clicking the XML button gives the complete SOAP request in XML format. From this complete XML, copy the part that corresponds to the required input parameter. For example, the required part for the Address parameter is shown below:

```xml
<obj1:buyFromAddress>
    <obj1:addressCode>XmlString{{Field }} </obj1:addressCode>
    <obj1:country>XmlString{{Field }} </obj1:country>
    <obj1:line1>XmlString{{Field }} </obj1:line1>
    <obj1:line2>XmlString{{Field }} </obj1:line2>
    <obj1:line3>XmlString{{Field }} </obj1:line3>
    <obj1:line4>XmlString{{Field }} </obj1:line4>
    <obj1:state>XmlString{{Field }} </obj1:state>
    <obj1:zipCode>XmlInt{{Field }} </obj1:zipCode>
</obj1:buyFromAddress>
```

In the above XML, the content shown in italics, that is, the value between the opening and closing tags should be replaced by the actual values to be sent at runtime as input.
**Data Type:** The data type of the input parameter sent to the web service operation. This is also the data type acceptable for mapping with the UDA.

**Select UDA:** Select the UDA whose value is assigned to the parameter specified in the **ParamName** column at runtime. (Or) Specify the constant value that will do the same. The UDAs that are created appear in the drop down list. The appropriate one can be selected.

Alternately, the user can also input a constant (literal) as the value for the input parameter.
Test item: You can specify the test values for the input parameter in this column.

- **Output tab**

Maps the output of the selected operation of the web service to the UDA. A complex output parameter can be mapped in the XML format with the UDA.

This table contains four columns:

**ParamName**: The output parameters of the selected web service operation are displayed here. Double-click on **Response** under the **ParamName** Column to expand the tree further.
Data type: The data type of the parameter received as a result of the web service operation. This is also the data type acceptable for mapping with the Output UDA

Select UDA: for selection of the UDAs, this will get the return values of the Web Service operation at runtime.

Test value: This value is displayed when the Test button is clicked. The Test button picks the Test Item values specified in the Input tab and calls the selected web service operation.
In some cases, the output can be an array which means the web service operation returns an unpredictable number of array elements. In such cases the mapping of discreet individual elements of the output parameter to UDAs becomes impossible. In such cases the user can map the parent array element to a string type UDA. The output is in xml format.

For example consider the following situation:
In the following figure, we see that the operation returns an array of objects (PO objects – Purchase order objects).
The user does not know the number of PO objects that will be returned. Hence, user cannot map the discreet individual parameters to UDA variables. The user can map the parent PO array element to a UDA.

The output in XML format as shown below (the output has been saved and shown in Internet explorer for better viewing):
In the above figure we can see the multiple POline elements.

- **Security Tab**

This is a web-service authentication tab. Specify the **User Name** and **Password** for accessing the web-service, if required.
- **Test tab**

Displays the SOAP response after the **Test** button is clicked.
There should be two outgoing arrows from the Infor Web Service node named **Done** and **Failed**. If the call to the web service is successful, the corresponding activity after **Done** is executed. Otherwise, the other activity is executed.

When a web service node is added in a process definition, the web service node adds some UDA variables automatically to the Process definition.
The UDA’s added are:

<table>
<thead>
<tr>
<th>UDA name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>__WS_ERROR_MESSAGE</td>
<td>It stores the error message if an error occurred when the web service was being executed.</td>
</tr>
<tr>
<td>__WS_STATUS</td>
<td>Stores the status of the web service.</td>
</tr>
</tbody>
</table>

To model using the Infor Action Type: **Infor ERP LN**

During analysis of the template, the decision will be made that a particular activity can best be executed using an **Infor ERP LN** session.

This section describes how to choose the best **Infor ERP LN** session and then how to model that **Infor ERP LN** session using the Infor Action Type.
Bear in mind that modeling activities for workflow is intended to present the user with a session best fit to the task to be carried out. In addition, you want the user to be able to perform tasks in the most efficient way. To ensure that the user can work efficiently, you can limit the options in the session to be used, so that the user is more focused on this task and cannot divert from the task.

To select the best Infor ERP LN session

To use Workflow in a meaningful way, you must link the best Infor ERP LN session to an activity. In other words, the session must exactly represent the action that the activity requires. That can mean that linking a main session to an activity is not useful, because the main session is too general and does not specify the particular action that must be performed. If a subsession is available, model the subsession to avoid ambiguity about the action that is required from the Workflow user.

To select the most suitable session to link to an activity, take the following steps:

1. Categorize the intended actions that must be carried out by means of the activities, for example, an order for sales, purchase, or manufacturing.

2. Find the main session that deals with this topic or category.

   In Infor ERP LN, most sessions that you start from the menu start in so-called multi-occurrence mode. As a result, you see, for example, a list of sales orders and details of that sales order only are shown if you double-click a sales order. This type of detail session is called the Synchronized dialog box of the main session. To use that session in your activity, you must model the main session with start-up type Single Occurrence and send attributes to that session. More information on this topic is provided in the description of the startup type model option.

   Check whether the required action can best be carried out from the main session. If you find a main session that fits, model that session. If you do not find an appropriate main session, go to Step 3.

3. Select the subsession that enables you to carry out the desired action and model the subsession.

   If the subsession contains insufficient information to carry out the task, you must model the main session. The Workflow user must then manually start the subsession from the Special menu.
To model the session

1. Start the iBPM Console and create a template.
2. Click the Infor Action icon to place a node for an Infor ERP LN session in the template.

Right-click the Infor Action and, on the shortcut menu, click Forms. The Infor Activity Node Properties dialog box appears:

On the Forms tab, choose action type Infor ERP LN.

Session Code

You can specify the session codes in the following two ways:

1. Manual method
2. Using the browse option

   Manual method:

   Specify the session code in the Session Code Field. The session code can be directly typed in the text field.

   No check on validity is performed, therefore, be sure to type the correct session code.

   Using Browse option:
1 Click the browse button next to the **Session Code** field; the **Session Codes** panel opens.

2 Select the package in the **Package** field; values in the **Module** field are populated based on the package specified.

3 Select a module in the **Module** field; sessions of the selected package and module are retrieved from the workflow repository.

   **Note:** Some combinations of Package and module can be empty, indicating that there are no sessions. In such cases the Session Codes list is empty.

   **Note:** Sessions repository is generated on ERP LN server using a session ttadv2285m000. For details please refer Chapter 5 in the Infor ERP LN Workflow Installation and Configuration Guide. (U9329A US)

4 Double-click on the row / record **or** select a row / record and click **OK**.

5 Instead, a user can also search for a session by entering the description or part of description of the session in the **Session Description** text field. Following scenarios are possible when you search for a session:
When the **Package** and **Module** fields are empty, the search is performed on all the sessions of the Package – Module combinations.

- When you specify the **Package** only, the search is performed on all the modules of the package.
- When you specify the **Package** and **Module**, search is performed on the sessions of the specified package – module combination.

### Company Number

If you have a multisite setup, you might want specific **Infor ERP LN** sessions always to run in a fixed company. For example, Finance sessions must always run in the Finance company you have defined.

By default, if you leave this field set to 0 (zero), the **Infor ERP LN** sessions start in the start-up company of the user who carries out the activity.

The execution in another company only works if the user is also in the normal menu can switch to that company. Therefore, check user and company authorizations prior to implementation.

### Authorization Type

If you start an **Infor ERP LN** session, by default, options such as **Insert**, **Copy**, and **Delete** are selected. If you determine that particular options are unnecessary in the step in the template, you can restrict the access to those options. If, for example, only existing data must be modified in that step and no new data must be entered or existing data deleted, you must model here the value **Modify/Print/Display**. If, for example, you want a user only to view data in this step, select **Display** in this field.

Within a session some menu choices and/or buttons only appear, when the authorization type is high enough. That means that with Display authorization for example a ‘Process’ button will be disabled or invisible in certain sessions (because for Processing you normally need at least Modify/Print/Display authorization).

By default, the subsessions that you start from a session inherit the authorization type for the modeled session. If you want subsessions to start with other authorization types, you must model that on the **Sub-Sessions** tab.
Start-Up Type

To specify how the selected Infor ERP LN session is presented at run time to the Workflow user, select a value in the Start-Up Type. If the functionality that the activity requires is present in the single-occurrence session, always select Single Occurrence Session.

In Infor ERP LN, most sessions that you start from the menu start in so-called multi-occurrence mode. As a result, you see, for example, a list of sales orders and only after you select and double-click a sales order can you view the details of that sales order. This type of detail session is called the Synchronized dialog box of the main session.

The main session partly controls these synchronized dialog boxes. Therefore, to use these dialog boxes in a workflow context, you must model the dialog boxes with the session code of the main session and start-up type Single Occurrence.

Because the synchronized dialog box is always about one record, you must define the key values as input attributes to the session. No input attributes are required only if you selected the Insert Record option.

In some rare cases, you might still want the user to work with one record, but need access to menu options only available in the main session. In that case, you must model the start-up type as Multi-occurrence Single Record and send the key values as input attributes.

Print, process, and update sessions are regarded as single-occurrence sessions.

Start Option – User Option

To enable the Workflow system to execute a command automatically when the session is started at run time, select a start option in the Start Option field. For example, you can start a session with the Insert Record command. The session then appears to the user as if you clicked Insert on the session. In addition, the Workflow system checks when a record was actually added to close the activity.

Be aware that if you use the start option Get Defaults, data sent as input for the session will be overwritten with the defaults.

If you select Specific as the start option, you can select a form command in the User Option field. Usually, however, modeling the session started
through **Special** menu command directly is preferable. In that way, you can send and retrieve data from that session, because you can only send and retrieve data from the session specified in the **Session Code** field.

If you select **Specific**, be sure to also send data to the session.

If you start an **Infor ERP** LN session, you can use the key combination CTRL+SHIFT+7 to view information about the session.

Below the **Form_commands** tag, you see the values to be used for the user option. For example, if you want the Generate function to run at session startup, you must enter the value `exec.user.3`.
Menu options

To ensure that the user focuses on the task to be carried out in the session, some menu options in the session are unavailable by default.

You can select the following menu options only if you think it would be useful for the task to have that menu option. You must model these options to be enabled at run time:

- **Browse Record**
  This menu option enables the user to browse between records in the started session. If you modeled that the session must start for a specific record, do not select this option.

- **Browse Group**
  This menu option enables the user to browse between groups of records in the started session. If you modeled that the session must start for a specific record, do not select this option.

- **Get Defaults**
  This menu option enables the user to use previous saved values to set the values of the fields in the session. However, this can disturb the values you modeled as input for a session. For this reason, Infor advises that you do not select this option.

- **Text**
  This menu option enables the user to add text in the session.

- **Filter**
  This menu option enables the user to use the **Filter** command on the **View** menu in the session.

- **Chart**
  This menu option enables the user to work with charts in the session.

- **Make Job**
  This menu option enables the user to place this session into a job.

**Note:** Be aware that you cannot enable menu options that are not enabled when you start the session from the Menu Browser. At runtime, you then receive an error. Therefore, the best way is to start the session from the menu and study the working and appearance before you model the session for Workflow.
Concept of attributes

Attributes, which are also known as user-defined attributes, are named variables defined for a particular template and are bound to the templates activities. You can use attributes to exchange data back and forth among Infor ERP LN sessions and the Workflow Management System. You can then also use the same attributes in other applications present in the template.

Together with the design of the template and specification of the activities, you must investigate the flow of data in the process and determine how to retrieve the data.

After you evaluate which data flows are necessary, you must design the attributes in detail. To design the attributes, answer the following questions for each activity:

- Does the activity (session) require data from other activities?
- Must the activity (session) provide data to other activities?
- Does the activity (session) require a single value of a single Infor ERP LN field?

The general task of attributes is to enable the transfer of data between Workflow and the Infor ERP LN sessions. In detail, you must differentiate between the following tasks:

- Transfer key data
- Evaluate decisions
- Display additional information in the Workflow client

The online Help of the workflow clients describes how to customize the view, so you can see attributes in your workflow client’s work list.

To initialize particular fields

To set particular session fields, for example, the order number, to a value based on data in earlier activities, you must bind the field to an attribute.

For sessions to maintain or display data, you can only send data to the key-fields of the session.

For sessions to process or print data, you can send data to all fields on the forms.
To use attributes you must define so-called attribute bindings on the Attributes tab of the Infor Activity Node Properties dialog box:

**Data Item Name**

In this field, you can select any UDA you defined that has a compatible domain to Infor ERP LN fields (Session fields also known as Baan Field). The domains accepted are:

- Date
- Float
- Integer
- String

Note that to avoid run-time errors you must validate the UDA type with the Infor ERP LN field type.

**Data Item Type**

This field displays the data type of the chosen UDA.

**Binding**

With the binding type is defined if the value of the UDA must be used as input for the session and/or must be output from the session.
Session Fields (Baan Field)

The Infor ERP LN field related to the session started, to which data must be sent or from which data must be retrieved.

You can specify the session codes in the following two ways:

1. Manual method
2. Using the browse option

Manually entering the Session Field

To identify the Infor ERP LN fields to which the attribute must be bound, take the following steps:

1. Start the Infor ERP LN session
2. Press CTRL+SHIFT+7. An XML tree appears in your browser with session information:
To model with Infor Workflow 7.4

XML tree:

```
<ttaad2500m000 Name="User Data">
  <Environment>
  </Environment>
  <Attributes>
    <ttaad200.user Name="User">
      <Value = "awever" />
      <Attributes>
        <Number = "1" xpos = "1" ypos = "3" />
        <Tab_order_nr = "1" />
        <Domain = "ttaad.user" />
        <Domain_type = "String" />
        <Domain_length = "8" />
        <Display_length = "8" />
        <Keypart = "true" />
        <Multi_occ = "true" />
        <Input_field = "true" />
        <Input_type = "input" />
        <State = "enable" />
        <Reference_Path = "" />
        <Zoom = "Session" />
        <Zoom_session_attr.zoomsessionS = "ttaad2500m000" />
        <Zoom_returnfield_attr.zoomreturnS = "ttaad200.user" />
        <Echo_field_attr.echo = "true" />
        <Mandatory = "true" />
        <Default_or_Previous = "previous" />
        <Default = "" />
        <Xpos_field_command = "0" />
        <Child_field_behind = "0" />
        <Dependent_on_parent = "0" />
        <Column = "1" />
        <Format = "" />
        <User_Format = "" />
      </Attributes>
    </ttaad200.user>
    <ttaad200.name Name="Name" />
    <ttaad200.utyp Name="User Type" />
    <ttaad200.pacc Name="Pck. Comb." />
    <ttaad200.comp Name="Company" />
    <Old_Formal_commands>
    </Old_Formal_commands>
    <Customizations>
  </Attributes>
</ttaad2500m000>
```
3 Search the tag that represents the field that you want to bind. This tag shows
the current values of the session. If, for example, you want to fill the User
field, you must model `ttaad200.user`.

4 Make sure that the Domain_type of the Infor ERP LN field is compatible with
the data type of the user-defined attribute to which you want to bind the
Domain_type.

Using Browse option

Click on the browse \ zoom button (…). It opens the list of fields. These fields
are retrieved from the repository on workflow server; this repository is filled
from Infor ERP LN. User has to double-click \ select and click OK button on
row \ record to select the required field.

NOTE:
- In order to use browse option a valid session code must have been
  selected in the Activity Tab. Only then the fields belonging to that session
code will be displayed here.
- If session code is not entered in the Activity Tab or if the session code
  entered is invalid, the Session Fields List will be empty.

This repository contains the fields of all forms and main table of the session.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
<th>Description</th>
<th>Key Field</th>
<th>Field Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>bus component</td>
<td>string</td>
<td>Bus Component</td>
<td></td>
<td>Input</td>
</tr>
<tr>
<td>commit time</td>
<td>date</td>
<td>Start Commit Time</td>
<td></td>
<td>Input</td>
</tr>
<tr>
<td>consumer</td>
<td>string</td>
<td>Consumer</td>
<td></td>
<td>Input</td>
</tr>
<tr>
<td>label</td>
<td>string</td>
<td>Sync Object</td>
<td></td>
<td>Input</td>
</tr>
<tr>
<td>sync method</td>
<td>enumerated</td>
<td>Synchronization Method</td>
<td></td>
<td>Input</td>
</tr>
<tr>
<td>sync object from</td>
<td>string</td>
<td>From</td>
<td></td>
<td>Input</td>
</tr>
<tr>
<td>sync object to</td>
<td>string</td>
<td>To</td>
<td></td>
<td>Input</td>
</tr>
<tr>
<td>where to start</td>
<td>enumerated</td>
<td>Where To Start</td>
<td></td>
<td>Input</td>
</tr>
</tbody>
</table>

Session Field list Columns

Field Name: Name of the field

Data Type: Data type the field belongs

Description: Field description
Key Field: Checked option indicates that it was the key field / primary key of table.

Field Type: The field nature or display type, like input or display field.

Mandatory

This check box is only applicable for output bindings. If this check box is selected, the workflow system expects a value to be returned from the session. If no value is returned, the activity stays in the user's work list and the user must re-execute the activity.

To retrieve a value from an Infor ERP LN session, you must select a record. Either one record is visible in the session, or in a multi-record session, you must select one record before you exit the session.

To retrieve field values for use in later activities

To use the value of a field in a later activity, you must bind the field to an attribute. The attribute can then later also be used in a decision, or the attribute value can be shown as part of the view of the workflow client.

You can retrieve data from all fields visible on the forms of the session and for all fields of the main table of the session.

To retrieve values, the Binding type is Output from Session. If the Mandatory check box is selected, the activity will not be completed if no value is returned from the Infor ERP LN session.

Segmented fields

Within Infor ERP LN a new type of field is introduced, the so-called segmented field.

A segmented field is in fact a combination of fields, although it is stored as a single field.

For example consider the ‘item’ field, which contains the item code. This field is built up of two segments. So in fact there is an ‘item.segment.1’ field and an ‘item.segment.2’ field.
For data transfer you must use the field segments instead of the entire field: if you use the entire field Workflow strips the leading and trailing spaces, and the field will therefore not match with the database values.

Concept of sub-session authorization

You can start additional sessions from practically every session in Infor ERP LN. Those sessions are then called subsessions. From those subsessions, you can call yet a third set of sessions. The Subsessions by ERP Session (tgbrg5162m000) session lists the possible subsessions for a particular Infor ERP LN session.
To carry out the workflow task, you can model, for example, that the user has full authorization for the session specified on the Activity tab. By default, that authorization type is inherited to the subsessions of the modeled session. However, in some cases, you might want to have a different authorization type valid for some of the subsessions. In that case, you must model those exceptions on the Sub-Sessions tab. The sessions that you start from one of the sub-sessions then inherits the new authorization type of that session. If you want to make an exception, you must also model that session with a specific authorization type on the Sub Sessions tab.

In this example, you modeled that the user must not be able start the Convert changes to Runtime (ttams2200m000) session.

**Session Code**

Here, you must enter the session code of the sub-session for which you want a different authorization type, and then for the session from which the subsession is started.

**Authorization**

Here, you must select the authorization type that will be applicable for the subsession entered in the field session code.

If you model No Authorization for a subsession, at startup of that subsession, a message appears and the startup is cancelled. As a result, if a session is part of the **Special** menu or is a button on the form, that menu item or button
will not be unavailable. The user only knows that they cannot start the session after choosing the menu option or pressing the button.

**Note:** you cannot model the Calculator (ttstpcalc) and Calendar (ttstpcalendar) subsessions. These 2 subsessions are displayed in the subsession tree, see the sample screenshot earlier in this section, but they are not real Infor ERP LN sessions. They actually start the MS Windows calculator and calendar.

**To run the Infor ERP LN session**

In the Workflow Client (Worktop\Webtop), all activities are shown. Each type of activity has another icon to indicate the type. The following Work List shows three activities of type Infor ERP LN Session:

- Display Logon Data
- Transport Orders
- Start Maintain User Data

The activities are also displayed in the so-called Work List view. See the following figure:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>State</th>
<th>Attachments</th>
<th>Priority</th>
<th>Received Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>fmlbd4100m100</td>
<td>SSA ERP LN</td>
<td>Active</td>
<td>0</td>
<td>Medium</td>
<td>4/7/2005 3:04:17 PM</td>
</tr>
<tr>
<td>gmpc1100m100</td>
<td>SSA ERP LN</td>
<td>Active</td>
<td>0</td>
<td>Medium</td>
<td>4/7/2005 3:22:59 PM</td>
</tr>
</tbody>
</table>

By default the columns displayed are about the standard activity data. However the end-user can add columns to show the “User Defined Attributes”:

- In the Worktop the end-user must manually type the names of the UDAs, UDAs which are marked to be displayed in worklist, see the User defined attribute tab on activity node.
- In the WebTop the UDAs are displayed and the end-user can select them.

If a UDA should not be visible in that dialog, then you should give it a name starting with “__”.

The use of the workflow functionality in the Worktop and WebTop is described in the online-Help of those clients.
All requests to start an Infor ERP LN session from the workflow clients are controlled by a so-called workflow daemon in the Infor ERP LN environment. This workflow daemon communicates with the 4GL-engine and together they handle the session to be started in the way you modeled.

However, if code constructions are used in the sessions that conflict with the workflow daemon control, errors will occur. More information about which type of code constructions are meant, you can find in the Appendix Customized sessions and Infor Workflow.

In general, errors that result from conflicting code constructions are not logged. The session either stops responding or fails completely, and you must then restart your workflow client.

Most errors that are the result of modeling mistakes can be found in the log-files log. events or log.wfdaemon. These log files describe, for example, whether a particular menu option was not enabled in the session.

**Note:** To see the content of the log files on your screen, you can start the ttstperfllog session and choose as device D (display).

**Workflow limitations**

Infor ERP LN software is predominantly Workflow-enabled, which means that the software can be controlled at run-time by the Workflow system. However, a small number of Infor ERP LN sessions have limitations if you want to incorporate the sessions in processes that are used by Workflow. A small number of Infor ERP LN applications cannot be used by Workflow at all.

The following categories of limitations can be distinguished:

- Sessions that are not Workflow-enabled
- Sessions that are 3GL instead of 4GL
- Multi main table sessions
- Particular types of subsessions
- Sessions that are part of hard-coded Workflow

This chapter describes the consequences for each limitation and describes the workarounds you can use if you want to model particular Infor ERP LN applications in Workflow processes.
Sessions that are not Workflow-enabled

You can use most Infor ERP LN sessions in Workflow without any problems. However, due to specific programming characteristics, a number of sessions can present problems when used in Workflow.

These sessions are not fully controlled by Workflow, which leads to incorrect data transfers, unsatisfactory messages, and eventually to an obstruction of the Workflow processes.

Part of those problems can be caused by the use of so-called bshell commands. These problems can surface as messages regarding sessions stopped by unknown process or a total block of the Workflow Client.

Recoding the session can resolve most of these problems. For more information, refer to the appendix.

Do not model problem sessions in a template. In this case, you must try to find another session with the same functionality and decide not to run this session in Workflow. In the definition of the session, you must specify that the session is Not Workflow Enabled.
During execution, the workflow daemon checks this field and then displays a message, if a session is not Workflow enabled.

**3GL sessions**

Some of the Infor ERP LN sessions that you can use in Workflow modeling are so-called 3GL sessions.

Most Infor ERP LN sessions are 4GL sessions. Like 4GL sessions, Workflow can start 3GL sessions, but cannot communicate fully with the Workflow daemon.

Therefore, the Workflow daemon does not know if a 3GL session is closed. In addition, you cannot transfer Workflow attributes from or to a 3GL session. To enable the Workflow daemon to know that a 3GL session was carried out, a question appears if you start a 3GL session from the Workflow Client that asks you whether the session was run successfully. The following figure provides an example:
If you click **Yes**, you can proceed to the next activity. If you click **No**, the activity remains in your Worklist Handler and you cannot proceed to any other activity until you carry out this activity.

Although you cannot use a 3GL session to transfer data from one **Infor ERP LN** session to another, including a 3GL session in a process can be useful due to the added functionality and because the session can display important information.

If, however, you use a 3GL session to change data, through startup of so-called detailed sessions, modeling the detail sessions is preferable. The reason for this is because you can then send and receive data from the session and also control the authorization type of the session.

**Multi Main Table sessions**

Within the **Infor ERP LN** product a new type of session is introduced. The new session type is called Multi Main Table (MMT) session. The following figure shows a sample MMT session.

To model such a session, you need to know, that every tab at the bottom is regarded as a subsession. So when you need different authorization types for the different tabs, then you need to model each as a subsession. In this
example the tabs are subsessions for “BOM”, “Routing”, “Product Features” and “Constraints”.

Mostly you would like to limit the actions of the end-user within workflow, so the end-user is focused on the task to be performed. The MMTable sessions do not fit in that strategy. For some of the MMT sessions you can find another session to execute the task needed.

For other MMT sessions you need to model them carefully to keep the end-user focused on the task, probably using subsession modeling to restrict certain features to be used.

To find out, which subsession codes you need to use, right-click the Infor ERP icon in your taskbar and click Processes on the shortcut menu. The Process browser is started. See the following figure for an example:

In this browser you can see, that for example, the subsession code for the BOM tab is “tipcf3110m0000”.

If you prefer that the end-user is only focused on “BOM” actions, then you can also model this BOM subsession instead of the multi-main table.

Data transfer to a multi-main table session is only possible to the fields in the main (top) part of the session. Those fields will also only be shown in the XML browser after you press CTRL+SHIFT+7 in the MMT session.

**Note:** for more information on MMT sessions, refer to “Multi Main Table sessions” in the Infor Web Help.

**Subsessions**

You can start the subsessions that you model in business processes directly by Workflow without having to carry out the main session first. However, if you model subsessions in such a way that they can be directly started by Workflow, you can run into problems if the subsessions are not properly prepared by the main session. As a result, you must send the right key field values to the subsessions by means of Workflow attributes.

Nevertheless, some subsessions will terminate abnormally regardless of when you close the subsessions, because the subsessions are not suitable to be directly started by Workflow.

In this case, you must either have the subsession changed or model the main session.

**Hard-coded Workflow**

Within Infor ERP LN there is so-called hard-coded Workflow. Hard-coded Workflow means that the sequence of activities in certain areas of Infor ERP LN is determined by the application design. In other words, the application logic determines whether a next step can be carried out. If you do not follow the sequence of sessions as described in the application logic, the application and the Workflow Management System will no longer be synchronized. To avoid this loss of synchronization, the processes must be modeled exactly as described in the application logic. If a predefined sequence of activities is modeled correctly, the Workflow Management System can handle these processes.

Modeling such processes within Workflow would mean that you probably would extend the basic application logic processes with manual tasks and/or approval steps.
Examples of areas that contain this kind of application logic are:

- Sales Order Management
- Purchase Order Management
- Warehouse Order management
- Project Structure

You can find the structure of the application logic in the online-help of the major sessions within the relevant area.

To model using the Infor Action Type: Call Program

The Call Program activity type enables you to include a call to an external program in a workflow template. For example, you can include a call to Microsoft Excel to create a spreadsheet, or you can refine the call and not only open Excel but also open an existing spreadsheet. You can call any program on your computer, but the program must also exist on the computer of the client who performs the activity.

The Call Program activity type has many of the same properties as other activity nodes. For a description of the other properties, refer to the BPM documentation.

To add a call to an external program to a workflow template, take the following steps:

5 Log on the iBPM Console and click the Template tab.
6 Select an existing template to edit or create a new template.
7 Select the Call Program activity type as described in “To select an Activity Type.” The following dialog box appears:
Enter the name of the program that the workflow calls. If calling the program is sufficient, you do not need to make any other entries.

8 Click Add to activate the fields.

9 Use the following fields to refine the call. The program automatically generates the step numbers.

**Type**

Select one of the following:

- Constant Value
- UDA Variable

**Value**

Enter a value to refine the call. For example, to call Microsoft Excel and open a particular file, enter the pathway and file name.

10 Enter any other steps in the order in which the steps are to be performed. If the steps are out of sequence, select an entry and use **Move Up/Move Down** to position the steps appropriately. To remove an entry from the list, select the entry and click **Remove**.

11 Click **OK** to save your entries.
To work with attachments

Attachments are a way to link documents with a running process. These documents can contain work instructions, plans, images, and so on. Attachments are available during run-time. For example, users who work with workflow clients can add, modify, and delete attachments.

Addition of an attachment to a process results in the upload of the attachment to the Workflow system. In this way, the attachment becomes available to all users that participate in the running Workflow process.

The following sections describe attachments in the context of the Infor ERP LN Worktop client. However, similar options are available in the Infor WebTop or Infor ERP LN client.

Add attachment

To add an attachment to a running workflow process:

1. Select an activity in the work list.
2. Right-click the activity and, on the shortcut menu, click Attachments… to view the attachments. The Attachments dialog box appears:
3 Click **Add**.

The **Add Attachment** dialog box appears:

4 Enter the **Name** and **File** location. The **Name** is the name of the attachment; the **File** is the location of an existing file.

5 Click **OK**.

**Note:** Select the **Overwrite if Exists** check box to overwrite the attachment file if the attachment (name and file are the same) already exists on the workflow server. This can be useful, if the local version of the file is more up to date than the existing version on the server. Leave this check box cleared if the attachment on the server must be preserved. The number of attachments is increased for the activity’s process, as shown in the following figure:
**Note:** When you open the **Attachments** dialog box again and the upload to the workflow server was successful, the attachment's file location is shown on the workflow server as shown in the following figure:

![Attachment dialog box](image1.png)

**Note:** You can add multiple attachments to a process. For each added attachment, neither the attachment's **Name** nor **File** must already exist in the available list of process attachments.

![Delete attachment](image2.png)
To delete an attachment, right-click the activity and, on the shortcut menu, click **Attachments**... to view the attachments. Select the attachment to delete and click the **Delete** button.

**View Attachment**

![View Attachment](image)

To view attachments, right-click the activity and, on the shortcut menu, click **Attachments**... to view the available attachments. Select the attachment to view and click **View**.

**Note:** If you view an attachment in this way, you cannot modify the attachment.

**Edit attachment**

![Edit attachment](image)

To edit an attachment, right-click the activity and, on the shortcut menu, click **Attachments**... to view the available attachments. Select the attachment to edit and click **Edit**.

**Note:** After you close the application that contains the attachment during the edit, the attachment is uploaded to the workflow server. If any user edits the
attachment during the process, other users can only view the attachment in the attachment’s state before being edited.

Attachments administration

During the execution of multiple processes in time, the number of attachments on the workflow server grows. The system administrator of the system on which the workflow server resides is responsible for the amount of attachments that are stored.

Applicable attachments to be removed from the server are those for which their process is closed. Information about closed processes can be obtained using the Administration console. For detailed information, refer to the Interstage (iBPM) documents referenced in the “About this guide” section of this document.

The attachments on the workflow server are tagged with process information. This enables each of the attachments to backtrack to their owner process. Each attachment is stored in the following format:

- `<File Name>_ProcessID_<ID>._File Extension`

  where:

  - `<File Filename>` is the original file name of the attachment without its extension, for example, Work Instruction, Design_Plan, and so on.
  - `_ProcessID_` is the text: _ProcessID_
  - `<ID>` is the actual ID of the process to which the attachment is linked, for example, 16318, 121223, and so on.
  - `<File Extension>` is the actual file extension of the original attachment, for example, .txt, .doc, and so on.

The attachments are stored in the following directory on the workflow server:

- `<Install Path>\attachments`

  The `<Install Path>` is the location where the workflow extension is installed.
Chapter 3
Workflow Triggering

This chapter describes how to use the Workflow Triggering part of the Infor Workflow for Infor ERP LN.

Overview

During the development of templates, an analysis is made of when a process should be started from a particular template.

The conclusion could be that processes started from a specific template must be initiated, for example, by table changes. To enable that possibility, the triggering functionality is realized. In addition, you can use triggers from sessions based on the application logic.

To implement triggering, a good analysis of the event must take place, which starts the process.

Workflow triggering is an implementation on top of more generic triggering functionality realized in the tools area. This process is described in greater detail in the User’s Guide for Triggering Infor ERP LN Server (U8762C US).
Workflow Triggering Configuration

This chapter describes how to configure a Workflow Trigger. For more detailed information about a particular session, you can click Using Session on the Help menu in the sessions.

The Workflow Trigger, which will be configured, is to ensure that when a new sales order is entered by means of EDI, a process starts to perform several basic checks.

Before you begin to configure, you must define a default text group for your Infor ERP LN user in the company in which you want to define the workflow trigger. To do this, you can attach a Default Text Group template to that user in the User Data (ttaad2500m000) session.

The best procedure is to have the template in i-Flow modeled before you configure the Workflow Trigger.

1 Fill in master data that will be applicable for all Workflow Triggers.

Start the Workflow Parameters (tgwfm0100m000) session:

![Workflow Parameters](image)

Click New to add a parameter set. See the following figure for an example.
The first three paths are to place files that are created during the generation of the triggers. There are error files and code files. The code files are used by the triggers at runtime.

The errors that appear during the exchange export with the Export Data (On a Regular Basis) session are written into a file in a subdirectory below **Table Output Files Path**, which looks as follows:

```
<TriggerName>.<Run Number>.e
```

The XML files generated at runtime are placed in a directory on the **Infor ERP LN** server and, when needed, can be automatically transferred through FTP to i-Flow server. To perform this transfer, you must fill the XML master data fields. Be aware that the FTP path must be a logical name known to the FTP server on i-Flow server. You must enter the FTP password in a separate dialog box, which ensures the encryption of the password.

When you select the **Keep XML Files on Server** check box, the files will stay on the ERP server after a successful FTP to the workflow server. You then must develop a procedure to clean up the directory occasionally.

2 Define the Workflow Trigger.

Start the Workflow Triggers (tgwfm1100m000) session.
Because the trigger is about a new sales order that arrives from EDI, the trigger must be defined as **Table Invocation**, and based on a create event in the Sales Order table (tdsls400) table.

After you also define the UDAs for the Workflow trigger and click **Save**, the generation process starts.

During that process, data is saved as Exchange Scheme and Tools-trigger information. The **Change Data** tab shows when a change and/or a generation were carried out.

Errors during generation are written to the log file log.tgwf.
3 Define the conditions for the Workflow Trigger.

In the Workflow Triggers session, click **Conditions** to start the Workflow Trigger Conditions (tgwfm1110m000) session.

In this session, you can define the conditions to be checked after detecting that the specified event on the specified table has occurred. For this trigger, the **Origin (tdsis400.corg)** field must have the value **3 (=EDI)**.

Defining a condition is not mandatory. If you do not define a condition, all events of the type specified with the trigger lead to starting a process. For this trigger, that means that all new sales orders would lead to a process being started.

4 Define the UDAs for the Workflow Trigger.

In the Workflow Triggers session, click **UDAs** to start the Workflow TriggerUDA’s (tgwfm1120m000) session.
In the process to be started, data from Infor ERP LN might be required. The process data is stored in so-called user-defined attributes (UDAs). To ensure the transfer of the data from the table on which the trigger is based to the triggered process, in this session, you define the link between the table fields and the UDAs. The UDA name is case sensitive.

5 Set up the audit for the involved table.

Because the Workflow Triggering functionality is based on the generic Triggering functionality, which in turn is based on auditing, you must turn on auditing for the involved tables.

To activate the audit trail for a table, you must link the table to the “workflow” audit profile of the package to which the table belongs. To do this you must take the following steps:

a Start the Audit Profiles (ttaud3110m000) session and select the “workflow” audit profile of the package to which the table belongs. For example: select the “td workflow” profile if you want to activate audit trail for the tdsls400 table.

Subsequently click **Table Settings by Profile** on the **Specific** menu. The Audit Tables by Profile (ttaud3120m000) session is started. See the following screenshots.

**Note:** If there is no “workflow” audit profile for the relevant package, you must first create one, for example, “td workflow”, and link it to the “workflow” audit
category. See the example in the screenshot. If the “workflow” category does not exist, you must create it in the Audit Categories (ttaud3100m000) session.

b Click **New** to add the table(s) to the profile. Set the Field Selection to “All” and the Audit Type to “Always”. In this way the events generated by Workflow will contain all fields (changed + unchanged) of the records involved. **Note:** if you set the Audit Type to “Changed”, the events only contain the primary key fields and the changed fields. This is not recommended for Workflow.

c Run the Create Runtime Audit Definitions (ttaud3200s000) session. Make sure that at least the **Audit Profiles** check box is selected.
Finally, log off and log on again to load the new audit settings. **Important:** to prevent the loss or corruption of data, all users must leave Infor ERP LN when new settings are converted to run time.

You are likely not permitted to run the above mentioned sessions. Ask the appropriate person in your organization to set up audit trail for the tables used in your Workflow triggers.

For more information on Audit Management, refer to the Infor Web Help.

6 Define the trigger on the template in i-BPM.

For the Workflow Trigger, an XML Schema Definition (XSD) file is created in the directory specified in the workflow master data session as the **Trigger Objects Path**. That file is called `<Triggername>.xsd`, and for the example, the file is called sales.xsd. This file must be placed on i-Flow server, to be used during the configuration of the Template trigger. You must transfer this file manually and place it in a location accessible by the web accessible.

The value in the XML Schema in this example is http://<HostName>:49950/ibpm/xml.xsd".

Start the iBPM Console and open the template for edit. Next, on the **Template** menu, click **Properties**. Click the **Triggers** tab.
On the **Event** tab, you must specify the XML schema reference, which is a URL to the XSD file on i-BPM server. Click **Retrieve**. The “Schema Loaded” message appears. If the path is incorrect, an error message appears.

You must specify the **Event Filter** to establish that only events intended for this trigger must be processed. For event name “sales,” for example, you must type the following in the **Event Filter** field:

```
toString(getXPath("/Event/ControlArea/eventName/text()"))="sales"
```

7 Click the **Data Mapping** tab:
On this tab, you can select the elements from the XSD, which refer to the UDAs defined in the configuration session and then link these UDAs to the UDAs defined in the template.

8 Enable the trigger for the template in i-Flow.

The final step is to activate the trigger for the template. From that moment on, you can start a process from this template if XML files arrive in the appropriate directory on i-Flow server. Be aware that to start a process from the template, the template needs not to be published.

Application Invocation

Note: The trigger you define with the steps described above is based on a table event (Trigger Type = Table Invocation). However, triggers can also be based on a call from an Infor ERP LN session (Trigger Type = Application Invocation).

The configuration procedure for both trigger types is nearly the same:

- Procedure steps 1, 6, and 7 are identical for both types.
- For a trigger that is based on Application Invocation you can omit step 3 and step 5.
- Steps 2 and 4 are different. See the following examples:
For a trigger that is based on Application Invocation the information for step 2 must be as follows:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigger Name</td>
<td>Invapp</td>
</tr>
<tr>
<td>Trigger Description</td>
<td>Invocation from application</td>
</tr>
<tr>
<td>Trigger Type</td>
<td>Application Invocation</td>
</tr>
<tr>
<td>Table Name</td>
<td></td>
</tr>
</tbody>
</table>

It is not possible to specify a table or an event.

For a trigger that is based on Application Invocation the information for step 4 must be as follows:

<table>
<thead>
<tr>
<th>Sequence Number</th>
<th>Table Field</th>
<th>UDA Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>amount</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>factor</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>number</td>
<td></td>
</tr>
</tbody>
</table>

Because no table is connected to the Workflow trigger, the defined UDAs cannot be linked to Infor ERP LN table fields.
In this case, UDAs receive values by means of the session script. Be aware that UDAs are case sensitive.

**Workflow triggering runtime**

To launch the triggers at runtime, you must schedule the trigger.

1. Start the Export Data (On a Regular Basis) session. You can start this session from the **Tools ➔ Integration Tools ➔ Exchange ➔ Export Module:**

2. You must fill in the fields as shown in this figure. In the **Exchange Scheme** field, enter the trigger name.

3. If you click **Continue**, the audit files will be processed for the specified start date-time to the current date-time.

   This session can be placed in a Job to be executed on a regular base.

   For more details on how to use the session, click **Help.**
The errors that appear during the use of this session are written into a file in a subdirectory below Table output Files Path, as specified in the Workflow Trigger master data, which looks like <Triggername>.<Run Number>.e.

During execution of the triggers, trigger XML files are transported between the Infor ERP LN server and i-Flow server. If the two systems are not connected, the trigger XML files present on the Infor ERP LN server must be transferred manually to i-Flow server.

If you have the option to keep the trigger XML files on the Infor ERP LN server activated, seeing which files are transferred and which are not is more difficult. The reason for this is because the normal working procedure is such that i-Flow removes the XML files after a successful start of the process.

To start a trigger from an Infor ERP LN session, you must include a function call (to datgrapi.trigger.do) in the program script of that session. For more details, refer to the sample code in the following section.

Sample code to launch a trigger from an Infor ERP LN program script

```plaintext
function main()
{
  long ret1

  ret1 = generate.trigger("credit",
                         "1001", "300055", "10000", "8100", "7900")
    | in real program script the values come for example
    from table-fields
    if ret1 = 0 then
      message("Ready")
    else
      message("Trigger creation failed")
    endif
}
```
function long generate.trigger(
    const string i.triggername,
    const string i.customerid,
    const string i.ordernumber,
    const string i.creditlimit,
    const string i.creditinuse,
    const string i.prev.creditinuse)
{
    long event    | tree containing trigger event
    long retl     | return value to be checked
    string dums   | dummy string

    | first specify type of event
    retl = datrgevent.simpleevent.create("customer",
        "creditLimitAlmostReached", event)
    if retl <> 0 then
        return(retl)
    endif

    | then fill all the UDA’s
    retl = datrgevent.simpleevent.set.value(event,
        "customerid", i.customerid)
    if retl <> 0 then
        return(retl)
    endif
    retl = datrgevent.simpleevent.set.value(event,
        "ordernumber", i.ordernumber)
    if retl <> 0 then
return(retl)
endif
retl = datrgevent.simpleevent.set.value(event,
    "creditlimit", i.creditlimit)
if retl <> 0 then
    return(retl)
endif
retl = datrgevent.simpleevent.set.value(event,
    "creditinuse", i.creditinuse)
if retl <> 0 then
    return(retl)
endif
retl = datrgevent.simpleevent.set.old.value(event,
    "creditinuse", i.prev.creditinuse)
if retl <> 0 then
    return(retl)
endif

| then invoke the trigger
retl = datrgeapi.trigger.do(i.triggername, event, dums, dums)
if retl <> 0 then
    return(retl)
endif

return(0) | OK
This chapter describes administrative tasks, such as Infor workflow Auditing, configuration, sample workflow import, and client configuration.

Workflow Configurator

When you install Infor Workflow 7.4, the installation program uses your entries to create the default configuration for the workflow programs. Usually, you need not have to make any changes to these settings. However, when you add or change servers, change your passwords, or if you want to perform troubleshooting, you can change the workflow configuration.

**Note:** Restart the Workflow services and the web server after making the changes

To open the Configurator

Configurator has been changed from the previous versions of the Workflow product. Till WorkFlow 6.2, **Configurator** was a standalone tool. Now it is a web application, which is accessible from the IBPM web application.

Complete the steps to access the **Configurator** page:

Use the following URL to access Interstage® Business Process Manager home page:
http://<Hostname>:<portno>/ibpm/

The following page opens:

1. Click the **Administration** button in the page shown in the (above) picture
2. The Interstage Business Process Manager Administration page opens:
3 Enter the **User ID** and **Password** and click Login.

4 The Configurator is available in **Process Definitions** tab. By default for every new login, the opening page for an Administrator login is **Process Definitions** tab. If you are in a different page then click the **Process Definitions** button.
5 Click the **Configurator** button; the Configurator page opens:

By default the first page opened for the configurator shows Data Sources configuration list page. This page lists the different Data Sources configured for this server.
The following links are available for Configurator:

a. Data Source – default first page shown.
b. Trigger Event Configuration
c. File Listener

Each configuration and its corresponding pages are described in detail in the following part of the document.

**Data source definition maintenance**

This program configures the Java Database Connectivity (JDBC) data sources. You can configure additional data sources and update existing data sources. Provide a unique name to each data source. For example, you can set up additional data sources for various application servers, such as Test and Production.

**To add and update a JDBC data source**

To add or maintain a JDBC data source, complete the following steps.

1. Open the Workflow Configurator as described above. By default the Data Source list page is shown.

2. If you are in a different configuration page such as Trigger Event Configuration or File Listener configuration, click Data Source button; the page shown above figure is displayed.

3. The following actions are possible with the Data Source Configurations:
   a. Create a Data Source configuration
      o Click the **Create New** button.
Enter the details as described below in the step C – **Update Data source** and click **Save** button as shown in the figure. The only difference between details while updating and creation of a data source is, while updating, the Data source name is not editable.
b. View Details of a Data Source configuration

4 Click on a data source name to view the details (see the above figure)
a. Update Data Source configuration

The Username, password, URL and Driver Name fields are editable. To update, modify the values in these fields and click **Update** as shown in the figure above.

- **DataSourceName**
  Specify the data source you want to add.

- **User Name**
  Specify the user ID to log on to the data source from the workflow server.

- **Password**
  Specify the password to log on to the data source.

- **URL**
  Enter the URL for the JDBC data source. Include the server name and the data library with the files and programs that generate the workflow data.

- **Driver Name**
  Select the appropriate class driver.

**Note:**

Every workflow server has a database as a backend for storing the workflow information. This backend database can be Oracle or MS SQL server or any other database supported by Infor Workflow. Hence by default the drivers to connect to backend on which workflow database is present are available on the workflow server. These drivers can be directly used to create a Data
source configuration for another database on the same database product used as backend for workflow server.

However, when we want to create a data source configuration to use a database on any other database, we will need the specific drivers to connect to that database. These drivers can be referred to as “Third party drivers”. These other third party drivers must be placed in

- `<TARGETDIR>\taskmanager\iflowjsp\WEB-INF\lib` and
- `<TARGETDIR>\lib\ext`

b. Test Data Source by connecting to data source

5 Go to the details page of a data source. Click **Test**. The test result is displayed as follows:
c. Delete Data Source configuration

Go to the details page of a data source. Click **Delete** to delete the currently displayed data source. The following dialog box appears:
6 Click **OK** to confirm deletion or **Cancel** to keep the configuration.

**To configure the XML File Listener**

The XML Listener is a Workflow component that looks for XML documents received from the server and notifies running workflow processes that new data is available.

The XML Listener also starts new workflow processes when the Listener receives new trigger data. The XML File Listener is an integral part of Infor Workflow. You cannot add another listener and you cannot delete the existing listener. You can customize some of the XML File Listener features.

By default, the XML Listener deletes XML documents that the Listener processes successfully and saves XML documents that cannot be processed.

You can change these defaults and, if you save the XML documents, specify the directory where they are saved. You can also adjust the frequency at which the XML Listener scans for new XML files.

To maintain the XML File Listener, take the following steps:

1 Click on the Configurator button in the Administration Console,
2. The page with links to different configuration pages opens.
3 Click on the **File Listener** button, the page with the existing file listener opens:

![Image of the Interstage Business Process Manager interface with the File Listener button highlighted.]

4 The page where you can modify the **XML File Listener** opens.

![Image of the Interstage Business Process Manager interface showing the XML File Listener Details page.]

5 Update the following information:

- **Path**
  The location of the XML File Listener. The installation populates this field.

- **Scan Interval**
  The duration, in milliseconds, with which the XML File Listener scans for new XML documents.

- **Stabilize Period**
  The time, in milliseconds, in which no changes are made to a file size and no modifications to a file. A file must be stable for this period before the file listener can open and process the file.

- **File Handler Class**
  The class that handles trigger processing. Accept the default.

- **On Success Delete**
  By default, the XML File Listener deletes the XML documents the Listener processes successfully. To save XML documents, clear this check box.

- **On Success Move**
  This field is active if you save XML documents.
  
  Enter the directory where the XML documents are saved.

- **On Error Delete**
  By default, the XML File Listener saves XML documents that Listener cannot process. Click this check box to delete the XML documents.

- **On Error Move**
  The installation program provides a default location for XML documents the program cannot process. You can specify a new location.

6 Click on **Update** to save your entries.

7 The following message is displayed on successful saving of the updated details. Follow the link to the “**File Listener Details**” for further modifications to the entry.

---

**Auditing the Process Instance**

**Audit Process Instance** functionality enables the administrator to monitor the execution of process instances. Typically, an audit involves the following activities:

- Tracking the User Defined Attributes (UDA).
The history of changes performed on a process instance.

The output of an auditing activity is an audit report. An audit report contains the following information:

- The person who performed the operation
- Operation performed on the process instance
- UDA on which change is performed or the UDA affected as a result of the operation
- Time stamp indicating the time and date when the operation is performed on the process instance and so on.

**Note:** To view the history of an UDA, you must enable the **Trackable** check box during the process definition modelling.

### Accessing the Audit Functionality

To access the Audit Process Instance functionality, complete the following steps:

Use the following URL to access Interstage® Business Process Manager home page:
http://<Hostname>:<portno>/ibpm/

The following page opens:

1. Click **Administration**, the login page opens:
2 Provide the User ID and Password.

3 Click Login, the Interstage® Business Process Manager-Administration page opens:
4. Click on the **Process Instance** tab.

5. Click Audit Process Instance, the **Audit Process Instances**:Filter Process Instance Criteria page opens:
The following table lists the fields on the **Audit Process Instances: Filter Process Instance Criteria** page:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Date of Creation            | The date and time when a process instance is created. You can use the calendar control to input the date and time. However, you can manually specify the date and time in the following format:  
  *Month Day, YYYY hr:minutes* |
| Process Definition Name     | The name of the process definition. You can use the logical operator for SQL query. Following are the operators for the field:  
  - Contains  
  - Does not Contain  
  - Starts with  
  - Does not start with  
  - Ends with  
  - Does not End with  
  - Equals  
  - Not Equals  
  - Like  
  - Not Like  
  You must select one of the above operators and specify the filtering criteria.  
  **Note:** The default operator is **Contains**. |
| Process Instance Name       | The name of the process instance. You can use the logical operator for SQL query. Following are the operators for the field:  
  - Contains  
  - Does not Contain  
  - Starts with |
### Field Name | Description
--- | ---
- Does not start with  
- Ends with  
- Does not End with  
- Equals  
- Not Equals  
- Like  
- Not Like  

You must select one of the above operators and specify the filtering criteria.

Note: **The default operator is** Contains.

---

**Process Status**

Status of the process Instance. The following are the possible values for the field:

- **All (Except archived):** Process instances in which all the activities are successfully executed or suspended. This will exclude only the Archived Process instances.

- **CLOSED/ABORTED/SUSPENDED:** Process instances whose status is either closed or aborted or suspended.

- **Error:** Process instances in which errors are encountered.

- **Running:** Indicates all the process instances that are still in running state. It means there is at least one activity yet to be completed in this process instance.

---

**Process Instance DB Status**

Indicates whether a process instance is archived. Following are the possible values for the field:

- **Archived:** The process instances that are archived. When you select this value **UDA Value** field is disabled, that is, user cannot filter the process instance based on the UDA value, you can use only the **UDA**
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>for filtering.</td>
</tr>
<tr>
<td><strong>Not Archived</strong>: The process instances that are not archived. When you select this value, you can filter the process instances using both the UDA Values and UDA Name.</td>
<td></td>
</tr>
</tbody>
</table>

| UDA Name | User Defined Attribute. |

<table>
<thead>
<tr>
<th>UDA Type</th>
<th>Data type of the UDA. Following are the possible value for the field:</th>
</tr>
</thead>
</table>
|          | • String
|          | • Integer
|          | • Long
|          | • Float
|          | • Big Decimal
|          | • Boolean
|          | • Date |

<table>
<thead>
<tr>
<th>UDA Value</th>
<th>The search criteria to filter the UDA. You can use the following logical operators to filter the UDA values:</th>
</tr>
</thead>
</table>
|           | • =
|           | • >=
|           | • <=
|           | • >
|           | • >
|           | • !=
|           | • LIKE
|           | • NOT LIKE
|           | • IN |

**Warning**: The operator selected in this field must be appropriate with respect to the Data Type selected in the UDA Name field.
For example, you cannot specify a **Like** operator for numeric values like **Integer**, **Float**, **Long**, **Big Decimal** and **Date** data types.

**Field Name** | **Description**
--- | ---
File Type | Indicates the format of the audit report. Following are the possible values for the field:
- **Text**: The audit report is generated in the RTF format.
- **PDF**: The audit report is generated in the PDF format.
Preview Audit Report | You can use this option to preview the audit report. This field has the following options:
- **Yes**: Click this option to preview the audit report in the HTML format.
- **No**: Click this option to write the report directly to the file, based on the value specified in the **File Type** field.

**Command** | **Description**
--- | ---
Get Audit Report | Generates the audit report. The format of the output is based on the value selected in the **File Type** field. You can preview the report before you print it to the file by selecting the **Yes** option in the **Preview Audit Report** field.
Clear | Clears all the fields on the form.

6 Specify the filtering criteria.

7 Click **Get Audit Report**. Following are the possible scenarios:

   a In case you select option **Yes** in the **Preview Audit Report** field, the audit report opens in a web page as follows:
In the above web page, click **Write To File** to generate a report in the format specified in the **File Type** field.

In case you select the option **NO**, in the **Preview Audit Report** field, a **File Download** dialog box appears.

- Click **Save** to save the audit report on your system.
- Click **Open** to open the audit report without saving it.
Client configuration

Infor Workflow 7.4 for Infor ERP LN utilizes two workflow client applications: Infor WebTop for Infor ERP LN and Infor ERP LN Worktop.

To enable both Webtop and Worktop users to access the workflow work list, you must first make these users available on the workflow server. For detailed information about user configuration, refer to the Worktop documentation referenced in the “About this guide” section of this document.

If users are authenticated to participate in workflows, the users can perform workflow-related configurations. For WebTop and Worktop, the following sections describe these configurations in detail.

Infor WebTop 8.4 for Infor ERP LN

For more information on WebTop 8.4, refer to the Infor Workflow for ERP LN Install guide for Webtop 8.4 configuration.

Infor Worktop 2.4.15

The Infor ERP LN Worktop client can serve as a workflow client such as the Infor WebTop 8.4 for Infor ERP LN.

To change the Workflow communication parameters of an existing Worktop document, you must first open the document. On the Tools menu, click Options to open the Options dialog box. A Workflow tab appears, in which you can modify Workflow settings.
In the Port field, you must specify the TCP port on which the Workflow Web server is listening. That is: the port number that was specified during the installation in the Web Server Port dialog box of the InstallShield Wizard.

Note: This dialog box also appears if you use the new document wizard to create a new document. For detailed information, refer to the Worktop documentation referenced in the “About this guide” section of this document.

Workflow update

Three versions of the iFlow/iBPM server are available. For each version, a specific Workflow integration component (iFlow.jar) used.

Note: By default, iFlow.jar build 87 exists in <worktop root>\lib\iflow_first and iFlow.jar build 110 in <worktop root>\lib\iflow_second. The latest iflow.jar for Workflow 7.4 will be downloaded automatically once the user connects to the server.
Alternative solution to download iFlow.jar

Open the following URLs:


This call downloads the iFlow.jar file to a location that you can specify. Location: <worktop root>/lib/iflow_first directory and under cache folder there will be a folder with workflow server name if not exists create and place this iFlow JAR.
Appendix A

Workflow-controlled Infor ERP LN session

If a Workflow user starts an Infor ERP LN session from the Workflow client, a dedicated communication between the Workflow client and the Infor ERP LN Server is established. This communication is a background process and is not visible to the Workflow user. However, to understand the necessary modifications, this section provides additional information.

If a Workflow user starts an Infor ERP LN session, the session-startup request is received by the so-called Workflow daemon. The workflow daemon communicates with the 4GL engine about the start of the session. The communication results in the visible start of the session in the way the session was modeled.

When the user starts a session, a process ID is generated. The communication is organized around that process ID.

If the Infor ERP LN session contains so-called bshell commands, the process ID is disturbed, in which case communication between the Workflow Daemon and the 4GL engine can also be disturbed.

The following session-internal commands and functions were detected to cause workflow-related problems:

- EXIT
- KILL
- ABORT
- EXECUTE(END.PROGRAM)
- EXECUTE(ABORT.PROGRAM)

To ensure that a customized Infor ERP LN session runs and ends correctly, check that the corresponding session does not contain bshell commands. Otherwise, follow the instructions on how to change, as described in the following section.

To change the Exit command

The **Exit** command is a bshell command and is often used to end Infor ERP LN sessions. The exit command must be replaced in all those Infor ERP LN sessions that must be workflow-controlled.

- Instead of the `exit()` command, use the `stop()` command.
- In code lines with the `exit(<Variable>)` phrase, use `export(<Variable>)` or `exit.val$`.

These modifications are only valid for Infor ERP LN 4GL session code. Infor ERP LN 3GL sessions are started by Workflow, but not controlled afterwards. Therefore the ending is outside the Workflow system.

To change the Kill command

The **kill** command is also a bshell command that is often used in the program code for Infor ERP LN sessions. The **Kill** command is used in two ways:

- To finish subsessions
- To stop the session itself

If you use a **Kill** command to stop the Infor ERP LN session itself, the process ID of that session is completely unregistered. The process ID indicates the object with which the Workflow Server communicates.

As a result, the Workflow Server loses the information on what object the last action was performed and, therefore, the Workflow Server cannot continue
the controlling process. This problem does not arise if you use the Kill command to finish subsessions.

To stop Infor ERP LN sessions, use the stop() command instead of the Kill command.

Change the Abort command

For the Abort command, the same explanations given for the exit command are applicable. Therefore, replace abort() commands with stop() commands in the session source and program code.

Change the Execute commands places

The execute (end.program) and the execute(abort.program) refer to choice sections in the Infor ERP LN session logic. Both expressions cause errors during the communication with the Workflow Engine if the commands are placed in particular program sections of the source code.

If the execute (end.program) command is placed in the choice.abort.program section, the Workflow Engine interprets this command as a successful finish although the program operating was canceled. This interpretation leads to a removal of the corresponding activity from the work list, which is incorrect.

If the execute (abort.program) command is placed in a choice.end.program section, the Workflow Engine interprets the phrase as a cancel.