software AG

Implementing E-form Support for BPM

Version 10.15

October 2022

software AG

Table of Contents

About this Guide

This guide provides information to software developers about how to integrate electronic forms (or e-forms) into business processes and user interfaces implemented using the webMethods BPMS infrastructure. This guide describes:

- General concepts and information about working with e-forms.

- How to configure and deploy an e-forms environment.

- How to implement and work with the supported e-form types.

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Narrowfont	Identifies service names and locations in the format *folder.subfolder.service*, APIs, Java classes, methods, properties.
Italic	Identifies:
	Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
Monospace font	Identifies:
	Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
|	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the | symbol.
[ ]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [ ] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Product Documentation

You can find the product documentation on our documentation website at https://documentation.softwareag.com.

In addition, you can also access the cloud product documentation via https://www.softwareag.cloud. Navigate to the desired product and then, depending on your solution, go to "Developer Center", "User Center" or "Documentation".

Product Training

You can find helpful product training material on our Learning Portal at https://knowledge.softwareag.com.

Tech Community

You can collaborate with Software AG experts on our Tech Community website at https://techcommunity.softwareag.com. From here you can, for example:

- Browse through our vast knowledge base.

- Ask questions and find answers in our discussion forums.

- Get the latest Software AG news and announcements.

- Explore our communities.

- Go to our public GitHub and Docker repositories at https://github.com/softwareag and https://hub.docker.com/publishers/softwareag and discover additional Software AG resources.

Product Support

Support for Software AG products is provided to licensed customers via our Empower Portal at https://empower.softwareag.com. Many services on this portal require that you have an account. If you do not yet have one, you can request it at https://empower.softwareag.com/register. Once you have an account, you can, for example:

- Download products, updates and fixes.

- Search the Knowledge Center for technical information and tips.

- Subscribe to early warnings and critical alerts.

- Open and update support incidents.

- Add product feature requests.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

1 Introduction to E-forms Integration

Overview

Electronic forms, or *e-forms,* are replacing paper processes in paper-intensive industries such as government, financial services, and education. Both small and large organizations often use dozens or even hundreds of forms to facilitate various aspects of their business processes—mortgage applications, employment applications, time sheets, invoices, order forms, and so on. Process participants use these e-forms as a primary user interface for entering and modifying process-related information.

The primary goal of webMethods electronic forms integration is to enable the use of e-forms in business processes and user interfaces implemented using the webMethods BPMS infrastructure. This guide describes:

■ General concepts and information about working with e-forms.

■ How to configure and deploy an e-forms environment.

■ How to implement and work with the supported e-form types.

You are responsible for obtaining and installing the supported e-form applications for creating, modifying, and managing your electronic forms; Software AG products do not provide this functionality.

For additional information about working with e-forms, see the following documentation:

■ *webMethods Service Development Help*

■ *webMethods BPM Process Development Help*

■ *webMethods BPM Task Development Help*

■ *webMethods CAF and OpenUI Development Help*

■ *webMethods Task Engine User's Guide*

General Concepts

The webMethods BPM suite supports e-forms created with the following applications:

■ Adobe LiveCycle

■ Microsoft Office InfoPath

Each of these applications enables you to create the following objects:

■ An e-form *template*—one or more files that describe the form structure, appearance, and behavior. A template carries no user-entered form data (although sample data may be present).

■ An e-form *instance*—a published version of the template, containing specific form data in a single file.

For more information about templates and instances, see the documentation for your e-form application.

Supported e-form templates can be imported into Software AG Designer where they are converted to a standard IS document type. You can use the IS document type to describe the input for a business process, as the business data for a task that expects an e-form document, or for any other supported IS document type use.

E-form templates and instances are also accessed at run time by the webMethods run-time components (Process Engine and Task Engine).

You can store your e-form templates in a content repository, in a file system, or even on a Web server. In all cases, Software AG Designer must have network access to the storage location at design time. Using a content repository for both templates and instances is the most straightforward approach.

During run time, the template and instance location must be available to the Task Engine if you have implemented the ability for a user to download an e-form from an e-form enabled task instance running in My webMethods. You can omit the requirement for access to the templates folder by attaching a copy of the template to the task application during design time; for more information, see the *webMethods BPM Task Development Help*.

If Integration Server is connected to a webMethods Broker at the time you create an IS document type from an e-form template, the resulting IS document type will be publishable to the webMethods Broker and will have an associated webMethods Broker document type.

To use an e-form instance as a run-time IS document type source that starts a business process, the following requirements apply:

■ Your e-form instances must be kept in a content repository folder.

■ You must not store instances and templates in the same folder.

■ You must create a separate instances folder for each e-form you work with.

■ A webMethods e-form environment must be configured and deployed to the Integration Server host where the process will run.

For more information about configuring and deploying an e-forms environment, see "Setting up an E-form Repository" on page 17.

For more information about working with IS document types, see the *webMethods Service Development Help*.

Suite Integration

E-form integration interacts with the following products in the webMethods product suite:

■ My webMethods Server—You can use My webMethods Server as your e-form repository for storing e-form templates and e-form instances through a JSR-170 interface.

■ My webMethods—You must use the Central Configurator interface in My webMethods to configure and deploy e-form environment definitions. If you implement e-form enabled task applications, users will interact with the tasks in My webMethods; if you have implemented the ability to do so, they can download and upload e-form documents from within the task user interface.

- Software AG Designer—You use Software AG Designer to import an e-form template and create a corresponding IS document type from it. You then use this IS document type to create e-form enabled business process models as well as e-form enabled task applications. You can also create e-form enabled CAF portlet applications in Designer.

- Integration Server/Process Engine—You use the Integration Server/ Process Engine as the run-time environment for webMethods business processes. The Integration Server host is the target for your deployed e-form environment definition, and the Process Engine monitors the e-form instance folder defined in the environment for the arrival of new e-form instances, which can be used to start a corresponding business process.

- webMethods Task Engine—If you have implemented e-form enabled task applications, the Task Engine manages these tasks and the e-form business data associated with the task. If you implement the ability to download an e-form from an e-form enabled task instance running in My webMethods, the Task Engine formats the e-form business data for downloading using the associated template, and, after uploading, modifies and saves the task business data with the modified data from the uploaded e-form.

About the E-form Repository

E-form integration requires communication with My webMethods Server as a JSR-170 repository containing an *instances folder* and, if desired, a *templates folder*. When configured as a JSR-170 repository, My webMethods Server must have network access to Software AG Designer and the webMethods run-time components (Integration Server, Process Engine, and Task Engine).

Templates and instances must be maintained in separate folders in the e-form repository. After you set up an e-form repository, you must configure one or more e-form environments in My webMethods and deploy these environments to the Integration Server host(s) where e-form enabled processes will be running.

By deploying an e-form environment, you are specifying the repository location of your e-form templates and e-form instances to the various components, so the templates and instances can be retrieved as needed during design time and run-time activities:

- During design time activities with Software AG Designer, the deployed e-form environment enables you to locate and select an e-form template for importation as an IS document type or as business data for a task.

- During run-time activity, the deployed environment identifies the e-form instance folder; the Process Engine continuously monitors this folder for the arrival of new instances of an e-form to start a new business process.

- Also during run time, if you implement the ability to download/upload an e-form from a task instance running in My webMethods, the Task Engine must have access to the e-form template location.

For more information about creating and deploying an e-form environment, see "Configuring and Deploying an E-form Environment" on page 23.

Other E-form Template Storage Options

In addition to storing an e-form template in a content repository, it is possible to use the following storage options:

- You can store e-form templates in a standard file system location, with the following requirements:

  - During design time, the location must accessible from the location where your Software AG Designer instance is running. Designer provides a file system browse feature so you can locate and select an e-form template in the file system for importation as an IS document type or as business data for a task.

  - During run time, the location must be accessible from your Task Engine host if you implement the ability to download an e-form from an e-form enabled task instance interface running in My webMethods. The Task Engine formats the e-form business data using the associated template for downloading.

    You can avoid this requirement by attaching a copy of the e-form template to an e-form enabled task application at design time. For more information, see the *webMethods BPM Task Development Help*.

- E-form templates can also be stored on a web server and made available with standard http access (the web server must be available to Designer at design time and to the Task Engine at run time).

  Note:
  You cannot use a file system location for the instances folder; the instances folder must always be located in My webMethods Server configured as a JSR-170 repository with network access to the run-time components.

General Usage

The most common scenarios for integrating e-forms into the webMethods product suite are:

- To start or provide data to a business process running in the Process Engine (just as with any IS document type).

- To provide business data to a task running in the Task Engine (just as with any IS document type).

- To provide users with download/upload access to forms and form data in a running task. This enables users to download an e-form, modify it offline, and then upload it back to the task later.

Implementation of all of the above capabilities are supported with wizards and drag-and-drop functionality within Software AG Designer.

It is also possible to implement e-form integration with generic composite application framework (CAF) portlets, although this must be done manually; for more information, see the *webMethods CAF and OpenUI Development Help*.

Using E-forms in a Business Process

This section provides an overview of the basic workflow for using e-forms with a business process; for specific information and procedures, see:

- *webMethods Service Development Help*

- *webMethods BPM Process Development Help*

- *webMethods BPM Task Development Help*

- *webMethods CAF and OpenUI Development Help*

- *webMethods Task Engine User's Guide*

≫ **To use e-forms in a business process, you follow these general procedures:**

1. Create a business process.

2. Create an IS document type from an e-form template with the Create New Document Type wizard.

   > **Note:**
   > The generated e-form document type is a fully compliant IS document type and can be used in the same ways as any other IS document type.

3. Drag the resulting IS document type onto the business process canvas to create a receive step, or drop it onto an existing receive step.

4. Specify the receive step e-form repository location.

5. If you want to integrate the e-form data with a task in the process, create a task step in one of the following ways (a process can contain multiple e-form enabled tasks):

   - Use the **CAF > Generate Task** menu option from the e-form document type, then:

     1. Configure the task as required (assignment, events, etc.)

     2. Use the Task UI Update wizard to add the e-form fields and upload/download buttons to the task interface.

   - Select an existing task, or implement an existing step in the process as a task, and then:

     1. Add the e-form IS document type as business data if not yet included.

     2. Configure the task as required (assignment, events, etc.)

     3. Use the Task UI Update wizard to add the e-form fields and upload/download buttons to the task interface.

6. Build and upload the process project.

7. Deliver an e-form instance to the repository instance folder (sourced from the correct e-form template).

The Process Engine monitors the specified instance folder in the repository; when a new instance of the e-form appears in the folder, the Process Engine automatically converts the e-form instance to an IS document based on the previously defined e-form template-to-IS document type relationship.

The receive step handles the e-form based IS document just as it would any other IS document, and the e-form data is added to the process pipeline.

If there is a task step in the process, when it is reached, a new task is queued and appears in the appropriate My webMethods inboxes, with e-form fields and e-form instance values in the Data view.

In My webMethods, the assigned task user can:

■   Work with e-form data in the task.

■   Download the e-form from the task for offline access (in actual e-form format).

■   Upload the modified e-form back into the task.

2 Setting up an E-form Repository

E-form Repository Considerations

This chapter describes how to set up My webMethods Server to serve as an e-form repository

You can find general information about e-form repositories in "Introduction to E-forms Integration" on page 9

Be aware of the following items as you set up your repository:

■ If you do not want to store e-form templates in your repository, you can store them in a file system or even on a web server. For more information, see "Other E-form Template Storage Options" on page 13. Each supported e-form template has a defined file name extension; for example, .xsn for InfoPath, and .xdp for LiveCycle. When you add e-form templates into the content repository templates folder, *you must ensure that you include those extensions with the name*.

■ You must create a separate e-form instance folder for each e-form document you work with. Each supported e-form instance has a defined file name extension; for example, .xml for InfoPath, and .xdp for LiveCycle. When you add e-form instances into the content repository instances folder, *you must ensure that you include those extensions with the name*.

■ Your internal data network and permissions configuration must allow Software AG Designer and the webMethods run-time components (Integration Server, Process Engine, and Task Engine) to be able to access the repository folders and read the templates and instances in them.

> **Note:**
> The Task Engine does not interact with instances directly; however, it is possible to implement a custom portlet within a task to upload e-form instances to the instances folder, in which case access to the instances folder would be required.

■ After you create your repository folders, you must configure an e-forms environment for each of your e-form repository folder locations, and deploy that environment to the Integration Server host(s) where your e-form enabled processes will be running, as described in "Configuring and Deploying an E-form Environment" on page 23.

■ If you plan to use WebDAV to connect to your repository folders, be sure to review the information in "WebDAV Considerations" on page 44.

Creating Repository Folders

If you use My webMethods Server as your repository, you are advised to create your e-form folders in the Public Folders section of the My webMethods Server taxonomy. Wherever you locate the templates and instances folders, you must determine and apply the folder permissions you want to use.

By default, Public Folders are configured to provide write access for all authentication users. You may want to restrict write access for the templates folder to template administrators only and provide read-only access for a wider set of users.

The instances folder will typically need write access for another set of users (generally not all users), without the ability to delete instances, which would be provided to instance administrators.

Aside from the two points mentioned above, there are no limitations to or requirements for the folder structure you implement. Creating aliases for these folders is not required; however, aliases are useful if it is likely that the location of the folders will change if the environment will be moved from testing to production. Aliases are also useful if you plan to create custom CAF portlets that interact with these folders.

When you create e-form folders in your content repository, keep the following points in mind:

■ Always locate your template folders and your instance folders in the same repository.

■ Do not mix templates and instances in the same folder. If you decide to use your repository to store both templates and instances, you must create separate folders for templates and instances.

■ Do not mix e-form documents in the same instances folder. You must create a separate instances folder for each e-form document you want to work with.

   For example, if you create an e-form document FormA, and another e-form document FormB, you must create separate instances folders for FormA and FormB.

■ Maintain different e-form template types in separate templates folders (this is recommended but not required).

Here is just one example of an e-form templates and instances folder arrangement for four e-forms, two created with LiveCycle and two created with InfoPath:

```
/Public Folders

    |_Instances_InfoPath

        |_InstancesA

            |_FormA.xml

        |_InstancesB

            |_FormB.xml

    |_Instance_LiveCycle

        |_InstancesC

            |_FormC.xdp

        |_InstancesD

            |_FormD.xdp

    |_Templates_InfoPath

        |_FormA.xsn

        |_FormB.xsn
```

```
      |_Templates_LiveCycle
```

```
          |_FormC.xdp
```

```
          |_FormD.xdp
```

Note:
The use of Adobe LiveCycle e-forms with task applications may require you to define a location for PDF versions of your LiveCycle forms. For more information, see "Adobe Design Time Considerations for Task Applications" on page 37 and "Making Adobe E-form Instances Available at Run Time" on page 40.

Repository Structure Considerations

Spend some time considering how you want to design your e-form repository hierarchy. For example, suppose you use two human resources (HR) e-form documents, and that you also use two purchasing department e-form documents. For business purposes, you might like to keep these forms stored in separate areas of your repository, subject to separate access permissions.

In your repository, you can create a repository path to a folder that contains a separate instances folder for each of your HR forms. For example:

```
/Public Folders/E-forms/HR_Instances/HRForm1/[InstanceFiles]
```

```
/Public Folders/E-forms/HR_Instances/HRForm2/[InstanceFiles]
```

You can also create a repository path to a folder that contains instances folders for your purchasing department forms. For example:

```
/Public Folders/E-forms/PUR_Instances/PURForm1/[InstanceFiles]
```

```
/Public Folders/E-forms/PUR_Instances/PURForm2/[InstanceFiles]
```

In this case, you would need to create four different e-form environments, two describing the path to each of the HR e-form instance folders, and two others describing the path to each of the purchasing e-form instance folders. The environments will be identical except for the path to the e-form folders, which will be different for each environment. You then deploy the four environments to the Integration Server host.

When you create your business process models, you will configure the receive step for each process to monitor the related e-form instances folder, which you select from the list of instances folders from the e-form environment you deployed to the Integration Server host.

Important:
When configuring your business process models for e-form use, do not configure more than one monitoring connection for each instances folder. In other words, you must maintain a one-to-one relationship between an instances folder and a single receive step.

When you plan your repository structure, you should also take e-form environment creation and deployment issues into account; for more information, see "Configuring and Deploying an E-form Environment" on page 23, especially "E-form Environment Considerations" on page 25.

Creating an E-form Repository Folder

The following procedure describes how to create new folders by logging in to My webMethods. You can also create new directories in the MWS Admin view in Software AG Designer by right-clicking on an existing folder and clicking **New > Folder**.

> **To create an e-form folder in My webMethods Server**

1. Log on to My webMethods as sysadmin (or as an appropriately authorized user).

2. Click the Public Folders link at the top of the page.

3. On the Tools menu, click **New > Folder.**

4. In the New Folder panel, type a folder name, such as Templates, or Instances.

5. Add a description of the folder if desired. If you are creating multiple folders for different instance types (LiveCycle and InfoPath), note the purpose in the description field.

6. Click **Create**.

7. Click the folder's Tools menu and modify the folder properties or permissions, if required.

8. Repeat steps 3 - 7 to create additional folders.

Managing Instance Files in the Repository

You are advised to develop a plan for managing the instance files that are delivered to your My webMethods Server repository. If you allow instance files to accumulate to significant numbers, excessive memory use will occur and My webMethods Server performance will degrade, resulting in increasingly slower response time.

Given the wide range of platform configurations on which My webMethods Server may be running, there is no specific limit on the number or size of e-form instance files. However, best practices recommend that you develop a manual or programmatic plan for regularly cleaning, purging, or archiving your instance folder files in My webMethods Server.

About Repository Connectivity

Your internal data network and permissions configuration must allow Software AG Designer and the webMethods run-time components (Integration Server, Process Engine, and Task Engine) to be able to access the repository folders and read the templates and instances in them.

If the repository server is not running, or network connectivity is interrupted:

■ You will not be able to import e-form templates stored in the repository into Software AG Designer.

- No new process model instances will be started in the Integration Server/Process Engine run-time environment.

- You will not be able to upload an e-form to a task instance running in My webMethods.

- You may not be able to download an e-form from a task instance running in My webMethods.

3 Configuring and Deploying an E-form Environment

Overview of E-form Environment Configuration

The e-form templates and instances you create are used by the webMethods product suite in the following ways:

- During design time, you use Software AG Designer to locate and select an e-form template for importation as an IS document type or as business data for a task.

- During run time, the Process Engine continuously monitors an instances folder for the arrival of new instances of an e-form to start a new business process.

- Also during run time, if you implement the ability to download an e-form from an e-form enabled task instance interface running in My webMethods, the Task Engine formats the e-form business data for downloading using the associated template, and, after uploading, modifies and saves the task business data with the modified data from the uploaded e-form.

When you are working with My webMethods Server configured as a JSR-170 content repository, you must inform these components of the correct e-form repository folders to use by defining an e-form environment in My webMethods, and then deploying that environment to the Integration Server(s) where e-form enabled processes will be running.

In other words, by deploying the e-form environment, you are specifying the repository location of your e-form templates and e-form instances to the various components, so they can be retrieved as needed during design-time and run-time activities.

You must maintain your e-form instances in My webMethods Server configured as a JSR-170 repository; however, you can locate your e-form templates elsewhere if you so choose. For more information, see "Other E-form Template Storage Options" on page 13.

About the Central Configurator Interface

This chapter provides specific information about configuring an e-form environment with the Central Configurator user interface. For additional general information about working with the Central Configurator interface, see the PDF publication *Configuring BAM*.

Determining Whether Central Configurator Information Is Installed

To define an e-form environment in My webMethods, you must have the Central Configurator user interface component installed with your My webMethods Server installation.

 To determine if the Central Configurator Interface is installed

1. In My webMethods: **Navigate > Applications > Administration > System-Wide > Environments > Define Environments**.

2. If this application path is not available, you must install the Central Configurator user interface before you can create an e-form environment.

Installing the Central Configurator Interface

≫ **To install the Central Configurator Interface**

1. Shut down all webMethods products.

2. Start the Software AG Installer.

3. Enter your credentials and specify the installation directory of your current installation of My webMethods Server.

4. On the component selection panel, select **My webMethods User Interfaces > Central Configuration**.

5. Complete the installation.

6. Start My webMethods Server and verify that you can navigate to the Central Configurator page as described in the previous procedure.

E-form Environment Considerations

A webMethods e-form environment describes the repository location of your e-form instances folders and, if desired, your e-form templates folder. It also defines the Integration Server host(s) where the environment is to be deployed. If you have multiple repositories, or multiple locations within a single repository, you will need to create and deploy an e-form environment for each.

You should plan to spend time designing your e-form repository structure and environments before you begin creating and deploying them. For more information about planning your e-form repository structure, see "Creating Repository Folders" on page 18.

Obviously, your deployment architecture can become quite extensive if you have a large number of e-form documents and business processes. The situation can become even more complex if you have multiple repositories or are working in a clustered or multiple Integration Server situation. You are advised to develop detailed deployment plans in advance of any major implementations. For further information about deployment with clustered or multiple Integration Servers, see "Deployment Considerations" on page 29.

Configuring an E-form Environment

You can create and deploy multiple e-form environments. For example, if your organization uses multiple paths to e-form documents in your repository, you must create and deploy an e-form environment for each one.

Similarly, if you have multiple repositories for your e-forms, you can create and deploy one or more environments for each repository location. For example, if you have different e-form repositories for different business units, you can create a separate environment for each repository.

Configuring an e-form environment consists of the following general actions:

- Define a repository server by specifying the physical connection information for the repository (also referred to as a *logical server* in Central Configurator), as well as the path to the e-form folders within the repository.

- Define one or more hosts where an Integration Server and Process Engine are running. This defines the target location(s) for deploying the environment.

- Map the repository server to the Integration Server host(s).

- Map the endpoint information for the host(s) by specifying a transport type and port for the host system(s).

- Verify the configuration.

≫ **To create an e-form environment**

1. Log on to My webMethods using an account that has permissions for configuring an e-form environment (the Administrator account has these permissions by default).

2. In My webMethods: **Navigate > Applications > Administration > System-Wide > Environments > Define Environments**.

3. In the Environments window, click **Add Environment**.

4. In the Environment Information window:

   a. Type an environment name, such as "InfoPath E-form Environment."

   b. Type a description for the environment, such as "Contains InfoPath E-form Templates and Instances folders."

5. Click **Save**.

6. In the Environments window, click the link created by the new environment in the **Environment Name** column.

7. On the **Design Servers** tab of the Environment Configuration window, click **Add**.

8. On the Add Logical Servers dialog box, click **E-forms Server** in the **Select Logical Servers** list. This specifies that you are creating an e-form environment.

9. Click **OK**.

10. Click the **Configure Servers** tab.

11. If it is not already expanded, expand the E-forms Server entry in the **Configuration** column.

12. Click the E-forms Configuration link.

13. Define the repository connection information by typing the following:

   ■ **Friendly Name**—this is configured by default as a concatenation of the repository server host name and server port. However, you can enter any text value (spaces are not allowed). This value is used to display a user-friendly name in Software AG Designer for the repository server that is hosting the e-form folders. This field has a character limit of 256.

   > **Note:**
   > If you deploy more than one environment to the same Integration Server host, the Friendly Name must be unique for each environment.

   ■ **Server Host**—the host name or IP address of the repository server. Enter only the host name or IP address; URL descriptors are not used.

   ■ **Server Port**—the e-form port number for the repository server. If you are using My webMethods Server as your repository, accept the default value of 10999. You can specify a different port number as described in "Changing the My webMethods Server Repository Port or Name" on page 33.

   ■ **Server Path**—the path within the server to the location of the e-form folders. If you are using My webMethods Server as your repository, accept the default value of mwsRepository. This field has a character limit of 1024. You can specify a different path name as described in "Changing the My webMethods Server Repository Port or Name" on page 33.

   ■ **Server User**—the name of an existing user account with permission to access the contents of the e-form folders. For My webMethods Server, the default value is Administrator. This field has a character limit of 256.

   ■ **Server Password**—the password for the specified user account. This field has a character limit of 256.

   ■ **Server Listening Path**—the path to the e-form folder that will contain the instances that will be used to start a new business process. For example, if you are using My webMethods Server as your repository, with an Instances folder in Public Folders, the path would be /Public Folders/Instances. This field has a character limit of 1024.

   > **Note:**
   > If you are creating an environment that defines only the location of your e-form templates, leave this value empty. If you do so, use caution with the environment. As it does not have an instance folder defined, it must not be used with a process model. Consider defining a friendly name such as, "Environment1_Templates_Only."

   ■ **Server Template Path**—the path to the e-form folder that contains the templates that will be used by Software AG Designer. For example, if you are using My webMethods Server as your repository, with a Templates folder in Public Folders, the path would be /Public Folders/Templates. This field has a character limit of 1024.

   > **Note:**

> If you are storing your templates in a different repository from your e-form instances (or in a file system), leave this value empty. If you are storing your templates in a different repository you must create a separate environment to define the location of your templates.

14. Click **Save**.

15. Click the **Define Hosts** tab. On this tab, you define one or more Integration Server hosts. When you deploy this environment, it will be deployed to the host(s) defined here.

16. Click **Add Host**.

17. On the Add/Edit Host dialog box:

    ■ Type a display name for the Integration Server host.

    ■ Type the host name or IP address of the Integration Server host.

18. Click **OK**. If you want to add additional Integration Server hosts, repeat steps 16 and 17.

19. Click the **Map Servers** tab.

20. Click the Edit icon in the **ACTIONS** column to map an individual logical server, or click **Map All**. This maps your repository server to the Integration Server host(s).

21. Click the **Map Endpoints** tab. The default values are already configured on this tab, and the tab is automatically marked as configured. Do not change the http transport type.

    Note:
    Accept the default port value of 15006 for hosts where you have only one Integration Server running on the specified host (unless you have a pre-existing port conflict). If you have two or more Integration Servers running on the same host hardware, you must take care to specify a separate port number for each Integration Server instance on that host. To change the e-form listener port definition, see "Changing Host Port Numbers" on page 32.

22. Click the **Map DB Pools** tab. There are no values to configure on this tab, and the tab is automatically marked as configured.

23. Click the **Validate** tab. If the configuration report indicates that you do not have a valid configuration, make changes to the environment configuration as needed and return to the **Validate** tab to re-validate the configuration.

    Note:
    It is possible to obtain a valid configuration report but still have a configuration that will not work correctly. For example, a typographical error in a folder path or user account name will still be accepted as valid, although it will not work correctly when deployed.

24. Click **Finish** to return to the Define Environments page.

Deploying an E-form Environment

You must deploy your e-form environment to the Integration Server host(s) where you will be running e-form enabled business processes. This enables Software AG Designer and the various run-time components (Integration Server, Process Engine, and Task Engine) to know where the e-form folders are located.

Deployment Considerations

Some forethought and planning must be applied if you working with e-form environments in a multiple or clustered Integration Server installation.

In these situations, exercise caution when deploying environments from the Central Configurator interface. If you deploy an e-form environment to more than one Integration Server/Process Engine host, you create a situation where you can start multiple (duplicate) business processes when an e-form instance arrives in an instances folder. Take care to avoid this situation.

In addition, when configuring your business process models for e-form use, do not configure more than one monitoring connection for each instances folder. In other words, you must maintain a one-to-one relationship between an instances folder and a single receive step.

Consider the following conditions:

■ *Where are my e-form enabled process models running?* When you upload an e-form enabled process model to an Integration Server/Process Engine host, the Process Engine monitors the e-form instance folder specified within the process for instances of a specifically named e-form document. If you upload the same process model to two or more Integration Server/Process Engine hosts that have the same e-form environment, a new business process instance will start on each host when a new instance of the named e-form document arrives in the monitored instances folder. Furthermore, you will have created multiple listener connections to each instances folder, which must be avoided.

■ *What e-form documents and folders am I monitoring?* When you define an e-form enabled process model, you specify an instances folder to monitor and a specific e-form document name to look for in the folder. Consider this example:

   ■ You create Process1 to monitor the Instances folder in your repository for arrival of E-formDocumentA, and upload Process1 to Integration Server/Process Engine host Alpha.

   ■ You create Process2 to monitor the same Instances folder in your repository for arrival of E-formDocumentB, and upload Process2 to Integration Server/Process Engine host Beta.

   ■ You deploy your e-form environment to both hosts Alpha and Beta. Both hosts monitor the same instances folder, but because the Process Engine on each host is monitoring for a separate e-form document, new process instances will run separately on each host, as expected.

   However, this implementation violates two requirements: 1) you will have created multiple listener connections to the Instances folder, which must be avoided, and 2) you have placed two separate e-form documents in the same instances folder, which must also be avoided.

In this situation, the correct approach would be:

- Create two separate instances folders, one for E-formDocumentA and one for E-formDocumentB.

- Configure two e-form environments, each one describing the path to only one of the instances folders.

- Deploy the E-formDocumentA environment to host Alpha where Process1 will run, and the E-formDocumentB environment to host Beta, where Process2 will run.

  This solution satisfies the requirements of "one listener connection per folder" and "one e-form document per folder," ensuring proper operation.

My webMethods Server Cluster Considerations

If you are operating My webMethods Server in a clustered environment, some considerations apply to the implementation of e-form support. In the case of a My webMethods Server cluster, do not connect to the Integration Server/Process Engine host through a load balancer because RMI (the method used to communicate) generally does not work in this configuration. Instead, configure a direct connection between a single My webMethods Server node and the Integration Server/Process Engine host.

Prerequisites

Prior to deploying an e-form environment, the following conditions must be met:

- The e-form environment must pass the Central Configurator validation step and a check mark must appear in the Ready to Deploy column in the Environments window.

- The targeted Integration Server must be running.

Deploying the Environment

≫ To deploy an e-form environment

1. Log on to My webMethods using an account that has permissions for deploying an e-form environments (the Administrator account has these permissions by default).

2. In My webMethods: **Navigate > Applications > Administration > System-Wide > Environments > Define Environments**.

3. In the Environments window, click the Deploy icon ➡ for the environment.

4. On the Deployment Results tab, click **Deploy All**. This action deploys the e-form environment to all of the Integration Server hosts defined in the environment.

5. Results of the deployment appear in the text box on the Deployment Results tab. If the deployment is unsuccessful

   a. Note the messages in the results text.

   b. Return to the Define Environments page.

   c. Make corrections to the environment as needed.

   d. Repeat steps 3 and 4.

Updating an E-form Environment

After you deploy an e-form environment, it is possible that some elements of the environment may change, such as port numbers or folder locations. In this case, you can enter the changes to the environment and then deploy the updates to the target hosts.

≫ **To update an e-form environment**

1. Log on to My webMethods using an account that has permissions for deploying an e-form environments (the Administrator account has these permissions by default).

2. In My webMethods: **Navigate > Applications > Administration > System-Wide > Environments > Define Environments**.

3. Make the necessary changes to your environment, as described in "Configuring an E-form Environment" on page 25.

4. Click **Finish** when you have entered all of your changes.

5. In the Environments window, click the Deploy icon ➡ for the environment.

   > **Important:**
   > The following step specifies that you click **Deploy Updates**. *Do not click* **Deploy All**. If you click **Deploy All**, you will create a new environment on the target hosts. Also note that because the Friendly Name is the unique key for an environment, if you change it and then deploy the environment to the same Integration Server host, a new environment will be created on the host regardless of the deployment method (**Deploy All** or **Deploy Updates**).

6. On the Deployment Results tab, click **Deploy Updates**. This action deploys updates in the e-form environment to all of the Integration Server hosts defined in the environment.

7. Results of the deployment appear in the text box on the Deployment Results tab. If the deployment is unsuccessful:

    a.   Note the messages in the results text.

    b.   Return to the Define Environments page.

    c.   Make corrections to the environment as needed.

    d.   Repeat steps 3 and 4.

Changing Host Port Numbers

When you define an Integration Server/ Process Engine host as part of an e-form environment, a default port of 15006 is specified. If you have a conflict with this port number, you can specify a different port number.

Note:
Do not change the default http transport type.

≫ **To specify a different host port number**

1. Log on to the Integration Server Administrator.

2. In Integration Server Administrator: **Security > Ports**.

3. If it is unused, delete or disable the entry for port 15006 as required by your environment. If you are using 15006 for other purposes, it can be left enabled.

4. Click the Add Port link and select a port type of webMethods/HTTP.

5. Click **Submit**.

6. In Regular HTTP Listener Configuration:

    a.   Click **Enable**.

    b.   Type the new port number.

    c.   Set the Package Name to WmPRT.

    d.   Click **Save Changes**.

7. In the Port List, click the Edit link in the **Access Mode** column for the newly created port.

8. Determine if the following services are present in the Allow List:

    ■   wm.prt.eforms:update

- wm.server:connect

- wm.server:disconnect

- wm.server:ping

9. If any of these services are not present, click the Add Folders and Services to Allow List link.

10. Select the appropriate ACL to display the missing services.

11. Select the missing service and click **Append Selected**.

12. Click **Save Additions**.

> **Note:**
> Be sure to verify that the port number in your e-form environment host definition matches this new port.

Changing the My webMethods Server Repository Port or Name

By default, the port number for accessing My webMethods Server as a JSR-170 repository is 10999, and the repository path name is mwsRepository.

≫ **To assign a different port number or path name**

1. Log on to My webMethods as a system administrator (sysadmin).

2. In My webMethods: **Administration Dashboard > Configuration > CAF Application Runtime Configuration**.

3. On the CAF Application Runtime Configuration page, locate the Administration application.

> **Tip:**
> Sort the **Application** column so that the Administration application appears at or near the top of the list.

4. Click the Administration application.

5. In the Configure Application panel, expand the **Web Application** entry if is not already expanded; click **Environment Entries**.

6. Do either or both of the following:

- For the rmiPort value, type a new port number.

- For the rmiRepositoryName, type a new path name.

> **Note:**

The following step will stop and restart the application.

7. Click **Apply**.

Important:

Be sure to modify any existing e-form environments that refer to the repository port number or path name that you have changed, as described in "Configuring an E-form Environment" on page 25.

4 Implementing E-forms

Adobe LiveCycle Implementation Considerations

The webMethods product suite supports interactive Adobe forms that comply with XML Forms Architecture (XFA). Interactive XML forms can be deployed as PDF or as an XML Data Package (XDP). An XDP file is an XML file that packages a PDF file in XML, along with XML form and template data.

The webMethods product suite requires that the form elements and data be presented in XDP format to satisfy both its design time and run-time requirements.

Adobe Forms come in two categories:

■ XFA-compliant, XML-based forms created, modified, and managed with Adobe LiveCycle Designer. These e-forms are the primary implementation for use with the webMethods product suite.

■ Acrobat forms created, modified, and managed with Adobe Acrobat. These forms must be imported into Adobe LiveCycle Designer before they can be used with webMethods product suite. For more information, see "Using Traditional PDF Forms Created with Adobe Acrobat" on page 42.

Creating XFA-Compliant Forms with LiveCycle Designer

Adobe LiveCycle Designer is a design tool for creating rich interactive e-forms that comply with XFA (XML Forms Architecture, sometimes referred to as XFA forms). You can design e-form templates and save them in PDF and XDP format. The target format of the forms depends upon who the end users are and how they will interact with the document and submit their data.

■ E-forms saved and distributed in the PDF format are worked upon by end users using the Adobe Reader software.

■ The XDP format provides greatly enhanced capabilities for rendering and submitting user data, and requires the Adobe Forms Server for its rendering and data submission.

The webMethods product suite requires e-form definition in XDP format. The typical steps involved in creating the design-time XDP from a PDF document are as follows:

1. Form creation—Create the e-form in LiveCycle Designer and save it as PDF (Static or Dynamic).

2. Enable Usage Rights—To be able to save and export form data, you must set the appropriate usage rights for the PDF file using Adobe Reader Extension. Note that usage rights may also be enabled using Acrobat Professional software.

3. Preview and Export—Open up the rights-enabled PDF form using Adobe Reader and then enter sample data into all the form fields. This step is particularly important because the exported data file is used to infer the XML schema of the form fields. The quality of the input data dictates how well the form fields are represented in the schema and it is advised that conforming data be entered for each of the form fields.

Additionally, care should be taken to ensure that the values match the data-type and format expected from the end users. For example, area code, phone number, and date fields may allow both formatted and non-formatted data; in this case, during preview, you are advised to enter the data in the format that is most likely to be used by the end-users. Be sure to enter data for non-required fields as well.

4. Save and export in XDP format—Finally, review the filled-in form and make sure that valid data is entered for all editable fields. Save as "XML Data package (xdp)" and export the form data as an XML file (the default extension of the exported file is .xdp). If you plan to use the e-form with a task application so the task user can upload and download e-forms, save the file in PDF format as well.

You can now place the new e-form template (the exported .xdp file from step 4 above) in the repository templates folder, or in other location as described in "General Concepts" on page 10. From there, the template can be imported into Software AG Designer as an IS document type or as business data for a task.

Limitations to Using XDP Files

There are some limitations to creating an IS document type using an XDP file. Software AG Designer examines the form data within the XDP and derives an XML schema from it. This inferred schema is then used to generate the IS document type.

An XML file is restricted in that it does not contain specific data type and formatting information for its elements, so the resulting document type is also less restrictive in terms of allowed data values. There are two ways to mitigate risks involved when creating IS document type using XDP:

■ Include sample data that best represents the formatting and type restrictions for each of the form fields. Also, make sure that data is supplied even for non-required form fields to eliminate ambiguity when schema elements are derived.

■ The user can optionally review the generated IS document type and edit document fields manually.

For more information on importing e-form templates and working with IS document types, see the *webMethods Service Development Help*.

Adobe Design Time Considerations for Task Applications

If you plan to configure a task application so the task user can upload and download Adobe LiveCycle e-forms, it is important to keep the following points in mind.

The XDP data file is used at design time to create the JCR template provider for the task download function. During run time, at the time of download, the XDP data file requires an associated PDF file so the e-form can be successfully opened in a PDF reader. You can save the template in PDF format at creation time, as noted in "Creating XFA-Compliant Forms with LiveCycle Designer" on page 36. You must store the PDF in a location accessible to Software AG Designer and the run-time components (Integration Server, Process Engine, and Task Engine).

The XDP stores the location of its associated PDF file as an href attribute inside the XDP. That href attribute must be a fully-resolvable URL; for example:

■ http://

■ ftp://

■ Absolute path: C:\FolderA\FolderB\File.pdf or /forms/folderA/folderB/File.pdf

■ Relative path: ../FolderA/FolderB/File.pdf.

Otherwise, the download will fail at run time.

The value of the href is set at export time (although you can manually edit the href after completing the export), and is derived based on the target location of the XDP, relative to the location of the PDF.

If an XDP file is moved from its original location, then the relative link to its associated PDF will be broken. In this case, the XDP file will not open in a PDF reader (in most cases, the PDF reader will prompt the user to browse for the PDF) and a JCR provider cannot be created.

Defining the Path Between and an XDP and a PDF File

There are two methods for ensuring the proper definition of a path to the PDF associated with an XDP file. The following explanations assume the files are located in My webMethods Server.

Manually Editing the Path

> **To manually edit the path to the PDF**

1. Define the template folder location somewhere in My webMethods Server (that is, the location where both the PDF and the exported XDP will reside). For example, this location can be a folder named "Templates" under Public Folders in My webMethods Server: /Public Folders/Templates/.

2. Place the PDF file in this location, and export the XDP file into the same location.

3. Open the XDP file in a text editor and modify the href value. At this point, the href will contain only the name of the PDF file; for example, href="filename.pdf". Change the value as follows:

```
New Value: href =
http://ServerName:8585/wm_mws_config/repository/mws/Public%20Folders/

Templates/filename.pdf
```

4. Save the file.

Mapping the Path

≫ **To map the path to the PDF**

1. Define the template folder location somewhere in My webMethods Server (that is, the location where both the PDF and the exported XDP will reside). For example, this location can be a folder named "Templates" under Public Folders in My webMethods Server: /Public Folders/Templates/.

2. Connect to this folder with WebDAV or with a utility that enables you to map to an FTP or WebDAV server.

   > **Note:**
   > The Microsoft WebDAV implementation (Map a Network Drive) works as expected when you open an existing e-form template, modify it, and then save it to the original location. However, when attempting to save a newly created file to a WebDAV location using the "Save As" function, if you observe that Microsoft WebDAV fails to complete this task, you may want to investigate other WebDAV implementations or network mapping utilities to avoid this problem.

3. After mapping the /Public Folders/Templates/ folder so that it appears as a local drive, export the XDP file and save it directly into the mapped drive.

In this case, the XDP automatically registers a link to its associated PDF file using an absolute path at the time of export; no manual intervention is necessary.

It is important to note that in this case, the PDF does not reside in My webMethods Server, but can be located completely independently of the XDP location (as long as the PDF location is accessible to Software AG Designer and the run-time components (Integration Server, Process Engine, and Task Engine).

Creating an XDP Form without an Associated PDF

The form author can choose to save a newly created form in XDP format without having to create a PDF. Such forms typically have a button that a user can click to submit their data in XDP format.

After the form design is finalized, the author can preview the document within LiveCycle Designer.

- In the preview mode, the form author enters sample data for every field in the form.

- The form author clicks the submit button and submits the sample data to the specified folder location.

- This submitted data is captured as an XDP file and saved into a location that is accessible to Software AG Designer.

Using an XML Schema as a Template Source

During form creation, it is possible to use an XML schema (as an .xsd file) as a template source. The form fields are bound to the XML elements at design time. If the form is designed entirely from an XML schema then it is possible to use the XML schema. There are some distinct advantages of using this method:

- It eliminates the need to provide the design time XDP file with the sample form data, which is used to infer a schema. At design time, when using the e-form import wizard to create an IS document type within Software AG Designer, the user selects the XML schema file (.xsd) that was used to create the form instead of the .xdp file.

- When an .xsd file is used to create an IS document type, the resulting document type fully encapsulates content and type restrictions for each of its elements per the schema.

Making Adobe E-form Instances Available at Run Time

To start a business process at run time, the webMethods product suite expects an XDP file that contains the user submitted data to appear in the monitored instances folder.

> **Important:**
> Adobe LiveCycle e-form instances placed in a monitored e-form instances folder *must* have a file name extension of .xdp.

When an e-form is ready for distribution, (either PDF or XDP), the form data can be submitted as described in the following two scenarios:

1st Scenario: PDF E-forms

1. Deploy the PDF e-form and make it available to the end users.

2. Users access the e-form using Acrobat or Reader software and fill in the form data.

3. After the form is filled in, the user must export the form data in XDP format. The XDP file serves up as the run-time artifact that must be copied into the instances folder to trigger a business process.

4. The user may optionally save the file with the data local to their environment.

2nd Scenario: XDP E-forms

1. Deploy the e-form to the Forms Server for general user access.

2. Customers access it via the web, download it, fill it in, and click the e-form's submit button.

3. The filled-in data is captured in XDP format and copied into the instances folder to trigger a business process.

> **Note:**
> As described above, the run-time XDP file is an XML file that consists of form data and a reference to the original PDF document. At run time, when a user downloads an e-form instance, the href tag will link the form to its original PDF document.

Using LiveCycle E-forms with Adobe Reader

For proper operation with LiveCycle e-forms, you are advised to configure your PDF reader application to display the PDF file in the reader application itself, and not within a browser (that is, as a plug-in). The following procedures are for Adobe Reader; other PDF reader applications have not been tested with the webMethods product suite e-form implementation.

> **To configure the Adobe Reader application**

1. Close all browsers.

2. Open Adobe Reader.

3. In Adobe Reader, **Edit > Preferences > Internet**.

4. Clear the **Display PDF in browser** check box.

5. Click **OK**.

6. Close Adobe Reader.

Behavior When Downloading an E-form from a Task

The user:

1. Logs into My webMethods.

2. Locates and open the task.

3. Clicks the download button.

Adobe Reader starts separately (that is, as a standalone application) and displays the e-form PDF. At this point, the user can save the file locally in PDF format, or export the data in XDP format. The file must be in XDP to be uploaded back to the task.

> **Note:**
> If you are using Mozilla Firefox, you may see a different behavior, depending on the version of Adobe Reader installed. With Adobe Reader 8.x.x, clicking the download button displays the File Save As dialog box, giving you the option to either open the file using Adobe Reader or to save the file locally as an .xdp. If you select the 'open with reader' option, the Adobe Reader application opens and displays the PDF.

Using Traditional PDF Forms Created with Adobe Acrobat

Forms that are created using the older Acrobat suite of products are not XFA complaint. Users who need to leverage such pre-existing PDF forms may be able to do so by importing these documents into LiveCycle Designer; however, depending upon the complexity and the version involved, some cleanup might be required.

For more details related to importing PDF forms into LiveCycle, refer to the LiveCycle documentation. After the document is imported into LiveCycle and the form design is finalized, then the e-form can be distributed as an XFA-compliant form.

Using Digital Signatures with Adobe LiveCycle E-forms

Digitally signing LiveCycle e-forms has two aspects:

- Document signatures that secure the look and feel of the form.

- Data signatures that secure form data at submission time.

The suggested approach to integrate signed LiveCycle forms with the webMethods product suite is to use document signatures.

Microsoft Office InfoPath Implementation Considerations

Microsoft Office InfoPath is an integrated application for developing and publishing e-forms from within the Microsoft Office suite of products. InfoPath e-forms are XML based and can be saved as XML files and published as XSN files. The webMethods product suite imports only XSN files for use as e-form templates.

Making InfoPath E-form Instances Available at Run Time

To start a business process at run time, the webMethods product suite expects an .xml file that contains the user submitted data to appear in the monitored instances folder.

> **Important:**
> Microsoft Office InfoPath e-form instances placed in a monitored e-form instances folder *must* have a file name extension of .xml.

Working with InfoPath Files in Non-Windows Environments

InfoPath is a Windows-only application. If your organization is working in a Windows-only environment, this should pose no problems. However, if portions of your webMethods product suite are running on other operating systems, such as UNIX or Linux, you may face some limitations.

For example, it should be possible to trigger a business process running on a UNIX system with an InfoPath e-form that arrives in the instances folder of your repository, assuming you have

correctly imported the e-form template, created an IS document type, and configured the business process to use it.

However, if a user working on a non-Windows operating system downloads an InfoPath e-form from a task, it will not be possible for the user to actually work with the e-form in that environment, as the InfoPath application can only run in a Windows environment.

Therefore, if you plan to enable users to download InfoPath e-forms, work with them in their local environment, and then upload them back to a task, you must ensure that those users log on to My webMethods from a Windows operating system.

■ Software AG Designer and the run-time components (Integration Server, Process Engine, and Task Engine) can be implemented on non-Windows operating systems even when used with InfoPath e-forms.

> **Important:**
> If you are running your installation of My webMethods Server/Task Engine on a UNIX or Linux platform, you must install version 1.2 or later of the `cabextract` utility into `/usr/bin` (or create a link to it within `/usr/bin`) on your My webMethods Server system to enable proper operation between the Task Engine and InfoPath e-forms. For further information about this utility, see http://www.cabextract.org.uk/.

■ InfoPath e-form templates and instances can be stored in My webMethods Server configured as a JSR-170 repository, running in a non-Windows operating system.

Other Solutions for Non-Windows Environments

If you expect that some of your users will be working on non-Windows computers and those users will be downloading and interacting with InfoPath e-forms, consider developing a process that will translate the e-form data in the pipeline into an Adobe form. That form can then be downloaded from a task onto a non-Windows platform, worked with in Adobe Reader, and then uploaded back into a task in the process.

Field Limitations

You might encounter the following issues when working with Microsoft Office InfoPath:

■ IS document types generated from an InfoPath XSD may contain field names that webMethods Broker will not propagate to non IS-clients.

■ InfoPath template fields intended to be mandatory are imported into an IS document type with the fields shown as optional. This applies to InfoPath template fields with the "Cannot be blank" property selected. When InfoPath renders the form into an XML Schema, these fields are shown as optional, which is how they are treated when they are imported into an IS document type.

This is caused by the way InfoPath handles attributes. Attribute elements are optional by default; in this case, the "use" attribute for the field in question should be set to "required." If this value is not specified, then the default value is "optional" and it is imported as such.

For more information, you can refer to these sources:

http://www.w3schools.com/Schema/el_attribute.asp

http://www.infopathdev.com/blogs/greg/archive/2004/09/16/The-Difference-Between-Optional-and-Not-Required.aspx

Using Digital Signatures with Microsoft InfoPath E-forms

When designing an InfoPath e-form, there are two generic actions involved when creating e-forms enabled for digital signatures:

■ You can enable digital signatures for the entire form or for specific parts of the form. When users sign the form, the digital signature locks the data so that it cannot be changed, and stores the view with the digital signature.

■ You can optionally configure a prompt to the user to sign the form upon submittal if the user attempts to submit the form without a digital signature (this feature exists only for signatures that sign the entire form). If you do not enable this prompt, it is possible for a user to save the form without signing it.

During design time, when an InfoPath e-form enabled for digital signatures is imported as an IS document type, the signature field is set to "required." If an instance of this e-form is submitted at run time without a digital signature, a run-time exception will appear in the Integration Server error log, indicating that the form data "does not conform to the Publishable Document Type."

When an InfoPath template associated with signature-enabled form is imported as an IS document type, you are advised to examine the properties of the signature field in the document. The signature field is typically set to be a required field. If you want to accept e-forms that are enabled for digital signature but are not digitally signed, you must manually change the field type in the IS document type to "optional" to avoid errors at run time.

WebDAV Considerations

Because the e-form environment typically spans multiple network locations, WebDAV can be used to simplify transferring e-form files from one network location to another.

■ The Microsoft WebDAV implementation (Map a Network Drive) works as expected when you open an existing e-form template, modify it, and then save it to the original location. However, when attempting to save a newly created file to a WebDAV location using the "Save As" function, if you observe that Microsoft WebDAV fails to complete this task, you may want to investigate other WebDAV implementations or network mapping utilities to avoid this problem.

■ When setting up a WebDAV connection or network mapping to a My webMethods Server folder such as /Public Folders/Templates/, use the sysadmin user account (or an appropriately authorized user account) when specifying credentials.

5 Synchronizing E-forms

Synchronizing E-forms and IS Document Types

Because e-form templates are converted into IS document types, you must be aware that any modification to either the e-form template or to the IS document type will result in mis-matched (or out of sync) documents.

■ After an e-form is imported and an IS document type is created from it, any modifications to the content or structure of the IS document type will cause it to be out of sync with the e-form template from which it was created. When an instance of the e-form template is received, it will no longer match the IS document type, and therefore will be unusable by the Process Engine.

■ The same situation will occur if you change the e-form template, and instances arrive in the repository from the modified e-form. The instances will no longer match the IS document type created from the initial template.

In both cases, you must:

1. Delete the existing IS document type.

   Note:
   The renaming action required in the following step is necessary only if the IS document type and its associated schema are used in more than one place. The file name of the e-form template is used to set the schema domain for the associated schema, and a unique file name will yield a unique schema domain. Integration Server maintains a central schema registry, so if a schema for a particular document type is used anywhere in the Integration Server, that definition will take precedence in all locations, regardless of subsequent imports. The only way to get the new definition to take precedence is to rename the imported e-form template, or to ensure that the old one is completely deleted from the Integration Server. Finding the old schema definition may not be easy because the original schema may be used in another package or folder.

2. Change the name of the e-form template you have modified; for example, if you have modified an e-form template named `my_template.xdp`, add a revision number, such as `my_template_r1.xdp`.

3. Use the New Document Type wizard to recreate the IS document type from the latest version of the e-form template.

If there are changes to the IS document type that will affect your process model (such as data structure changes), additional procedures are required, as follows:

Note:
Changes to field attributes in the e-form template do not require the following actions.

You must:

■ Delete the existing process model receive step that uses the IS document type.

■ Drag the recreated IS document type to the process canvas to create a new receive step in process model and configure the step accordingly.

■ Map fields from the receive step to the pipeline as needed.

■ Build and upload the process model.

For more information about working with IS document types, see the *webMethods Service Development Help*.

Maintaining E-form Templates and IS Document Types

After you import an e-form template and create an IS document type with it, you must maintain that e-form template in its original import location for as long as the resulting IS document is used by the webMethods product suite.

When the e-form and its resulting IS document type is no longer used by the webMethods product suite, you can safely delete or remove the e-form template from its import location.

Similarly, do not delete or remove the IS document type created when you import an e-form template until that IS document type is no longer used by the webMethods product suite. For more information about working with IS document types, see the *webMethods Service Development Help*.

About Monitoring E-form Listeners

When you configure a receive step in a business process model to use an IS document type created from an e-form instance, you specify two pieces of information:

■ The name of the e-form being used with the receive step.

■ The location of the instances folder where new instances of the e-form will be delivered.

When the process model is generated, the Process Engine creates a listener component that monitors the named instances folder for the arrival of a new instance of the named e-form. That listener component is enabled when the process model is enabled.

When the listener detects the arrival of a new instance, a flow service begins the process of converting the e-form into the correct IS document type, according to the relationship defined when the e-form template was initially imported. The resulting IS document type is then passed to the receive step.

If the listener component is not running, no new processes instances will be started even if e-form instances are arriving in the monitored folder.

Be aware of the following key points:

■ Do not configure two or more process models to monitor the same instances folder. There must be a one-to-one relationship between an instance folder and a single listener component.

■ If the repository server on which the monitored listener folder is located stops running, or if network connectivity to the repository server is lost, the listener component will stop running and must be restarted manually from the Process Engine dashboard; for more information, see "Monitoring and Starting E-form Listeners" on page 48.

■ If you are using My webMethods Server as a repository, and an e-form instance is delivered into a monitored instances folder when My webMethods Server is not running (for example,

by WebDAV connection), the e-form instance will not trigger the monitoring process when the listener component is restarted.

Monitoring and Starting E-form Listeners

You can monitor the operational status of e-form listener components with the Process Engine Dashboard; you can start listener components from the Dashboard as well.

≫ **To monitor or start an e-form listener**

1. Log on to the where the monitoring processes are located.

2. Click **Packages > Management**.

3. Locate the WmPRT package and click the Home Page icon 🏠.

4. Select **Processes** in the Process Engine pane on the left.

5. In the left pane of the Process Engine > Processes screen, select the process model that you want to work with.

   Process Engine displays information about the versions of the selected process model in the right pane.

6. To view more information about a specific process model version, click the name of version; details are displayed in the right-hand pane.

7. The **E-form Content Repository Listeners** section lists listeners associated with the process model version.

   Note:
   E-form Content Repository Listeners are displayed only if listeners are configured for the process model version.

8. To view details about a listener, click the listener name.

9. Status icons indicate whether a listener is:

   ■ Running (🟢 status icon)

   ■ Stopped (🔴 status icon)

10. If a listener is stopped, click the Restart link to restart it. The Process Engine displays messages about the outcome of the restart at the top of the page.