IBM® Workplace Forms™ Server — API

Version 2.6.1

Installation and Setup Guide

G325-2590-00

Note

Before using this information and the product it supports, read the information in “Notices,” on page 35.

First Edition (September 2006)

This edition applies to version 2.6.1 of IBM Workplace Forms Server — API (product number L-DSED-6RFRFZ) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Introduction

Welcome to the IBM® Workplace Forms™ Server - API. This manual outlines the installation and configuration of the API, and provides instructions for redistributing the API with applications you develop.

The Workplace Forms Server - API is available for C, Java™, and COM. For more detailed information about the functions provided by the API, refer to the appropriate version of the *IBM Workplace Forms Server — API User's Manual.*

About This Guide

This guide is divided into two sections based on whether you are installing the API under Windows® or UNIX®. Each section contains complete information for installing, configuring, testing, and redistributing the API.

Refer to the Table of Contents at the beginning of this manual to find the appropriate sections for your platform.

Backwards Compatibility

IBM Workplace Forms Server - API version 2.6.0 supports XFDL version 5.0 and later. If a form of a version earlier than 5.0 is opened, the API will process it as a version 5.0 form, and will apply the version 5.0 rules to the form.

Furthermore, the API is compatible with extensions (.ifx files) compiled with the PureEdge-branded API of version 5.0 or later. Extensions compiled with earlier versions of the API will not work with this version of the API.

Note: If the API reads in a version 5.1 or earlier form with the UFL_SERVER_SPEED_FLAGS on, and then tries to verify a signature that covers a toolbar item with no options, the API will always return an invalid result regardless of the true state of the signature.

You can avoid this problem by designing your forms to include an empty custom option in the toolbar item.

Document Conventions

This guide uses "*xx*" or "*xxx*" in place of the two or three digit version number of the API. In particular, these placeholders appear when referring to file names, folders, and directories that contain the API's version number.

Similarly, when specifying path names, angle brackets <> act as placeholders to denote generic folder or directory names whose actual names may vary depending on your operating system or installation choices. In particular, the following placeholders appear throughout this guide:

• *<API Installation Folder>*	The folder in which you installed the API.
• *<WIN OS>*	The Windows folder. For example, C:\WINNT.
• *<WIN SYS>*	The Windows system folder. For example, C:\WINNT\system32.

- *<UNIX OS>* The name of the UNIX operating system you are using.

What is the Workplace Forms API?

The IBM Workplace Forms Server - Application Programmer Interface (API) is a collection of specialized functions that you can use to develop applications that interact with XFDL forms. The API provides a C and Java interface for Windows and UNIX computers, as well as a COM interface for Windows platforms.

The API is divided into two sections, reflecting two sets of functionality:

- **The Form Library —** The Form Library is a collection of functions that enables you to access and manipulate XFDL forms as structured data types. These functions provide your applications with a means of reading and writing forms, retrieving information contained in graphical form elements, and assigning information to these items.
- **The FCI Library —** The FCI library is a collection of specialized functions that enables you to develop your own functions. XFDL forms can then access these custom functions, thereby extending the capabilities of the Internet Commerce System, without requiring an upgrade to either your forms software or to the form description language (XFDL).

  Note: The FCI Library is not available when using the COM interface.

Skill Level Required To Use the API

Any moderately experienced programmer will find the Workplace Forms Server API convenient and easy to use. However, the skill level required to develop a particular application may be quite high. This document is intended for developers who have a working knowledge of:

- Extensible Forms Description Language (XFDL) and syntax. Refer to the Extensible Forms Description Language Specification for more information.
- Java or C programming language and syntax.
- The Java or C application development environment.
- Any COM-compliant programming language and development environment.

Platform Requirements

The API supports the following platforms:

- Windows 2000 Server sp4
- Windows 2003 Server Standard Edition
- AIX® version 5.1, 5.2, or 5.3 on an IBM pSeries® server.
- Sun Solaris 9 and 10 running on a 32-bit Sparc processor.
- Red Hat Enterprise Linux® 3.0 Enterprise Server, 3.0 Advanced Server, 4.0 Enterprise Server, or 4.0 Advanced Server on 32-bit Intel-based Server

For more specific system requirements, refer to the following website:

http://www.ibm.com/support/docview.wss?rs=2357&uid=swg27008289.

Using the API With Windows

This section explains the installation and configuration process.

Setting Up the API

To install and configure the API under Windows, you must complete a number of steps. The following table outlines those steps:

1.	Run the installation program.
2.	Review and update the configuration file.
3.	Set up your development environment to work with the API. • Under C • Under Java
4.	Run the sample applications to test your API installation.

Installing the API with Previous Versions of the Viewer or Designer

This version of the API is not compatible with version 6.0 or earlier of the PureEdge-branded Viewer or Designer. If you want to continue using a version of the Viewer or Designer prior to 6.1, you must copy some files *before* you install this version of the API:

1. Locate the following files in the *<WIN SYS>* folder:
   - pe_cc.dll
   - pe_java.dll
   - uwi_java.dll
2. If you have a version of the PureEdge-branded Viewer prior to 6.1 installed, copy those files to the following folders:
   - *<Viewer Installation Folder>*
   - *<Viewer Installation Folder>*/Modules/
3. If you have a version of the PureEdge-branded Designer prior to 6.1 installed, copy those files to the following folder:
   - *<Designer Installation Folder>*

You can now install the new version of the API without affecting the Viewer or Designer.

Installing the API

The API is bundled as a self-extracting zip file. To install the API:

Double-click PEAPI*xxx*_WINDOWS.EXE, located on the disk provided.

You have now installed the API for Windows. This includes components for both the C and Java development environments.

Note: The install location of the API has changed. The new path is: \Program Files\IBM\Workplace Forms\Server\2.6\API

The API installation also provides the installation files for Deployment Server and Webform Server. To install and configure Webform Server or Deployment Server, see their respective administration guides.

Location of Installed Components

The API installation program installs files to support both Java and C. The following table lists important API components and their locations:

Location	File Name	Description
<*API Installation Folder*>	licenses.txt	The license agreement.
<*API Installation Folder*>	readme.txt release_notes.txt	Text files containing last-minute information about the API.
<*API Installation Folder*>	redist.txt	Lists the only files that you are licensed to distribute with your applications.
<*API Installation Folder*>\ Lib\Java\	pe_api.jar	A JAR file for compiling applications that use the API.
<*API Installation Folder*>\ Lib\Java\	uwi_api.jar	A JAR file for compiling applications that require backward compatibility.
<*API Installation Folder*> \Lib\ msc32\	pe_cc.lib	Enables your C applications to link to the API.
<*API Installation Folder*>\ Include\	.h files	The API header files.
<*API Installation Folder*>\ Samples\	multiple files	A set of sample applications that use the API functions.
<*API Installation Folder*>\ redist\	multiple files	A collection of the files you will need to redistribute with your application.

Copy the System Files

The API installation copies all of the API files to your computer. However, before you can use the API you must copy some of those files to your Windows System folder, as follows:

1. Locate the following folder:

   ```
   <API Installation Folder>\redist\msc32\
   ```

2. Select all of the files in the folder and copy them to:

   ```
   <WIN SYS>
   ```

   Note that you do not need to copy the IBM sub-directory.

Register the COM DLL

If you plan to use the COM API, you must first register the pe_com.dll file with Windows. To do this:

1. Open a DOS shell or command prompt.
   - For example, under Windows 2000 select **Start** → **Programs** → **Accessories** → **Command Prompt**.
2. On the command line, type:

   ```
   regsvr32 <path to pe_com.dll>
   ```

- For example:

  ```
  regsvr32 C:\WINNT\system32\pe_com.dll
  ```

This will register the COM DLL and associated files.

Update the PureEdgeAPI.ini File

The PureEdgeAPI.ini file determines which applications use which version of the API. This ensures that older applications do not attempt to use newer APIs, which may not be compatible.

To update the PureEdgeAPI.ini file:

Go to the Windows folder:

- If no PureEdgeAPI.ini file exists, create the file with a text editor and add the following lines:

  ```
  [API]
  7.0.* = C:\Program Files\IBM\Workplace Forms\Server\xx\API\redist
     \msc32\PureEdge\xx
  ```
- If a PureEdgeAPI.ini file exists from a previous installation, update the file to include the following lines:

  ```
  [API]
  7.0.* = C:\Program Files\IBM\Workplace Forms\Server\xx\API\redist
     \msc32\PureEdge\xx
  ```

Configure the API

The API relies on a configuration file to set a number of properties. The configuration file, called prefs.config, is normally stored in the following locations:

```
c:\Documents and Settings\<username>\Application Data\PureEdge\
   API x.x\Prefs\prefs.config
c:\Documents and Settings\All Users\Application Data\PureEdge\
   API x.x\Prefs\prefs.config
```

When determining the configuration, the API first looks at the configuration file in the user specific folder then the file in the *All Users* folder. The API will read both configuration files, but any settings in the user specific folder will override settings in the *All Users* folder.

If you have not installed any other Workplace Forms products, you may need to create these configuration files and directories.

The configuration file uses tag value pairs to set the following properties:

Tag	Setting
overrideDefaultPathToNetscapeProfile	The path to your Netscape certificate store. If the path is not provided, the API will attempt to locate the certificate store automatically.
overrideDefaultPathToNetscapeSecurityLibraries	The path to your Netscape libraries. To determine this path, search for nss3.dll.
javaPath	The path to your Java virtual machine. If you do not provide a path, the API will attempt to locate the Java VM automatically. In most cases, automatic detection will succeed under Windows.

Tag	Setting
checkCRLDistribution Points	An **on** or **off** value. If **on**, the API will check CRL distribution points whenever verifying a digital certificate. This requires Internet access, and will slow the verification process. If **off**, the API will only use locally stored CRLs to verify certificates.
entrustProfile	The name of the Entrust profile the API should use when working with Entrust signatures.
javaVM	Sets the default Java Virtual Machine. Once set, the API will use that JVM if it is available, otherwise it will locate the latest JVM installed on the computer. Set this to the string that the JVM uses to register itself. For example, ″Sun VM <version>″. The version is guaranteed to include a major and minor number, and may include further information, such as a maintenance number, build number, and so on. Since it can be difficult to get an exact match, you can use the * wildcard in the version string. For example, you might use the following string: `Sun VM 1.4*` This will match any version beginning with 1.4, such as ″1.4.2_03 JDK″ or ″1.4 JDK″. When there are multiple matches, the match that will be selected is indeterminate, so it best to be specific. Additionally, the API always chooses the JDK over the JRE. For example, if you search for ″Sun VM 1.4*″, and you have ″Sun VM 1.4.2_03 JDK″ and ″Sun VM 1.4.2_03 JRE″ installed, the API will use the JDK. Note that this setting is for Windows only.

jvmOptions.*x*	Allows you to pass configuration options to the Java VM. Each option must have its own numbered entry as shown:
	`jvmOptions.1 = <config setting 1>`
	`jvmOptions.2 = <config setting 2>`
	By default, the API does not pass any configuration options.

javaIgnoreUnrecognized	An **on** or **off** value. If **on**, the Java VM will ignore unrecognized options used in the *javaOptions* setting. If **off**, the Java VM will report unrecognized options in the *javaOptions* setting. Default is **off**.

To configure the API, simply open the prefs.config file in a text editor and set the preferences accordingly. If you do not include a particular setting, the default is used instead. For example, your prefs.config file might look like this:

```
checkCRLDistributionPoints = off
entrustProfile = Server
```

In this case, the API will automatically detect both the location of the Netscape Profile and the location of the Java virtual machine. However, the API will not check CRL distribution points and will use the default configuration options for the Java VM. Finally, the API will use the ″Server″ profile with Entrust.

Setting Up the C Development Environment

The Windows version of the API supports 32-bit applications using a library and a dynamically linked library (DLL).

However, before you can use the API, you must first configure your development environment to work with it. To make this easier, IBM provides template files for Microsoft® Visual C++ (version 5.0 or later) that are configured to work with the API libraries.

To create a Visual C++ application using the API libraries, copy the appropriate project templates from the following location to your current working directory and begin building your application:

- The Form project templates are located in the folder:
  ```
  <API Installation Folder>\Samples\Msc32\Form\Template\
  ```
- The FCI project templates are located in the folder:
  ```
  <API Installation Folder>\Samples\Msc32\FCI\Template\
  ```

Note: If these project templates are not appropriate for your needs, refer to the C edition of the *IBM Workplace Forms Server - API User's Manual* for instructions on how to set up your own development environment.

Setting Up the Java Development Environment

To use the Java API you must:

- Configure your integrated development environment to use the API.
- Configure your computer to run applications that use the API.

To configure your integrated development environment to use the Java API:

Add the following files to your **CLASSPATH**:

```
<WIN SYS>\PureEdge\xx\java\classes\pe_api.jar
<WIN SYS>\PureEdge\xx\java\classes\uwi_api.jar
```

To configure your computer to run applications that use the API:

1. From the **Start** menu, select **Settings** → **Control Panel** → **System**.
2. On the **System Properties** sheet, click the **Advanced** tab.
3. Click **Environment Variables**.
4. Under **System Variables**, select the **CLASSPATH** variable and click **Edit**.
   - If the **CLASSPATH** variable does not exist in your list of environment settings, you must create it: click **New** and type **CLASSPATH** in the **Variable Name** field.
5. In the **Variable Value** field, enter the location of the following components:
   - pe_api.jar
   - pe_api_native.jar
   - uwi_api.jar
   - uwi_api_native.jar
   - current directory (represented by a period)

6. Click **OK**.
7. Click **OK**.

Setting Up the COM Interface

Before you can make calls to the COM API, you must include the IFS_COM_API.tlb library in your project. The steps to do this will vary depending on the programming language you are using.

For example, if you are using Microsoft Visual Basic, you can include the type library as follows:

1. From the **Project** menu, select **References**.
2. In the **References** dialog, select **InternetForms API**.
3. Click **OK**.

For specific information about including a type library in your project, refer to the documentation for your development environment.

Running the Sample Applications

The API includes the following sample applications:

- Calculate Age Sample
- Form Sample (C and Java only)
- FCI Sample (C and Java only)
- JSP Sample (Java only)

These applications are provided to demonstrate the use of many of the functions in the API, and are useful for testing your API installation and development environment.

Note: To view the forms provided with these sample applications, you must have a licensed or evaluation copy of IBM Workplace Forms Viewer installed on your computer.

The Calculate Age Sample Application

The Calculate Age sample application demonstrates the most commonly used functions in the API.

This application reads a form called CalculateAge.xfd into memory, and reads the values for the year, month, and day of birth from the form. It then writes those values into a series of hidden fields that are used by a compute to determine the person’s age. Once the values are written, the application writes the form as Output.xfd.

Running Calculate Age Under C

Compile the source code file calculateAge.c to produce an executable called calculateAge.exe. Then follow these steps to run the Calculate Age sample application:

1. Using the Viewer, open the form called CalculateAge.xfd located in the following folder:

```
<API Installation Folder>\Samples\Msc32\Form\Demo\Calculate_age\
```

2. Change the name and birth date if you want, then save and close the form.
3. Run CalculateAge.exe.
   - This creates a form called Output.xfd in the current folder.
4. Open the Output.xfd form, and check the result in the **Age** field.

Running Calculate Age Under Java

Follow these steps to set up and run the Calculate Age sample application under Java:

1. Using the Viewer, open the form called CalculateAge.xfd located in the folder:
   ```
   <API Installation Folder>\Samples\Java\Form\Demo\Calculate_age\
   ```
2. Change the name and birth date if you want, then save and close the form.
3. Using your Java compiler, compile the file called calculateAge.java found in the current folder.
   - This produces a new class file called calculateAge.class.
4. Run calculateAge.class.
   - This creates a form called Output.xfd in the current folder.
5. Open the Output.xfd form, and check the number in the **Age** field.

Running Calculate Age For COM

The Calculate Age sample for COM is written in Visual Basic and the sample project files were generated using Microsoft Visual Basic 6.0. To run the sample, follow these steps:

1. Using the Viewer, open the form called CalculateAge.xfd located in the folder:
   ```
   <API Installation Folder>\Samples\Msc32\COM\Demo\Calculate_age\
   ```
2. Change the name and birth date if you want, then save and close the form.
3. Load the Calculate Age project files in Microsoft Visual Studio and compile them.
   - This produces a new executable called CalculateAge.exe.
4. Run CalculateAge.exe.
   - This creates a form called Output.xfd in the current folder.
5. Open the Output.xfd form, and check the number in the **Age** field.

The Form Sample Application

The Form library sample application demonstrates more of the functions available in the API.

This application reads into memory an input form called FormSample.xfd. It then reads the current date and birth date values from the form and sets a formula on the **Age** field that automatically computes the user’s age. Next, the application determines the user’s gender based on the radio buttons in the form. It then replaces these buttons with an image representing the user’s gender. Finally, it duplicates the address field to create a mailing address field. When complete, the application saves the changes made to FormSample.xfd in a new form called Output.xfd.

Running the Form Sample Under C

Compile the source code file formSample.c to produce an executable called formSample.exe. Then follow these steps to set up and run the Form sample application:

1. Using the Viewer, open the form called FormSample.xfd located in the folder:

   ```
   <API Installation Folder>\Samples\Msc32\Form\Demo\Sample_application\
   ```
2. Notice the radio buttons used to indicate gender and the empty **Age** field. You may now close the Viewer.
3. Run the sample application formSample.exe.
   - This produces a new form called Output.xfd in the current folder.
4. Use the Viewer to compare the two forms and note the changes produced by the application.

Running the Form Sample Under Java

Follow these steps to set up and run the FCI sample application:

1. Using the Viewer, open the form called FormSample.xfd found in the folder:

   ```
   <API Installation Folder>\Samples\Java\Form\Demo\Sample_application
   ```
2. Notice the radio buttons used to indicate gender and the empty **Age** field. You may now close the Viewer.
3. Using your Java compiler, compile the file called formSample.java found in the current folder.
   - This produces a new class file called formSample.class.
4. Run the sample application formSample.class.
   - This produces a new form called Output.xfd in the current folder.
5. Use the Viewer to compare the two forms.

The FCI Sample Applications

The API includes two different samples that demonstrate the FCI, depending on whether you are working in C or Java.

Running the FCI Sample Under C

The FCI sample application for C uses the FCI methods to create a function called *multiply* that can be called from within any form. This function multiplies two values and returns the result.

To run the sample:

1. Compile the source code file fciExtension.c located in the folder:

   ```
   <API Installation Folder>\Samples\msc32\FCI\Demo\Multiply\
   ```
2. Move fciExtension.ifx into the Forms System global extensions folder:

   ```
   <API Installation Folder>\PureEdge\xx\system
   ```
3. Using the Viewer, open the form fciSample.xfd, located in the folder:

   ```
   <API Installation Folder>\Samples\msc32\FCI\Demo\Multiply
   ```
4. Fill out fciSample.xfd. The result appears in the **Result** field.

Running the FCI Sample Under Java

The FCI sample application for Java uses the FCI methods to create a function called **convertDate**. This function takes two arguments, the date and the locale, and converts the date into the specified country's language and format. The application displays the converted date as a value for the **Converted Date** label. In this sample, the JAR file containing the extension was embedded into the form.

To run the FCI sample application:

1. Open the form called FCISampleWithJar.xfd found in the folder:

   ```
   <API Installation Folder>\Samples\Java\FCI\Demo\Convert_date\
   ```
2. Enter a date in the **Date** field. Then, select a language and date conversion format.
3. Click the button **Click me to convert the date**.
   - The converted date appears on the form.

Note: For information on how to build JAR (Java archive) files, refer to the Java Edition of the *IBM Workplace Forms Server - API User's Manual*. For information on embedding JAR files into XFDL forms, refer to the online help for the Designer.

The JSP Sample Application

The JSP sample application is only available for Java. The sample demonstrates how to integrate XFDL code with JSP. This application consists of three files: jspget.jsp, getlabel.txt, and jsppost.jsp. To run the application, you need to place the above files into a J2EE compliant web server's JSP folder. For more information, refer to the Java edition of the *IBM Workplace Forms Server - API User's Manual* and to your web server's documentation.

Distributing Your Applications

If you plan to distribute applications that call the API, you must ensure the following:

- That your package includes the correct API files.
- That the PureEdgeAPI.ini file is set up correctly on each computer.
- That the prefs.config file is set up correctly on each computer.
- That signature support is set up correctly on each computer.
- That each computer has the correct environment variable settings.

Redistributing API Files

When you redistribute the API, you must copy the complete contents of the following folder, including the sub-folders, to the target's Windows System folder:

```
<API Installation Folder>\redist\msc32\
```

Setting Up the PureEdge.ini File

The PureEdgeAPI.ini file determines which applications use which version of the API. This ensures that older applications do not attempt to use newer APIs, which may not be compatible.

The PureEdgeAPI.ini file must be placed in the Windows folder. If no PureEdgeAPI.ini file exists on the target computer, you should install your own. If a PureEdgeAPI.ini file already exists, you should update the file to include an entry for your application. For more information about setting up the PureEdgeAPI.ini file for your application, refer to the *IFSInitialize* function in the *IBM Workplace Forms Server - API User's Manual.*

Setting Up the prefs.config File

The prefs.config file controls the configuration of the API. When you distribute an application that uses the API, you must also distribute a configuration file for that application. This file ensures that the API is configured properly to work with your application. (For more information about creating the configuration files, see ″Configure the API″.)

You can install the configuration file in any of the following locations:

```
c:\Documents and Settings\<username>\Application Data\PureEdge\
   <application> <version>\Prefs\prefs.config
c:\Documents and Settings\All Users\Application Data\PureEdge\
   <application> <version>\Prefs\prefs.config
c:\Documents and Settings\<username>\Application Data\PureEdge\
   API <version>\Prefs\prefs.config
c:\Documents and Settings\All Users\Application Data\PureEdge\
   API <version>\Prefs\prefs.config
```

These locations are listed in order of precedence, from highest to lowest. When loading the configuration information, the API first checks the lowest precedence location, then works its way to the highest. The configuration settings are loaded from each location, but settings in a higher precedence location override those in a lower precedence location.

In these paths, the <application> and <version> are determined by the parameters of the *IFSInitialize* function. You must call this function from your application to initialize the API, and as part of that you must supply a name and version for your application. Use the same name and version in your path.

For example, if you called the *IFSInitialize* function with a name of ″CustomApp″ and a version of ″1.0″, you would install the configuration file in the following location for *All Users*:

```
c:\Documents and Settings\All Users\Application Data\PureEdge\
   CustomApp 1.0\Prefs\prefs.config
```

For more information about the *IFSInitialize* function, refer to the *IBM Workplace Forms Server - API User's Manual*.

Setting the CLASSPATH

Before a computer can run applications that use the Java API or use Java extension files, you must set your computer's CLASSPATH environment variable to include the following files:

```
<WIN SYS>\PureEdge\xx\Java\Classes\pe_api.jar
<WIN SYS>\PureEdge\xx\Java\Classes\pe_api_native.jar
<WIN SYS>\PureEdge\xx\Java\Classes\uwi_api.jar
<WIN SYS>\PureEdge\xx\Java\Classes\uwi_api_native.jar
```

Uninstalling the API

To uninstall the API, use the **Add or Remove Programs** tool available in Windows' Control Panel. However, you must also review the installation instructions and manually delete the files that you moved to new locations. For example, when you installed the API, you copied a number of files from the <*API Installation Folder*>\redist\msc32\ directory to the <*WIN SYS*> directory. You will need to remove these files by hand.

Using the API With UNIX

This section explains the installation and configuration process.

Setting Up the API

To install and configure the API under UNIX, you must complete a number of steps. The following table outlines those steps:

1.	Install the API.
2.	Review and update the configuration file.
3.	Set up your development environment to work with the API. • Under C • Under Java
4.	Run the sample applications to test your API installation.

Installing the API

To install the API for UNIX, you must obtain the appropriate UNIX installation file and transfer it to your UNIX computer. You will find this compressed file on your API installation CD:

- **Solaris**: WFServer_*xxx*_API_Solaris.bin
- **Linux**: WFServer_*xxx*_API_Linux.bin
- **AIX**: WFServer_*xxx*_API_AIX.bin

Note: If transmitting or receiving installation files over an ftp server, ensure that the files are sent as binary and not ascii. If you get an error when running the file, it may be because the file is not of the proper format. If this happens, use the `dos2UNIX` command to reformat the file, as shown:

```
dos2UNIX <file to convert> <new filename>
```

To install the API:

1. Log in to your UNIX computer as root.
2. Transfer the appropriate installation file to your UNIX computer.
3. Run the installer by typing:
   ```
   ./ <installation file> -console
   ```
4. Select a language.
5. Page through the license agreement by pressing **Enter**.
6. Once you have read the license agreement, type 1 to accept the agreement, and then press Enter.
   - This produces a number of directories containing the API.
7. Switch to the directory called /usr/lib by typing:
   ```
   cd /usr/lib
   ```
8. Copy the redistributable files:
   ```
   cp -r -p <API Installation Folder>/redist/<Operating System>/* .
   ```

9. If you are using Solaris, copy the 8859-1%UCS-2BE.so file and the UCS-2BE%8859-1.so file from the installation directory to the /usr/lib/iconv directory.
10. Ensure that users other than root can access the files copied to the /usr/lib directory. The files need read permission and the directories need read and execute permission. The following commands set the necessary permissions:

    ```
    chmod -R a+r+x /usr/lib/PureEdge
    chmod a+r /usr/lib/libpe_cc.so /usr/lib/libpe_java.so
       /usr/lib/libuwi_java.so
    ```

11. If you are using Solaris, you must also set permissions for the Solaris files, as shown:

    ```
    chmod a+r /usr/lib/iconv/8859-1%UCS-2BE.so /usr/lib/iconv/UCS-2BE%8859-1.so
    ```

12. Create or update the PureEdgeAPI.ini file in the /etc directory. The file should contain an entry for the API as shown:

    ```
    [API]
    7.0.*=/usr/lib/PureEdge/70
    ```

    - If you did not install the API in the /usr/lib directory, use the appropriate directory in the above setting. For example, if you installed the API in /home/jdoe then you would use /home/jdoe/PureEdge/70.
13. If multiple users will be using the API, ensure all of them have read access to the PureEdgeAPI.ini file. For example, you can make the file readable by all users with the following command:

    ```
    chmod a+r /etc/PureEdgeAPI.ini
    ```

You must now configure the API.

Note: The API installation includes the Deployment Server and Webform Server installation files. To install and configure Webform Server or Deployment Server, see their respective administration guides.

Location of Installed Components

The API installation program installs files to support both Java and C. The following table lists important components of the API and their location:

Location	File Name	Description
<*API Installation Folder*>	licenses.txt	The license agreement.
<*API Installation Folder*>	readme.txt release_notes.txt	Text files containing last-minute information about the API.
<*API Installation Folder*>	redist.txt	Lists the only files that you are licensed to distribute with your applications.
<*API Installation Folder*> /Lib/Java/	pe_api.jar	A JAR file for compiling applications that use the API.
<*API Installation Folder*> /Lib/Java/	uwi_api.jar	A JAR file for compiling applications that require backward compatibility.
<*API Installation Folder*> /Include/	.h files	The API header files.
<*API Installation Folder*> /Samples/	multiple files	A set of sample applications that use the API functions.
<*API Installation Folder*> /redist/	multiple files	A collection of the files you will need to redistribute with your application.

Location	File Name	Description
<API Installation Folder> /utils/*<UNIX version>*/	PEVersion	A utility for determining the version of an API file.

Configuring the API

The API relies on a configuration file to set a number of properties. The configuration file, called prefs.config, is normally stored in the following location:

```
/home/<username>/.PureEdge/API x.x/prefs/prefs.config
```

For web server applications that are not run as a specific user, prefs.config is stored in:

```
/etc/PureEdge/API x.x/prefs/prefs.config
```

If you are using the API in conjunction with Webform Server, a prefs.config file may also be stored in:

```
/etc/PureEdge/Webform Server x.x/prefs/prefs.config
```

Note: When creating these directories, remember that paths which include spaces must be enclosed in quotation marks. Also, ensure you create permissions to access these directories using `chmod a+r`.

Once you have installed the API, you can create a configuration file in this location. The configuration file uses tag value pairs to set the following properties:

Tag	Setting
overrrideDefaultPathToNetscapeProfile	The path to your Netscape certificate store. If the path is not provided, the API will attempt to locate the certificate store automatically.

Tag	Setting

javaPath	The path to your Java virtual machine. This path is not automatically detected under UNIX.
	To locate the JVM, search for one of the following files:
	• libjvm.so (newer versions of Java)
	• libjava.so (older versions of Java)
	(Note that under AIX, the filenames will end in .a rather than .so.)
	For example, the following command would locate the libjvm.so file:
	`find /usr -name libjvm.so -print`
	Common locations under Solaris include:
	`/usr/java/jre/lib/sparc/client/libjava.so`
	`/usr/java/jre/lib/sparc/server/libjvm.so`
	`/usr/java/jre/lib/sparc/libjvm.so`
	A common location under AIX is:
	`/usr/java131/jre/bin/classic/libjvm.a`
	Typically, /usr/java is a symbolic link to the Java base directory. You can use symbolic links when specifying the path to the JVM.
	If you are using Java version 1.4 or later, there will be two versions of the JVM: one with HotSpot and one without. The HotSpot version typically offers better performance and reliability.

checkCRLDistribution Points	An **on** or **off** value. If **on**, the API will check CRL distribution points whenever verifying a digital certificate. This requires Internet access, and will slow the verification process. If **off**, the API will only use locally stored CRLs to verify certificates.
entrustProfile	The name of the Entrust profile the API should use when working with Entrust signatures.
javaOptions	A space delimited list of configuration options that will be passed to the Java VM when it is started by the API. By default, the API does not pass any configuration options.
javaIgnoreUnrecognized	An **on** or **off** value. If **on**, the Java VM will ignore unrecognized options used in the *javaOptions* setting. If **off**, the Java VM will report unrecognized options in the *javaOptions* setting. Default is **off**.

To configure the API, simply open the prefs.config file in a text editor and set the preferences accordingly. If you do not include a particular setting, the default is used instead. For example, your prefs.config file might look like this:

```
javaPath = /usr/java/jre/lib/sparc/libjava.so
checkCRLDistributionPoints = off
```

In this case, the API will automatically detect the location of the Netscape Profile and will use the Java VM in the */usr/java/jre/lib/sparc/* directory. However, the API will not check CRL distribution points, will not work with Entrust (since no profile is specified), and will not send any configuration options to the Java VM.

Note: To ensure that your applications run under the API, be sure to set the ***LD_LIBRARY_PATH*** or ***LIBPATH*** environment variables, as described in ″Setting Up the C Development Environment″ and ″Setting Up the Java Development Environment″ .

Setting Up the C Development Environment

To set up your computer to run applications that use the API:

- On a Linux or Solaris system, include the following directories in the **LD_LIBRARY_PATH** environment variable:

  ```
  /usr/lib/
  /usr/lib/PureEdge/xx/system/
  <path to directory containing the Netscape Security Libraries>
  ```

- On an AIX system, include the following directories in the **LIBPATH** environment variable:

  ```
  /lib
  /usr/lib
  /usr/lib/PureEdge/xx/system
  <path to JVM>/jre/bin/classic
  <path to JVM>/jre/bin
  <path to directory containing the Netscape Security Libraries>
  ```

  Note: To find the Netscape Security Libraries directory, search for the libnss3.so file.

Setting Up the Java Development Environment

To configure an integrated development environment to work with the Java API:

Include the following files in your **CLASSPATH** environment variable:

```
/usr/lib/PureEdge/xx/java/classes/pe_api.jar
/usr/lib/PureEdge/xx/java/classes/uwi_api.jar
```

To set up your computer to run applications that use the API:

1. Include the following files in your **CLASSPATH** environment variable:

   ```
   /usr/lib/PureEdge/xx/java/classes/pe_api.jar
   /usr/lib/PureEdge/xx/java/classes/pe_api_native.jar
   /usr/lib/PureEdge/xx/java/classes/uwi_api.jar
   /usr/lib/PureEdge/xx/java/classes/uwi_api_native.jar
   ```

2. On a Linux or Solaris system, include the following directory in the **LD_LIBRARY_PATH** environment variable:

   ```
   /usr/lib/:/usr/lib/PureEdge/xx/system/
   <path to directory containing the Netscape Security Libraries>
   ```

   On an AIX system, include the following directories in the **LIBPATH** environment variable:

   ```
   /lib
   /usr/lib
   /usr/lib/PureEdge/xx/system
   /<path to JVM>/jre/bin/classic
   /<path to JVM>/jre/bin
   <path to directory containing the Netscape Security Libraries>
   ```

   Note: To find the Netscape Security Libraries directory, search for the libnss3.so file.

Running the Sample Applications

The API includes the following sample applications:

- Calculate Age Sample
- Form Sample
- FCI Sample
- JSP Sample (Java only)

Except for the JSP sample, these applications are available for both C and Java. The applications are provided to demonstrate the use of many of the functions in the API, and are useful for testing your API installation and development environment.

Note: To view the forms provided with these sample applications, you must have a licensed or evaluation copy of IBM Workplace Forms Viewer installed on your computer.

The Calculate Age Sample Application

The Calculate Age sample application demonstrates the most commonly used functions in the API.

This application reads a form called calculateAge.xfd into memory, and reads the values for the year, month, and day of birth from the form. It then writes those values into a series of hidden fields that are used by a compute to determine the person's age. Once the values are written, the application writes the form as output.xfd.

Running Calculate Age Under C

To set up and run the Calculate Age sample under C:

1. Switch to the directory:
   ```
   <API Installation Folder>/samples/<UNIX OS>/form/demo/calculate_age/
   ```
2. To view XFDL forms you must use a computer running Windows. FTP the file calculateAge.xfd to a computer running Windows.
3. In Windows, use the Viewer to open calculateAge.xfd.
4. Change the name and birth date if you want, then save and close the form.
5. If you changed the values in the form, FTP the saved form back to the original directory on your UNIX computer.
6. Create the *Makefile* by typing **xmkmf** at the command prompt and press **Enter**.
7. At the command prompt, type **make** and press **Enter**.
   - This will build the application file (calculateAge).
8. To run the sample application, type **./calculateAge** and press **Enter**.
   - This will create a form called output.xfd.
9. FTP the file output.xfd to a computer running Windows.
10. In Windows, use the Viewer to open the output.xfd form and check the number in the **Age** field.

Running Calculate Age Under Java

To set up and run the Calculate Age sample under Java:

1. Switch to the directory:

   ```
   <API Installation Folder>/samples/java/form/demo/calculate_age/
   ```

2. To view XFDL forms you must use a computer running Windows. FTP the file calculateAge.xfd to a computer running Windows.
3. In Windows, use the Viewer to open calculateAge.xfd.
4. Change the name and birth date if you want, then save and close the form.
5. If you changed the values in the form, FTP the saved form back to the original directory on your UNIX computer.
6. Using your Java compiler, compile the file called calculateAge.java found in the current folder.
   - This produces a new class file called calculateAge.class.
7. Run calculateAge.class.
   - This will create a form called output.xfd.
8. FTP the output.xfd form to a computer running Windows.
9. In Windows, use the Viewer to open the output.xfd form and check the number in the **Age** field.

The Form Sample Application

The Form library sample application demonstrates more of the functions available in the API.

This application reads into memory an input form called formSample.xfd. It then reads the current date and birth date values from the form and sets a formula on the **Age** field that automatically computes the user's age. Next, the application determines the user's gender based on the radio buttons in the form. It then replaces these buttons with an image representing the user's gender. Finally, it duplicates the address field to create a mailing address field. When complete, the application saves the changes made to formSample.xfd in a new form called output.xfd.

Running the Form Sample Under C

To set up and run the Form sample under C:

1. Switch to the directory:

   ```
   <API Installation Folder>/samples/<UNIX OS>/demo/sample_application/
   ```

2. To view XFDL forms you must use a computer running Windows. FTP the file **formSample.xfd** to a computer running Windows.
3. In Windows, use the Viewer to open formSample.xfd.
4. Notice the radio buttons used to indicate gender and the empty **Age** field. You may now close the Viewer.
5. Create the *Makefile* by typing **xmkmf** at the command prompt and press **Enter**.
6. At the command prompt, type **make** and press **Enter**.
   - This will build the application file (formSample).
7. To run the sample application, type **./formSample** and press **Enter**.
   - This produces a new form called output.xfd in the current folder.
8. FTP the file output.xfd to a computer running Windows.
9. In Windows, use the Viewer to compare the two forms.

Running the Form Sample Under Java

To set up and run the Form sample under Java:

1. Switch to the directory:

   ```
   <API Installation Folder>/samples/java/form/demo/sample_application/
   ```

2. To view XFDL forms, you must use a computer running Windows. FTP the file **formSample.xfd** to a computer running Windows.
3. In Windows, use the Viewer to open formSample.xfd.
4. Notice the radio buttons used to indicate gender and the empty **Age** field. You may now close the Viewer.
5. Using your Java compiler, compile the file called formSample.java found in the current folder.
   - This produces a new file called formSample.class.
6. Run the sample application formSample.class.
   - This produces a new form called output.xfd in the current folder.
7. Use FTP to transfer output.xfd to your Windows computer, then compare the two forms in the Viewer to see the changes that were made by the sample application.

The FCI Sample Application

The API includes two different samples that demonstrate the FCI, depending on whether you are working in C or Java.

Running the FCI Sample Under C

The FCI sample application for C uses the FCI methods to create a function called *multiply* that can be called from within XFDL forms. This function multiplies two values and returns the result.

To set up and run the FCI sample under C:

1. Switch to the directory:

   ```
   <API Installation Folder>/samples/<UNIX OS>/fci/demo/multiply/
   ```

2. Create the *Makefile* by typing **xmkmf** at the command prompt and press **Enter**.
3. At the command prompt, type **make** and press **Enter**. This will build the application file and the extension file (fciExtension.ifx).
4. Move the file fciExtension.ifx to the /usr/lib/PureEdge/*xx*/extensions/ directory.
5. To run the sample application, type **./sampleap** and press **Enter**. The following line will appear:

   ```
   Multiplying 24 by 13 gives 312.000000
   ```

   - This indicates that the application executed successfully and multiplied the two numbers using the **multiply** function.

Running the FCI Sample Under Java

The FCI sample application for Java uses the FCI methods to create a function called *convertDate*. *convertDate* takes two arguments, the date and the locale, and converts the date into the specified country’s language and format. The application displays the converted date as a value for the Converted Date label.

Follow the steps below to set up and run the FCI sample application:

1. Switch to the folder:

   ```
   <API Installation Folder>/samples/java/fci/demo/convert_date/
   ```

2. To view XFDL forms, you must use a computer running Windows. FTP the file fciSampleWithJar.xfd to a computer running Windows.
3. In Windows, use the Viewer to open fciSampleWithJar.xfd.
4. Enter a date in the **Date** field. Then, select a language and date conversion format.
5. Click the **Click me to convert the date** button.
   - The converted date appears on the form.

Running the JSP Sample Application

The JSP sample application is only available for Java. The sample demonstrates how to integrate XFDL code with JSP. This application consists of three files: jspget.jsp, getlabel.txt, and jsppost.txt. To run the application, you need to place the above files into a J2EE compliant web server's JSP folder. For more information refer to the Java edition of the *IBM Workplace Forms Server - API User's Manual* and to your web server's documentation.

Distributing Your Applications

If you plan to distribute applications that call the API, you must ensure the following:

- That your package includes the correct API files.
- That the PureEdgeAPI.ini file is set up correctly on each computer.
- That the prefs.config file is set up correctly on each computer.
- That each computer has the correct environment variable settings.

Redistributing API Files

To redistribute the API, you must copy the complete contents of the following directory, including the sub-directories, to the target's /usr/lib/ directory:

```
<API Installation Folder>/redist/<UNIX Type>/
```

Where <*UNIX Type*> is one of: Linux, AIX, or SunOS.

Setting Up the PureEdgeAPI.ini File

The PureEdgeAPI.ini file is located in the following directory:

```
/etc
```

If no PureEdgeAPI.ini file exists, you should install your own. If a PureEdgeAPI.ini file already exists, you should update the file to include an entry for your application. For more information about setting up the PureEdgeAPI.ini file for your application, refer to the IFSInitialize function in the *IBM Workplace Forms Server - API User's Manual*.

Setting Up the prefs.config File

The prefs.config file controls the configuration of the API. When you redistribute the API with your appplication, you must also install an application specific configuration file on the target computers. You should set this file to configure the API properly for your application. For more information about creating the prefs files, see ″Configure the API″.

You must install the configuration file in the following locations:

```
/home/<username>/.PureEdge/<application> <version>/
   prefs/prefs.config
```

or

```
/etc/PureEdge/prefs/prefs.config
```

Both folders are checked for configuration information, and the settings in the user specific folder override the settings in the *All Users* folder. However, if you do not require user specific settings, you only need to install the configuration file in the *All Users* folder.

In these paths, the <application> and <version> are determined by the parameters of the *IFSInitialize* function. You must call this function from your application to initialize the API, and as part of that you must supply a name and version for your application. Use the same name and version in your path.

For example, if you called the *IFSInitialize* function with a name of ″CustomApp″ and a version of ″1.0″, you would install the configuration file in the following location:

```
/home/<username>/PureEdge/CustomApp 1.0/prefs/prefs.config
```

For more information about the *IFSInitialize* function, refer to the *IBM Workplace Forms Server - API User's Manual*.

Setting the Environmental Variables

Before a computer can run applications that use the API, you must set its **CLASSPATH** environment variable to include the following files:

```
/usr/lib/PureEdge/xx/java/classes/pe_api.jar
/usr/lib/PureEdge/xx/java/classes/pe_api_native.jar
/usr/lib/PureEdge/xx/java/classes/uwi_api.jar
/usr/lib/PureEdge/xx/java/classes/uwi_api_native.jar
```

Additionally, on a Linux or Solaris system, include the following directory in the **LD_LIBRARY_PATH** environment variable:

```
/usr/lib/:/usr/lib/PureEdge/xx/system/
```

On an AIX system, include the following directories in the **LIBPATH** environment variable:

```
/lib
/usr/lib
/usr/lib/PureEdge/xx/system
/<path to JVM>/jre/bin/classic
/<path to JVM>/jre/bin
```

Note: If you did not install the API in the /usr/lib directory, use the appropriate directory in the above settings. For example, if you installed the API in /home/jdoe then you would use /home/jdoe/PureEdge/xx/system.

Uninstalling the API

To remove the API from your UNIX system, first run the uninstall script. After this is complete, review the installation instructions and manually delete the files and directories that you moved from their original install locations to new locations.

Using the API with WebSphere Application Server

IBM WebSphere® Application Server (WAS) is a popular application deployment environment that is often used to run applications created with Workplace Forms Server — API. In many Workplace Forms products, WAS and the API are installed together. However, if you are installing the API into a pre-existing WAS environment, some additional configuration must be done.

To use the API with WAS:

1. Install Workplace Forms Server — API.
2. Set the WebSphere variables (see "Setting the WebSphere Variables").
3. Modify the PureEdgeAPI.ini file (see "Creating the PureEdgeAPI.ini File").
4. Set the Java process definitions (see "Setting the Java Process Definitions" on page 28).
5. Set the shared libraries (see "Setting the Shared Libraries" on page 28).
6. Set the server class loader (see "Setting the Server Class Loader" on page 29).

Setting the WebSphere Variables

1. In the **WebSphere Application Server Administration Console**, go to **Environment → WebSphere Variables | Manage WebSphere Variables**.
2. To set the scope, click **Browse Nodes** and select the node that contains the server you are using.
   - For example, **WebSphere_Portal** or **server1**.
3. Click **Apply**.
4. Click **New** to create an environment variable.
5. Under **Name**, type **WFS_API_DIR**.
6. Under **Value**, type the directory in which you installed the API files.
   - For example, `C:\Program Files\IBM\Workplace Forms\Server\2.6\API\redist\msc32\`
7. Repeat steps 4-5 to create another environment variable called **WFS_API_LIB_DIR** and point it to the API's Java libraries directory.
   - For example, ${WFS_API_DIR}/PureEdge/70/java/classes

Creating the PureEdgeAPI.ini File

Note: You may have already completed this task as part of the API install.

1. In a text editor, create a file called PureEdgeAPI.ini.
2. Include the following two lines:

   ```
   [API]
   *=<API Install Dir>\redist\msc32\PureEdge\xx
   ```

   Note that you should replace *<API Installation Dir>* with the actual path to the API installation directory and *xx* with the version number of the API.
   - For example: `*= C:\Program Files\IBM\Workplace Forms\Server\2.6\API\redist\msc32\PureEdge\70`

   For more information about this file, refer to the description of the *initialize* method in the *IBM Workplace Forms Server — API: Java User's Manual*.

3. Save the PureEdgeAPI.ini file to the <*WIN*> (Windows) or *etc* (UNIX) directory that contains the API files.

Setting the Java Process Definitions

1. Open one of the following:
   a. In WAS 6.0.2, open **Servers** → **Application servers** → **<*server_name*>** → **Java and Process Management** → **Process Definition** → **Environment Entries|Custom Properties**.
   b. In WAS 5.1.1, open **Servers** → **Application servers** → **<*server_name*>** → **Process Definition** → **Environment Entries**.
2. In this location, add or update the following properties:
   - On Windows:
     – Add a **PATH** property and point it to the API directories that contain .dll files.
       - For example, ${WFS_API_DIR};${WFS_API_DIR}/PureEdge/70/system
   - On AIX:
     – Add a **LIBPATH** property and point it to the API directories that contain .so files.
       - For example, ${WFS_API_DIR}:${WFS_API_DIR}/PureEdge/70/system
   - On Linux:
     – Add a **LD_LIBRARY_PATH** property and point it to the API directories that contain .so files.
       - For example, ${WFS_API_DIR}:${WFS_API_DIR}/PureEdge/70/system
   - On Solaris:
     – Add a **LD_LIBRARY_PATH** property and point it to the API directories that contain .so files.
       - For example, ${WFS_API_DIR}:${WFS_API_DIR}/PureEdge/70/system

Note: If the **LD_LIBRARY_PATH**, **LIBPATH**, or **PATH** properties have already been created, simply add the API directory paths to the existing properties. Remember to use the correct separator; Windows uses a semicolon while UNIX platforms uses a colon.

Setting the Shared Libraries

1. In the **WebSphere Application Server Administration Console**, go to **Environment** → **Shared Libraries**.
2. To set the scope, click **Browse Nodes** and select the node that contains your server.
3. Clear the **Server** field and click **Apply**.
4. Click **New** to create a shared library.
5. Under **Name**, type **WFS_API_LIB**.
6. Under **Classpath**, list the following files (one per line):
   - ${WFS_API_LIB_DIR}/pe_api.jar
   - ${WFS_API_LIB_DIR}/pe_api_native.jar
   - ${WFS_API_LIB_DIR}/uwi_api.jar
   - ${WFS_API_LIB_DIR}/uwi_api_native.jar
7. Click **OK**.

Setting the Server Class Loader

1. Open one of the following:
   a. In WAS 6.0.2, open **Servers** → **Application servers** → **<*server_name*>** → **Java and Process Management** → **Classloader**.
   b. In WAS 5.1.1, open **Servers** → **Application servers** → **<*server_name*>** → **Classloader**.
2. If there is already a classloader for your application server, go to step 5. Otherwise, go to step 3.
3. To create a new classloader, click **New**.
4. Under **Class loading mode**, select **Parent Last** and click **OK**.
5. Select the classloader for your application server and click **Libraries**.
6. Click **Add**.
7. Under **Library Name**, select **WFS_API_LIB**.
8. Click **OK**.
9. Save all changes to the master configuration.

Appendix: Additional Usage Notes

This appendix lists additional minor form and application design notes for Workplace Forms API:

- The Viewer and API cannot read forms that have a byte-order mark (BOM). Attempting to do so will produce the following error: *Unable to load form: Invalid document structure*.

  To prevent this problem: If you use a text editor to edit your forms, configure your editor so it does not add a BOM to UTF-8 documents. (See your text editor's documentation for details.)

  To remove a BOM from a form: Open the form in the Designer and save it.
- Workplace Forms products no longer support the Netscape *browser*, but do support digital certificates provided by the Netscape browser's NSS certificate store.
- If you are using a server application that notarizes signatures (that is, you are using the Authenticated Clickwrap or HMAC signature engine), when the user submits a signed form to the server, the CryptoAPI digital certificate may display a prompt on the server. Because the prompt is displayed on the server, where no person can respond to it, the server (or a thread on the server) may appear to hang.
- If you use Mozilla-based web browsers, you should set up your server modules to specify that the web browser clear its cache before it posts a form submission. Some versions of Mozilla-based browsers cache posts. As a result, if a post of a form is interrupted for some reason and you submit the form a second time, the browser sends the cached (and incomplete) first post.
- The COM interface does not correctly support output parameters when using VBScript. Because of this, you must set all output parameters to null.

  In cases in which you would normally use vbNull, you must instead use "1", which is the constant value of vbNull. If you do not use "1", the function you are calling will not work correctly.
- Do not use relative paths to start applications that use the API under AIX 5.1. If you do, the API will not initialize properly.
- When building an application that uses the API under UNIX, ensure that you link with the libpe_cc.so library.
- Under Solaris, you must link with the -Blocal linker option (or equivalent). This makes all of the symbols within the program local, so that loaded shared libraries cannot resolve against those symbols. If this linker flag is not given, and the application is using the Java support, one of the shared libraries loaded by the Java VM resolves against symbols defined in libpe_cc.so and causes a segmentation fault when that function is called.

Where to Find More Information

The *IBM Workplace Forms Server - API User's Manual* is available in the following versions:

- C Edition
- COM Edition
- Java Edition

Each manual introduces you to the features of the API, including an overview of the XFDL form memory structure. It also teaches you how to use the functions available in the API and provides many examples of their use in the appropriate programming language.

Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Office 4360
One Rogers Street
Cambridge, MA 02142
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

AIX
IBM
Workplace
Workplace Forms

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Program Number: 5724-N08

Printed in USA

G325-2590-00