Oracle® Fusion Middleware

Oracle WebCenter Forms Recognition Installation Guide

12c (12.2.1.3.0)

E93585-01

July 2018

Documentation for Oracle WebCenter Forms Recognition administrators that describes how to install Forms Recognition.

ORACLE®

Oracle Fusion Middleware Oracle WebCenter Forms Recognition Installation Guide, 12c (12.2.1.3.0)

E93585-01

Contents

1 About WebCenter Forms Recognition

Oracle WebCenter Forms Recognition is a learning-based intelligent document recognition (IDR) solution that can recognize, categorize and extract information from any type of document. WebCenter Forms Recognition uses intelligence - not templates - to effectively locate, extract, and link data to back-end systems and processes, to provide the industry's highest level of document recognition and data extraction. Together with Oracle WebCenter Enterprise Capture and Oracle WebCenter Content: Imaging, WebCenter Forms Recognition offers a pre-packaged, end-to-end invoice processing solution for Oracle's Financial Management applications to facilitate processing large volumes of invoices with high upfront data extraction accuracy to minimize the need for human intervention.

Within the WebCenter Forms Recognition suite, the Designer application enables you to define and customize the automatic processing of incoming documents, for example, which document classes are relevant in your enterprise as well as which information is to be extracted from the classified documents. All custom settings are saved in a Forms Recognition project file.

To process large volumes of documents, WebCenter Forms Recognition organizes documents into batches, which are defined in the Forms Recognition project file. The project files and stored settings are automatically forwarded to the WebCenter Forms Recognition Runtime Server for production processing.

The WebCenter Forms Recognition Runtime Server runs unattended as a server process in the background. Several mechanisms ensure that the system is stable, meaning that it can automatically recover from most error situations. Multiple instances of Forms Recognition Runtime Server can be started simultaneously in a network or on a single machine. These instances cooperate and allow for optimal load distribution, and high availability.

Batches that cannot be automatically processed in their entirety by the Forms Recognition Runtime Server are forwarded to the quality assurance application, Verifier, or Web Verifier, for manual correction.

The Web Verifier application module allows users to verify documents with no software installed on the client side. Web Verifier can be used via a supported browser on any client machine to verify documents. This requires installation and configuration of the project and batches on the database platform. From version 11.1.1 Oracle WebCenter Forms Recognition features a database platform for Forms Recognition applications. It is possible to store project and authentication information in the Forms Recognition database. This solution allows for central management of storage and backup and thus provides for easier security, better connectivity of your applications, and higher flexibility of your personnel.

2 System Requirements

2.1 Operating Systems

Using WebCenter Forms Recognition requires a complete and successful installation of the software on a Windows server or workstation running one of the following operating systems:

- Microsoft Windows Server 2008 R2 (IPv4 and IPv6)
- Microsoft Windows Server 2012 R2
- Microsoft Windows 7
- Microsoft Windows 10
- VMware ESX 4.1 is certified

Note: WebCenter Forms Recognition requires that .NET Framework 4.5.2 to be installed on the Forms Recognition server or workstation. This remains true even where a later version of the .NET Framework is installed.

2.2 WebCenter Forms Recognition Database

The WebCenter Forms Recognition database has been certified to run on the following database platforms:

- Oracle Database 11*g* R2
- Oracle Database 12c R1
- Microsoft SQL Server 2008 R2
- Microsoft SQL Server 2012 R2
- Microsoft SQL Server 2014 SP2

Note: The 32-bit Oracle Database Client is required on all workstations or servers where one or more WebCenter Forms Recognition components are installed, and where the Forms Recognition database resides on an Oracle Database.

2.3 Internet Information Services (IIS)

Using WebCenter Forms Recognition Web Verifier will require the following software applications installed on the Forms Recognition web server:

- .NET Framework 4.5.2
- Internet Information Server (IIS)

Note: Ensure the IIS installation includes the **ASP.NET** application development role service, and the **IIS Management Console** role service.

The Web Verifier client requires a supported browser. Refer to the *Oracle WebCenter Forms Recognition Certification Matrix* for full details of supported browsers and versions.

3 Pre-Installation of WebCenter Forms Recognition

Before installing WebCenter Forms Recognition, ensure that you perform all the pre-installation steps for a smooth installation.

Note: Oracle recommends engaging an appropriately trained implementation partner to ensure the successful installation of the WebCenter Forms Recognition software.

3.1 Before Installing WebCenter Forms Recognition

Before starting the installation, make sure that you have local administrator rights on the target machine. During the installation, a number of DLLs will be copied to the Windows system directory and registered with the operating system. The WebCenter Forms Recognition database will be created on the Oracle or SQL Server database servers. The install process requires administrative privileges and access to the Windows registry.

3.2 WebCenter Forms Recognition Database

WebCenter Forms Recognition version 12c (12.2.1.3.0) is able to store the following core information directly in the Forms Recognition database, instead of the file system:

- Documents
- Batches (jobs)
- Project references
- Web Verifier configuration
- Batch/document lock handling
- Users, groups, roles and relationships
- Application level user licensing

Prior to installation of WebCenter Forms Recognition, ensure that you perform the appropriate database preparation steps.

3.2.1 Oracle Database Checklist

WebCenter Forms Recognition will need the following items taken care of prior to the installation of the software:

1. Install the 32-bit Oracle Database Client libraries on all computes where you want to run Forms Recognition. This includes all Runtime Server instances, and all Verifier and Designer workstations.
2. Create a new Oracle instance for Forms Recognition.
3. Create a new user with a password.
4. Assign sufficient rights to the above user:
    a. Allow for increased growth of data.
    b. Allow for insertion, modification, and deletion of data.

c. Allow for table, views, etc. creation.

5. Administrative database accounts with rights to create, modify, and delete tables. Windows Authentication can be used if the user performing the installation has administrative rights to the database server.
6. A designated user database account that will be used by Forms Recognition to access the database, add, modify, and delete data. Windows Authentication can be used if the user performing the installation has the appropriate rights to the database server.

3.2.2 SQL Server Checklist

WebCenter Forms Recognition will need to create the following items prior to the installation of the software:

1. An administrative database account with rights to create, modify, and delete tables. Windows Authentication can be used if the user performing the installation has administrative rights to the database server.
2. A designated user database account that will be used by Forms Recognition to access the database, add, modify, and delete data. Windows Authentication can be used if the user performing the installation has the appropriate rights to the database server.

4 Installing WebCenter Forms Recognition

4.1 Software Installation

Important: As the Forms Recognition database resides in an Oracle Database, the 32-bit Oracle Database Client must be installed prior to installing WebCenter Forms Recognition. The 32-bit Oracle Database Client is required, even if the installation is being performed on a 64-bit version of Windows.

To install WebCenter Forms Recognition:

1. Browse to the installation folder and run **setup.exe**

   Note: English and German are the supported installation languages. The installer gets its language settings from the regional settings of the operating system. The installation defaults to English if a language other than English or German is detected

2. Click **Next** to continue
3. If .Net Framework is not installed, setup will prompt you to install it; select **Let the setup install .Net Framework Version 4.5.2 (Recommended)**.
4. Click **Next** to continue
5. Select the installation type: **Complete** or **Custom**

   Complete: Installs the complete WebCenter Forms Recognition application suite including all components and all OCR engines (FineReader 10, FineReader 11, Kadmos 5 and Cleqs Barcode). The default install location (..\Program Files (x86)\Oracle\WebCenter Forms Recognition) and the default program groups are used.

   Custom: Enables you to install only the components you will use, and to specify the install location and program group

6. For a complete installation choose **Complete** and click **Next** then continue to step 5 below

   For a custom installation choose **Custom** and click **Next** then continue to *Section 4.1.1: Selecting Custom Installation*.

7. Specify the desired **Program Folder**. This defines where the Forms Recognition menu shortcuts will be located within the Windows Start menu. The default location is **Start Programs Oracle WebCenter Forms Recognition**
8. Click **Next**
9. Review the Current Settings. You can change any settings, if required, by using the **Back** button
10. Click **Next** to begin the complete installation and install all Forms Recognition applications and all optional components
11. Continue to *Section 4.1.2: WebCenter Forms Recognition Database Setup*

4.1.1 Selecting Custom Installation

If you selected **Custom** on the **Setup Type** screen, you can specify the installation directory to use and select the features and other components to be installed.

For a custom installation:

1. First, specify the installation directory. Click the **Change…** button to choose the folder where Forms Recognition will be installed
2. Click **Next**
3. In **Feature Selection** dialog box, select the desired **Applications** by checking the box beside those to be installed
4. In the **OCR Engines** list, you can select the OCR engines to be installed

   Only components selected during the installation will be available. However, you can install additional components later. See *Section 4.9: Adding or Removing Forms Recognition Components*.

Optional Components: OCR Engines	
FineReader 10	Supporting Chinese/Korean/Japanese characters in addition to English, German, Italian, French, and Spanish. Converts paper-based or scanned images into editable text
FineReader 11	The FineReader11 engine is fully integrated and supports OCR of several additional languages. The FineReader 11 engine features a number of general improvements in the quality of OCR output relative to the FineReader 10 engine: ▪ Receipt Mode, ▪ Improved auto-orientation, ▪ Improved OCR of amounts with leading/trailing asterisks, ▪ Improved OCR of amount with leading dollar sign $.
Cleqs Barcode	Reads handwritten and machine-printed data and barcode information. It reads 18 types of barcodes
Kadmos 5.0	Used for handwriting recognition
QualitySoft	It supports both grayscale and color images, and recognizes 19 different types of barcodes.

5. Click **Next**
6. Specify the desired **Program Folder**. This defines where the Forms Recognition menu shortcuts will be located within the Windows Start menu. The default location is **Start Programs Oracle WebCenter Forms Recognition**
7. Click **Next**
8. Review the Current Settings. You can change any settings, if required, by using the **Back** button
9. Click **Next** to begin the installation and install the selected Forms Recognition applications and components
10. Continue to *Section 4.1.2: WebCenter Forms Recognition Database Setup*

4.1.2 WebCenter Forms Recognition Database Setup

After the selected applications and components have been installed, the **Database Setup Options** are displayed in the installation wizard.

If you want to complete the installation without performing the database setup (e.g. on an additional Runtime Server machine or a Verifier workstation), select the **Do Not Install Database** option and proceed to *Section 4.1.3: Completing the WebCenter Forms Recognition Setup*, otherwise:

1. Select the desired database server, **Oracle** or **SQL Server**, then click **Next**
2. Enter the database username and password created in *Section 3.2: WebCenter Forms Recognition Database* above, then click **Next**
3. Enter the **Database Server Name** in the appropriate format for the selected database server:

   Oracle Database: *<database server name>*:*<database port>*/*<instance name>*

   e.g. `databaseServer.us.oracle.com:1521/orcl`

   SQL Server: *<database server name>*
4. Click **Next**

   The installation wizard will search for the specified database server, connect to it and initialize the WebCenter Forms Recognition database.

   If you already have a WebCenter Forms Recognition database installed, this database will be overwritten by this installation process. In that case, a notification would be displayed to warn you of deletion:

   A WebCenter Forms Recognition Database has been detected. If you continue, the Database will be overwritten. It is strongly recommended that the existing Database be backed up before continuing.

   If you want to save your existing database, back it up before continuing. This also applies for the installation of a newer version on the same machine.

   If you want to keep your current installation but want to install Forms Recognition, you can keep your existing database by copying all of the core configuration files from the existing installation folder to the new setup folder.

4.1.3 Completing the WebCenter Forms Recognition Setup

The final part of the installation shows the **Performed Tasks** and confirms the components that have been installed.

1. Click **Next** to continue
2. Optionally select the **Create Desktop Shortcuts for Applications** checkbox
3. Click **Finish** to complete the installation

Note: **WebCenter Forms Recognition** 12c (12.2.1.3.0) supports the French language for the Verifier and Learnset Manager applications. To enable the French language, select the language on the **Formats** tab of your system's Regional and Language Options.

4.2 Manually Creating Database Objects (Post Install)

It is also possible to install the database manually. This can be due to corporate policies. In such an instance, perform the following steps to install and configure the database manually:

1. Launch Windows Explorer and navigate to the installation folder

   Navigate to *<installerFolderLocation>***\FirstPart\Database\CreationScripts**

   There are two folders, **Oracle** and **SQL Server**. Each folder contains database scripts to execute that will create the tables, views, indexes, and default data values.

2. Execute the following steps appropriate to your database platform

   For Oracle Database

   - Follow the steps outlined in *Section 3.2.1: Oracle Database Checklist* above
   - Using SQL*Plus or SQL Developer, log into the database schema where the Forms Recognition tables will be located
   - Execute the **BrwCreateDatabase.sql** script

   For SQL Server

   - Follow the steps outlined in *Section 3.2.2: SQL Server Checklist* above
   - Log into the database with Administrator rights
   - Create a new database
   - Execute the **BrwCreateDatabase.sql** script

3. Navigate to *<installerFolderLocation>***\FirstPart\Database\UpdateScripts**

   Again, you will find folders for the **Oracle** and **SQL Server** scripts.

4. Execute the following steps appropriate to your database platform

   ---

   Note: If you refrain from executing the steps outlined below, an error message is generated when executing the Update Scripts.

   For Oracle Database

   - Edit the **BRW_Upgrade_Database.sql** script and replace all instances of **TargetDBSchemaName** with the name of the database schema created in *Section 3.2.1: Oracle Database Checklist* above
   - Execute the **BRW_Upgrade_Database.sql** script against the Forms Recognition database schema

   For SQL Server

   - Edit the **BRW_Upgrade_Database.sql** script and replace all instances of **TargetDatabaseName** with the name of the database user created in *Section 3.2.2: SQL Server Checklist* above
   - Execute the **BRW_Upgrade_Database.sql** script

5. Check that the database tables have been created correctly and no errors were reported on execution of the database scripts

6. There are several configuration components that require modification. Navigate to the WebCenter Forms Recognition installation folder. By default this is located in **C:\Program Files (x86)\Oracle\WebCenter Forms Recognition**

7. Navigate to the **WebCenter Forms Recognition Web Server** folder and open the **Web.config** file in a text editor, such as Notepad

8. Search for the connection string in the file:

   ```
   <connectionStrings></connectionStrings>
   ```

9. Modify the connection string to connect to the database:

   For Oracle Database

   ```
   <connectionStrings>

   <add name="Entities"
   connectionString="metadata=res://*/Entity.ORAEntities.csdl|
   res://*/Entity.ORAEntities.ssdl|res://*/Entity.ORAEntities.msl;
   provider=EFOracleProvider; Provider Connection String='Data
   Source=OracleServerName;User Id=Oracle;Password=Oracle'"
   providerName="System.Data.EntityClient" />

   </connectionStrings>
   ```

   For SQL Server

   ```
   <connectionStrings>

   <add name="Entities"
   connectionString="metadata=res://*/Entity.Entities.csdl|
   res://*/Entity.Entities.ssdl|res://*/Entity.Entities.msl;
   provider=System.Data.SqlClient; provider connection
   string=&amp;quot;Data Source=DBINSTANCE\SQLEXPRESS; Initial
   Catalog=SQLServerDatabaseCatalog;Integrated Security=false;User
   ID=sqlServer;Password=sqlServer;MultipleActiveResultSets=True&amp;quot
   ;" providerName="System.Data.EntityClient" />

   </connectionStrings>
   ```

10. Navigate to the **WebCenter Forms Recognition\Bin\bin** folder
11. There are 6 other configuration files that require changing as with the web.config. These are:
    - DstDsr.exe.config
    - DstHost.exe.config
    - DstSlm.exe.config
    - Brainware.System.Project.exe.config
    - DstVer.exe.config
    - DstWkBrw.exe.config

    Open each one in Notepad to make the appropriate changes below.
12. Search for the connection string in the file and modify the connection string to connect to the database. Refer to Step 9 above for examples.

For Oracle Database Only

13. For Oracle Database, it is required to make one more addition to the .Net Framework installation for the Oracle connection string above to work:
    a. Navigate to the Windows folder using Windows Explorer
    b. Navigate to **Windows\Microsoft.NET\Framework\v4.0.30319\CONFIG**
    c. Open the **machine.config** file for editing and locate the **DbProviderfactories** tag

d. Add the line indicated below on a single line in the .config file. **Do not delete any existing data:**

```
<system.data>
   <DbProviderFactories>
         <add name="EF Oracle Data Provider"
         invariant="EFOracleProvider" description="EF Provider
         for Oracle testing"
         type="EFOracleProvider.EFOracleProviderFactory,
         EFOracleProvider, Version=1.0.0.0, Culture=neutral,
         PublicKeyToken=def642f226e0e59b"/>
   </DbProviderFactories>
</system.data>
```

4.3 Configuring the WebCenter Forms Recognition Database (Post Install)

After the installation of WebCenter Forms Recognition with database, there are additional configuration steps that are required.

1. Check the project file names. The project file name will be used to display the available project lists in Web Verifier
2. Review the list of users in the projects
    a. All usernames and passwords must be consistent throughout all project files
    b. Each user must have their own username and password – user IDs cannot be shared
3. Export the users from the project file into the Forms Recognition database. This will now make users available to access projects via the Web Verifier.
4. Log into the User table (known in Oracle Database as **USER_**) and for each user, add a Forename and Surname into the Forms Recognition database
5. Create a Runtime Server instance for the project, or import existing Runtime Server settings, and configure again the Forms Recognition database (creating a job and linking to the Forms Recognition database).

You are now ready to use WebCenter Forms Recognition with database.

For information on how to configure a project for a WebCenter Forms Recognition instance with database, or to migrate file system batches to the Forms Recognition database, see the *WebCenter Forms Recognition Runtime Server User's Guide*.

4.4 Checking the Installation

The installation was successful if WebCenter Forms Recognition runs without errors.

To check for the correct installation of components:

1. Launch **Start** → **Programs** → **Oracle** → **WebCenter Forms Recognition** → **Tools** → **Components Version Info.**
2. From the menu, select **View** → **Components General Info**. This displays a list of installed components:
3. Check the list for:

- Completeness of components
- Homogeneity of build numbers
- Installation paths

4. All components (Cro*.dll, Cdr*.dll and Bwe*.dll) should have been registered automatically during the installation. If some of them seem to be missing, try to register them manually by executing the following Windows batch files:
   - **RegCro.bat** located in <*Installation Folder*>**\Components\Cairo**
   - **RegCdr.bat** located in <*Installation Folder*>**\Components\Cedar**
   - **BweReg.bat** located in <*Installation Folder*>**\Components\Bwe**

If the automatic registration does not work, try to register manually using the program **regsvr32.exe** from the Windows system directory.

If this does not help, create a copy of the components list using the command **File** > **Save to File** in the **Cairo and Cedar Component Version Info** utility. Submit an error report, the components list, and the log files located in the <*Installation Folder*>**\Bin\bin\Log** folder to Oracle Customer Support.

4.5 Migrating Existing Project Files

Project files designed in the earlier version of Forms Recognition must be converted to the current version format before they can be used in the current version.

The conversion process is fully automated and is done by Forms Recognition Designer. To convert project files created with earlier versions of Forms Recognition, do the following:

1. Launch the Designer application
2. Select **Load Project…** from the **File** menu
3. On the **Load Project** dialog box, browse to the project file location and double click on the project file that you want to convert
4. Login to the project as an administrative user for the project
5. Click **OK** to launch the automatic project conversion process. The conversion takes from a few seconds to a few minutes, depending on the size of the project
6. Once the conversion is completed, enter Designer's normal train mode by selecting the **Train Mode** option from the **Options** menu
7. Click the **Learn Documents** (lightbulb) toolbar button to relearn the project
8. Save the project. The project is ready for use in the current version of WebCenter Forms Recognition

---

Important: A project and learnset backup should always be taken before migrating project files.

For information on how to configure a project for a WebCenter Forms Recognition database, see the *WebCenter Forms Recognition Designer User's Guide*.

4.6 Removing WebCenter Forms Recognition Version 12.2.1.3.0

WebCenter Forms Recognition can be uninstalled by using the Windows Control Panel. It is important to stop all running services using the Task Manager before uninstalling the application. To remove Forms Recognition:

1. Launch the Windows Control Panel
2. Navigate to **Control Panel** **Programs** **Programs and Features**
3. On the **Uninstall or Change a Program** list, select **Oracle WebCenter Forms Recognition 12.2.1.3.0**
4. Click **Uninstall**
5. Follow the on-screen instructions
6. After un-installation, reboot your computer

4.7 Repairing a WebCenter Forms Recognition 12.2.1.3.0 Installation

The WebCenter Forms Recognition installer may be used to repair a copy of Forms Recognition that has stopped working properly. Factors that could cause an installation to malfunction include:

- Accidental deletion of application files
- Missing registry entries
- Corrupted application files
- Malicious attacks on a machine housing Forms Recognition

To repair WebCenter Forms Recognition:

1. Launch the Windows Control Panel
2. Navigate to **Control Panel** **Programs** **Programs and Features**
3. On the **Uninstall or Change a Program** list, select **Oracle WebCenter Forms Recognition 12.2.1.3.0**
4. Click **Change**
5. On the Setup dialog box, select **Repair** then click **Next**. This will reinstall all program components that were installed by the previous setup
6. Click **Finish** when setup is completed

4.8 Manually register components

The installation process automatically registers the Dynamic Link Libraries (DLL's) components. For troubleshooting purposes, you can manually register WebCenter Forms Recognition components. Complete the following steps to register the components:

1. Execute the *<Installation Folder>***\WebCenter Forms Recognition Web Server\Components\Cairo\RegCro.bat** file.
2. Execute the *<Installation Folder>***\WebCenter Forms Recognition Web Server\Components\Cedar\RegCdr.bat** file.
3. Execute the *<Installation Folder>***\WebCenter Forms Recognition Web Server\Components\Bwe\RegBwe.bat** file.

4.9 Adding or Removing Forms Recognition Components

WebCenter Forms Recognition is a product suite consisting of the following applications:Error! Unknown document property name.Error! Unknown document property name.

- WebCenter Forms Recognition Runtime Server
- WebCenter Forms Recognition Designer
- WebCenter Forms Recognition Verifier
- WebCenter Forms Recognition Web Verifier

The Forms Recognition installer uses a modular approach that enables you to add or remove applications from a machine.

To modify an existing WebCenter Forms Recognition installation:

1. Launch the Windows Control Panel
2. Navigate to **Control Panel Programs Programs and Features**
3. On the **Uninstall or Change a Program** list, select **Oracle WebCenter Forms Recognition 12.2.1.3.0**
4. Click **Change**
5. On the Setup dialog box, select **Modify** and click **Next**
6. In the **Feature Selection** dialog box, select or clear the desired components
7. Click **Next**.
8. In the **Icons on Desktop** page, complete the following sub-steps.
    a. Optional. Select **Create desktop shortcuts for applications**.
    b. Click Finish.
9. Setup adds (if checked) or removes (if unchecked) the components
10. Click **Finish** when setup completes

4.10 Password Encryption in Core Configuration Files

The application architecture of WebCenter Forms Recognition makes it very important to be able to hide sensitive security information, such as database access passwords, stored in Forms Recognition core configuration files or custom project configuration files.

Password encryption is optional and former configuration files with unencrypted passwords will still work with no issues.

Note: The maximum number of characters allowed to encrypt is 30. Passwords longer than 30 characters would not be encrypted.

Below are the steps to encrypt the database connection password for the core Forms Recognition configuration files:

1. Open one of the Forms Recognition configuration files, for example, *<Installation Folder>***\Bin\bin\DstDsr.exe.config**, in a text editor.
2. Locate the connection string and the password part of the string, example:

```
<connectionStrings>

        <add name="Entities"
        connectionString="metadata=res://*/Entity.Entities.csdl|
```

```
        res://*/Entity.Entities.ssdl|
        res://*/Entity.Entities.msl;provider=System.Data.SqlClient;provider
        connection string=&amp;quot;Data Source=MYSQLSRV;Initial
        Catalog=DatabaseName;Integrated Security=false;User ID=alexey;
        Password=MyPassword;MultipleActiveResultSets=True&amp;quot;"
        providerName="System.Data.EntityClient" />

    </connectionStrings>
```

3. Modify the password, replacing it with any number of asterisk (*) characters, example:

```
<connectionStrings>

        <add name="Entities"
        connectionString="metadata=res://*/Entity.Entities.csdl|
        res://*/Entity.Entities.ssdl|
        res://*/Entity.Entities.msl;provider=System.Data.SqlClient;provider
        connection string=&amp;quot;Data Source=MYSQLSRV;Initial
        Catalog=DatabaseName;Integrated Security=false;User ID=alexey;
        Password=********;MultipleActiveResultSets=True&amp;quot;"
        providerName="System.Data.EntityClient" />

    </connectionStrings>
```

Note: The number of asterisks used is not important.

4. Using the Command Line, run the *<Installation Folder>***\Bin\bin\DstCrypt.exe** tool with the following arguments:

   ```
   DstCrypt.exe /text "<password>" >> my_encrypted_password.txt
   ```

   Note: You could add the line above to a new .bat file created in the *<Installation Folder>***\Bin\bin** directory and double click on it. This should produce a new file with the name "my_encrypted_password.txt" in the same directory where the executable is located.

5. Open the resulting text file. It will contain text like in the example below. Copy the section indicated in bold type in the following example that represents the encrypted password:

   ```
   Text MyPassword encoded to
   Y652CeXVdMtdNtbnBD2itCEmfFFyHf9IGsN2psi6svy/MsKp8nKUgv2P7M37uu5rNo3V7wkH5795A5z6WG
   ox/KEm60l6AG9flX+B5miOQg7iOgJCBqoHrsAbICHzm2EJbCkaMp1oUcvtP+8hpeJVMlBpD+QkfLlithUX
   INhWaCM=
   ```

6. Locate the **appSettings** section of the DstDsr.exe.config file and add the new **EncrPwd** key to this section, assigning the encrypted sequence above to the value of the key:

```
<appSettings>

        <add key="EncrPwd"
        Y652CeXVdMtdNtbnBD2itCEmfFFyHf9IGsN2psi6svy/MsKp8nKUgv2P7M37uu5rNo3V7wkH579
        5A5z6WGox/KEm60l6AG9flX+B5miOQg7iOgJCBqoHrsAbICHzm2EJbCkaMp1oUcvtP+8hpeJVMl
        BpD+QkfLlithUXINhWaCM=

</appSettings>
```

7. Save the DstDsr.exe.config file.
8. When required, apply steps 1 - 7 to the other core configuration files, which represent different applications. These are:
   - For **Runtime Server** *<Installation Folder>*\Bin\bin\DstHost.exe.config
   - For **Learnset Manager tool** *<Installation Folder>*\Bin\bin\DstSlm.exe.config

- For **Verifier**<*Installation Folder*>\Bin\bin\DstVer.exe.config
- For **Workdoc Browser** <*Installation Folder*>\Bin\bin\DstWkBrw.exe.config
- For **Supervised Learning in Web Verifier** <*Installation Folder*>\Bin\bin\Brainware.System.Project.exe.config
- For **Web Verifier** <*Installation Folder*>\Oracle WebCenter Forms Recognition Web Server\web.config

---

Note: Corrupted or incorrect encryption key or an incorrect password in the web.config file will entail a 'Login failed' error message when trying to open the Web Verifier.

4.11 AP Packaged Project INI File Encryption

WebCenter Forms Recognition allows the user to encrypt a password (or any other value) within the AP Packaged Project's INI file. RSA encryption is used, which contains a public key and a private key.

The public key can be distributed to anybody that needs to encrypt strings and store them in the project INI file, for example, a WebCenter Forms Recognition administrator. Refer to *Section 4.11.2: Project INI File Encryption for the Integrator* below for information about how to encrypt a password using the public key.

Public key example:

```
<RSAKeyValue><Modulus>vJ+W7SuXuvOrWVoy4tPrbfLCuoHElo750cpTuEzLPk6iz6bHAodPVgLFaOEK+XMMS2G5
z+6961vuQsDGUt+O1Ag1PiTXCa6rrAaeCaaDO4HI8Mmpw0OkUZEfCZpTTYCYQPfZlgokwomF6VDSB9dlUS430IT0gc
tQY1b5iM4MqT0=</Modulus><Exponent>AQAB</Exponent></RSAKeyValue>
```

Only the project developer should know the private key. It will be coded into the project script to enable the decryption of the encrypted INI settings at runtime. Refer to *Section 4.11.1: Project INI File Encryption for the Project Developer* below for information about how to use the private key in project script to decrypt and use an encrypted password from the project INI file.

Private key example:

```
<RSAKeyValue><Modulus>vJ+W7SuXuvOrWVoy4tPrbfLCuoHElo750cpTuEzLPk6iz6bHAodPVgLFaOEK+XMMS2G5
z+6961vuQsDGUt+O1Ag1PiTXCa6rrAaeCaaDO4HI8Mmpw0OkUZEfCZpTTYCYQPfZlgokwomF6VDSB9dlUS430IT0gc
tQY1b5iM4MqT0=</Modulus><Exponent>AQAB</Exponent><P>8SRHEvT5Bn2paRHSDR9yCQb7WGYE9PbeHzuqwH
6iWa0LNYJrSrhhUeCEpwlPLQWQq10KmMZgG0+Br4nuBMmMHQ==</P><Q>yD7l9fjB/MJWYaV3LcEzY286Q+Xvo74i6
THvHkKqB1NKYGcN9xF9d8XbiUQNgBZ/4F02T6mFeYDO32KFVRXHoQ==</Q><DP>nRDTFn7nwRmSgfRwi8minkyk5DQ
3IFO35EIZ+x3Ao4Z52ZWkStwDz6/c12vR3XJVg7irkU0NBlzoDK1bklSw5Q==</DP><DQ>B3xieGmORva05/2ZkPpS
A3ubAALOjJ6FC5a0S7tOQ+vXMfdoTD45JIsfA+ipYIp2yVpyt1OtC7fHBA7Y0S95QQ==</DQ><InverseQ>4S1xqlX
K9f1rawGCbFWOVp6lz1fCoQ8RfyDE87/G/pUilHRJV2acBAcngY3c/MRMKrXQb8lx99k7dENUYc8ywQ==</Inverse
Q><D>KAL6cwkCQKgbuvKFRNSLZmFOqV2JpB5kI/p1U+0GWAs6Qi4wnPqy+53O3naOa2faPctXLSKJqvlvSz21VDMUC
syphvOSxBtc1cZHJp4ueQPA7u+qrIJaDY1RhlAVoqNfCJFX6+McVJ+I/X+mZOCtdUaCuAoNn014UYOaMujYDQE=</D
></RSAKeyValue>
```

Note: WebCenter Forms Recognition does not include tools for generating RSA encryption keys. The examples provided above can be used for demonstrating/testing, but it is the responsibility of the implementer to generate and provide the appropriate public and private keys in the required XML format.

4.11.1 Project INI File Encryption for the Project Developer

Where a database password is encrypted in the project INI file it is necessary to decrypt it at runtime, and then use the decrypted password in the database connection string to make the connection to the database. An example of how this would typically appear in the project INI file is:

```
SQL_VL_01_ConnectionString=Provider=OraOLEDB.Oracle.1; Persist Security Info=True;User
ID=WFR;Data Source=ORCL

SQL_VL_01_ConnectionPassword=puejB5SQNCFGgwe6MRoWc1G1y7qX8xSAhgUZjhN6JolhYdKIxla7vLMU4bYmG
9V3Ayxualp/ObgXRqnSAmGsGF1FPZXktRmf58SXbnCDXmYrYgp8eS3IaqiLUPrhTiRCvfr8ZsMtK+3usmahfxpESUO
Q7MZf36suWV4V3sBf9Xw=
```

Note: The **ConnectionString** setting does not contain the password. Instead, the database password is stored in its encrypted form in the **ConnectionPassword** setting.

The following script example shows how the encrypted password is retrieved from the INI file, decrypted, and then added to the connection string, resulting in a fully formed connection string that can be used to make a connection to the database through ADO.

Note: You must add a reference to the **CdrCrypt** COM object in the project script page.

```
Dim theCedarCryptographyHelper As New CdrCrypt.RSACodecInt

Dim strEncryptedPassword As String

Dim strOpenPassword As String

Dim strPrivateKey As String

strPrivateKey =
"<RSAKeyValue><Modulus>vJ+W7SuXuvOrWVoy4tPrbfLCuoHElo750cpTuEzLPk6iz6bHAodPVgLFaOEK+XMMS2G
5z+6961vuQsDGUt+O1Ag1PiTXCa6rrAaeCaaDO4HI8Mmpw0OkUZEfCZpTTYCYQPfZlgokwomF6VDSB9dlUS430IT0g
ctQY1b5iM4MqT0=</Modulus><Exponent>AQAB</Exponent><P>8SRHEvT5Bn2paRHSDR9yCQb7WGYE9PbeHzuqw
H6iWa0LNYJrSrhhUeCEpwlPLQWQq10KmMZgG0+Br4nuBMmMHQ==</P><Q>yD7l9fjB/MJWYaV3LcEzY286Q+Xvo74i
6THvHkKqB1NKYGcN9xF9d8XbiUQNgBZ/4F02T6mFeYDO32KFVRXHoQ==</Q><DP>nRDTFn7nwRmSgfRwi8minkyk5D
Q3IFO35EIZ+x3Ao4Z52ZWkStwDz6/c12vR3XJVg7irkU0NBlzoDK1bklSw5Q==</DP><DQ>B3xieGmORva05/2ZkPp
SA3ubAALOjJ6FC5a0S7tOQ+vXMfdoTD45JIsfA+ipYIp2yVpyt1OtC7fHBA7Y0S95QQ==</DQ><InverseQ>4S1xql
XK9f1rawGCbFWOVp6lz1fCoQ8RfyDE87/G/pUilHRJV2acBAcngY3c/MRMKrXQb8lx99k7dENUYc8ywQ==</Invers
eQ><D>KAL6cwkCQKgbuvKFRNSLZmFOqV2JpB5kI/p1U+0GWAs6Qi4wnPqy+53O3naOa2faPctXLSKJqvlvSz21VDMU
CsyphvOSxBtc1cZHJp4ueQPA7u+qrIJaDY1RhlAVoqNfCJFX6+McVJ+I/X+mZOCtdUaCuAoNn014UYOaMujYDQE=</
D></RSAKeyValue>"

strEncryptedPassword = DicVal("01" & "ConnectionPassword", "SQL")

If Len(strEncryptedPassword) > 0 Then

  strOpenPassword = theCedarCryptographyHelper.Decode(strEncryptedPassword, strPrivateKey)

End If

If Len(strOpenPassword) > 0 Then

  strConnection = strConnection + ";Password=" + strOpenPassword

End If
```

4.11.2 Project INI File Encryption for the Integrator

As an implementer or WebCenter Forms Recognition administrator, you simply need encrypt (for example) the database password and add the encrypted value to the project INI file. To encrypt a value, such as the database password:

1. Open the Windows Command Line
2. Navigate to the <*Installation Folder*>**\Bin\bin** directory in the Command Line
3. Execute the following command, replacing the <*myPassword*> and <*publicKey*> placeholders with the actual values for your environment:

   `DstCrypt.exe /text <myPassword> /key "<publicKey>" >> <output text file name>`

For example:

```
DstCrypt.exe /text MyPassword /key
"<RSAKeyValue><Modulus>vJ+W7SuXuvOrWVoy4tPrbfLCuoHElo750cpTuEzLPk6iz6bHAodPVgLFaOE
K+XMMS2G5z+6961vuQsDGUt+O1Ag1PiTXCa6rrAaeCaaDO4HI8Mmpw0OkUZEfCZpTTYCYQPfZlgokwomF6
VDSB9dlUS430IT0gctQY1b5iM4MqT0=</Modulus><Exponent>AQAB</Exponent></RSAKeyValue>"
>> my_encrypted_password.txt
```

Note: Notice that the public key value is contained in quotes but the password is not.

The text file specified in the command (e.g. my_encrypted_password.txt) will now contain the encrypted text string for the password.

4. Add the encrypted password to the **ConnectionPassword** setting in the project INI file. For example:

```
SQL_VL_01_ConnectionPassword=puejB5SQNCFGgwe6MRoWc1G1y7qX8xSAhgUZjhN6JolhYdKIxla7v
LMU4bYmG9V3Ayxualp/ObgXRqnSAmGsGF1FPZXktRmf58SXbnCDXmYrYgp8eS3IaqiLUPrhTiRCvfr8ZsM
tK+3usmahfxpESUOQ7MZf36suWV4V3sBf9Xw=
```

Note: Remember to remove the **`Password=<`*`database password`*`>;`** component from the setting in the corresponding **ConnectionString** setting.

4.12 Silent Installations

A silent install mode is provided for situations where the same configuration of WebCenter Forms Recognition is to be installed on several machines, for example, Verifier workstations. The use of a configuration file removes the necessity to go through the installation dialog on each machine.

4.12.1 Silent Install.ini

The configuration settings for the silent installation are read from the **Silent Install.ini** file in the WebCenter Forms Recognition installation directory. The directory contains an example file, which must be edited before performing a silent installation.

The file contains six sections:

- General
- Applications
- OCR Engines
- Additional
- Database Configuration
- DB Credentials

It is allowed to delete single entries or complete sections.

However, it is not allowed to use options without the section name. If any information is deleted from the **Silent Install.ini** file, the Setup uses the default values as described.

Name	Description
[General]	Determines how and where WebCenter Forms Recognition is to be installed.
EULA	The end-user agreement needs to be accepted on each machine where the software is installed. If this section is not modified in the silent

	install.ini file, the silent installation would be canceled. 0: Accepted 1 (DEFAULT): Not accepted
Path	Indicates where the application should be installed. The pathname should not have a final backslash. Example: `Path = "C:\Program Files (x86)\Oracle"`
MoveComponentsIfRequired	If an older version of the application is installed, this indicates whether to use the existing component folder or whether to move the old components into the new directory prior to installation. 0: Use existing component folder. 1 (DEFAULT): Move components to the new path.
CreateDesktopIcons	0 (DEFAULT): Don't create desktop shortcuts. 1: Create desktop shortcuts.
InstallWibuKey	0: Skip Wibukey driver installation. 1 (DEFAULT): Install Wibukey drivers.
StopIfDotNetIsNotFound	0: If the required .Net Framework is not found on the system the installation proceeds. The Features (WebVerifier, Database Connection...) will not be installed. 1 (DEFAULT): If the required .Net Framework is not found on the system, the installation will be aborted.
[Applications]	Defines which applications are to be installed. Note that it is permissible to skip all applications if, for example, only the extraction components are to be installed.
Designer	0: Skip installation of the Designer application. 1 (DEFAULT): Install the Designer application.
Verifier	0: Skip installation of the Verifier application. 1 (DEFAULT): Install the Verifier application.
Runtime Server	0: Skip installation of the Runtime Server application. 1 (DEFAULT): Install the Runtime Server application.
Web Verifier	0: To skip installation of Web Verifier application. 1 (DEFAULT): Install Web Verifier application. See also option **StopIfDotNetIsNotFound**
[OCR Engines]	Defines which OCR engines are to be installed. It is permissible to skip all the engines.
FineReader10	0: Skip installation of ABBYY FineReader 10. 1 (DEFAULT): Install ABBYY FineReader 10.
FineReader11	0: Skip installation of ABBYY FineReader 10. 1 (DEFAULT): Install ABBYY FineReader 10.
Kadmos5	0: Skip installation of Kadmos 5 engine. 1 (DEFAULT): Install Kadmos 5 engine.
Cleqs	0: Skip installation of Cleqs engine. 1 (DEFAULT): Install Cleqs engine.
QualitysoftBarcode	0: Skip installation of Quality Soft Engine 1 (DEFAULT): Install Quality Soft Engine

[Additional]	Additional files to install.
Demo Files	0: Skip installation of the demo project files. 1 (DEFAULT): Install demo project files.
[Database Configuration]	Configures an existing database server. See also option **StopIfDotNetIsNotFound**.
DBServerType	1: SQL Server database will be configured. 2: Oracle Server database will be configured. 3 (DEFAULT): No database will be configured.
If there is any wrong information for the following options, DBServerType will be set to 3.	
UseDBConfIniFile	Text file name that contains Database Connection String. If this option is empty, credentials will be taken from [DB Credentials] section. If there is neither a configuration file nor a [DB Credentials] section, **DBServerType** will be set to 3 (no database) internally. (DEFAULT empty string).
[DB Credentials]	This section can be used instead of a configuration file. (DEFAULT database configuration will be skipped, if option **UseDBConfIniFile** is not set)
Only for SQL Server Usage (See Option **DBServerType** and **UseDBConfIniFile**)	
SQLServerWindowsAuthent	0 (DEFAULT): No Windows Authentication will be used for DBA. 1: Windows Authentification will be used for DBA.
SQLServerAdminUser	DBA account name (See also option **SQLServerWindowsAuthent**). (DEFAULT empty string).
SQLServerAdminPassword	DBA account password (See also option **SQLServerWindowsAuthent**) (DEFAULT empty string).
For both Database Server Types (See Option **DBServerType** and **UseDBConfIniFile**)	
DBUserWindowsAuthent	0 (DEFAULT): No Windows Authentication will be used for DB user. 1: Windows Authentification will be used for DB user.
DBUserName	DB user account name (See also option **DBUserWindowsAuthent**) (DEFAULT empty string).
DBUserPassword	DB user account password (See also option **DBUserWindowsAuthent**) (DEFAULT empty string).
DatabaseServerPath	Name of the database. Usually it is specified as: **For Oracle Database:** *<database server>:<database port>/<instance name>* **For SQL Server:** *<MachineName>\<InstanceName>* (DEFAULT empty string).

5 Configure WebCenter Forms Recognition Web Verifier

Windows Internet Information Server (IIS) hosts Web Verifier.

Web Verifier requires the following windows components to be enabled.

Common HTTP Features

- Static Content
- Default Document
- Directory Browsing
- HTTP Errors

Application Development

- ASP.NET 4.5
- .NET Extensibility
- ISAPI Extensions
- ISAPI Filters

Health and Diagnostics

- HTTP Logging
- Request Monitor

5.1 Preparing Internet Information Server (Post Install)

One of the preconditions for working with WebCenter Forms Recognition Web Verifier is the installation of the Internet Information Server. Windows Server 2008 R2 works with IIS 7. Windows Server 2012 R2 works with IIS 8.5.

To perform the installation of the IIS application, use the following links for more information:

For Windows Server 2008 R2

http://learn.iis.net/page.aspx/29/installing-iis-7-on-windows-server-2008-or-windows-server-2008-r2/

For Windows Server 2012 R2

http://www.iis.net/learn/install/installing-iis-85/installing-iis-85-on-windows-server-2012-r2

5.1.1 Configuring IIS 7 or IIS 8

The Internet Information Server (IIS) is used for executing the WebCenter Forms Recognition Web Verifier. To configure IIS 7 or IIS 8 as the Forms Recognition Web Server:

1. Launch the IIS Manager (**Start** → **Administrative Tools** → **Internet Information Services (IIS) Manager**)

2. Expand the <*machine name*> node, then expand the **Sites** node in the navigation tree
3. Right-click **Default Web Site**
4. Select the **Add Application…** menu item
5. In the dialog window, enter the **Alias** you want to use to gain access to this Web virtual directory. Oracle recommends setting the alias as **WebVerifier**.
6. Set the **Physical path** to <*Installation Folder*>**\WebCenter Forms Recognition Web Server**
7. Click **OK**
8. Double click the **Default Document** icon for the WebVerifier application
9. Click **Add…** in the **Actions** menu
10. Enter **Login.aspx** in the resulting dialog, then click **OK**
11. Right-click on the **WebVerifier** node and choose **Add Virtual Directory…**
12. In the dialog box, set the **Alias** to **WL**
13. Set the **Physical Path** to <*Installation Folder*>**\Components\Cedar\WL**
14. Click **OK** to close the dialog
15. Select the **Application Pools** node in the navigation tree
16. Select **DefaultAppPool** from the list of application pools, then click **Advanced Settings…** in the **Actions** menu
17. Ensure all of the following settings are as follows:
    - Enable 32-Bit Applications: **True**
    - Managed Pipeline Mode: **Integrated**
    - Identity: **NetworkService**
18. Click **OK**
19. Using Windows Explorer, create the following three subfolders:
    - ..\Windows\Temp**Bwe**
    - ..\Windows\Temp**CdrDBCache**
    - ..\Windows\Temp**DIST**
20. For each of the subfolders created in the previous step, ensure that the **NETWORK SERVICE** user specified in the application pool is assigned the **Full Control** permission.

To complete the configuration for Web Verifier, Data Execution Prevention (DEP) should be disabled. To disable DEP, execute the following command in the Command Line:

```
bcdedit.exe /set {current} nx AlwaysOff
```

Finally, restart the server.

5.2 Modify the instanceName when using multiple web servers

To ensure that all the web servers' instances are using the same WebCenter Forms Recognition database, their instance name should be unique.

To configure unique instance name across different steps, complete the following steps.

1. Open *<Installation Folder>***\WebCenter Forms Recognition Web Server\Web.config**.
2. Search for `<system.controllers><client instanceName="Web Verifier" remoteObjectRenewalTimeout="180"></client>.`
3. Change `instanceName="Web Verifier"` to `instanceName="Web Verifier [xx]"` where xx is unique across the systems.
   Example `instancename="Web Verifier 01"` for the first server.
   Example `instancename="Web Verifier 02"` for the second server.

5.3 Modify the database connection string for Web Verifier

Complete the following steps to modify database connection string for Web Verifier.

1. From the *<Installation Folder>***\WebCenter Forms Recognition Web Server** directory, open the **Web.config** file in a text editor.
2. Search for the <connectionStrings> element.
3. For a SQL Server database, modify the following values.
   a. Set **Data Source** to the data source.
   b. Set **Initial Catalog** to the SQL Server database catalog.
   c. Set **User ID** to the service account user ID.
   d. Set **Password** to the service account password.
4. For an **ORACLE** database, modify the following values.
   a. Set **Data Source** to the data source.
   b. Set **User ID** to the service account user ID.
   c. Set **Password** to the service account password.
5. Save and close the file.
6. Create a copy of the **Web.config** file and move it to *<Installation Folder>***\Bin\bin.**
7. Rename the **Web.config file** copy to **Brainware.System.Project.exe.config**.

5.4 Configure server security for Web Verifier

5.4.1 Preparing the User Context (SQL Server)

It is necessary for the user of the user context in which the Web Verifier is running in IIS to have the proper rights to access the SQL Server database. By default, the Web Verifier runs under the NETWORK SERVICE user context; hence the same should be allowed to access the database.

If you selected **Windows Authentication** during the installation of WebCenter Forms Recognition, you will need to add the domain username to the SQL Server database in addition to the NT AUTHORITY\NETWORK SERVICE.

Steps to add NETWORK SERVICE to SQL server:

1. Open Microsoft SQL Server Management Studio
2. Expand the local computer name, select **Security** **Logins**
3. Right click **Logins**, select **New Login**
4. On **Login Properties**, under **General**, click **Search**
5. Enter **NETWORK SERVICE** and then click **Check Names**
6. Click **OK**
7. Select **sysadmin** (*public* is selected by default) for **Server Roles**
8. Click **OK**
9. The **NT AUTHORITY \NETWORK SERVICE** has been added to SQL server.

After adding Network Service to SQL server, make sure that the IIS is running under NT AUTHORITY \NETWORK SERVICE by opening the IIS Manager:

1. Click **Start**, and then select **Control Panel**
2. Select **Administrative Tools**, and then click **Internet Information Services (IIS) Manager**
3. In the **Connections** panel, expand the server node and click **Application Pools**
4. On **Application Pools**, select the application pool that you want to specify an identity, and then click **Advanced Settings** in the **Actions** panel
5. For the identity property, the built-in account should be **NetworkService**
6. If it does not contain NetworkService, click **Set…** to open the **Application Pool Identity** dialog box
7. Select the **Built-in account** option and select **NetworkService** account from the list

5.4.2 Setting Permissions for Forms Recognition Projects Execution

All WebCenter Forms Recognition projects are located in a file system folder. The Web Verifier sources this path from the Forms Recognition database. Forms Recognition projects are loaded by the **Brainware.System.Project.exe** process. This process cannot load the projects until it has the appropriate permissions for the projects folder. In this case, it is necessary to grant permission to the **Network Service** Windows user for this folder by performing the following steps:

1. Using Windows Explorer, navigate to the projects folder
2. Right-click on the folder and select **Properties**
3. In the dialog window, select the **Security** tab
4. Add the **Network Service** user to the list

5.5 Configure Internet Explorer for Web Verifier

This section describes security configuration for the Web Verifier client.

1. In Internet Explorer, select **Tools** **Internet Options**
2. Select the **Security** tab, and then click **Custom Level…**

3. Locate **ActiveX controls and plug-ins** in the list and ensure the following options are enabled:
   - Binary and script behaviors
   - Run ActiveX controls and plug-ins
4. Locate **Scripting** in the list and ensure the following options are enabled:
   - Active scripting
   - Allow status bar updates via script *(Optional. See note below)*
5. Click **OK** to save the browser settings

Note: The **Allow status bar updates via script** option is required to display information about batches, documents, current filters, and the page number in the Internet Explorer status bar. The Web Verifier client will function normally without this option enabled, but batch, document, filter, and page information will not be visible in the Internet Explorer status bar.

5.6 Configuring Single Sign-On Authentication for Web Verifier

Web Verifier has been enhanced to support Single Sign-On (SSO) user authentication. The SSO implementation will intercept the user's login request and collect the user credentials and authenticate the user, or accept the user as already authenticated.

The SSO functionality is implemented as a generic solution to work with any SSO implementation and configuration that provides the user credential information via an HTTP header.

Note: Configuration of the SSO service provider is out of scope of this document. Refer to the provider's product documentation.

5.6.1 Enabling Single Sign-On Authentication

To enable external SSO authentication for Web Verifier, the following changes are required in the **web.config** file:

1. Check the **<verifier.webclient>** section for the **<httpHeaderBasedSso>** parameter.
2. Set the flag for the **enabled** setting to **true**:

   ```
   <httpHeaderBasedSso loginHeader="idUserName" enabled="true"
   sessionHeader="ShibSessionID" />
   ```

3. Set the **loginHeader** value to the HTTP header attribute name, which is returned by the SSO service. For example:

   ```
   loginHeader="remoteuser"
   ```

4. Set the **sessionHeader** parameter to the default session ID header used by the SSO provider.

5.6.2 SSO Session vs. Web Verifier Client Session

There are two different sessions to consider when SSO is used; the SSO session and the Web Verifier Client (WVC) session. The SSO session generally has longer life as it might be used for other applications and not only for WVC.

The WVC and SSO sessions are both renewed with every server request, for example, field validation or opening the next batch. They are not renewed with client-side actions such as zooming in on the image or typing a value into form field without validating it.

Note: When SSO authentication mode is used, the WVC session timeout must be set to a smaller value than the SSO session timeout. Otherwise, if the SSO session times out while the WVC session is still active, the user will be re-directed to the SSO login page, and any changes made in Web Verifier will be lost. To achieve this, the following parameter must be set appropriately:

```
<sessionState timeout="20"/>
```

Refer to the product documentation of your SSO provider for details on how to configure the SSO session duration.

5.7 Configuring Windows Authentication for Web Verifier

The Web Verifier application allows you to login with your Windows user account. In this case, the password that is shared with Windows will be used to login into Web Verifier.

To use this option, you first need to configure the server.

Note: Only Windows Authentication access will be possible after this option is configured. However, when logged in to Web Verifier via Windows Authentication, it will be possible to use the re-login menu option to login e.g. as an administrator in order to perform certain administrative tasks.

5.7.1 Prerequisites

- Before starting to configure IIS, make sure that the Web Verifier application is working properly using an existing project user account
- Back up the web.config file

5.7.2 Configuration for IIS 7.5

To configure Windows Authentication access to Web Verifier with IIS 7.5:

1. Open **Authentication** settings in IIS group for the WebVerifier application
2. Enable **Windows Authentication** and disable all other authentication methods
3. Close all of the running browser sessions prior to accessing the WebVerifier application
4. Add the Windows user to the database. Refer to the *WebCenter Forms Recognition Designer User Guide* for information on how to do this

5.7.3 Changes to the Web.config File

It is highly recommended to have two versions of the web.config file; one for standard authentication and other for Windows Authentication. This will simplify switching between modes.

The following list shows required steps to convert standard web.config to a web.config with Windows Authentication enabled:

1. Make a backup copy of the existing web.config file
2. Change **<authentication>** section (located in the <configuration><system.web>) to the following:

   ```
   <authentication mode="Windows">
   ```

3. Remove the following line:

   ```
   <forms loginUrl="Login.aspx" defaultUrl="BatchView.aspx" />
   ```

   This is a child of the <authentication> section, and is only needed for standard authentication.

4. Change **<authorization>** section (located in the <configuration><system.web>) from **deny** to **allow**:

   ```
   <authorization>
       <allow users="?"/>
   </authorization>
   ```

5. Add the **enableSessionState** attribute to **<pages>** section (located in the <configuration> <system.web>):

   ```
   <pages enableSessionState="true">
   ```

6. Remove all **<location>** sections (located in the <configuration> section right before <appSettings>). Those sections look like the following:

   ```
   <location path="WL">
       <system.web>
           <authorization>
               <allow users="*" />
           </authorization>
       </system.web>
   </location>
   ```

5.7.4 Reverting Back to Standard Authentication

To switch from Windows Authentication mode back to standard authentication mode, the following adjustments to IIS are required:

1. In Internet Information Services (IIS) Manager, open **Authentication** settings in IIS group for the WebVerifier application
2. Disable **Windows Authentication** and enable both, **Anonymous Authentication** and **Forms Authentication**
3. Replace the web.config file with the backup made prior to enabling Windows authentication for Web Verifier

5.7.5 Configure cookies for Web Verifier

To make sure the browser sends cookies over https network, following changes are required.

1. Open **Web.config** from the *<Installation Folder>***WebCenter Forms Recognition Web Server** directory in a text editor.
2. In the <configuration> element, search for the following line.
   <system.web>
3. Under <system.web>, add the following line.
   `<httpCookies requireSSL="true" />`
4. To apply forms authentication, search for the following line.
   `<forms loginUrl="Login.aspx" defaultUrl="BatchView.aspx" />`
5. Add the requireSSL attribute.
   Example `<forms loginUrl="Login.aspx" defaultUrl="BatchView.aspx" requireSSL="true" />`
6. Save and close the file.
7. Optional. To prevent other applications from accessing Web Verifier cookies, deploy Web Verifier in one of the following ways.
   a. As the root level website.
   b. As the only web application under a website in IIS.

5.7.6 Configuring SSL for Web Verifier

For information how to set up SSL on your Internet Information Server (IIS) machine, refer to the Microsoft Support article *How to Implement SSL in IIS* (http://support.microsoft.com/kb/299875).

5.8 Use Traditional Chinese

If Chinese as UI language has been configured, then Web Verifier uses Simplified Chinese by default.

To use Traditional Chinese, following steps has to be configured.

1. Create a backup of *<Installation Folder>***WebCenter Forms Recognition Web Server\bin\Resources\zho** directory.
2. Copy all the files from *<Installation Folder>***WebCenter Forms Recognition Web Server\bin\Resources\cmn** directory to *<Installation Folder>***WebCenter Forms Recognition Web Server\bin\Resources\zho** directory.
3. Open *<Installation Folder>***WebCenter Forms Recognition Web Server\Web.config** file in a text editor.
4. Search
   `<add key="LanguageDisplayName_ZHO" value="□□□□" />`
   and replace with
   `<add key="LanguageDisplayName_CMN" value="□□□□" />`
5. Save and close the file.
6. Restart any open Web Verifier sessions.

5.9 Access WebCenter Forms Recognition Web Verifier

WebCenter Forms Recognition Web Verifier application will be accessible by the address http://localhost/WebVerifier/login.aspx

5.10 Enabling New Columns for Batch View

Four additional columns are available to hold additional information on batches:

- Batch.ExternalGroupId - default display name: "User Group"

  data type: The Group ID that has been assigned to a batch is relating to security. Batches can be assigned to user group via a unique ID. For example, German invoices belong to Group 1 and English invoices belong to Group 2. When in a shared service center, you could hide all German invoice batches from English Verifiers.

- Batch.ExternalBatchId - default display name: "Batch Group"

  data type: It allows the developer to uniquely identify the batch. For example, external system ID, storage box ID, etc.

- Batch.TransactionId - default display name: "Transaction"

  data type: It allows the developer to synchronize a newly created batch of documents with another external system. It can be used to identify originators of batch of documents.

- Batch.TransactionType - default display name: "Transaction Type"

  data type: It allows the developer to synchronize a newly created batch of documents with another external system. It can used to identify the types of documents (Invoices, Claim forms etc.) in batches or source of document (Email, Scanned etc.)

These table columns are not project or application specific and therefore cannot be configured in Designer or Verifier or RTS applications.

By default, these columns will be invisible. To configure the columns' visibility for Web Verifier, adjust the batch columns' attributes in the batchColumnVisibility section of the web.config file appropriately as described in Appendix A.

The values of the columns can only be set via the Project Script (PostimportBatch). Check the SQL scripts in the installation folder to activate the displaying of those columns. After enabling one or all of the additional columns in database, it applies to all application modules.

The additional columns can be enabled with columns customized.

For Oracle Database

Syntax:

```
exec sp_SetGlobalApplicationSetting('<column setting name>', '<column name to display>', <enabled>)
```

Examples:

```
exec sp_SetGlobalApplicationSetting('SysAppBatchColumnExternalGroupId', 'User Group', 1)

exec sp_SetGlobalApplicationSetting('SysAppBatchColumnExternalBatchId', 'Batch Group', 1)

exec sp_SetGlobalApplicationSetting('SysAppBatchColumnTransactionId', 'Transaction', 1)

exec sp_SetGlobalApplicationSetting('SysAppBatchColumnTransactionType', 'Transaction Type', 1)
```

For SQL Server

Syntax:

```
exec sp_SetGlobalApplicationSetting '<column setting name>', '<column name to display>',
<enabled>
```

Examples:

```
exec sp_SetGlobalApplicationSetting 'SysAppBatchColumnExternalGroupId', 'User Group', True

exec sp_SetGlobalApplicationSetting 'SysAppBatchColumnExternalBatchId', 'Batch Group',
True

exec sp_SetGlobalApplicationSetting 'SysAppBatchColumnTransactionId', 'Transaction', True

exec sp_SetGlobalApplicationSetting 'SysAppBatchColumnTransactionType', 'Transaction
Type', True
```

Note: For setting up the Group ID column, due to the security control, make sure the group ID is matching with the ID created for the users.

5.11 Changing Custom Column Names

After you have enabled new custom columns following the instructions in Section 5.10 *Enabling New Columns for Batch View*, you may want to give them more meaningful names.

5.11.1 Custom Column Names for Web Verifier

To change the custom column names for the Web Verifier application:

1. Using Windows Explorer, navigate to *<Installation Folder>***\WebCenter Forms Recognition Web Server\Bin\Resources\eng**
2. Open the file **Brainware.Verifier.WebClient.resx** in a text editor such as Notepad
3. Change the name of the four items below by adjusting the <value> parameter as indicated in bold type:

```
<data name="TEXT_EXTERNALBATCH_NAME" xml:space="preserve">
    <value>External Batch ID</value>
</data>

<data name="TEXT_EXTERNAL_GROUP_ID" xml:space="preserve">
    <value>User Group</value>
</data>

<data name="TEXT_TRANSACTION_ID" xml:space="preserve">
    <value>Transaction ID</value>
</data>

<data name="TEXT_TRANSACTION_TYPE" xml:space="preserve">
    <value>Transaction Type</value>
</data>
```

For the other application languages, repeat the steps 1 - 3 above using the **Brainware.Verifier.WebClient.resx** file from the appropriate folder under *<Installation Folder>***\WebCenter Forms Recognition Web Server\Bin\Resources**

5.11.2 Custom Column Names for the Verifier Application

For the Verifier client application, custom column names can be changed via SQL Script. Execute the below mentioned script for each column name you want to change:

For Oracle Database

Syntax:

```
exec sp_SetGlobalApplicationSetting('<column setting name>', '<column name to display>', <enabled>)
```

Examples:

```
exec sp_SetGlobalApplicationSetting('SysAppBatchColumnExternalGroupId', 'User Group', 1)

exec sp_SetGlobalApplicationSetting('SysAppBatchColumnExternalBatchId', 'Batch Group', 1)

exec sp_SetGlobalApplicationSetting('SysAppBatchColumnTransactionId', 'Transaction', 1)

exec sp_SetGlobalApplicationSetting('SysAppBatchColumnTransactionType', 'Transaction Type', 1)
```

For SQL Server

Syntax:

```
exec sp_SetGlobalApplicationSetting '<column setting name>', '<column name to display>', <enabled>
```

Examples:

```
exec sp_SetGlobalApplicationSetting 'SysAppBatchColumnExternalGroupId', 'User Group', True

exec sp_SetGlobalApplicationSetting 'SysAppBatchColumnExternalBatchId', 'Batch Group', True

exec sp_SetGlobalApplicationSetting 'SysAppBatchColumnTransactionId', 'Transaction', True

exec sp_SetGlobalApplicationSetting 'SysAppBatchColumnTransactionType', 'Transaction Type', True
```

6 Configure global application settings

6.1 Optional. Configure workflow history reporting

Workflow history reporting can be activated for documents, fields, table cells, classification, learning, and OCR and document separation. Changing these settings takes immediate effect and applies to all users.

For Oracle Database

For documents

```
exec sp_SetGlobalApplicationSetting ('SysAppHistoryReportingActivatedForDocument', 'True',
1)
```

For fields

```
exec sp_SetGlobalApplicationSetting ('SysAppHistoryReportingActivatedForField', 'True', 1)
```

For fields and table cells

```
exec sp_SetGlobalApplicationSetting ('SysAppHistoryReportingActivatedForTableCell',
'True', 1)
```

For classification

```
exec sp_SetGlobalApplicationSetting ('SysAppHistoryReportingActivatedForClass', 'True', 1)
```

For OCR and document separation

```
exec sp_SetGlobalApplicationSetting ('SysAppHistoryReportingActivatedForPage', 'True', 1)
```

For learning

```
exec sp_SetGlobalApplicationSetting ('SysAppHistoryReportingActivatedForLearning', 'True',
1)
```

For SQL Server Database

For documents

```
exec sp_SetGlobalApplicationSetting 'SysAppHistoryReportingActivatedForDocument', 'True',
True
```

For fields

```
exec sp_SetGlobalApplicationSetting 'SysAppHistoryReportingActivatedForField', 'True',
True
```

For fields and table cells

```
exec sp_SetGlobalApplicationSetting 'SysAppHistoryReportingActivatedForTableCell', 'True',
True
```

For classification

```
exec sp_SetGlobalApplicationSetting 'SysAppHistoryReportingActivatedForClass', 'True',
True
```

For OCR and document separation

```
exec sp_SetGlobalApplicationSetting 'SysAppHistoryReportingActivatedForPage', 'True', True
```

For learning

```
exec sp_SetGlobalApplicationSetting 'SysAppHistoryReportingActivatedForLearning', 'True',
True
```

6.2 Optional. Disabling (and Enabling) Batch Deletion

By default, it is possible to delete batches using the Management Console and through the Designer application. You can disable batch deletion through either or both of these applications by executing the following SQL command against the Forms Recognition database:

For Oracle Database

To disable batch deletion in Designer:

```
exec sp_SetGlobalApplicationSetting('SysAppBatchDeletionDisabledInDesigner', 'True', 1)
```

To enable batch deletion in Designer:

```
exec sp_SetGlobalApplicationSetting('SysAppBatchDeletionDisabledInDesigner', 'True', 0)
```

To disable batch deletion in the Management Console:

```
exec sp_SetGlobalApplicationSetting('SysAppBatchDeletionDisabledInRTS', 'True', 1)
```

To enable batch deletion in the Management Console:

```
exec sp_SetGlobalApplicationSetting('SysAppBatchDeletionDisabledInRTS', 'True', 0)
```

For SQL Server

To disable batch deletion in Designer:

```
exec sp_SetGlobalApplicationSetting 'SysAppBatchDeletionDisabledInDesigner', 'True', True
```

To enable batch deletion in Designer:

```
exec sp_SetGlobalApplicationSetting 'SysAppBatchDeletionDisabledInDesigner', 'True', False
```

To disable batch deletion in the Management Console:

```
exec sp_SetGlobalApplicationSetting 'SysAppBatchDeletionDisabledInRTS', 'True', True
```

To enable batch deletion in the Management Console:

```
exec sp_SetGlobalApplicationSetting 'SysAppBatchDeletionDisabledInRTS', 'True', False
```

Note: These settings will affect all users. You must restart Designer or the Management Console (as appropriate) for the settings to take effect.

6.3 About modifying the URL expiration time for Web Verifier

You can modify the URL expiration time for Web Verifier. Changing these settings takes immediate effect and applies to all users.

For Oracle

To modify the URL expiration time for Oracle, in SQL*Plus or Oracle Management Console, in database, complete the following step.

Specify the expiration time in seconds for the second parameter.

```
exec sp_SetGlobalApplicationSetting ('SysAppUrlSignatureExpirationPeriod', '300', 1)
```

For SQL Server

To modify the URL expiration time for SQL Server, in Microsoft SQL Server Management Studio, in database, complete the following step.

Specify expiration time in seconds for the second parameter.

```
exec sp_SetGlobalApplicationSetting 'SysAppUrlSignatureExpirationPeriod', '300', True
```

7 Configuring WebCenter Forms Recognition

7.1 Configuring Applications

There are some main configuration parameters to be accounted for. See *Appendix A: Web.config Options and Associated Resource File Parameters* for more information**.**

7.1.1 Configuring the Oracle WebCenter Forms Recognition Database Connection String

1. Open the application configuration file *<Installation Folder>***\WebCenter Forms Recognition Web Server\web.config** in a text editor, such as Notepad
2. Find the following string:

```
<connectionStrings>

<add name="Entities" connectionString="metadata=res://*/Entity.Entites.csdl|
res://*/Entity.Entites.ssdl|
res://*/Entity.Entites.msl;provider=System.Data.SqlClient;provider connection
string=&quot;Data Source=NEO\SQLSERVER2005;Initial
Catalog=Oracle_verifier_work;Integrated Security=false;User ID=developer;
Password=123456;MultipleActiveResultSets=True&quot;"
providerName="System.Data.EntityClient" />
```

3. Modify the connection string in accordance with your database settings
4. Replace the connection string in the Brainware.System.Project.exe config file with the one configured within the web.config file

Note: These two connection string entries must be identical in order to assure the availability of all Web Verifier functionalities associated with the Knowledge base.

7.2 Server Security Configuration

7.2.1 Registering COM Components

After applying a patch, locate and run the **Register Web Server.bat** as an administrator. It is located in the *<Installation Folder>***\WebCenter Forms Recognition Web Server\bin** folder. For registering this component:

1. Right-click on the **Register Web Server.bat** file
2. Select **Open** from the context menu

7.2.2 Encrypting Sections with the aspnet_regiis Tool

If you want to protect the data stored in the configuration file, perform the following steps:

7.2.2.1 Pre-configuring:

1. Find the **Brainware.System.AppConfiguration.dll** file in the <*Installation Folder*>**\Bin\bin** directory
2. Register this assembly in the GAC using the command:

   gacutil -I Brainware.System.AppConfiguration.dll

7.2.2.2 Encryption of the web.config file:

1. Use the aspnet_regiis command-line tool. This tool is located at:

   %WinDir%\Microsoft.NET\Framework\v2.0.50727\aspnet_regiis.exe

2. For encrypting a particular section of the configuration file, you can use the **-pe** option when executing the aspnet_regiis tool

   For example, for encryption of the **connectionStrings** section use:

   aspnet_regiis -pe connectionStrings -app/MyApp

Note: The **-app** option is used to specify the application's virtual path.

7.2.2.3 Decryption of the web.config file:

1. For decryption of a configuration section, use the following command:

   aspnet_regiis -pd connectionStrings -app/MyApp

7.3 Virus Check

Please note that the settings for the virus checker on the Web Server exclude the [Local Temp Folder]/CdrDbCache directory (Batch and the Common Learnset folders) from the locations, which are checked for viruses. This is due to performance considerations.

8 Security

8.1 WebCenter Forms Recognition Security

8.2 File System Security

Although WebCenter Forms Recognition does provide application-level security, the product relies on integrated Windows file system security built into the underlying operating system for file system access.

Forms Recognition uses operating system files (.sdp, .dat, .wdc, .sdb, etc.) to store all application and project data. A combination of shared and NTFS permissions are used to protect application data.

NTFS file and folder permissions are used to control the type of access that a user, group, or application has to folders and files. This includes everything from reading the contents of a folder or a file to modifying a folder's contents and/or executing individual files. There are five basic NTFS file permissions and six folder permissions:

File Permission	**Access Granted**
Read	Allows the user or group to read the file and view its attributes, ownership, and the permissions set.
Write	Allows the user or group to overwrite the file, change its attributes, view its ownership, and view the permissions set.
Read and Execute	Allows the user or group to run and execute the application. In addition, the user can perform all duties allowed by the Read permission.
Modify	Allows the user or group to modify and delete a file including performing all of the actions permitted by the Read, Write, and Read and Execute NTFS file permissions.
Full Control	Allows the user or group to change the permission set on a file, take ownership of the file, and perform actions permitted by all of the other NTFS file permissions.
Folder Permission	**Access Granted**
Read	Allows the user or group to view the files, folders, and subfolders of the parent folder. It also allows the viewing of the folder attributes, ownership, and permissions.
Write	Allows the user or group to create new files and folders within the parent folder, view folder ownership and permissions, and change folder attributes.
List Folder Content	Allows the user or group to view the files and subfolders contained within the folder.
Read and Execute	Allows the user or group to navigate through all files and subfolders, and to perform all actions allowed by the Read and List Folder Contents permissions.
Modify	Allows the user to delete the folder and perform all activities included in the Write and Read & Execute NTFS folder permissions.
Full Control	Allows the user or group to change permissions on the folder, take ownership of it, and perform all activities included in all other permissions.

The difference between NTFS file and folder permissions is the **List Folder Contents** folder permission. NTFS folder permissions enable system administrators to limit a user's ability to browse through a tree of folders and files. This is useful for securing a specific directory such as an application directory. A user must know the name and location of a file to read or execute it when this permission is applied to its parent

folder. However, in a WebCenter Forms Recognition environment, client applications in the product suite, instead of Windows Explorer, are used to process project data. The intent of file and folder permissions is to minimize the probability of accidental or malicious data destruction.

Shared permissions serve for purposes similar to NTFS permissions: They help protect files from unauthorized access. If you are a member of the Administrators or Power Users group, you can share folders on a local computer so that users on other computers can access those folders over the network. By assigning shared folder permissions to any shared folder, you can restrict or allow access to those folders over the network. Use NTFS folder permissions if the shared folder is located on a NTFS drive. NTFS permissions are effective on the local computer and over the network.

For more information regarding folder permissions, refer to *Appendix B: File Permissions Matrix*.

8.3 Accounts and File Access Security

Access to project data in a WebCenter Forms Recognition implementation should be granted using a combination of Discretionary Access Control (DAC) and Role-based Access Control (RBAC).

The Discretionary Access Control model allows the owner of objects or resources (in this context, a System Administrator) to control who accesses them and what operations they can perform. For example, a System Administrator who creates a share called **Projects** to hold data pertaining to a particular WebCenter Forms Recognition project can control and dictate (per the organization's security policy and business rules) who can access the items within the share.

The Role-Based Access Control model, also referred to as a non-discretionary model, makes access decisions based on the rights and permissions granted to a role or groups, instead of an individual. In this model, System Administrators create roles (or groups), and then assign rights and permissions to the role (or group) instead of directly to a user; users are then placed into a role (or group) and inherit the rights and permissions assigned to the role (or group).

The following table lists the recommended groups and accounts that should be created for each implementation of WebCenter Forms Recognition:

Group / Account Name	**Purpose**
Forms Recognition Project Users	Global group containing all users designated as a Forms Recognition project designer and/or data verifier within an organization.
Forms Recognition Admin	Global group containing all users designated as a Forms Recognition System Administrator within an organization. This group should be added to the local Forms Recognition group on all RTS servers and RTS Remote Admin workstations.
Forms Recognition	Local group used to grant access to local Forms Recognition resources; the Forms Recognition Admin global group should be added to its membership. Create this group on all Forms Recognition Server and RemoteAdmin machines
Forms Recognition Users	Local group used to grant access to the project directory. Add the global group Forms Recognition Project Users to its membership. Create this group on the Forms Recognition server housing the project directory.
Forms	Service account used to start the WebCenter Forms Recognition Service Manager. This

Recognition RTSsvc	user should be a member of the Forms Recognition Admin global group and the local Administrators group on all Forms Recognition servers and Remote administration machines.

The following table lists the groups and accounts, assigned permissions, and the folders/objects on which the permissions should be applied for each implementation of *WebCenter Forms Recognition*:

Group / Account Name	**Permission Type: Shared**	**Permission Type: NTFS**	**Folder/Objects Assigned On**
Forms Recognition	Full Control	Full Control	*<Project Path>*
Forms Recognition Users	Change	Modify	*<Project Path>*

For a comprehensive list of security settings and options, see *Appendix B: File Permission Matrix*.

8.4 Access to Project Data

WebCenter Forms Recognition uses a hierarchical file structure to store project-related data. The project directory is at the highest level of this structure.

All Forms Recognition components (including services, applications, license engine, and users) need appropriate access rights to the project directory and all of its subfolders.

See *Section 8.2: File System Security* above for details on how to enable access to project data.

Once Forms Recognition has been installed, configured, and prepared for production, appropriate file access security should be applied to the project directory before releasing the implementation to the general user community. A correct application of file access security can prevent unauthorized access to project data while granting access to authorized users.

To apply file access security to the Forms Recognition project directory:

1. Launch Windows Explorer on the server containing the project directory
2. Locate the project folder, right-click the folder name, and select **Properties**
3. In the **Properties** dialog box, go to the **Sharing** tab
4. Click **Share this folder**
5. In the **Share** name field, type a name for the share
6. Click **Permissions**. In the **Share Permissions** dialog box, do the following tasks, and then click **OK**:
   a. Add the local Forms Recognition group with **Full Control** permission
   b. Add the local Forms Recognition Users group with **Change** permission
   c. Add the local Administrators group with **Full Control** permission
   d. Remove the **Everyone** group
7. Go to the **Security** tab
8. Do the following tasks and click **OK** when finished:
   a. Add the local Forms Recognition group with **Full Control** permission

b. Add the local Forms Recognition Users group with **Change** permission
c. Add the local Administrators group with **Full Control** permission
d. Remove the **Everyone** group

Note: The Forms Recognition and Forms Recognition Users groups are local groups. The Forms Recognition local group should be created on all WebCenter Forms Recognition servers and RemoteAdmin machines; the Forms Recognition Users local group is only required on the server storing the project data. For an explanation of these groups, see *Section 7.4: Accounts and File Access Security*.

8.5 Configuring the Service Account for WebCenter Forms Recognition

8.5.1 Running WebCenter Forms Recognition on a Domain Network

WebCenter Forms Recognition Runtime Server service utilizes a Windows service that runs in the server background. This service manages the operation of Runtime Server instances, and processing of documents automatically.

When running Forms Recognition on multiple servers located on a domain network, it is recommended that the WebCenter Forms Recognition Runtime Server service be assigned a domain user against the Windows service. This will allow Forms Recognition to communicate with all servers running the service across the domain.

The service account used in WebCenter Forms Recognition is also given permission to any file/folder shares across the servers to allow the Runtime Server service access to all project related files.

Note: Do not use the service account to log into the system, either locally or through Remote Desktop Connection, Configure the Security Settings for the “Deny log on locally” and “Deny log on through Remote Desktop Services” policies in Windows on the system running the services.

8.5.2 About the service account for System Monitoring

The System Monitoring service is used to send email notification to selected users to notify of any errors, or warnings, that any Runtime Server instance may raise during its operation.

The service user account used for System Monitoring should have sufficient rights to be able to send emails on the server and domain.

9 Configuring Runtime Component

Runtime Service Manager (RTS) must be configured before using the WebCenter Forms Recognition.

The RTS Remote Administration Microsoft Management Console (MMC) snap-in enables you to start and stop multiple WebCenter Forms Recognition Runtime Servers remotely from a single workstation on the network. The WebCenter Forms Recognition installation creates a default console, called Runtime Server Administration that you can use to configure the RTS Remote Administration MMC snap-in.

9.1 Configuring the Runtime Service Manager Service

Below are the steps required for configuring the Runtime Service Manager service. Administrator rights are required to perform these steps:

1. Select **Start** → **Administrative Tools** → **Services**
2. Locate and double-click the **WebCenter Forms Recognition Runtime Service Manager**
3. On the **General** tab, under **Startup type**, select **Automatic** from the drop-down list
4. Go to the **Log On** tab
5. Under **Log on as**, select **This account**
6. Click **Browse…**
7. Find and add the domain user with appropriate and sufficient permissions for Forms Recognition processing network access rights (e.g. WFR RTSsvc), and then click OK
8. Type the domain password for the user in the fields provided
9. Click **OK**, and then close the Services window

9.2 Configuring the RTS RemoteAdmin MMC Snap-in

The installation of WebCenter Forms Recognition creates a default console, called WebCenter Forms Recognition Service Manager that you can use to configure the Forms Recognition RTS RemoteAdmin MMC snap-in.

Important: Before configuring the RemoteAdmin MMC snap-in, make sure that the steps outlined in *Section 8.1: Configuring the Runtime Service Manager* above have been performed and the WebCenter Forms Recognition Runtime Service Manager is started. Unless the service has been started, the MMC will not connect to the machine.

1. Make sure the WebCenter Forms Recognition Service Manager service is running. This lets you connect by MMC to the machine.
2. Identify one free configurable port available in any TCP/IP network or the internet across firewalls.

Note: The default port number is 50607.

3. Verify one of the following prerequisites.
    a. The administration workstation resides on the same LAN segment as the RTS services.
    b. In a subnet network, a name resolution system is in place to allow clients on one subnet to locate resources on another subnet.
4. Launch the WebCenter Forms Recognition Service Manager MMC snap-in by selecting **Start** → **Programs** → **Oracle** → **WebCenter Forms Recognition** → **Runtime Service** → **Management Console** on the desktop of the target machine
5. Right-click the **Runtime Server Administration** node and select **New RTS Group** from the context menu
6. On the **New Group** dialog, type a group name and click **OK**
7. Expand the **Runtime Server Administration** node
8. Right-click the group you created, and then select **New Machine…**

Note: There may be a short pause before the Group Management dialog is displayed. This is because the RemoteAdmin MMC Snap-in is searching the domain for machines where the WebCenter Forms Recognition Runtime Service Manager service is running.

9. In the **Domains** dropdown, select the domain where the machine being configured is located
10. Select or enter the name of the WebCenter Forms Recognition server to be added and click **OK**
11. Right-click on the machine name and select **New** → **RTS Instance…**
12. On the **New RTS Instance** dialog, type the instance name and then click **OK**

For information on how to configure project settings for a Forms Recognition instance, see the *WebCenter Forms Recognition Runtime Server User's Guide*.

9.3 Running Multiple Web Verifier and RTS Instances

9.3.1.1 Problem Description

When running more than approximately 12 concurrent Web Verifier user sessions, or more than approximately 14 Runtime Service instances, the system may start experiencing lack of Windows desktop heap resources and the extra user sessions or RTS instances can fail with different internal memory allocation errors.

9.3.1.2 Problem Cause

The default Windows OS setting of desktop heap size for the non-interactive Windows station often appears to be too low to host multiple simultaneously running Web Verifier or Runtime Service instances with extensive script engine utilization.

9.3.1.3 Recommended Configuration Changes

1. Open Windows Registry Editor
2. Browse to the key:

   `[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\SubSystems\Windows]`
3. Check the third argument of the **SharedSection** parameter:

   `SharedSection=1024,3072,`**`512`**

   512 (KB) is the default value that causes the issue as described in the problem description above.

   Note that for some OS versions, this default setting can be different. The table below gives the recommended values for this setting:

Number of RTS and/or Web Verifier Instances*	**Desktop Heap Size in KB**
1 - 10	512
11 - 24	1024
25 - 36	1536
37 - 48	2048
49 - 60	2560
61 - 72	3072

   * This represents the total number of simultaneously running Runtime Server instances **plus** the number of concurrent Web Verifier users running on the same physical server. For example, a server running 4 RTS instances with 8 concurrent Web Verifier users connected would total 12, so the "11 – 24" setting would be observed.
4. After modifying this parameter, restart the server.

9.4 Advanced Logging

The standard Runtime Server log includes system level resource information and, in case of system failure, special error logs.

9.4.1 Application Log files

Each log file contains the following information.

Entry #	**Description**
1	Type of message (info, warning, error, etc)
2	Severity of message
3	Time logged
4	Process ID (PID)
5	Overall used/available physical memory in KB
6	Overall used/available virtual memory in KB
7	Used physical/virtual memory by this Runtime Manager in KB
8	Process handles used by this Runtime Manager
9	GDI resources/UserObjects used by this Runtime Manager
10	Message text

In the <*Installation Folder*>**\Bin\bin\Log** folder, the log files for the different WebCenter Forms Recognition components can be found as follows:

Log file name	Description
V_ *.log	For Verifier messages, e.g. any custom script errors
VA_*.log	For Advanced Verifier messages
VL_*.log	Local Verifier messages.
H_ *.log	Runtime Server Host (Dst Host) messages for a single RTS instance.
L_*.log	For Learnset Manager messages, e.g. when the user triggers document learning, or when a backup of the Learnset is taken, etc.
D_*.log	Designer messages including script errors
U_*.log	Web Verifier and external application messages
S_*.log	Service Manager (DstMgr) messages, such as start and end of service, restart, or failures.
I_*.log	Component log files for all applications.
M_*.log	System Monitoring (DstEvent) messages. Holds all system messages and can log error messages across all server machines and hosts.

Error log files

In the event of system or application failures, WebCenter Forms Recognition creates an additional error log file named C_<Process ID>*_yyyymmdd*.log

10 Enabling Additional OCR Engine Languages

WebCenter Forms Recognition supports many OCR engine languages. The following table lists down the OCR engine languages supported by FineReader 10 and 11.

Supported OCR languages	h2	h3
Bulgarian	Italian	Romanian
Chinese Simplified *	Japanese *	Russian *
Chinese Simplified + English *	Japanese + English *	Slovak
Czech	Korean *	Slovenian
Danish	Korean + English * [Only with FineReader 11]	Spanish
Digits	KoreanHangul	Swedish
Dutch	Latvian	Turkish
Estonian	Norwegian	Ukranian *
Finnish	NorwegianBokmal	Vietnamese *
French	NorwegianNynorsk	CMC7
German	Polish	E13B
Greek *	Portuguese Brazilian	
Hungarian	Portuguese Standard	

* These languages require support of double byte and extended ASCII character sets. To avoid performance loss, do not use more than one DBCS language in a project.

10.1 Enabling a Language for an OCR Engine

Note: If not already done, you first have to enable the support of double byte and extended ASCII character sets, (Greek, CJK, Russian, Hebrew) for your system, as described in *Section 10.2: Adding an Input Language for Windows 7 or Windows Server 2008* below.

A language can only be processed by WebCenter Forms Recognition if it is installed on the server machine, and if it is present in the FineReader directory:

1. Exit all Forms Recognition applications
2. On the Forms Recognition servers, stop the WebCenter Forms Recognition Runtime Server services
3. Ensure the appropriate FineReader language file(s) are present in the folder:

   *<Installation Folder>***\Components\Cairo\FineReaderX** on the Forms Recognition server for FineReader 10, or *<Installation Folder>***\Components\Cairo\FineReaderX1\Data** for FineReader 11

   If necessary, the FineReader language files can be copied from the WebCenter Forms Recognition installation media, from the folder:

*<Installer Root>***Bin\WLPostInstall\LangFile\FineReaderX** for FineReader 10, or *<Installer Root>***Bin\WLPostInstall\LangFile\FineReaderX1\Data** for FineReader 11

4. Restart the WebCenter Forms Recognition Runtime Server services
5. Restart the client application

10.2 Adding an Input Language for Windows 7 or Windows Server 2008

1. Select **Start** → **Control Panel** → **Clock, Language, and Region** → **Region and Language**
2. Click the **Keyboards and Languages** tab
3. Click **Change keyboards…**
4. Under Installed services, click **Add…**
5. Expand the language item you want to add, then expand **Keyboard**
6. Select the input languages you want to add
7. Click **OK** to confirm

Appendix A Web.config Options and Associated Resource File Parameters

The table below contains some items that can be modified in the *<Installation Folder>***\WebCenter Forms Recognition Web Server\web.config** file with regards to enabling, disabling or otherwise customizing certain features.

Option	Description
ADOCommandExecutionTimeOut	Optional attribute. Timeout in seconds for database stored procedures execution. If not specified timeout from database connection string is used. `<client ADOCommandExecutionTimeOut=10></client>`
AllowAccessToDocumentsToIndexOnly	This option controls whether navigation is enabled only for documents for indexing (those with states from enabled workflow input states). This option only takes effect when "Disable navigation to valid documents" is set to *True* in settings. When set to *False* (or not included in web.config) WVC works as before allowing navigation to out-of-workflow documents. **Default Value:** `false`
BatchViewPageSize	The number of batches to display on Web Verifier in the batch list. Any batches exceeding that count are divided into other navigation pages. The default value is **20**, allowing for up to 20 batches to be shown in the Web Verifier batch list. `<add key="BatchViewPageSize" value="20" />`
ClientSideDocumentCacheSize	A parameter that allows the number of pages to cache in the current document. **Default Value:** `0`

connectionStrings	Configuration connects to database.
	`<connectionStrings>`
	`<add name="Entities" connectionString="…" providerName="System.Data.EntityClient" />`
	`</connectionStrings>`

DialogWidth	It defines the width of message boxes in pixels **Default Value:** `400`
document cacheSize in <document.controller> element	It specifies the number of workdoc objects to pre-load. This accelerates opening documents within the batch. Pre-loading cannot be disabled but can minimize the number of pre-loaded documents to 2, that means one current and one pre-loaded. **Default Value:** `5`
document maxPagesToPreload in the <document.controller> element	It defines the number of document pages to pre-load. First and last pages always pre-load, and remaining cache slots fill with pages that have field candidates starting from the lower index. The following actions take place on page images when a document loads in the background. ▪ Pre-load the page ▪ Convert the page to PNG

	▪ Save the page to the database **Default Value:** `5`
DocumentViewPageSize	The number of folders to display in the Document Tree view, when selecting *Show Selected Batch*. The default value, 4, denotes 4 folders to display in *Show Selected Batch* view; any additional batches are shown in subsequent navigation panels. `<add key="DocumentViewPageSize" value="4"/>` **Default Value:** `10`
EnableProfiler	Enables the Web Verifier profiler. `<add key="EnableProfiler" value="true|false" />` The profiler collects and records time taken by user actions such as commands and their internal sub-operations. **Default Value:** `False`
externalGroupIdColumn	Whether the external group ID column displays in Web Verifier **Default Value:** `False`
externalBatchNameColumn	Whether the external batch name column displays in Web Verifier **Default Value:** `False`
focusChanged	Whether to enable the focusChanged event for fields in the verification view. The event will be triggered when the user presses the ENTER key in a field. This setting has no effect on the FocusChanged event even if the <mouseClicked> attribute is set to true **Default Value:** `False`
HelpLink	`Links to Web Verifier Help`
httpHeaderBasedSso	Enables the Single Sign-On (SSO) user authentication using an HTTP header sent by an SSO service. The header value contains the SSO authenticated user name. This parameter is used by the SSO service to send logged in user name. Setting the *enabled* attribute to **true** enables the Web Verifier SSO feature. **Note:** On a system using standard authentication, the *enabled* setting should be set to **false**. The *loginHeader* value corresponds to the HTTP header attribute name that contains the SSO authenticated user. *sessionHeader* is SSO service dependent. For example, the *headerName* parameter is **ShibSessionId** for a Shibboleth provider. Example: `<httpHeaderBasedSso loginHeader="remoteUser" enabled="true" sessionHeader="ShibSessionID" />`
inactiveUserTimeout	Required attribute. It is not used to control user session timeout. The user session timeout is controlled by the <sessionState Timeout. parameter.
inspectionTimeOut	**Required attribute.** It is not used to control user session timeout. The user session timeout is controlled by the <sessionState Timeout. parameter. **Default Value:** `00:05:00`

instanceName	The name of the Web module that will be shown to have access to the batch list. `<client instanceName="Web Verifier"></client>`
itemCopied	Whether to enable itemCopied event. **Default Value:** `False`
LanguageDisplayName_[ISO]	This optional set of keys is located in the <appSettings> section of the web.config file. The [ISO] part of the key needs to be replaced by the three letters name of the folder in the Web Verifier resources folder. This key can be used to customize the language display names of the language selection drop-down menu. To display the correct string for Simplified Chinese, the following should be added: `<add key="LanguageDisplayName_ZHO" value="□□□□"/>` **Note:** As soon as this parameter is defined it will override the system language name.
licensePath	The location of the shared license file, reference documentation regarding configuration. This should point to the License Share file. `<project licensePath="C:\Program Files (x86)\Oracle\WebCenter Forms Recognition\License\Runtime.lic" mpdDistance="19" mpdThreshold="60" />`
level	It defines the tracing level **DEBUG**: Trace all information and error messages. **ERROR**: Trace error messages only.
loadInSeparateProcess	Required attribute. Read only. The value is **true** only.
mouseClicked	Whether to enable the mouseClicked event of fields and tables in the verification view in indexing mode. **Default Value:** `False`
pathToProjectExe	The location of the Forms Recognition Designer module (DstDsr.exe). **Default Value:** `pathToProjectExe=" C:\Program Files (x86)\ Oracle\WebCenter Forms Recognition\Bin\Bin\`
ReinitScriptEngineAfterScriptErrors	By default, this attribute is always set to false. This will recover the script engine whenever a script error occurs in Web Verifier application. **Default Value:** `false`
remoteObjectRenewalTimeout	Optional attribute. Remote object references are renewed at this time period (in seconds). Defaults to 180. Minimum accepted value is 30. The lower the number the faster unused objects free memory but this can lead to errors for long running commands. One can increase this value if some actions (i.e. field validation) take a while to finish with remoting error. Note that this value should be set in both web application config file and Brainware.System.Project.exe config file `<client remoteObjectRenewalTimeout=45></client>` **Default Value:** `30`
SavePageImageToDatabase	It specifies if page images extracted from document file blobs needs to be saved back to the database.
sessionState Timeout	The *sessionState Timeout* parameter controls the timeout for a user

	session. The value represents the number of minutes that a user is allowed to be inactive before the session is ended. **Note:** Session timeout should be set to a value less than that of the SSO session. Refer to the product documentation of your SSO provider for details how to configure the SSO session duration. **Default Value:** `20`
ShowExtendedErrorMessages	Set this attribute to true to enable stack trace information in the error messages appearing in Web Verifier. Messages are written to the Trace Log file. Allowable values are **true** and **false**. `<add key="ShowExtendedErrorMessages" value="true"/>` **Default Value:** `true`

slogan	A text message that can be displayed on the Web Verifier browser header with corporate messages / announcements / Corporate Slogan.
	`<verifier.webclient>`
	`<company slogan="This is a corporate message" />`

SYSTEM_LONG_DATE_FORMAT	This special XML key is located in each resources /[3-letters language]/Brainware.Verifier.WebClient.resx file.
	
	This key contains the date formatting pattern for the last access date column presentation within the batch list for that language. It is optional, and if set to *Empty*, then the system default formatting is applied.
	
	For Chinese Traditional and Simplified languages, the date format to be used is YYYY-MM-DD, and the time format is 24 based without any Chinese characters.
	
	Example for the Chinese:
	`<data name="SYSTEM_LONG_DATE_FORMAT" xml:space="preserve">`
	`<value>yyyy-MM-dd, hh:mm:ss</value>`
	`</data>`

transactionIdColumn	Whether the transaction ID batch column displays in Web Verifier **Default Value:** `false`
transactionTypeColumn	Whether the transaction type batch column displays in Web Verifier **Default Value:** `false`
tabPressed	Whether to enable the tabPressed event on fields and tables in the verification view in indexing mode. **Default Value:** `false`
tabCellSelected	Whether to enable the tableCellSelected event. **Default Value:** `false`
usePath	**Required attribute.** Enable/disable using pathToProjectExe parameter. Set this attribute to false to set pathToProjectExe parameter is current directory. **Default Value:** `true`
waitLoadTimeOut	**Required attribute.** Timeout for initial loading of project.exe. This parameter is used with enable option: loadInSeparateProcess = true **Default Value:** `00:01:00`

Appendix B File Permission Matrix

The table below displays the various file permissions that are used within WebCenter Forms Recognition.

Role / Group	Description
Administrators	Administrator user with full access rights to all application modules and features.
Developers	The groups of users that develop, maintain, and enhance projects.
Learnset Manager	Typically one user in the organization responsible for maintaining the project learnsets.
Advanced Verifiers	Several users responsible for identifying documents for improvements to the project learnset.
Standard Verifiers	Data entry clerks responsible for document correction.
RTS Service User	The service account responsible for running the service for automatic document processing. Configured only on the server machines.

Directory	Groups	NTFS Permissions						
		Full Control	Modify	Read & Execute	List Folder Content	Read	Write	No Access
Root Batch Folder	Administrators Developers Learnset Manager Advanced Verifiers Standard Verifiers RTS Service User	●	●	●	●	●	●	
Common Folder	Administrators Developers Learnset Manager Advanced Verifiers	●	●	●	●	●	●	
	Standard Verifiers RTS Service User							●
Global Project and Global Learnset	Administrators Developers Learnset Manager RTS Service User	●	●	●	●	●	●	
	Advanced Verifiers Standard Verifiers			●	●	●		
Local Project and Local Learnset	Administrators Developers Advanced Verifiers	●	●	●	●	●	●	
	Learnset Manager RTS Service User Standard Verifiers							●

ASE Pool	Administrators Developers RTS Service User	●	●	●	●	●	●	
	Learnset Manager Advanced Verifiers Standard Verifiers			●	●	●		
ASSA CSV File	Administrators Developers RTS Service User	●	●	●	●	●	●	
	Learnset Manager Advanced Verifiers Standard Verifiers							●

Appendix C Registry Options

The table below contains some items that can be modified in the Registry concerning enabling/disabling/customizing certain features.

Option	Description
ErrorTraceDir	The ErrorTraceDir registry key is available for those customers who wish to place the component tracing logs in a different location than the default *<Installation Folder>***\Bin\bin\Log** folder. The registry key allows the administrator to place the logs in a specific folder location separate from the core product logs. The registry setting is only applicable for the component logs, not for the core product logs. To configure a new location for Component Logs, follow the instructions outlines below: 1. Launch Windows Registry Editor 2. Navigate to: HKEY_LOCAL_MACHINE\SOFTWARE \[Wow6432Node \]Oracle\ErrorTrace 3. Create a new REG_SZ (String Value key) and call it **ErrorTraceDir** 4. Modify the new key created and enter the filepath location for component logs to be entered. Verify that the path entered exists and the service account/user has sufficient permissions to write to that location. For the change to take place, exit all WebCenter Forms Recognition applications, and stop any services running on the machine related to Forms Recognition, then launch the application and all new component logs will be written in the desired location.
HideBatchReleaseDialog	This key allows Oracle Support to disable the Batch Release dialog box within the Verifier, where the business does not require prompting users on next task. The registry value can be used to determine the next action carried out by users. The default action of the Batch Release dialog box is to verify the next invalid batch. When the dialog is suppressed, this value is maintained. To change to a different action, use the Batch Release dialog box once, then change the setting accordingly and click OK. To create the registry key to suppress the Batch Release confirmation screen, follow the instructions below: 1. Launch Windows Registry Editor 2. Navigate to: HKEY_LOCAL_MACHINE\SOFTWARE \[Wow6432Node \]Oracle\Cedar 3. Create a new REG_DWORD (DWORD Value key) and call it **HidebatchReleaseDialog** 4. Modify the new key created and enter the one of the following values: 0 - to enable the confirmation screen (default) 1 - to disable/ hide the confirmation screen For the changes to take place, exit all Forms Recognition applications, and then launch the applications again. To view that the change has been implemented: 1. Launch Verifier 2. Verify the batch to completion – no dialog box should appear
All	The ErrorTrace registry Key was introduced into core product logs to

	provide additional trace information on any errors or warnings in the system. The default value after installation is to record errors only related details. Modify the registry values to set the value from **0** to either **1**, **2**, or **3**: 1 - Only Errors (Default) 2 - Errors & Warnings 3 - Errors & Warnings & Information To configure ErrorTrace All value: 1. Launch Registry Editor 2. Navigate to: HKEY_LOCAL_MACHINE\SOFTWARE \[Wow6432Node \]Oracle\ErrorTrace 3. Create a new DWORD registry variable for **All**, set to the appropriate value of either **0**, **1**, **2**, or **3** 4. Close the Registry Editor
MaximumDiskspaceUsageMB	This registry value controls the amount of disk space allocated for component level logs on this server / workstation in MB. Setting this value to **0** has the same effect as if the value is not created at all, which is “deactivated”. To configure the maximum disk space usage: 1. Launch Registry Editor 2. Navigate to: HKEY_LOCAL_MACHINE\SOFTWARE \[Wow6432Node \]Oracle\ErrorTrace 3. Create a new DWORD registry variable for **MaximumDiskspaceUsageMB**, set to the appropriate value in MB 4. Close the Registry Editor
TotalDaysToKeepFiles	This registry value maintains the number of days the old component level logs are kept by the Forms Recognition server. Setting this value to **0** has the same effect as if the value is not created at all, which is “deactivated”. To configure the number of days to keep files: 1. Launch Registry Editor 2. Navigate to: HKEY_LOCAL_MACHINE\SOFTWARE \[Wow6432Node \]Oracle\ErrorTrace 3. Create a new DWORD registry variable for **TotalDaysToKeepFiles**, set to the appropriate number of days to maintain logs 4. Close the Registry Editor