Blueprint 8.0

Manual Upgrade Guide

2/27/2017

blueprint

Contents

Blueprint Manual Upgrade Guide

Overview

This *Blueprint Manual Upgrade Guide* is applicable if you have already installed Blueprint and you want to manually upgrade to a newer version. If you want to install Blueprint for the first time, please refer to the *Blueprint Installation Guide*. If you want to automatically upgrade Blueprint instead (recommended), see the *Blueprint Upgrade Guide*.

Important Notices

Upgrade duration

This upgrade includes data-migration processes that will lock Blueprint until they have completed (the login screen and REST API will not be available). In most cases, these processes will require less than 30 minutes; however, for very large Blueprint deployments, this process may take several hours.

Backing up your data

It is strongly recommended that you back up your Blueprint data prior to starting the upgrade.

64-bit application pool

To upgrade to v7.0, the application pool on the Blueprint web server must be set to 64-bit:

1. Open IIS Manager.
2. In the *Connections* pane, click Application Pools
3. Right-click the Blueprint application pool and click **Advanced Settings**
4. Ensure *Enable 32-bit Applications* is set to *False*

Installing services

Job services can be manually installed if functionality for job execution and/or HP ALM legacy support is needed. For more information on installing job services, see Installing services.

> **Tip:** Job services are already installed and configured if **Job Services** is listed as an option in the *Instance Administration Console* within Blueprint. HP ALM is already installed if the 32-bit Blueprint service is listed in the *Services* window on your system (*Administrative Tools*).

System Requirements

Please refer to the *Blueprint Installation Guide* for detailed information about Blueprint System Requirements.

blueprint

Upgrade Steps (update)

Step 1: Back up your data

Warning: It is strongly recommended that you back up your database and web site before starting the upgrade.

To back up your data before performing an upgrade:

1. Stop the Blueprint application pool and Blueprint web site.

   Note: Your Blueprint application pool and Blueprint web site may have different names, depending on what you chose during installation.

2. Stop any installed components, including:

   - Job services
   - The HP ALM application pool and web site (formerly known as HP QC)

   Note: This step only applies if any job services or the HP ALM web site have been installed.

3. Back up the following data:

   - Blueprint database
   - Blueprint website, including:
     - the **`web.config`** files
       - Blueprint web.config
       - AdminStore web.config
       - FileStore web.config
       - AccessControl web.config
       - ConfigControl web.config
     - logging configurations (**`Logging.config`** files for the above config files)
     - any customized logo files (for example, **`customlogo.png`**)
     - any custom dictionary files (**`dictionary.dct`**)

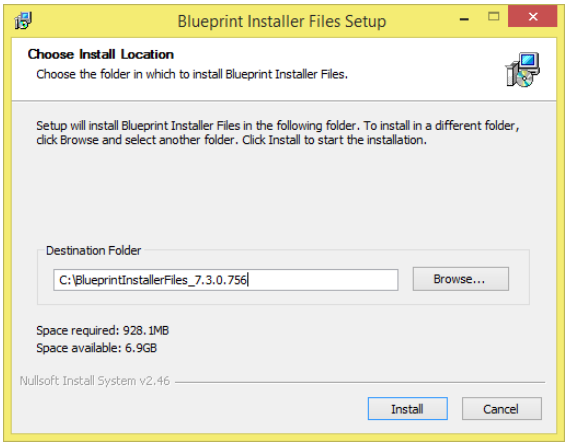
- **JobExecutorService** folder. This includes the **`BluePrintSys.RC.JobExecutor.exe.config`** for each job service that has been installed. It also includes the **`BluePrintSys.RC.JobExecutor32.exe.config`** if you have configured 32-bit job services to support HP Quality Center.

Step 2: Deploy the setup files to the web application server

The recommended installation procedure is to use the Blueprint Configuration Wizard to create a Blueprint settings file that will be used with an installation script.

Run the **`BlueprintInstallerFiles`** executable file to extract the installer files.

The following Installer Files Setup Wizard appears:



Note: The file extraction location should be different from the where Blueprint will be installed.

Click **Install** to extract the files.

After this process is complete, click **Close**.

Step 3: Upgrade the Blueprint web site and create the auxility site

Here is a summary of the steps that must be completed to configure the web application:

- Create the Application Pools:
  This step involves creating two Application Pools, one for each of the two sites' set of services.
- Create the Sites:
  This step involves creating the auxiliary site, and adding application pools to the Blueprint site.
- Configure the key and connection strings:
  This step tells the application where the databases are located and how to communicate with them.

Below is an example of the commands that can be run to configure the web application:

Important: The commands must be run with Administrator privileges on the web application server. Refer to the Configuration Utility Command Reference section for more information about these commands.

1. Create the Application Pool for the primary site:

```
blueprintwebcmd.exe /object APPPOOL /command ADD /apppoolname
[Blueprint_PrimaryServices] /userid [acme\rrunner] /password
[pAsswOrd]
```

2. Upgrade the primary site:

```
blueprintwebcmd.exe /object SITE /command UPGRADE /wsname [Blueprint]
/serviceapppoolname [Blueprint_PrimaryServices]
```

3. Create the Application Pool for the auxiliary site and services:

```
blueprintwebcmd.exe /object APPPOOL /command ADD /apppoolname
[Blueprint_AuxiliaryServices] /userid [acme\rrunner] /password
[password]
```

4. Configure the connection string for the auxiliary site:

```
blueprintwebcmd.exe /object AUXSERVICEDBCONFIG /command SET
/adminstoragedatasource [DBSERVER\INSTANCE01]
/adminstoragecatalogprefix [Blueprint] /wsname Blueprint_Auxiliary
```

> **Note**: Your AdminStorage database must have the `_AdminStorage` suffix. This suffix is automatically added to the value provided with the `/adminstoragecatalogprefix` command-line argument.

5. Create the auxiliary site:

```
blueprintwebcmd.exe /object SITE /command ADDAUX /wsname [Blueprint_
Auxiliary] /port [9101] /dir ["C:\Program Files\Blueprint Software
Systems\Blueprint\Web_Auxiliary"] /apppoolname [Blueprint_
AuxiliaryServices]
```

Step 4: Create the new service databases

Here is a summary of the steps that must be completed to configure the database:

- Create the database:
  These steps involve creating a shell for each of the databases (FileStore, which stores files such as artifact attachments, and AdminStore, which is an interface to administrative functionality). Each is a blank container and does not contain any content.
- Initialize the database:
  These steps create the database schemas and pre-populate the databases with the data that is necessary to start using Blueprint.

Below is an example of the commands that can be run to configure the databases:

> **Important**: The commands must be run with Administrator privileges on the web application server.

> **Note**: If you want to use *Windows Authentication* instead of database authentication, you can set the `integratedsec` parameter to *TRUE* in the commands below and omit the `userid` and `password`

parameters. Refer to the Configuration Utility Command Reference for more information about these commands.

1. Create the AdminStore database:

```
blueprintdbcmd.exe /object ADMINSTOREDB /command ADD /datasource
[DBSERVER\INSTANCE01] /catalogprefix [BlueprintDB] /integratedsec
FALSE /userid [dbadmin] /password [pAssw0rd]
```

Note: The `ADD` command automatically appends `_AdminStorage` to the database name.

2. Initialize the AdminStore database:

```
blueprintdbcmd.exe /object ADMINSTOREDB /command INIT /datasource
[DBSERVER\INSTANCE01] /catalogprefix [BlueprintDB] /integratedsec
FALSE /userid [dbadmin] /password [pAssw0rd]
```

Note: The catalog that will be updated is `[BlueprintDB]_AdminStorage`.

3. Create the FileStore database:

```
blueprintdbcmd.exe /object FILESTOREDB /command ADD /datasource
[DBSERVER\INSTANCE01] /catalogprefix [BlueprintDB] /integratedsec
FALSE /userid [dbadmin] /password [pAssw0rd]
```

Note: The `ADD` command automatically appends `_FileStorage` to the database name.

4. Initialize the FileStore database:

```
blueprintdbcmd.exe /object FILESTOREDB /command INIT /datasource
[DBSERVER\INSTANCE01] /catalogprefix [BlueprintDB] /integratedsec
FALSE /userid [dbadmin] /password [pAssw0rd]
```

Note: The catalog that will be updated is `[BlueprintDB]_FileStorage`.

5. Stop and then restart the Blueprint application pool and Blueprint web site.

Note: Your Blueprint application pool and Blueprint web site may have different names, depending on what you chose during installation.

Step 5: Upgrade web.config

If you are installing v7.0 for the first time, modify the Blueprint web.config so that it is aware of the auxiliary services.

1. Configure the keys in the Blueprint web.config files:

```
blueprintwebcmd.exe /object SERVICEKEYCONFIG /command SET /wsname
[Blueprint] /auxSiteUrl [localhost] /auxsitePort [9101]
```

> **Note**: `http://` and the port will be joined to the `auxSiteUrl` value automatically.

2. Configure the connection string in the primary services' web.config files:

```
blueprintwebcmd.exe /object SERVICEDBCONFIG /command SET /wsname
[Blueprint] /bpdatasource [DBSERVER\INSTANCE01] /bpcatalog [Blueprint]
/adminstoragedatasource [DBSERVER\INSTANCE01]
/adminstoragecatalogprefix [Blueprint] /filestoragedatasource
[DBSERVER\INSTANCE01] /filestoragecatalogprefix [Blueprint]
/isnewinstallation [true]
```

> **Note**: Your AdminStorage and FileStorage databases must respectively have the `_AdminStorage` and `_FileStorage` suffix. The suffix is automatically added to the value provided with the respective `/adminstoragecatalogprefix` and `/filestoragecatalogprefix` command-line argument.

3. Configure the connection string for the auxiliary site:

```
blueprintwebcmd.exe /object AUXSERVICEDBCONFIG /command SET /wsname
[Blueprint_Auxiliary] /adminstoragedatasource [DBSERVER\INSTANCE01]
/adminstoragecatalogprefix [Blueprint]
```

> **Note**: Your AdminStorage database must have the `_AdminStorage` suffix. This suffix is automatically added to the value provided with the `/adminstoragecatalogprefix` command-line argument.

Step 6: Add users to the databases

Add security by adding the Blueprint Server User to the databases and granting appropriate permissions. Consider the following examples:

```
blueprintdbcmd.exe /object USER /command ADD /datasource
[DBSERVER\INSTANCE01] /catalog [BlueprintDB]_AdminStorage /integratedsec
FALSE /userid [dbadmin] /password [pAssw0rd] /nuseridentity LocalService
```

```
blueprintdbcmd.exe /object USER /command ADD /datasource
[DBSERVER\INSTANCE01] /catalog [BlueprintDB]_FileStorage /integratedsec
FALSE /userid [dbadmin] /password [pAssw0rd] /nuseridentity LocalService
```

The database that the user requires access to is through the corresponding application pool. The `/nuseridentity` parameter is the identity of the application pool

Application Pool	Database
Blueprint	Blueprint
Blueprint_AuxiliaryServices	Blueprint_AdminStorage
Blueprint_PrimaryServices	Blueprint Blueprint_AdminStorage Blueprint_FileStorage.

Step 7: Upgrade the Blueprint database and web site

1. Run the following command:

   Important: The command must be run with Administrator privileges.

   ```
   blueprintdbcmd.exe /object DB /command UPGRADE /catalog [Blueprint]
   /datasource [DATABASESERVER\INSTANCE01] /integratedsec FALSE /userid
   [dbadmin] /password [pAssw0rd]
   ```

2. Integrate any custom edits from the **`web.config`** backup into the new **`web.config`** file by merging the two files.

   Some parameters may include the connection string, ALM configuration path(s) and/or timeouts.

   For more information on editing **`web.config`** parameters, see the *IT Administration Guide.*

3. Integrate any custom edits from the **`Logging.config`** backup into the new **`Logging.config`** file by merging the two files.
4. Integrate any custom edits from the **`BluePrintSys.RC.JobExecutor.exe.config`** backup into the new **`BluePrintSys.RC.JobExecutor.exe.config`** file for any installed job service.
5. Copy and paste any custom **`dictionary.dct`** files into new `dictionary` folder.
6. Stop and then restart all installed job services.
7. Restart the HP Quality Center application pool and web site.
8. Restart the Blueprint application pool and Blueprint web site.

   Note: Your Blueprint application pool and Blueprint web site may have different names, depending on what you chose during installation.

Step 8: Run the HTML Migration Utility

1. Locate the **`HTMLMigrationUtility.exe.config`** file within your Blueprint installation.
   This file can be found in the **`\Blueprint\Setup\HtmlMigrationUtility\`** directory.
2. Update the **`HtmlMigrationUtility.exe.config`** file to point to your database.
   In the **`<connectionStrings>`** section, add the **`connectionString`** value as found in your Blueprint **`web.config`** file.
3. Run **`HtmlMigrationUtility.exe`**

Note: When executed, this process will lock Blueprint until it has completed (the login screen and REST API will not be available). In most cases, this process will require less than 30 minutes; however, for very large Blueprint deployments, it may take several hours.
If the process was interrupted before completion, it can be rerun.

Step 9: Run the Global Search Indexing Utility

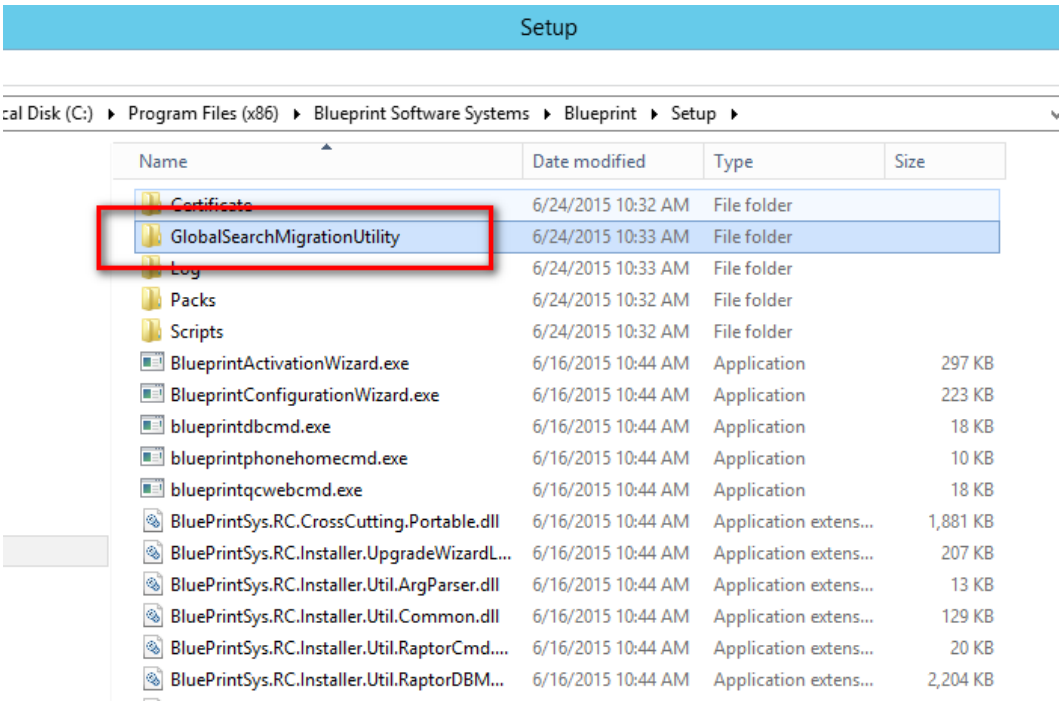
Note: The HTML Migration Utility, as outlined in the previous step, must be completed before proceeding with running the Global Search Indexing utility.

Note: This Step must be completed if you are upgrading from a version of Blueprint earlier than 6.2

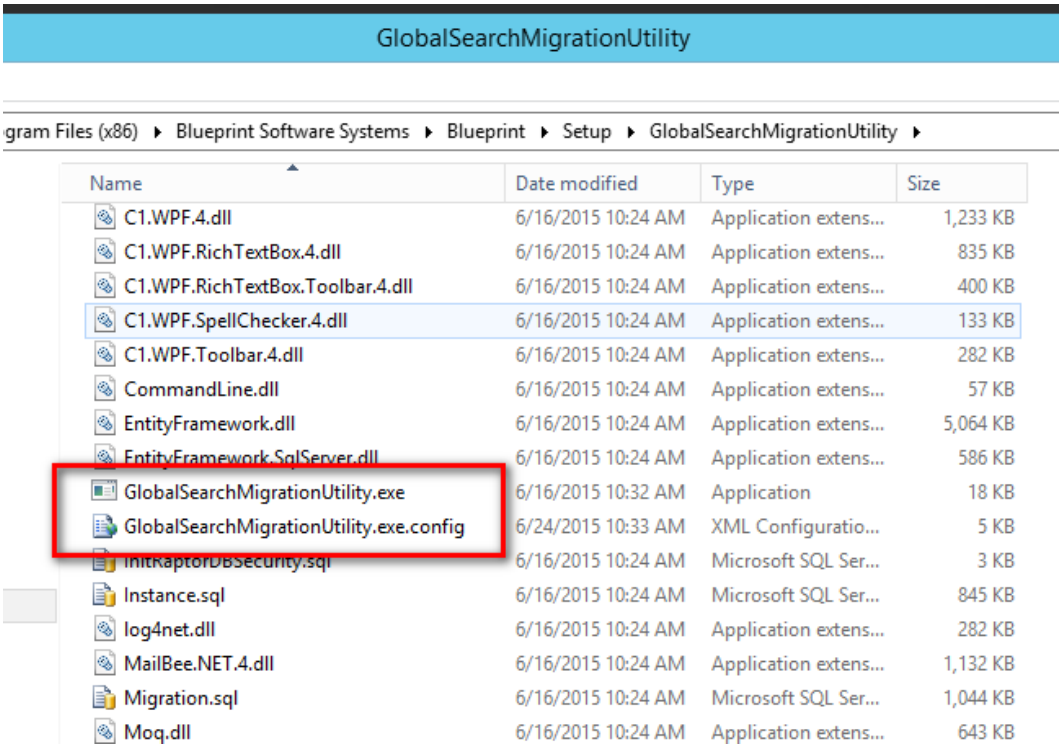
Run the Global Search Indexing Utility on the database

Once the Blueprint Database has successfully been upgraded, it must be indexed for Global Search.

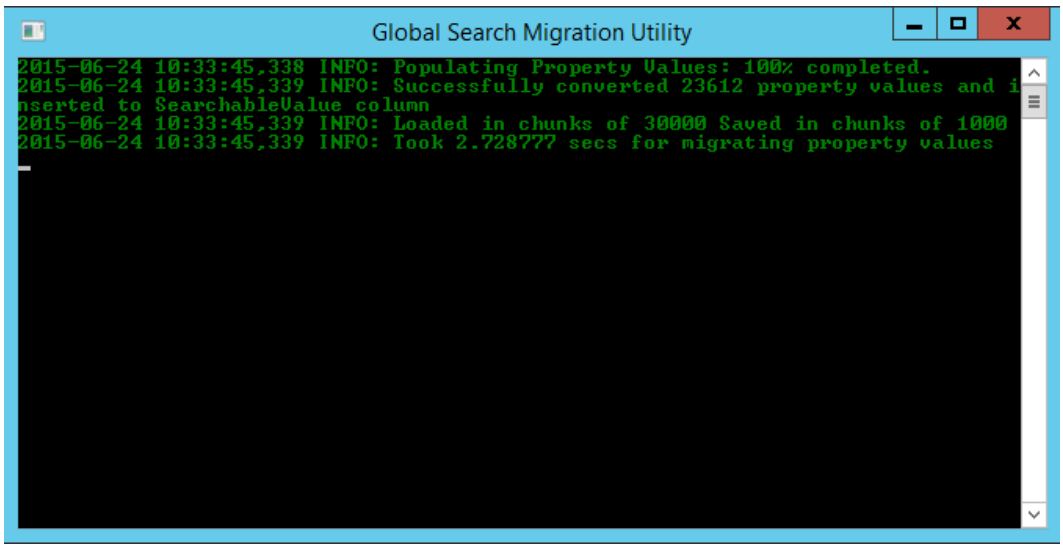
1. Navigate to the folder where Blueprint was installed.



2. Update **`GlobalSearchMigrationUtility.exe.config`** with the connection string from web.config.



3. Run **`GlobalSearchMigrationUtility.exe`**



Note: The utility may run for several hours depending on the size of the database.

Step 10: Run the FileStore Migration Utility

Note: The HTML Migration Utility, as outlined in the previous step, must be completed before proceeding with running the FileStore migration utility.

To improve robustness and scalability, the FILESTREAM feature was deprecated in v7.0. Migration to the new Blueprint_FileStorage database is handled by the FileStore Migration Utility. The migration will keep your old data intact. This step is now mandatory for this release; FILESTREAM data will be removed after upgrading.

1. Locate the **`FileStoreMigrationUtility.exe.config`** file within your Blueprint installation. This file can be found in the **`\Blueprint\Setup\FileStoreMigrationUtility\`** directory.
2. Update the **`FileStoreMigrationUtility.exe.config`** file to point to your database. In the **`<connectionStrings>`** section, add the **`connectionString`** value as found in your Blueprint **`web.config`** file.
3. Run **`FileStoreMigrationUtility.exe`**

> **Note**: When executed, this process will lock Blueprint until it has completed (the login screen and REST API will not be available). If the process is interrupted before completion, it can be rerun, and will resume with the next file to migrate. In most cases, this process will require less than 10 minutes; however, for very large Blueprint deployments, it may take many hours (for example, in an unoptimized test environment, 125 GB of data, representing 275,000 objects, required over 12 hours).

The utility, by default, will migrate all contents from FILESTREAM to the FileStorage database to completion. Users with policy-based maintenance windows can use the **`-d <duration>`** attribute and value to define a migration interval, in minutes. In the following example, the utility will continue to migrate files for 30 minutes (that is, it will potentially *begin* migrating a new file up to the 30-minute mark):

```
FileStoreMigrationUtility -d 30
```

Step 11: Unlock IIS configuration sections

Use the AppCmd to modify your IIS configuration by unlocking the appropriate sections. In the Windows command prompt, run the following commands, replacing **`Blueprint_Auxiliary`** with the name of your Auxiliary site:

> **Important**: The command must be run with Administrator privileges.

```
%windir%\system32\inetsrv\appcmd.exe unlock config Blueprint_Auxiliary
/section:system.webServer/security/authentication/windowsAuthentication
/commit:apphost
```

```
%windir%\system32\inetsrv\appcmd.exe unlock config Blueprint_Auxiliary
/section:system.webServer/security/authentication/basicAuthentication
/commit:apphost
```

```
%windir%\system32\inetsrv\appcmd.exe unlock config Blueprint_Auxiliary
/section:system.webServer/security/authentication/digestAuthentication
/commit:apphost
```

```
%windir%\system32\inetsrv\appcmd.exe unlock config Blueprint_Auxiliary
/section:system.webServer/security/authentication/anonymousAuthentication
/commit:apphost
```

Step 12: Configure SearchService and ArtifactStore

Change SearchService and ArtifactStore to services in IIS

1. Open IIS Manager.
2. In the *Connections* pane, expand the server that is running Blueprint.
3. Expand the svc folder, found under the Blueprint site.
4. In the svc folder, right-click the SearchService folder, and select **Convert to Application**.
5. In the Add Application window, click **Select...**
6. In the Select Application Pool window, select *Blueprint_PrimaryServices*, then click **OK**.
7. Click **OK**.
8. Repeat the above steps to convert the ArtifactStore folder to an application that uses the same *Blueprint_PrimaryServices* application pool.

Modify the SearchService web.config

1. Right-click the SearchService application, then click **Explore**.
2. Open **`web.config`** in a text editor.
3. In the **`<appSettings>`** section, ensure the **`AccessControl`** and **`ConfigControl`** values are the same as those in the AdminStore **`web.config`** file.
4. In the **`<connectionStrings>`** section, modify the **`<add name="Blueprint">`** key, so that its **`connectionString`** value matches that of the **`<add name="RaptorMain">`** key in the AdminStore **`<web.config`** file.

Modify the ArtifactStore web.config

1. Right-click the ArtifactStore application, then click **Explore**.
2. Open **`web.config`** in a text editor.
3. In the **`<connectionStrings>`** section, modify the **`<add name="RaptorMain">`** key, so that its **`connectionString`** value matches that of the **`<add name="RaptorMain">`** key in the AdminStore **`<web.config`** file.

Step 13: Configure the Blueprint client

Configuring elevated trust in-browser

Blueprint must be configured to run with elevated trust in-browser before you can use some advanced features, such as:

- screen capture capabilities
- pasting images into diagrams
- Visio integration, such as importing and exporting diagrams
- rich text table integration with other applications

Elevated trust in-browser can be configured manually on each client machine, or the configurations can be pushed to Windows computers in a centralized manner.

Using group policy to push the elevated trust-in browser configurations to Windows computers in a centralized manner

The recommended way to configure Blueprint to run with elevated trust in-browser is to use Group Policy. Group Policy allows IT Administrators to push configurations to Windows computers in a centralized manner.

For overview information about Group Policy, refer to Microsoft's Group Policy documentation at: http://technet.microsoft.com/en-us/windowsserver/bb310732.aspx.

To configure Blueprint to use elevated trust in-browser, your Group Policy configuration must do the following:

1. Set the following registry value:

   `HKEY_LOCAL_MACHINE\Software\Wow6432Node\Microsoft\Silverlight\AllowElevatedTrustAppsInBrowser`

   should be set to

   0x00000001.

   To learn more about setting a registry value through Group Policy, please refer to the Microsoft documentation that explains how to configure a registry item at: http://technet.microsoft.com/en-us/library/cc753092.aspx.

2. Download the elevated trust in-browser package from the Blueprint Customer Portal.
3. Add the **`publicBlueprintCertificate2017.cer`** certificate to the Trusted Publishers Store.

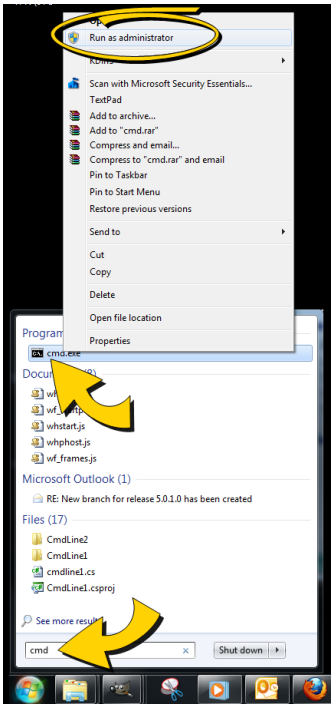
   To learn more about adding a certificate through Group Policy, please refer to the Microsoft documentation that explains how to deploy certificates by using group policy (http://technet.microsoft.com/en-us/library/cc770315(v=ws.10).aspx).

Manually configuring a computer to run with elevated trust in-browser

To configure Blueprint to run with elevated trust in-browser, perform the following steps on each client:

1. Download the elevated trust in-browser configuration files.
2. Unzip the package and note the directory where the files are located.
3. Run **`cmd.exe`** as Administrator.

    1. Click the Windows **Start** menu and type **cmd.exe** into the search bar.
    2. Right-click the **cmd.exe** program that appears under the Programs heading and then select **Run as administrator**:



3. When the confirmation dialog appears, click **Yes**.

   The **cmd.exe** application launches with Administrator privileges:

4. Use the **cd** command to navigate to the folder where you unzipped the files.

   For example:

   ```
   cd c:\temp\elevated_trust
   ```

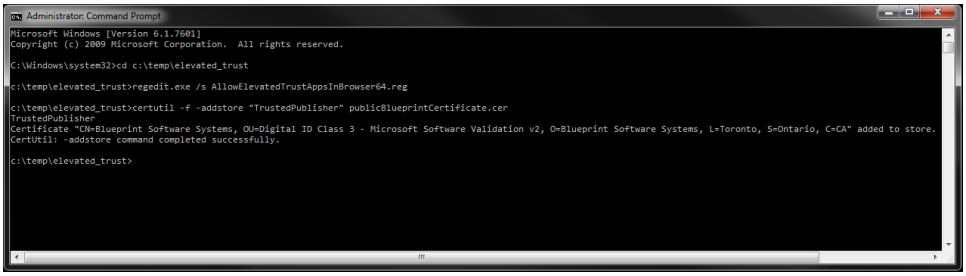
5. Enter the following commands to allow elevated trust to run on your local machine:

   ```
   regedit.exe /s AllowElevatedTrustAppsInBrowser64.reg
   ```

6. Run the following `certutil` command to apply the Blueprint public certificate:

   ```
   certutil.exe -f -addstore "TrustedPublisher"
   publicBlueprintCertificate2017.cer
   ```

   Here is an example of the commands run on a 64-bit operating system:



7. Restart your web browser for the changes to take effect.

Appendix

Configuration utility command reference

Tip

You can type the following commands to view more information about the command parameters:

```
blueprintwebcmd.exe /help
blueprintdbcmd.exe /help
blueprintqcwebcmd.exe /help
```

Web Application Server Configuration Parameters (`blueprintwebcmd.exe`)

Parameter	Description	Default	Example
/object	Defines the object type of the command. This parameter can be set to one of the following values: ▪ SITE - administration of web sites ▪ APPPOOL - administration of application pools ▪ DBCONFIG - administration of database connection configuration sections ▪ SERVICEKEYCONFIG - administration of key configuration sections ▪ SERVICEDBCONFIG - administration of database connection configuration in services		

Parameter	Description	Default	Example
/command	Defines the command to perform. This parameter can be set to one of the following values: ▪ LIST - list the existing objects in the server ▪ ADD - add new objects to the server ▪ ADDAUX - add new objects to the auxiliary server ▪ DELETE - delete the object from the server ▪ START - start the object instance on the server ▪ STOP - stop the object instance on the server ▪ SET - sets the specific configuration values (currently supported only by DBCONFIG object)		
/wsname	Defines the name of the site. This should be the same as the application pool name.	Blueprint	Blueprint
/wsid	Defines the ID of the site.		25
/port	Defines the port number used for the site.		8080
/auxSiteUrl	Defines the ID of the site.	Blueprint_ Auxiliary	Blueprint_Auxiliary
/auxSitePort	Defines the port number used by the auxiliary site.		9101
/httpHostname	Hostname for the site		
/https	Flag specifying SSL install of the site		
/sslcert	Name of IIS server certificate		
/sslcertstore	Name of IIS certificate store		
/dir	Defines the location of the Blueprint installation.		C:\Program Files\Blueprint\Web
/apppoolname	Defines the name of the application pool.	Blueprint	Blueprint
/serviceapppoolname	Defines the name of the primary services application pool.	Blueprint_ PrimaryServices	Blueprint_ PrimaryServices
/datasource	Defines your database and instance names.		DBSERVER\INSTANCE01
/catalog	Defines the name of the database.	Blueprint	

Parameter	Description	Default	Example
/adminstoragedatasource	Defines the AdminStorage database.		
/adminstoragecatalogprefix	Defines the initial catalog prefix for AdminStorage. (Note the _ AdminStorage suffix is automatically added to this value, and should be omitted from the command-line value.)		Blueprint (which becomes Blueprint_ AdminStorage)
/filestoragedatasource	Defines the FileStorage database.		
/filestoragecatalogprefix	Defines the initial catalog prefix for FileStorage. (Note the _FileStorage suffix is automatically added to this value, and should be omitted from the command-line value.)		Blueprint (which becomes Blueprint_FileStorage)
/integratedsec	Defines whether or not Windows security is used. This parameter can be set to one of the following values: ▪ TRUE ▪ FALSE If /integratedsec is set to FALSE, you must specify a /userid and /password.		
/userid	Defines the username of the Service Account/Application Pool user.		
/password	Defines the password of the Service Account/Application Pool user.		
/multisite	Flag used for multi-site deploy		

Database Server Configuration Parameters (`blueprintdbcmd.exe`)

Parameter	Description	Default	Example
/object	Defines the object type of the command. This parameter can be set to one of the following values: ▪ DB - administration of database ▪ FILESTOREDB - administration of FileStore database ▪ ADMINSTOREDB - administration of AdminStore database ▪ USER - administration of database user		

Parameter	Description	Default	Example
/command	Defines the command to perform. This parameter can be set to one of the following values: ▪ ADD - add new object to the server ▪ INIT - initialize the object. Currently supported only for DB object. ▪ UPGRADE - upgrade the object. Currently only supported for DB object.		
/datasource	Defines your database and instance names.		DBSERVER\INSTANCE01
/catalog	Defines the name of the service database.	Blueprint	BlueprintDB
/catalogprefix	Defines the initial catalog prefix for the service database. (Note the _AdminStorage suffix is automatically added to this value, and should be omitted from the command-line value.)		Blueprint (which becomes Blueprint_ AdminStorage)
/integratedsec	Defines whether or not Windows security is used. This parameter can be set to one of the following values: ▪ TRUE ▪ FALSE If /integratedsec is set to FALSE, you must specify a /userid and /password.		
/userid	Defines the username of the *Database System Administrator* user. This parameter is only required if /integratedsec is set to FALSE.		
/password	Defines the password of the *Database System Administrator* user. This parameter is only required if /integratedsec is set to FALSE.		
/nuseridentity	Defines the username of the *Blueprint Server User*.		acme\rrunner

HP ALM Legacy Connector Configuration Parameters (`blueprintqcwebcmd.exe`)

Parameter	Description	Default	Example
/object	Defines the object type of the command. This parameter can be set to one of the following values: ▪ SITE - administration of web sites ▪ APPPOOL - administration of application pools ▪ DBCONFIG - administration of QC connection configuration sections		

Parameter	Description	Default	Example
/command	Defines the command to perform. This parameter can be set to one of the following values: ▪ LIST - list the existing objects in the server ▪ ADD - add new object to the server ▪ DELETE - delete the object from the server ▪ START - start the object instance on the server ▪ STOP - stop the object instance on the server ▪ SET - sets the specific configuration values (currently supported only by CONFIG object)		
/wsname	Defines the name of the site. This should be the same as the application pool name.		HPALMLegacyConnector
/wsid	Defines the ID of the site.		25
/port	Defines the port number used for the site.		8081
/httpHostname	Defines the host name for the site.		
/dir	Defines the location of the connector installation.		C:\Program Files\Blueprint\QcLiteWeb
/apppoolname	Defines the name of the application pool. This should be the same as the site name.		HPALMLegacyConnector
/userid	Defines the username of the Service Account/Application Pool user.		
/password	Defines the password of the Service Account/Application Pool user.		
/multisite	Flag used for multi-site deployment.		

Phone Home Configuration Parameters (`blueprintphonehomecmd.exe`)

Parameter	Description	Default	Example
/usageinfo	Defines whether usage information is being sent to Blueprint. This parameter can be set to one of the following values: ▪ TRUE ▪ FALSE		
/systeminfo	Defines whether system information is being sent to Blueprint. This parameter can be set to one of the following values: ▪ TRUE ▪ FALSE		
/wsname	Defines the name of your Blueprint web site.		
/userid	Defines the username of the *Database System Administrator* user.		
/password	Defines the password of the *Database System Administrator* user.		

Installing services

Note: Only Blueprint database users can manually install the Windows services. For more information about how to add a database user, see Adding a database user (command line).

The following Windows services are available to install if needed:

- **Job services (recommended)**
  This functionality is necessary to perform the following jobs in Blueprint: document generation, exporting artifacts to ALM systems and test generation.
- **Legacy support for HP ALM versions 12 and earlier (optional)**
  If you require support for HP ALM version 12 or earlier, setup of this component is necessary.

  Note: HP ALM support is only available for COM library. It is not available for the REST API.

To install the Windows-based services:

- Continue to the single-server section to host job services and the Blueprint database on the same server.
- Continue to the distributed-server section to host job services and the Blueprint database on separate servers.

Adding a database user (command line)

You can add any of the following database user types using the `/nuseridentity` parameter:

- Local Service: **LocalService**
- Local System: **LocalSystem**
- Network Service: **NetworkService**
- A Windows services account. Example: **blueprint\jsmith**.

Add a database user:

```
blueprintdbcmd.exe /object USER /command ADD /datasource
[DBSERVER\INSTANCE01] /catalog [BlueprintDB] /integratedsec FALSE /userid
[dbadmin] /password [pAssw0rd] /nuseridentity LocalService
```

Setting up services (single-server setup)

You can set up one or both of the following:

- 64-bit job services
- HP ALM legacy support

Setting up 64-bit job services (single-server)

Installing 64-bit job services involves completing the following steps:

- Step One: Configuring 64-bit job services
- Step Two: Finalizing the 64-bit job services setup

Step One: Configuring 64-bit job services

1. Open the following configuration file in a text editor: **BluePrintSys.RC.JobExecutor.exe.config**
2. Make sure that the value in the `<add key="Service.Jobs">` tag is as follows:

```
<add key="Service.Jobs" value="
DocGen,TfsExport,HpAlmRestExport,TfsChangeSummary,HpAlmRestChangeSumma
ry,TfsExportTests,HpAlmRestExportTests" />
```

3. Make sure the `<add key="Service.Name">` tag specifies the 64-bit job service as follows:

```
<add key="Service.Name" value="Blueprint Job Service (64 bit)" />
```

> **Note:** If you are setting up the second 64-bit job service in this step, specify the `Service.Name` value as `Blueprint Job Service 2 (64 bit)` instead of the value listed above.

4. Replace the following `connectionString` value with the Blueprint database connection string:

```
<connectionStrings>
<add name="InstanceContainer" connectionString="
metadata=res://*/Models.Instance.csdl|res://*/Models.Instance.ssdl|res
://*/Models.Instance.msl;provider=System.Data.SqlClient;provider
connection string=&quot;Data Source=.\MSSQLSERVER;Initial
Catalog=Blueprint;Integrated
Security=True;Pooling=True;MultipleActiveResultSets=True&quot;"
providerName="System.Data.EntityClient" />
</connectionStrings>
```

> **Note:** `Data Source` must specify the SQL instance name and the Blueprint instance name. If your SQL instance has a name that is different from **`MSSQLSERVER`** and/or your Blueprint instance is not named **`Blueprint`**, you need to change the value(s).

Step Two: Finalizing the 64-bit job services setup

1. Install both 64-bit job services by running the following command with your user name and password:

```
BluePrintSys.RC.JobExecutor.exe -c Install -a User -u [USERNAME] -p
[PASSWORD]
```

> **Note:** To install the service using the default Windows account, run the following command instead:
>
> `BluePrintSys.RC.JobExecutor.exe -c Install -a LocalService`

2. Start the 64-bit job services with the following command:

```
BluePrintSys.RC.JobExecutor.exe -c Start
```

You have successfully deployed the 64-bit job services.

> **Important:** Repeat steps one and two to configure the second 64-bit job service.

Setting up HP ALM legacy support (single-server)

Installing HP ALM legacy support involves completing the following steps:

- Step One: Setting up the HP ALM legacy support connector
- Step Two: Configuring 32-bit job services for HP ALM legacy support
- Step Three: Finalizing the HP ALM legacy support setup

Step One: Setting up the HP ALM legacy support connector

1. Set up the HP ALM application pool by running the following command:

```
blueprintqcwebcmd.exe  /object AppPool /command ADD /apppoolname
HPQCLegacyConnector
```

2. Set up the HP ALM web site by running the following command (where the number after `/port` is `QcLiteWeb`'s port number):

```
blueprintqcwebcmd.exe  /object Site /command ADD /wsname
HPQCLegacyConnector /dir "C:\Program Files\Blueprint Software
Systems\Blueprint\QcLiteWeb" /port [8081] /apppoolname
HPQCLegacyConnector
```

3. Start the HP ALM application pool by running the following command:

```
blueprintqcwebcmd.exe  /object AppPool /command START /apppoolname
HPQCLegacyConnector
```

4. Start the HP ALM web site by running the following command:

```
blueprintqcwebcmd.exe  /object SITE /command START /wsname
HPQCLegacyConnector
```

5. Set the HP ALM key for Blueprint with the following command (where the number after `/port` is `QcLiteWeb`'s port number):

```
blueprintqcwebcmd.exe /object Config /command SET /dir "C:\Program
Files\Blueprint Software Systems\Blueprint\Web" /port [8081]
```

You have successfully set up the HP ALM legacy support connector.

Step Two: Configuring 32-bit job services for HP ALM legacy support

1. Open the following configuration file in a text editor: **BluePrintSys.RC.JobExecutor32.exe.config**
2. Make sure that the value within `<add key="Service.Jobs">` is as follows:

```
<add key="Service.Jobs" value="QcExport,QcChangeSummary,QcExportTests"
/>
```

3. Make sure the `<add key="Service.Name">` tag includes the Blueprint HP ALM legacy job service as follows:

```
<add key="Service.Name" value="Blueprint HP QC Legacy Job Service (32 bit)" />
```

4. Replace the following `connectionString` value with the Blueprint database connection string:

```
<connectionStrings>
<add name="InstanceContainer" connectionString="
metadata=res://*/Models.Instance.csdl|res://*/Models.Instance.ssdl|res://*/Models.Instance.msl;provider=System.Data.SqlClient;provider connection string=&quot;Data Source=.\MSSQLSERVER;Initial Catalog=Blueprint;Integrated Security=True;Pooling=True;MultipleActiveResultSets=True&quot;"
providerName="System.Data.EntityClient" />
</connectionStrings>
```

> **Note:** `Data Source` must specify the SQL instance name and the Blueprint instance name. If your SQL instance has a name that is different from **`MSSQLSERVER`** and/or your Blueprint instance is not named **`Blueprint`**, you need to change the value(s).

Step Three: Finalizing the HP ALM legacy support setup

1. Install the 32-bit service by running the following command with your user name and password:

```
BluePrintSys.RC.JobExecutor32.exe -c Install -a User -u [USERNAME] -p [PASSWORD]
```

> **Note:** To install the service using the default Windows account, run the following command instead:
>
> `BluePrintSys.RC.JobExecutor32.exe -c Install -a LocalService`

2. Next, start the 32-bit job executor Windows service with the following command:

```
BluePrintSys.RC.JobExecutor32.exe -c Start
```

You have successfully set up legacy support for HP ALM.

Setting up Windows-based services (distributed-server setup)

Setting up services on a separate machine (distributed-server) involves these steps:

- Step One: Configuring services
- Step Two: Testing the connection to the database
- Step Three: Finalizing the job services setup

Step One: Configuring services

You have the option of setting up the following services:

- 64-bit job services
- HP Quality Center legacy support

Configuring the 64-bit services

1. Open the following configuration file in a text editor: **BluePrintSys.RC.JobExecutor.exe.config**
2. Make sure that the value in the `<add key="Service.Jobs">` tag is as follows:

```
<add key="Service.Jobs" value="
DocGen,TfsExport,HpAlmRestExport,TfsChangeSummary,HpAlmRestChangeSumma
ry,TfsExportTests,HpAlmRestExportTests" />
```

3. Make sure the `<add key="Service.Name">` tag specifies the 64-bit job service as follows:

```
<add key="Service.Name" value="Blueprint Job Service (64 bit)" />
```

> **Note:** If you are setting up the second 64-bit job service in this step, specify the `Service.Name` value as `Blueprint Job Service 2 (64 bit)` instead of the value listed above.

4. Replace the following `connectionString` value with the Blueprint database connection string:

```
<connectionStrings>
<add name="InstanceContainer" connectionString="
metadata=res://*/Models.Instance.csdl|res://*/Models.Instance.ssdl|res
://*/Models.Instance.msl;provider=System.Data.SqlClient;provider
connection string=&quot;Data Source=.\MSSQLSERVER;Initial
Catalog=Blueprint;Integrated
Security=True;Pooling=True;MultipleActiveResultSets=True&quot;"
providerName="System.Data.EntityClient" />
</connectionStrings>
```

> **Note:** `Data Source` must specify the SQL instance name and the Blueprint instance name. If your SQL instance has a name that is different from **`MSSQLSERVER`** and/or your Blueprint instance is not named **`Blueprint`**, you need to change the value(s).

> **Important:** Repeat all of the steps above to configure the second 64-bit job service.

Setting up HP ALM legacy support

Setting up legacy support for HP ALM involves the following steps:

- Step One: Setting up the HP ALM legacy support connector
- Step Two: Configuring the 32-bit job services

STEP ONE: SETTING UP THE HP ALM LEGACY SUPPORT CONNECTOR

1. Set up the HP ALM application pool by running the following command:

```
blueprintqcwebcmd.exe  /object AppPool /command ADD /apppoolname
HPQCLegacyConnector
```

2. Set up the HP ALM web site by running the following command (where the number after `/port` is `QcLiteWeb`'s port number):

```
blueprintqcwebcmd.exe  /object Site /command ADD /wsname
HPQCLegacyConnector /dir "C:\Program Files\Blueprint Software
Systems\Blueprint\QcLiteWeb" /port [8081] /apppoolname
HPQCLegacyConnector
```

3. Start the HP ALM application pool by running the following command:

```
blueprintqcwebcmd.exe  /object AppPool /command START /apppoolname
HPQCLegacyConnector
```

4. Start the HP ALM web site by running the following command:

```
blueprintqcwebcmd.exe  /object SITE /command START /wsname
HPQCLegacyConnector
```

5. Set the HP ALM key for Blueprint with the following command (where the number after `/port` is `QcLiteWeb`'s port number):

```
blueprintqcwebcmd.exe /object Config /command SET /dir "C:\Program
Files\Blueprint Software Systems\Blueprint\Web" /port [8081]
```

You have successfully set up the HP ALM legacy support connector.

STEP TWO: CONFIGURING THE 32-BIT JOB SERVICES

Complete the following instructions, pasting the copied files onto your target machine:

1. Open the following configuration file in a text editor: **BluePrintSys.RC.JobExecutor32.exe.config**
2. Make sure that the value within `<add key="Service.Jobs">` is as follows:

```
<add key="Service.Jobs" value="QcExport,QcChangeSummary,QcExportTests"
/>
```

3. Make sure the `<add key="Service.Name">` tag includes the Blueprint HP ALM legacy job service as follows:

```
<add key="Service.Name" value="Blueprint HP QC Legacy Job Service (32
bit)" />
```

4. Replace the following `connectionString` value with the Blueprint database connection string:

```
<connectionStrings>
<add name="InstanceContainer" connectionString="
metadata=res://*/Models.Instance.csdl|res://*/Models.Instance.ssdl|res
://*/Models.Instance.msl;provider=System.Data.SqlClient;provider
connection string=&quot;Data Source=.\MSSQLSERVER;Initial
Catalog=Blueprint;Integrated
Security=True;Pooling=True;MultipleActiveResultSets=True&quot;"
providerName="System.Data.EntityClient" />
</connectionStrings>
```

> **Note:** `Data Source` must specify the SQL instance name and the Blueprint instance name. If your SQL instance has a name that is different from **`MSSQLSERVER`** and/or your Blueprint instance is not named **`Blueprint`**, you need to change the value(s).

Step Two: Testing the connection to the database

> **Note:** This step must be performed for each job service you intend to set up.

This testing procedure must validate the configuration of the following values:

- The `Service.Name` value is unique and no other job service has this name on the current machine
- The `connectionString` value is valid and the job service can connect to Blueprint database.

If the test is not successful, you must specify the correct value(s) in the job service configuration file and re-attempt the test.

To test the job services connection to the Blueprint database:

- Run the following command (where `[JobServiceExecutableFile]` is the name of the job service executable file):

```
[JobServiceExecutableFile] -c test
```

> **Note:** The 32-bit job services file name is typically **BluePrintSys.RC.JobExecutor32.exe** and the 64-bit job services file name is typically **BluePrintSys.RC.JobExecutor.exe**.

Step Three: Finalizing the job services setup

> **Note:** This step must be performed for each job service you intend to set up.

This is the final step in setting up job services on a separate machine from the Blueprint database.

To finalize the setup:

1. Install the job service by running the following command (where `[JobServiceExecutableFile]` is the name of the job service executable file):

```
[JobServiceExecutableFile] -c install
```

2. Start the job service by running the following command (where `[JobServiceExecutableFile]` is the name of the job service executable file):

```
[JobServiceExecutableFile] -c start
```

You have deployed job services.

To verify whether job services have been successfully installed and configured:

1. Log on to Blueprint.
2. Open the *Instance Administration Console* from the **Menu** 

.

   The *Instance Administration Console* appears.
3. Click **Job Services**.

   The *Job Services* screen appears.

Any job services that have been successfully installed and configured appear in the *Job Services* list. Information about the configured service name and supported jobs also is available in the list.