

precisely

EngageOne Designer

Installation and Configuration Guide

Version 6.6 Service Pack 11



Table of Contents

1 - Introduction

About EngageOne Designer.....	4
The repository.....	4
Securing sensitive information.....	5

2 - Hardware and software requirements

Hardware requirements.....	7
Software requirements.....	8

3 - Installing and configuring Designer

About the installers	14
Running the client install in silent mode.....	16
Deployment scenarios.....	17
Client/Server and Distributed considerations.....	22
Configuring for application virtualization.....	24
Installing and configuring advanced search support.....	26
Configuring project services.....	46

4 - Start-up & login

Logging into Designer.....	49
Resources for samples /confidence tests.....	49

5 - Repository selection

Repository selection.....	51
---------------------------	----

6 - Preventative maintenance and best practices

Best practices.....	53
---------------------	----

1 - Introduction

This manual describes how to install and configure EngageOne Designer. It is primarily intended for system administrators responsible for deploying EngageOne Designer.

In this section

About EngageOne Designer.....	4
The repository.....	4
Securing sensitive information.....	5



About EngageOne Designer

Designer consists of the design editors and publication tools plus a repository within which all design resources are managed. Access to all system features and control of the workflow is carried out within the Designer client.

Depending on your license configuration you can create a multi-user Client/Server environment, or a Distributed environment where the Designer is installed on a separate machine to your SQL Server.

In the Client/Server environment the client software is copied to a network location, the server control programs are installed for use by the local machine, and resource repository is created (or referenced). Client machines must then install from this location.

The Distributed environment is similar to the Client/Server setup, except that your existing SQL Server is used and the Designer can be installed on a different machine to the SQL Server.

For more details about environments see the required setup type under “Installation” on page 56.

The repository

The first installation of Designer at a site must create a new resource repository. This uses Microsoft SQL Server as its database. If you already have SQL Server, or SQL Server Express you can create your repository in the existing environment.

Note: that SQL Server Express database is not recommended for a client/server installation with more than five users.

The repository consists of a database and a filestore. The database is used to provide revision control for the repository and will be in the default location as specified in the SQL Server. The filestore is where the design files and resources are stored, and is specified as a folder location.

If you are re-installing the software make sure the Upgrade option is checked in the Install wizard. This will ensure that the existing repository is preserved correctly.

Setting the option to compact the database will defragment the database and rebuild its index. This will reduce the size of the database, but can take some time, especially for large repositories. Be aware that performance might also be affected.

Securing sensitive information

Encoding of sensitive information such as passwords, connection strings, etc., is implemented in Designer using the following encryption algorithms:

- Designer login passwords are hashed using the SHA512 algorithm; the hash codes are stored in the Designer database.
- The database connection string in the application server configuration file is encrypted with the Windows Data Protection API (DPAPI) using the Triple-DES algorithm.
- The **EncryptPassword.exe** utility uses AES128 encryption for values in the Advanced Search settings file.

2 - Hardware and software requirements

This section lists requirements in terms of hardware and software need to run the Designer software.

In this section

Hardware requirements.....	7
Software requirements.....	8



Hardware requirements

The following list shows the minimum requirements. As with most software, the more resources you have available the more quickly and effectively Designer will run.

	Designer client	Application Server	SQL Server
Processor	2.4 GHz	4x 2.4 GHz	4x 2.4 GHz
Hard disk	4 GB	50 GB	50 GB
RAM	1 GB	4 GB	8 GB
Monitor	20"		
Graphics	1600 x 1200		
Colors	24 bit true		

Software requirements

Microsoft Windows

Designer is supported on the following Windows 64 Bit platforms only:

- Microsoft Windows Server 2019 Standard
- Microsoft Windows Server 2016 Standard
- Microsoft Windows Server 2012 R2 Update (KB2919355)
- Microsoft Windows 10
- Microsoft Windows 8.1 Update (KB2919355)

Microsoft SQL Server

Designer is supported only with the latest available Cumulative Update of 64 Bit Microsoft SQL Server:

- Microsoft SQL Server 2017
- Microsoft SQL Server 2016 SP2
- Microsoft SQL Server 2014 SP3
- Microsoft SQL Server 2012 SP4

Refer to the Microsoft support site for details and installation downloads of the latest SQL Cumulative Update release for each SQL Server version.

Note: It is recommended that you use Microsoft SQL Server as your design repository database.

SQL Server Collation

The Designer SQL Server database collation must be set to **SQL_Latin1_General_CP1_CI_AS** before Designer installation. No other SQL Server collation is supported.

Designer Performance on Windows Server 2016 / Windows 10

By default, Windows Defender scans each file copy / execution in the directories listed below. As a result, a performance degradation will be experienced compared to earlier Windows versions.

To optimise performance on Windows Server 2016 / Windows 10, the following directories should be added to the Windows Defender exclusion list.

- Designer Filestore share
- Designer Server directory

- Designer Client directory
- Designer Publish Localcache

Required Software

Designer requires the following components, which will be installed if necessary. If a later version is detected, the component will **not** be installed.

Designer Client client/server	*Microsoft Visual C++ 2015-2019 Redistributable Microsoft .NET Framework 4.8 Resource Access Service
Application Server client/server, distributed	Microsoft .NET Framework 4.8 Microsoft Shared Management Objects for SQL Server Microsoft System CLR Types for SQL Server Resource Access Service
SQL Server distributed	Microsoft Shared Management Objects for SQL Server Microsoft System CLR Types for SQL Server

*Also includes the Windows Update for Universal C Runtime in Windows (KB2999226), which requires that other updates have already been installed. See <https://support.microsoft.com/en-gb/kb/2999226> for details.

NOTE: By default, the Resource Access component is installed in the Windows **Program Files** folder. If you wish to install the Resource Access Service elsewhere, the installer for this component can be found on the release media.

Ghostscript

If you are going to use EPS images or the external documents feature to import PDF or EPS, you must have Ghostscript installed. Designer has been tested on Windows with Ghostscript 9.27 (32 bit only).

Citrix and Windows Terminal Services

Designer can be installed on Citrix and Windows Terminal Services Server for use with application virtualization.

To enable virtualization configure the Designer client file `EventHub.xml` entry as follows:

```
EnableCitrixMode="true"
```

Microsoft Internet Explorer

Some Designer components make use of modules provided by Microsoft Internet version 11 or higher and this software must be available prior to installing Designer material.

PDF viewer

Adobe Reader DC (Latest update version) is recommended as the Designer publication preview PDF viewer. Windows 10 users may use the Windows Edge PDF plug-in reader.

Windows OpenSSH

If you are going to use SFTP to transfer HIP files securely, you must have Windows OpenSSH version 7.6 or higher. Public/private key pairs must be generated and made available for your Windows account and remote target.

Upgrading SQL Server 2000 repositories to SQL Server 2012 and later

If you are working with a repository created in SQL Server 2000 it is not possible to upgrade your repository directly to an environment running SQL Server 2012 or later. To overcome this issue it is necessary to upgrade the repository from SQL Server 2000 to SQL Server 2008.

To upgrade a repository created in SQL Server 2000:

1. If there is no instance of SQL Server 2008, then install SQL Server 2008 Express edition. Note that this is available from the Microsoft web site.
2. Install Designer and connect to the SQL Server 2008 instance when prompted during the installation.
3. Restore the repository created in SQL Server 2000 using Repository Configuration Tool.
4. Backup the repository using Repository Configuration Tool.
5. Start the Repository Configuration tool and connect to the SQL Server 2012 instance.
6. Using the Repository Configuration tool, restore the repository backup from step 4.

Other SQL Server considerations

The default installation of Designer requires SQL Server to be configured in SQL Mixed Authentication mode. If you use a firewall you must ensure that the SQL Server ports are open. If you are running a distributed install ensure that your Database Engine has the TCP/IP protocol enabled.

Snapshot limitations

Snapshots created within an SQL Server 2000 or later environment cannot be imported directly to SQL Server 2012. To overcome this limitation it is necessary to perform the following steps:

1. Import the SQL Server 2000 snapshot to SQL Server 2008 based repository.
2. Export the required objects from your SQL Server 2008 based repository.
3. Import the snapshot created in the previous step into your SQL repository.

Issues with Windows Firewall

Windows Firewall is automatically enabled in versions of Windows from 7 onwards. This will block some of the Designer executable files from running. You can either:

- unblock these files when prompted by Windows Firewall
 - cockpit.exe
 - g1eventhub.exe

Note: that you will only have to do this once, when the files are first used.

Or

- turn off Windows Firewall. This is not advisable unless you are using another firewall.

Note: that with a firewall you must ensure that the SQL Server ports are open.

3 - Installing and configuring Designer

In this section

About the installers	14
Running the client install in silent mode.....	16
Deployment scenarios.....	17
Client/Server and Distributed considerations.....	22
Configuring for application virtualization.....	24
Installing and configuring advanced search support.....	26
Configuring project services.....	46



About the installers

This chapter describes the installers provided with Designer, their function, and possible deployment scenarios using the installers.

Important points to note:

- Software installed prior to Designer Service Pack 11 must be uninstalled before running the installers detailed below. However, repository upgrade can be performed on repositories created prior to Designer Service Pack 11.

In a scenario where you have not uninstalled a prior version before installing Designer Service Pack 11, both versions will be installed; however, the earlier version of the software will not function. In this situation, best practice is to:

- Uninstall both versions and reinstall the new version of Designer.
 - Take a backup of the repository and all the configuration files example, SSO, secure settings, project services, etc, before migrating to any new version of Designer.
- If you have installed the Advanced Search feature prior to Designer Service Pack 11, it must be uninstalled before reinstalling the latest version.

The installers distributed with Designer are:

- ***setup-designer-server.exe***

Installs the Designer repository and application server software allowing you to:

- Change the installation folder from the default, if required
- Connect and configure your repository

- ***setup-designer-client.exe***

Installs the Designer client software allowing you to change the application server name and port, if required.

Note that any prerequisite software is installed automatically, when needed. It is recommended that the latest windows updates have been applied before installation.

The following deployment scenarios are possible using the installers:

- Client/Server - install the Designer client(s) and server on separate machines, refer to [Client/Server deployment](#) on page 17 for further information.
- Distributed - install the application server and SQL Server on different machines, refer to [Distributed deployment](#) on page 19 for further information.

IMPORTANT: The Windows account under which the software is installed must have Administrator access to the local machine.

Installing the Designer client silently

You will typically push the latest Designer client software to your user's machines in an enterprise deployment scenario. You can achieve this by installing the client software in silent mode; refer to [Running the client install in silent mode](#) on page 16 for further details.

Install authenticity

To ensure the release installation is authentic and has not been tampered with, the installers and the installed resources are digitally signed.

Upgrading

You will be able to upgrade your Designer software as from the Designer Service Pack 11 onwards using the installers described in this chapter.

Note: for previous versions of the software, you will need to uninstall the existing software before installing the latest version.

You are advised to backup your Designer data before performing any upgrade/uninstall.

Running the client install in silent mode

To run the Designer client install in unattended mode, copy *setup-designer-client.exe* from the release media *install* folder to your local folder.

Include the setup command using the following syntax in your install script:

```
.\setup-resource-access-service.exe /s /v"/qb ALLUSERS="\1\""  
.\setup-designer-client.exe /l"language-code" /s /v"/qb ALLUSERS="\1\""  
SERVERNAME=server-name PORTNUMBER=port"
```

Where:

- *SERVERNAME* - the server from which the Designer server application is installed.
- *PORT* - the listening port for the application server. The default port number is 37719.
- *language-code* - the language version of the client to be installed.

Permissible *language-code* values are:

- 1033 (US English)
- 1031 (German)
- 1034 (Spanish)
- 1036 (French)
- 3084 (French-Canadian)

Example:

```
.\setup-designer-client.exe /l"1033" /s /v"/qb ALLUSERS="\1\""  
SERVERNAME=DESAPPSERVER PORTNUMBER=37719"
```

Note: the command above must be entered as a single command line.

Although the install process will automatically install all prerequisite software, it is important to note that the following applications may require a restart:

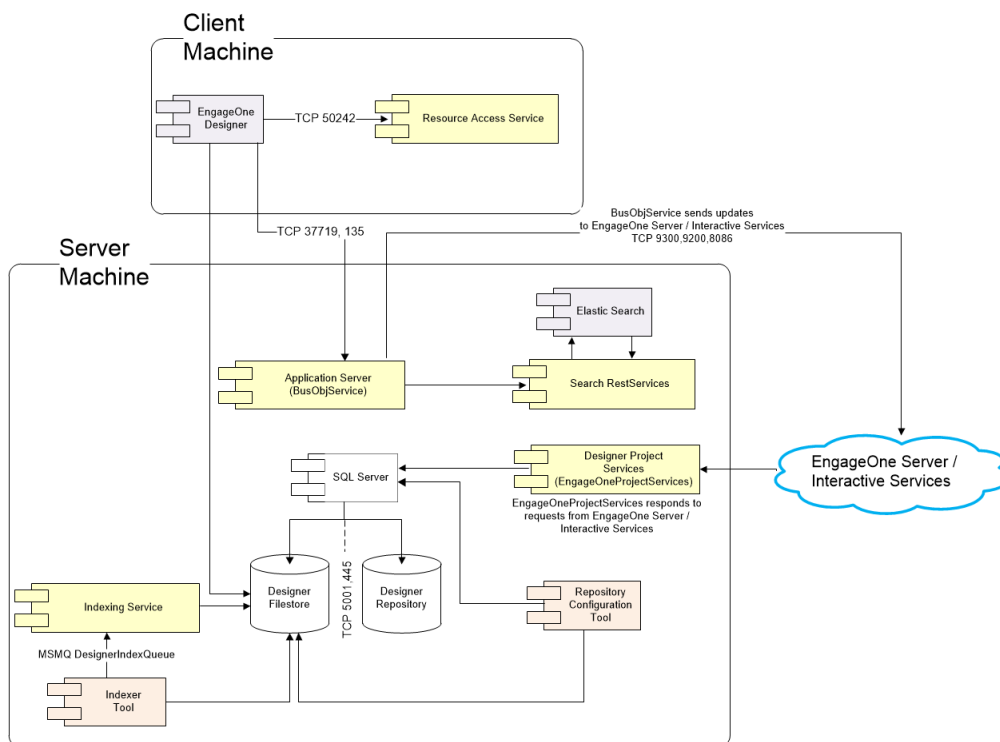
- .Net Framework 4.8
- Microsoft Visual C++ 2015-2019 Redistributable

It is therefore recommend that these components are installed before any silent mode install.

Deployment scenarios

Client/Server deployment

In a client/server deployment the Designer repository is located on a separate machine with one or more client machines running Designer and accessing the server machine.



The Administrator will log-in in to Designer to create user accounts for other users. The Administrator should then notify other users of their account details and the location from which the client software can be installed.

The sections that follow detail the steps you need to follow to complete a client/server deployment of Designer.

Server install

On the server machine, from the release media's *install* folder locate and run *setup-designer-server.exe*, this program will:

1. Installs the Designer server software.
2. Guide through a number of standard install wizard pages.
3. Install prerequisite software as required.
4. Allow you to change the installation directory.
5. Configure the Designer repository.

The Repository Configuration Tool

The Designer Repository Configuration tool allows you to set up your repository. This application is supplied with the product distribution material and can be run separately, following the installation of the server software.

To run the Repository Configuration Tool:

1. Select the **EngageOne Designer Repository Configuration Tool** option available from your Windows Start menu.
2. In the **Connect to SQL Server** dialog box click the **Browse** button to select the server that is being used for the repository from the **SQL Server** list – or local if it is on your machine.
3. If you are connecting to an SQL server that requires its own login, i.e. not the Windows user name and password, clear the **Use Windows authentication** check box and enter the **SQL Login Name** and **Password**.
4. Click **Connect**.

Refer to the Designer User's Guide for detailed information about the Repository configuration tool.

For a Client/Server environment you must first create a folder for the filestore and be aware of Sharing and Security options. For example, read/write access must be granted to all users running Designer and the Windows account that the SQL Server is running under. See [Client/Server and Distributed considerations](#) on page 22 for more details.

Client install

On your client machine(s), from the release media's *install* folder locate and run *setup-designer-client.exe*, this program will:

1. Install prerequisite software, as required.
2. Installs the Designer client software.
3. Assign application server name and port number.
4. Allow you to change the installation directory.

Distributed deployment

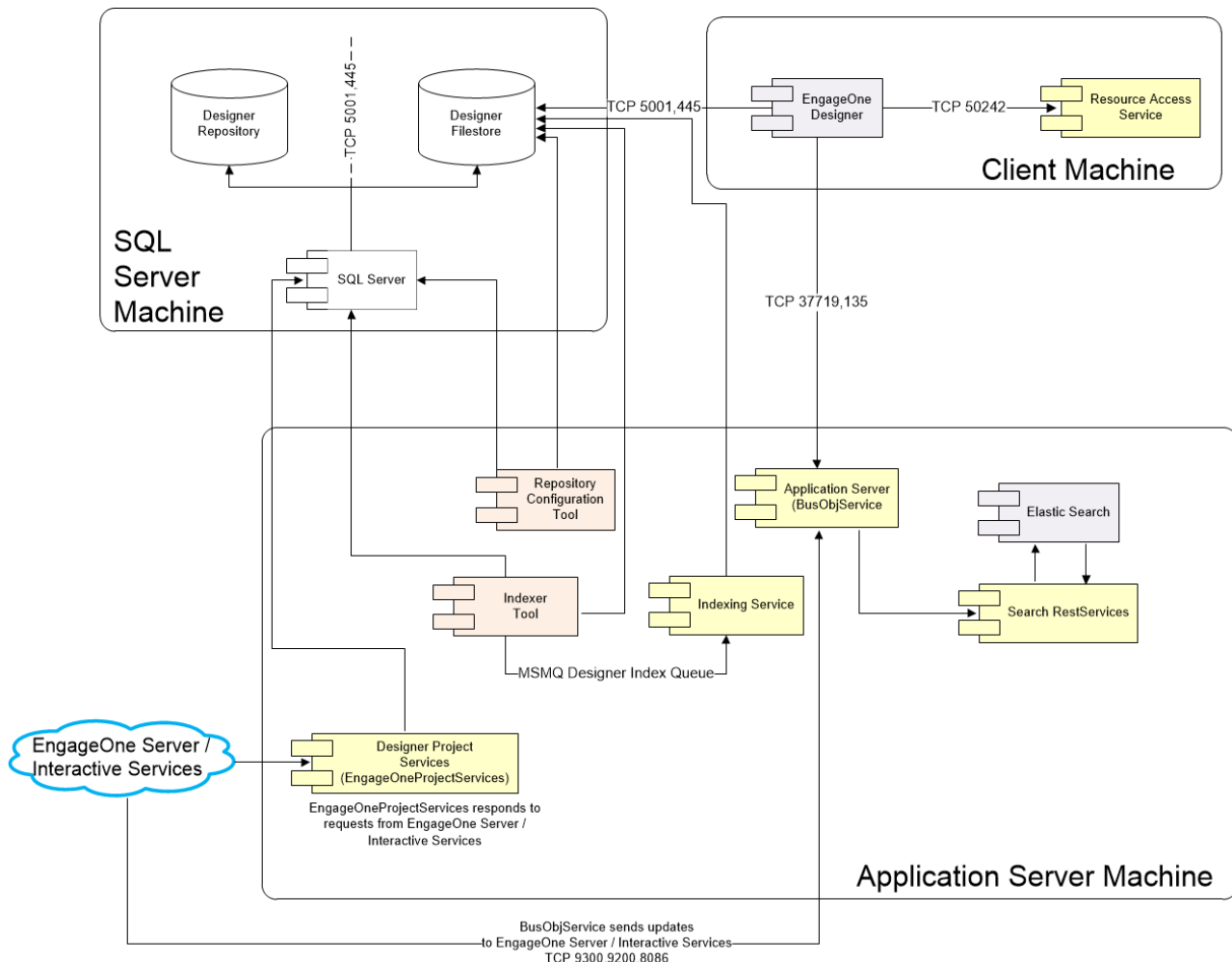
In a Distributed environment the Application Server is installed on a different machine to the one running your SQL Server.

NOTE: Your SQL Server instance must be configured to allow connections from other computers; this is typically achieved by enabling the TCP/IP protocol via the **SQL Server Configuration Manager**.

You must first create a folder for the filestore and share, and grant read/write access as with the Client/Server install – see **Client/Server and Distributed considerations** on page 22 for details.

Note that the filestore does not have to reside on the same machine as the SQL Server; however, the SQL Server Windows service account must have read/write access to the filestore in order to support the Import/Export Snapshot feature.

Example distributed installation



Configure the SQL Server

In a distributed deployment scenario it is important to note that TCP/IP connections in the SQL configuration manager must be enabled.

1. Locate the *ConfigureSQLServer.SQL* file in the release media sub-directory *\install*.

This script configures the SQL Server Login that can be used by the Application Server to connect to SQL Server - the default Login name is *DOC1Login*. Note that the Login is not configured with SQL Server admin rights.

The script also configures options required to support upgrading databases from older versions of the software and support the import of snapshots created from earlier versions of the software.

You can control these options by setting the optional configuration flags inside the script; therefore, it is important to review the script before proceeding.

2. Edit the *ConfigureSQLServer.SQL* script and specify the required Login name and password. Execute the script and check the messages logged. The table below lists messages you may see and the appropriate action to take.

Updating password for existing SQL Server login [LOGIN_NAME]	A Login with the required SID already exists so the Login name specified in the script was not applied. The specified password was applied. The Login name displayed in the message and the specified password can be used when configuring the Application Server.
Using existing SQL Server login [LOGIN_NAME]	A Login with the required SID already exists so the Login name specified in the script was not applied. A blank password was specified so the password for the existing Login was unchanged. The Login name displayed in the message and its associated password can be used when configuring the Application Server.
The server principal LOGIN_NAME already exists.	A Login with the specified name already exists but it does not have the required SID. Delete the Login manually or specify a new Login name in the script.

Application server

1. Uninstall any existing Designer Application Server software before Designer 6.6 Service Pack 11.
2. Locate and run the *setup-designer-server.exe* in the release media's *\install* folder; this program will:
 - a. guide through several standard install wizard pages.
 - b. install prerequisite software as required.

- c. allow you to change the installation directory.
 - d. configure the Designer repository.
3. On the **SQL Server Connection** page, you can:
 - use **SQL Server Authentication** with the Login credentials specified in the **Configure SQL Server** section.
 - use **SQL Server Authentication** with a SQL Server Login that has admin rights.
 - select **Use Trusted Connection** if the current Windows login has admin rights in SQL Server.
 4. On the **Repository Setup** page, install a new repository or upgrade an existing repository as appropriate. When installing a new repository, note that the **Database Backup Location** is accessed by SQL Server on the Database Server. It could be a local path on the Database Server or a UNC path accessible to SQL Server.
 5. On the **Application Server Component** page, when configuring the connection between the Designer Application Service and SQL Server, you can:
 - select **SQL Server Authentication with specific credentials** and enter the Login credentials specified in the **Configure SQL Server** section, or credentials for a SQL Server Login that has admin rights.
 - select **Windows Authentication** and enter credentials for a Windows login that has admin rights in SQL Server.
 6. Continue with the installation, following on-screen instructions.

Client install

On your client machine(s), from the release media's *install* folder locate and run *setup-designer-client.exe*, this program will:

1. Install prerequisite software, as required.
2. Installs the Designer client software.
3. Assign application server name and port number.
4. Allow you to change the installation directory.

Client/Server and Distributed considerations

In a Client/Server or Distributed environment the filestore should either be on the server machine, or on another server machine that is accessible to all clients and the server.

Note: that the client and server must be installed on separate machines.

The database will automatically be made available via the SQL Server. However, you must ensure that the filestore and setup-designer-client.exe are available to all users who will be accessing it.

Use the Sharing and Security option in Windows Explorer to:

- share the folder
- give full security control to the appropriate users/groups.

You will also need:

- write access to your working folder. By default this is your local 'temp' folder.
- access to the registry entries:

– on first installation and when upgrading

```
HKEY_LOCAL_MACHINE\SOFTWARE\GROUP1
HKEY_CURRENT_USER\Software\Group1\DOC1 Series
```

- permission to install any new fonts used.
- permissions for the Windows account that the SQL Server is running under. Note that if you have installed the SQL Server to run under the Local System account then you will need to change this to a named Windows user account. This is required because the Designer Backup, Restore and Snapshot functions execute within the SQL Server to transfer files to and from the filestore.

Note: when specifying locations during installation, the full UNC path must be given, e.g. \\server\ecs\filestore.

Advanced search

If you wish to use the advanced search feature, it is important to note that the service account for the search indexing service must have read/write access to the Designer filestore.

Securing data transfer in a Client/Server environment

To ensure client/server data is securely transferred you can configure all data communications to use secure channels. This mode of data transfer contains a set of security protocols that provide identity authentication and secure communication through encryption. To activate this method of data transfer you will need to modify the following files:

- BusObj.config on the server machine
- Processor.dll.config on all client machines.
- ResourceAccessService.exe.config on all client machines

Note: by default these configuration files are shipped with Secure Channel deactivated.

To implement secure channel data communications:

1. On the server machine use the following path to locate the BusObj.config file from:

```
<EngageOne Install Path>\Server\
```

2. Edit the BusObj.config file using a text editor such as notepad and ensure that the channel tag's secure attribute is set to true, as shown in the code snippet below:

```
<system.runtime.remoting>
  <customErrors mode="off"/>
  <application>
    <channels>
      <channel ref="tcp" secure="true" port="37719">
        <serverProviders>
          <formatter ref="binary" />
        </serverProviders>
      </channel>
    </channels>
```

3. On the client machine(s) use the following path to locate the Processor.dll.config file from:

```
<EngageOne Install Path>\Client\
```

4. Edit the Processor.dll.config file using a text editor and ensure that the channel tag's secure attribute is set to true, as shown below:

```
<configuration>
  <system.runtime.remoting>
    <application>
      <channels>
        <channel ref="tcp" secure="true">
          <clientProviders>
            <formatter ref="binary" />
          </clientProviders>
```

```

    </channel>
  </channels>

```

5. On the client machine(s) use the following path to locate the ResourceAccessService.exe.config file:

```
<EngageOne Install Path>\Resource Access Service\
```

6. Edit ResourceAccessService.exe.config file with a text editor and ensure that the channel tag secure attribute is set to true, as shown below:

```

<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>
    <add key="ResourceServer" value="localhost" />
    <add key="ResourceServerPort" value="50241" />
    <add key="Protocol" value="tcp" />
    <add key="ScratchDir" value="" />
  </appSettings>
  <system.runtime.remoting>
    <application>
      <channels>
        <channel ref="tcp" port="50242" secure="true">
          <serverProviders>
            <formatter ref="binary" />
          </serverProviders>
        </channel>
      </channels>
    </application>
  </system.runtime.remoting>
</configuration>

```

Note: EngageOne Resource Access Service must be restarted after the ResourceAccessService.exe.config file has been edited.

7. On the server machine restart EngageOne Template Designer Application Service using the Windows Start button, use run to execute services.msc. Locate EngageOne Template Designer Application Service and restart the service.
8. Restart Designer on the client machines.

Configuring for application virtualization

Designer can be installed as a single-instance server using Citrix application virtualization. This allows multiple users to connect to and run the Designer remotely using web protocols.

Installing Designer on Citrix and Windows Terminal Services Server

Run the required installation as normal – Client/Server or Distributed – on the Citrix or Windows Terminal Services Server machine. You must then edit the configuration file `EventHub.XML` in the installation folder and set the option `EnableCitrixMode` to `True`.

Installing and configuring advanced search support

The advanced search is an **optional** feature offering sophisticated search capabilities. You can create complex searches to find specific asset types and their associated properties using Designer's built-in search query language. This section outlines the steps necessary to install and configure this option.

Note: The advanced search is restricted to the repository selected when logging into Designer. Note that the advanced search is enabled only if you have installed the required components by running the search service installer.

Note: Repository alias names must be unique where a single instance of Elasticsearch is used to support Designer's advanced search feature. Care must therefore be taken when defining repository alias names using the Repository Configuration Tool particularly where your operating environment contains multiple Designer installations using a single Elasticsearch instance. Bear in mind that repository alias names are treated in a case insensitive manner by Elasticsearch.

Note: If you have installed the Advanced Search feature prior to Designer Service Pack 11, it must be uninstalled before reinstalling the latest version.

Installation steps:

- Ensure your hardware requirements match the recommendations according to your operational requirements, refer to [Additional hardware recommendations](#) on page 27
- Ensure the prerequisite software is installed, refer to [Additional software requirements](#) on page 28
- Enable Microsoft Message Queuing (MSMQ), refer to [Enabling MSMQ](#) on page 29
- Run the search service installer, refer to [Running the search service installer](#) on page 31
- Optionally customize searchsettings.json according to your operational requirements, refer to [Customizing searchsettings.json parameters](#) on page 35
- Use the indexer tool to index assets in your repository, refer to [Maintaining search indexes](#) on page 33

Additional hardware recommendations

Services associated with the advanced search can be installed on the Designer application server. In this scenario additional hardware resource must be made available.

Use the following information as a guide:

- Increase the available memory by 2GB. An additional 2GB of disk space is also recommended for each repository you intend to search.
- If you have a single repository:
 - increase the CPU cores by a minimum of 2.
- If you have multiple repositories:
 - add an extra core per heavily used repository,
 - increase the available memory by 4GB.

The search feature is generally not very demanding in terms of hardware. However, some operations such as the initial indexing of a repository, branching projects and restoring snapshots, can be very CPU intensive. If these operations are likely to occur often, for example, on an hourly basis, we would recommend you provision an additional core for that repository in addition to the guidance above.

In cases where there are multiple active repositories, we would recommend that you install the new search related services onto their own server.

For example:

- A 4 core 2.4GHz CPU server with 8GB RAM and 60GB would support approximately 400 concurrent users across two active repositories.
- A 6 core 2.4GHz CPU server with 8GB RAM and 60GB would support approximately 400 concurrent users across four active repositories.

Additional software requirements

The advanced search feature requires the following software to be installed:

JVM

- Oracle/OpenJDK 1.8.0u111
- Oracle/OpenJDK 9 - 14

Note:

- Oracle/OpenJDK 15 is NOT supported.
- `JAVA_HOME` environment variable must be assigned to the path of the JDK.

Microsoft

- Microsoft Message Queuing (MSMQ). Refer to [Enabling MSMQ](#) on page 29.
- Microsoft Powershell version 4.0 or higher.

Elastic Search

- Elastic search 7.10.1.

Note: Elasticsearch is packaged as part of the search service and can be installed while installing the search service. Refer to [Running the search service installer](#) on page 31.

Enabling MSMQ

To enable **Microsoft Message Queuing** (MSMQ):

1. Launch the **Server Manager**.
2. Navigate to **Manage > Add Roles and Features**.
3. Click **Next** in the **Before You Begin** screen.
4. Select **Role-based** or **Feature-based** installation and click **Next**.
5. Select the server where to install the feature and click **Next**.
6. Do not select any server roles. Click **Next**.
7. In Features, expand **Message Queuing > Message Queuing Services** and select **Message Queuing**.
8. Continue running the wizard adding any other roles required by MSMQ (if applicable).
9. Click **Install** to start installation.

Note: the message queuing service (Message Queuing) must be restarted after installation.

The search service installer

A single installer zip (SearchServicesWin64.zip) is included with the distribution material to support the advanced search feature, this file can be found at the following location:

```
\Features\search
```

Run the installer to set up:

- The Elasticsearch engine. Note that the installation of this component is optional.
- The search ReST service.
- The indexing service.

Note: prior to running the installer Microsoft Message Queuing (MSMQ) must be enabled. Refer to [Enabling MSMQ](#) on page 29 for further information.

Deployment configurations

The installer supports the following deployment configurations:

- The Designer server, search services and Elasticsearch on a single machine.
- The Designer server on one machine with search services and Elasticsearch on another.
- The Designer server, search services and Elasticsearch on separate machines.

Note: The Elasticsearch install is distributed with the Designer install material in the packages folder of the installer zip (SearchServicesWin64.zip).

Installing Elastic Search on a separate server to Search Services

A script to install elastic search on its own on a separate server to the search services is not provided.

In this scenario, you must:

1. Unzip elasticsearch-7.10.1-windows-x86_64.zip onto the other server.
2. Modify **elasticsearch.yml**.
3. Add the server's **network.host**.
4. Run **elasticsearchservice.bat**.

About the search indexing service

The Search Indexer is a Windows service responsible for the indexing of Designer asset properties and content, it is identified in the Windows Task Manager Service list as:

```
EngageOneSearchIndexerService.
```

It is important to note, that the service account for the search indexing service must have read/write access to the Designer filestore in all deployment configurations.

The service:

- Consumes indexing requests from an MSMQ queue and sends the extracted information to an Elasticsearch instance specified in the searchsettings.json file.
- Logs information into the Windows Application log, with the logging level controlled by the Logging section in the .json file. Refer to LogLevel [Customizing searchsettings.json parameters](#) on page 35 for further information.
- May be installed on the Designer application server but it should be installed into a different folder to that of the other Designer server executables.
- Is un-installed by using the search un-installer, refer to [The search uninstaller](#) on page 39

The execution of the installer is governed by the supplied searchsettings.json file, you may need to update the json file according to your specific requirements before running the installer. Refer to [Customizing searchsettings.json parameters](#) on page 35 for detailed information.

Running the search service installer

To run the advanced search installer:

1. Open a Powershell window with administrator privileges.

Note: Use Powershell version 4.0 or later. It is also important to note that MSMQ should be enabled when upgrading to Powershell 4.

2. The search feature zip file is located in your release media's:

```
\Features\search\ServicesWin64.zip
```

Note: You will need to copy the zip file to a new folder, for example: `c:\Search` and extract the contents of the zip file to the new folder. The location you choose to extract the installer will depend on your chosen Designer deployment configuration.

3. Check, and modify where necessary, the `searchsettings.json` file in line with your specific operating requirements. Refer to [Customizing searchsettings.json parameters](#) on page 35 for detailed information.

Note: that the installer will install the services based on the `installPath` setting defined in the json file but will not create the designated folder. You must create the install folder manually. The default setting in this scenario would be: `c:\search\install`.

4. From the folder chosen to extract the installer, type:

```
.\install.ps1 searchsettings.json
```

and press **ENTER** to start the install.

Note: This script requires the execution policy to allow the running of unsigned scripts, so you may need to execute the following before running the script:

```
Set-ExecutionPolicy -ExecutionPolicy Bypass -Scope Process
```

5. Press **ENTER** when prompted to install the required services.
6. Copy the searchsettings.json file to the Designer server install folder and restart the **EngageOne Template Designer Application Service**.

Note: The default location for Designer server installation is:

```
<EngageOne Install Path>\Server
```

Important Note: Upgrading from 6.6.10 to 6.6.11

For release 6.6.11, we have upgraded Advanced Search to benefit from the enhanced security and support provided by Elasticsearch version 7.10.1. If you are upgrading from 6.6.10 to 6.6.11, you must first uninstall the 6.6.10 search services using the 6.6.10 uninstall.ps1 script.

You will not be able to install the 6.6.11 Advanced Search feature if an Elasticsearch 5.2.2 service is detected. Once the 6.6.11 services are installed you will need to run the IndexerTool.exe (refer to [Maintaining search indexes](#) on page 33) to re-index the repository. If you do not upgrade the Advanced Search services, asset indexing and advanced search will not function correctly, and you will find failure events in the Windows event logs of the Designer client computer.

Maintaining search indexes

The indexer tool (indexertool.exe) allows you to maintain search indexes. This utility is provided with the distribution material at the following location:

```
<EngageOne Install Path>\Server\
```

Use the indexer tool to perform the following operations:

- Repository Indexing
- Deleting the index of the repository.
- Displaying help about indexer tool.

Note: The <RepositoryName> referred to in the indexertool commands that follow is normally the alias name of the repository from the repository list in the repository configuration tool. If the repository list has not been setup, then the repository name is the database name.

Note: The advanced search index is automatically updated with new objects/changes to existing objects once the advanced search service is installed, configured, and executing normally

Repository indexing

The command for indexing the database is:

```
indexertool.exe /r:<RepositoryName>
```

where <RepositoryName> is the name of the repository to be indexed and is case-sensitive.

This command is to be used in the following scenarios

- Indexing a newly created repository
- Indexing a repository restored from backup
- Repairing an existing repository index.

You will typically want to do this when the index of the repository gets out of synchronization with the changes in the repository. This can occur as a result of network connection failures, etc. indexer service being down etc. Running the indexer tool with the repository name again will ensure that the latest changes to assets are indexed and available for use with the Advanced Search feature.

It is important to bear in mind, that if for any reason the indexing service is not active then any modifications to assets during this period will not be visible in the advanced search. In this scenario, you will need to run the indexertool to synchronize the changes.

Index deletion

The command for deleting an index is:

```
indexertool.exe /delete /r:<Repositoryname>
```

where <RepositoryName> is the name of the repository to be indexed and is case-sensitive. A typical example where this command can be used, is to delete an obsolete index for a decommissioned repository.

Help about the indexer tool

The command for displaying help about the indexertool is:

```
indexertool.exe /?
```

Customizing searchsettings.json parameters

It is important to note that the latest version of searchsettings.json file must be copied to Designer server install folder.

The default location for Designer server installation is:

```
<EngageOne Install Path>\Server
```

The following services must be restarted each time a change is made to the searchsettings.json file:

- EngageOne Template Designer Application Service
- EngageOne Template Designer Search Indexer Services
- EngageOne Template Designer Search ReST Services

searchsettings.json

Section	Name	Description	Default value	Comments
Logging				
	IncludeScopes	Determines whether log scope information is shown. Not used by the service but may be useful to turn on if debugging.	false	
EventLog				
	Logging	This controls the level of logging going to the Windows event log. See the 'Logging' section above for settings.		
LogLevel				
	Default	Log level of the service specific messages captured while the service is running	Warning	Can be Debug, Information, Warning or Error

Section	Name	Description	Default value	Comments
	System	Log level of the system messages captured while the service is running	Warning	As above
	Microsoft	Log level of the various Microsoft assembly messages captured while the service is running	Warning	As above
ServiceSettings				
Search	searchServiceUrl	The ReST service endpoint	http://localhost:8086/search	If you wish to use an HTTPS connection you must manually install the appropriate certificates on the Designer application server and SIS server as appropriate.
	searchEngineUrl	The Elasticsearch cluster endpoint	http://localhost:9200	
	searchEngineTransportPort	The Elasticsearch TCP Transport port number	9300	Sets the <code>transport.tcp.port</code> setting in the Elasticsearch configuration file
Indexing				
	queueName	The indexing service queue name	.\PRIVATE\$\DesignerIndexQueue	
	maxIndexingThreads	The maximum number of threads available to the indexing service	4	The maximum number of concurrent index requests the indexing service will attempt to service.

Section	Name	Description	Default value	Comments
	indexRequestTimeoutSecs	The request timeout limit (in seconds) used when sending indexing requests to Elasticsearch.	300	If this setting is less than 1 then the default Elasticsearch request timeout setting is left unchanged.
Security.TLS				
	enabled	Enable or disable REST layer security	true	Set to true to enable TLS (HTTPS)
	keystoreLocation	Path to the keystore file, relative to the config/ directory		Must be set for TLS. E.g. keystore_node1.pfx
	keystoreAlias	Optional alias if keystore contains multiple certificates		If none specified the first certificate will be used
	keystorePassword	Keystore password		Encrypted password. Generated by the EncryptPassword.exe utility
	truststoreLocation	Path to the truststore file, relative to the config/ directory.		Must be set for TLS. E.g. truststore.pfx
	truststoreAlias	Optional alias if truststore contains multiple certificates		If none specified the first certificate will be used
	truststorePassword	Truststore password		Encrypted password. Generated by the EncryptPassword.exe

Section	Name	Description	Default value	Comments
Security.Authentication				
	mode	The authentication mode. valid settings are 'basic' or 'engageone'	basic	Basic authentication or EngageOne authentication modes are available.
	userName	Basic authentication user name		If the mode is basic this is the basic authentication user name. In the default configuration file this is set to system . If the mode is engageone this is the EngageOne security services username, commonly set to servicesuser .
	userPassword	Basic authentication user password (encrypted)		Used during service to service calls. Both userName and userPassword must be set. The password must be encrypted using the EncryptPassword.exe utility present in the searchservices folder of the release media. The encrypted value in the shipped settings file is password .
	securityServiceUrl	The URL of the EngageOne security service		Must be set when selecting the EngageOne authentication mode

The search uninstaller

From a Powershell window, in the folder chosen to extract the installer, type:

```
.\uninstall.ps1 searchsettings.json
```

and press ENTER.

Note: the un-installer will run using the settings configured in searchsettings.json.

Troubleshooting search issues

The information provided in this section is intended for administrators responsible for troubleshooting and rectifying search related issues.

Enabling debug logging for the search services

The search services can be configured to log more detailed information in the Windows Event log for diagnostic purposes.

1. **Locate the `searchsettings.json` files within the `IndexerService` and `SearchService` installation folders.**
2. **Update the `LogLevel` settings from `Warning` to `Debug`, as shown below:**

```
"LogLevel": {  
  "Default": "Debug",  
  "System": "Debug",  
  "Microsoft": "Debug"  
},
```

3. **Save the files and restart the `EngageOneSearchServices` and `EngageOneSearchIndexerService` services to pick up the new settings.**

Restore the `LogLevel` settings after performing diagnostics.

Messages

Failed to start elastic search service

Error message	Failed to start 'elasticsearch-service-x64' service
---------------	---

REST error code	Not applicable
-----------------	----------------

Occurs when:	First time installation of .\install.ps1 searchsettings.json
--------------	--

Troubleshooting information

The issue has been observed due to a lag between the request to start the Elastic Search Service and the Search Service actually starting on some Windows Server environments.

A second execution of `.\install.ps1 searchsettings.json` will usually resolve the issue.

Failed to start service 'EngageOne Template ...

Error message	Failed to start service 'EngageOne Template Designer Search ReST Services (EngageOneSearchServices) '.
---------------	--

REST error code	Not applicable
-----------------	----------------

Occurs when:	Running the search feature Powershell installer, e.g. .\install.ps1 searchsettings.json
--------------	---

Troubleshooting information

The issue has been observed when the search service default port, 8089, is already in use by another service, or application on the computer the search service is being installed on.

Changing the port number in the searchsettings.json config file to an unused one and then retrying to install should resolve this error

Incompatible search services

Error message	Incompatible search services have been found
---------------	--

REST error code	Not applicable
-----------------	----------------

- Occurs when:
- Running IndexerTool.exe to re-index the repository
 - Performing an Advanced Search in the Designer client

Troubleshooting information

Uninstall the older 6.6.10.x search services and install the 6.6.11.x or later Advanced Search services.

Indexer service Request timeout error

Error message	Failed to write index for <ASSET>: Maximum timeout reached while retrying request
---------------	---

REST error code	Not applicable
-----------------	----------------

- Occurs when:
- Indexing, typically when using the IndexerTool to index a Designer repository.
 - Indexing (save, branch, etc.). In this scenario, the index will lose its synchronization.

Troubleshooting information

The error will be logged in the Indexer service event log.

The problem occurs when Elasticsearch takes too long to process an indexing request.

Actions:-

- Review your Elasticsearch installation to improve indexing speed.
 - Increase the indexRequestTimeoutSecs setting in the searchsettings.json file located in the indexer service installation location.
-

Error message	The indexer service is taking too long to respond.
---------------	--

REST error code	Not applicable
-----------------	----------------

Occurs when: This error generally occurs when there are access issues between the search indexing service and the Designer filestore.

Troubleshooting information

The error will be logged in the Indexer service event log.

Check that the service account for the search indexing service has read/write access to the Designer filestore.

Repository index not found

Error message	Repository index not found Could not find the index for repository. Please contact your system administrator
---------------	---

REST error code	400
-----------------	-----

Occurs when:	The database is not indexed
--------------	-----------------------------

Troubleshooting information

In the Search/Indexer service event log:

An error will be logged when save/branch type operations are performed in designer client.

Other areas to investigate:

- **Services** - EngageOne Template Designer Search ReST services
- **Services** - EngageOne Template Designer Search Indexer services
- Check if any index requests are queued by viewing:
Computer Management/Private Queues/DesignerIndexQueue/Queue messages
- Run `indexertool.exe` from:

```
..\Designer\Server
```

Refer to [Maintaining search indexes](#) on page 33 for further information.

Search index synchronization

Error condition	<ul style="list-style-type: none"> • Search indexes are out of synchronization • Inconsistent search results • No error messages logged.
-----------------	---

REST error code	Not applicable
-----------------	----------------

Troubleshooting information

- Check:
Services /EngageOne Template Designer Search ReST services
- Check if any index requests are queued by viewing:
Computer Management/Private Queues/DesignerIndexQueue/Queue messages

Search service connection error

Error message	Connection error Error connecting to the search service. Please contact your system administrator
---------------	---

REST error code	Not applicable
-----------------	----------------

Occurs when:	<ul style="list-style-type: none"> • Search REST Service is down or is inaccessible at the configured address. • Indexing (save, branch, etc.). In this scenario, the index will loose its synchronization.
--------------	---

Troubleshooting information

In the Designer event log go to:

Event viewer/application and services log/cockpit

The following log entry is observed:

Failed to connect to the search service: Error while copying content to a stream. --> Unable to read data from the transport connection: An existing connection was forcibly closed by the remote host. --> An existing connection was forcibly closed by the remote host.

Note: the event log for the Search/Indexer service will not create a log entry in this scenario.

Search service reported an error

Error message	The search service reported an error, please contact your system administrator
---------------	--

REST error code	500
-----------------	-----

Occurs when:

- Elastic Search service is down.
 - Indexing (save, branch, etc.). In this scenario, the index will lose its synchronization.
 - The Designer search services have not been upgraded from 6.6.10 to 6.6.11 or later.
-

Troubleshooting information

In the Designer event log go to:

When searching: An internal server error occurred. Please contact your system administrator.

In the Search/Indexer service event log:

Unable to connect to the remote server...

Other areas to investigate:

- Check if :

Services /Elastic Search Service

has been started, if not start the service.

- Check if any index requests are queued by viewing:

Computer Management/Private Queues/DesignerIndexQueue/Queue messages

Configuring project services

Use the Designer Repository Configuration Tool - Project Services tab to enable the review and approval of Designer publications by EngageOne Portal users. The Project Services tab allows you to import service settings from a properties file(deploy.properties), available from your EngageOne Administrator. Refer to the table below for detailed information:

Project Services tab	deploy.properties field	Comments
Local service URL	<code>designer.services.url</code>	
Key store	<code>tls.key.store.location</code> <code>tls.key.store.password</code>	Optional, configured if your organization uses TLS security. If configured an encrypted password is set in the properties file but is not visible on the Project Services tab in Designer.
Trust store	<code>tls.trust.store.location</code> <code>tls.trust.store.password</code>	Optional, configured when using TLS security. If configured an encrypted password is set in the properties file but is not visible on the Project Services tab in Designer.
Alias	<code>designer.tls.key.alias</code>	Optional, configured when using TLS security
Project services URL	<code>core.services.url</code>	
Security services URL	<code>security.services.url</code>	
Service user name	<code>system.user.username</code> <code>system.user.password</code>	An encrypted password is set in the properties file but is not visible on the Project Services tab in Designer

About security certificates for project services

When operating in TLS mode the Designer Project services require multiple certificates that can be provided in one or two PFX certificate files:

1. A Truststore containing the certificates necessary to communicate with the EngageOne security and Project services.

Note that EngageOne Project services certificate must have the following Extended Key Usage extensions:

- TLS Web Server Authentication
- TLS Web Client Authentication

2. A Keystore containing the certificate to be used for the Designer project service.

This should contain:

- Public key
- Private key
- TLS Web Server and Client Extended Key Usage extensions
- A unique alias that will be used in the EngageOne deployment.properties file, designer.tls.key.alias field. Note that this must be different from the alias used in the Truststore.

The certificates can be combined into a single store as long as the Designer Project services certificate alias is unique.

4 - Start-up & login

In this section

Logging into Designer.....49
Resources for samples /confidence tests.....49



Logging into Designer

The login screen will appear requesting a user name and password for the Designer. The first time you log on (in all environments) you should specify the default Administrator account name and password. Both are 'admin' (ensure lower case characters are used).

If Designer has previously been configured to allow multiple repositories to be used then use the Options button to select the one you want to use.

Resources for samples /confidence tests

Two publications are provided with the release material. These can be used as samples on which to base your own publications. They are also used as the source for the confidence tests supplied with Generate – see the “Confidence testing” sections in the Generate Release Notes.

The publications are supplied as individual snapshots and can be imported into Designer by using the Tasks/Import/Snapshot option in Designer and browsing to `\snapshots` from the Designer installation media. See “Snapshots” in the Designer Users Guide for more information.

5 - Repository selection

In this section

Repository selection.....51



Repository selection

The **EngageOne Designer Repository Configuration Tool** option is available from your Windows Start menu.

Note: You must log out of Designer before changing the repository. In a Client/Server environment you must ensure that all clients are also logged out.

- **To switch the repository database server:** connect to an appropriate server in the SQL Server Connection page. Then set up the repository using the Repository Setup page.
- **To use a different repository:**
 - either -
use the Repository Configuration Tool and specify a different repository in the Repository Setup page
 - or –
select the required repository when logging in (only if configured – see “Switching the repository database” above).

6 - Preventative maintenance and best practices

In this section

Best practices.....53



Best practices

Designer repository backups/restore and snapshots

Perform regular backups of your Designer repository. The backup process takes a copy of your entire repository protecting against the possibility of data loss or corruption. Use the restore process to recover the required Designer repository from your backups. Additionally, the snapshot feature allows you to capture and export a project or any of the objects within it such as a folder, a publication or Active Content. You can capture the current state or any previously labeled version. The snapshot can then be imported into a Designer repository as a new project. This enables you to copy objects from one Designer repository to another; for instance when you need to transfer resources to software support for analysis. Refer to the Repository Management section in the Designer Users guide for detailed information on these features.

Working with projects and branching.

Version control allows you to bring together resources and objects in projects for a common purpose, such as grouping together all the resources used in a publication for a particular marketing campaign or for new development. A project can be subdivided into folders if required, giving more structure to the organization of your work. You can branch a project, folder, or anything within them, at any time. This will create a new project which could, for example be used for separate development, leaving the original unchanged.

During branching deleted items are not carried forward to the new branch. This can help improve performance where many deletions have been made within a project. In this scenario it is recommended you branch your project and work with the new branch.

Refer to the Version Control section in the Designer Users guide for detailed information on projects and branching.

Working with interactive documents containing localized paragraphs

It is important to note that mandatory prompts must be common to all language variants in document designs containing localized paragraphs, otherwise, unexpected behavior will occur.

For example, a document design contains French and English paragraph variations, as follows:

- The English document contains the following interactive prompts:
 - A, B and C

- The French document contains the following interactive prompts:
 - A,B,C and D(mandatory)

In this scenario, an Interactive user will not be able to submit the English document because the document logic will detect that prompt D requires a value. Note that prompt D will not be visible for entry in the English paragraph variation.

Notices



Copyright © 1993, 2021 Precisely. All rights reserved

This publication and the software described in it is supplied under license and may only be used or copied in accordance with the terms of such license. The information in this publication is provided for information only, is subject to change without notice, and should not be construed as a commitment by Precisely. To the fullest extent permitted by applicable laws Precisely excludes all warranties, representations and undertakings (express or implied) in relation to this publication and assumes no liability or responsibility for any errors or inaccuracies that may appear in this publication and shall not be liable for loss or damage of any kind arising from its use.

Except as permitted by such license, reproduction of any part of this publication by mechanical, electronic, recording means or otherwise, including fax transmission, without the express permission of Precisely is prohibited to the fullest extent permitted by applicable laws.

Nothing in this notice shall limit or exclude Precisely liability in respect of fraud or for death or personal injury arising from its negligence. Statutory rights of the user, if any, are unaffected.

*TALO Hyphenators and Spellers are used. Developed by TALO B.V., Bussum, Netherlands Copyright © 1998 *TALO B.V., Bussum, NL *TALO is a registered trademark ®

Encryption algorithms licensed from Unisys Corp. under U.S. Patent No. 4,558,302 and foreign counterparts.

Security algorithms Copyright © 1991-1992 RSA Data Security Inc

Datamatrix and PDF417 encoding, fonts and derivations - Copyright © DL Technology Ltd 1992-2010

Barcode fonts Copyright © 1997 Terrapin Solutions Ltd. with NRB Systems Ltd.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

Artifex and the Ghostscript logo are registered trademarks and the Artifex logo and Ghostscript are trademarks of Artifex Software, Inc.

This product contains the Regex++ library Copyright © 1998-2000 Dr. John Maddock

PostScript is a trademark of Adobe Systems Incorporated.

PCL is a trademark of Hewlett Packard Company.

Copyright (c) 2000 - 2015 The Legion of the Bouncy Castle Inc. (<http://www.bouncycastle.org>)

PStillDll (c) Dipl.-Ing. Frank Siegert, 2005-2018

PStill is a trademarked term, registered with the German patent and trademark office

This product contains RestSharp, version number 105.2.3, which is licensed under the Apache License, version number 2.0. The license can be downloaded from

<http://www.apache.org/licenses/LICENSE-2.0>. The source code for this software is available from <http://restsharp.org>.

This product contains Json.NET, version number 9.0.1, which is licensed under the MIT License. The license can be downloaded from

<http://github.com/JamesNK/Newtonsoft.Json/blob/master/LICENSE.md>. The source code for this software is available from <http://www.newtonsoft.com/json>.

ICU License - ICU 1.8.1 and later Copyright (c) 1995-2006 International Business Machines Corporation and others All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

This product contains Elasticsearch, which is licensed under the Apache license, version number 2.0. The license can be downloaded from <http://www.apache.org/licenses/>. The source code for this software is available from <https://github.com/elastic/elasticsearch>.

This product contains Elasticsearch.Net, which is licensed under the Apache license, version number 2.0. The license can be downloaded from <http://apache.org/licenses/>. The source code for this software is available from <https://github.com/elastic/elasticsearch-net>.

This product contains NEST, which is licensed under the Apache license, version number 2.0. The license can be downloaded from <http://apache.org/licenses/>. The source code for this software is available from <https://github.com/elastic/elasticsearch-net>.

This product contains Antlr4cs Runtime, which is licensed under the BSD-3-Clause. The license can be downloaded from <https://raw.githubusercontent.com/tunnelvisionlabs/antlr4cs/master/LICENSE.txt>. The source code for this software is available from <http://www.antlr.org>.

This product contains Log4net. The license for log4net can be downloaded from <https://www.apache.org/licenses/LICENSE-2.0>. The source code for this software is available from https://logging.apache.org/log4net/download_log4net.cgi.

Otherwise all product names are trademarks or registered trademarks of their respective holders.

Support

Click [here](#) for full EngageOne Compose documentation and access to your peers and subject matter experts on the Knowledge community.



1700 District Ave Ste 300
Burlington MA 01803-5231
USA

www.precisely.com

© 1993, 2021 Precisely. All rights reserved.