



Customer Engagement

EngageOne[®] Compose

EngageOne[®] Designer

Installation and Configuration Guide

Version 6.6 Service Pack 9



Table of Contents

1 - Introduction

About EngageOne® Designer	4
The repository	4

2 - Hardware and software requirements

Hardware requirements	6
Software requirements	7

3 - Installing and configuring Designer

About the installer	13
Stand alone or Client/Server	14
Standalone install	14
Client/Server install	15
Distributed install	17
Client/Server and Distributed considerations	22
Configuring for application virtualization	25
Installing and configuring advanced search support	25
Configuring project services	43

4 - Start-up & login

Logging into Designer	45
Resources for samples /confidence tests	45

5 - Repository selection

Repository selection	47
----------------------	----

6 - Preventative maintenance and best practices

Best practices	49
----------------	----

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

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 either a single user Standalone environment, 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 Standalone version the Designer is installed for immediate access to a local resource repository, which is created if required.

In the Client/Server environment the client software is copied to a network location and the server control programs are installed for use by the local machine. A resource repository is created (or referenced) as with the Standalone install. 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.

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	6
Software requirements	7

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.

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

Software requirements

Designer is supported on the following platforms only:

- Microsoft Windows Server 2019 Standard (64 Bit)
- Microsoft Windows Server 2016 Standard (64 Bit)
- Microsoft Windows Server 2012 R2 Update (KB2919355) (64 Bit)
- Microsoft Windows Server 2008 R2 (64 Bit)
- Microsoft Windows 10 (64 Bit)
- Microsoft Windows 8.1 Update (KB2919355) (64 Bit)
- Microsoft Windows 7 SP1 (32 and 64Bit)

It is supported with the following 64 bit versions of Microsoft SQL Server only:

- Microsoft SQL Server 2017 CU13
- Microsoft SQL Server 2016 SP2
- Microsoft SQL Server 2014 SP2
- Microsoft SQL Server 2012 SP4

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.

Standalone	*Microsoft Visual C++ 2015 Redistributable Microsoft .NET Framework 4.6.1 Microsoft Shared Management Objects for SQL Server Microsoft System CLR Types for SQL Server Resource Access Service
Designer Client client/server	*Microsoft Visual C++ 2015 Redistributable Microsoft .NET Framework 4.6.1 Resource Access Service
Application Server client/server, distributed	Microsoft .NET Framework 4.6.1 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 9.25 (32 bit only).

Citrix and Windows Terminal Services

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

Microsoft SQL Server

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

Note: that this is not a requirement if you choose a Standalone install, as limited user repository software (SQL Server Express) can be downloaded from the Microsoft web site.

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.

Adobe Reader

Adobe Reader DC 15.0 or higher must be available to read previews of your publications. You can use the link on the contents window to download Adobe Reader.

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.

Install issues on Windows Server

If Microsoft SQL Server is installed prior to Designer, then the install HTA application may be prompted for the HTML application to be associated.

After associating the HTML application, the HTA will need to be closed and reopened in order to use the Designer installation links.

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 installer	13
Stand alone or Client/Server	14
Standalone install	14
Client/Server install	15
Distributed install	17
Client/Server and Distributed considerations	22
Configuring for application virtualization	25
Installing and configuring advanced search support	25
Configuring project services	43

About the installer

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

A contents window is launched automatically when the `setup.hta` file is launched from the installation media.

The installer allows you to select the following installation types:

- **Standalone** or **Client/Server** on the current machine.
- **Distributed** - this options allows you to install the Application Server and the Repository Configuration Tool separately, refer to [Distributed install](#) on page 17 for further information.

The installation process will then commence. Follow the instructions given by the install wizard.

Stand alone or Client/Server

Stand alone or Client/Server installation is available from the installer's content window by selecting the **Install Designer ...** option.

You will be presented with one or more install preparation pages, click **Next** when prompted.

On the **Customer Information** page enter:

- A user name and organization to be associated with the installed software.
- The type of access to run the Designer:
 - **All users**
 - **Only for me**

On the **Setup Type** page select option to specify the type of Designer environment you want to create:

- **Standalone** – creates a single user environment.
- **Client/Server** – installs the server and creates a copy of the client software ready for installation by client systems.

Note: if you already have an existing installation, you will only be able to upgrade it. To change between client/server and standalone run the installation for previous version of Designer and remove the existing installation. Install afresh as required.

The install process will gather details to configure the resource repository. In most cases, the default settings can be used for this. Refer to the sections that follow for more detailed information.

However, if you want to use an existing repository and filestore you may need to tailor the configuration of the repository.

Note: it is recommended that the latest windows updates have been applied before installation.

Standalone install

Following the **Setup Type** page, subsequent install wizard pages guide you through the stand alone install process.

Client/Server install

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.

To run the Repository Configuration Tool:

1. From the Start menu select `Programs\PBBI CCM\Repository Configuration Tool`
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.

On the client machine browse to the location where the Client installer software was installed on the Server machine and from the `\Server\Client Diskset` directory run the `setup.exe` program. This runs a simple installation wizard that installs the client software and registers the repository location.

Note that the default location for the client diskset is:

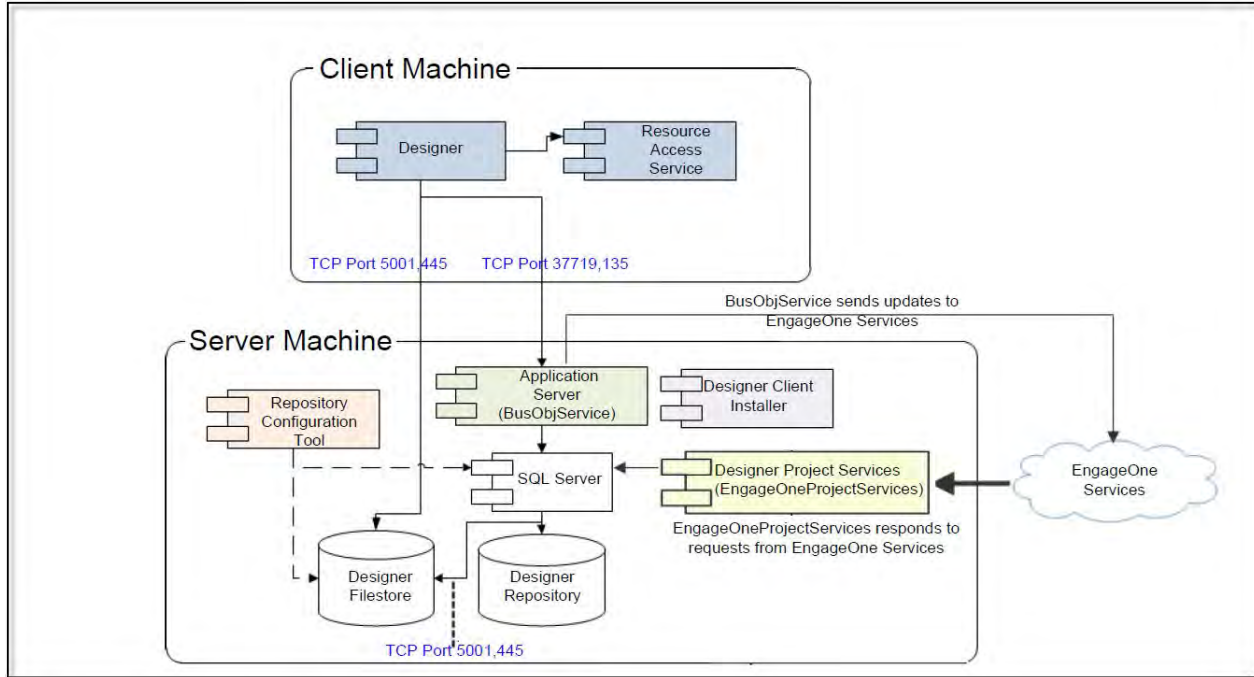
```
c:\Program files\PBBI CCM\DOC1\Designer\Client Diskset
```

and on 64 bit hardware:

```
C:\Program Files (x86)\PBBI CCM\DOC1\Designer\ClientDiskSet
```

Once installed the Administrator can login to Designer and 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.

Example Client/Server installation



Distributed install

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

NOTE: remote connections must be enabled for the SQL server installed in your distributed environment. Refer to the SQL server documentation for details on how to do this.

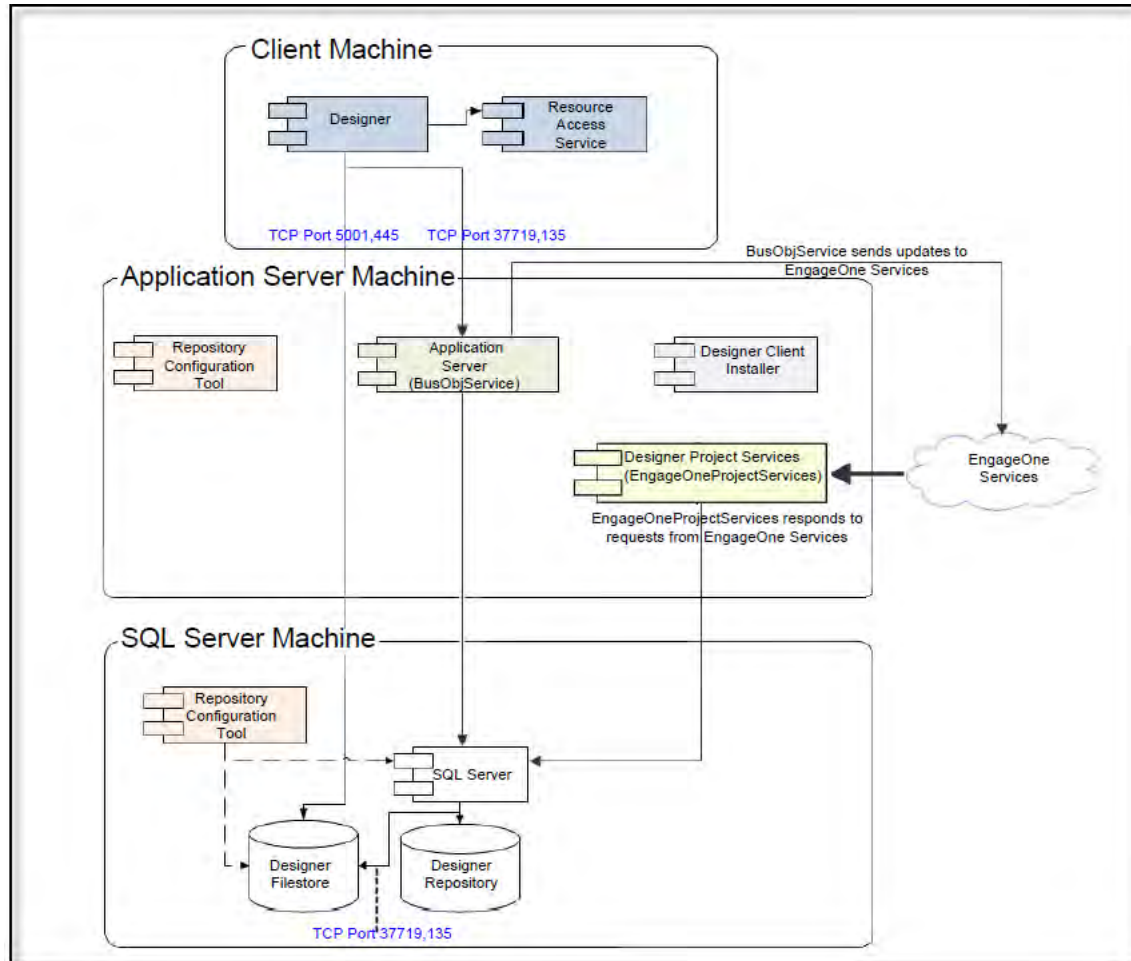
You must first create a folder for the filestore and share, and grant read/write access as with the Client/Server install – see [Client/Server and Distributed considerations](#) on page 22 for details.

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.

There are 2 options available for distributed install.

- Install Repository Configuration Tool on the SQL Server machine, then install Application Server on the Application Server machine. Refer to [Distributed install Option 1](#) on page 19
- Run a SQL script on the SQL Server machine, then install Designer (Client/Server setup type) on the Application Server machine. Refer to [Distributed install Option 2](#) on page 20

Example distributed installation



Distributed install Option 1

Install the Repository Configuration Tool on the SQL Server

1. Launch the `setup.hta` file from the installation media on the SQL Server and select the Install Repository Configuration Tool option in the contents window. The installation material will be copied to the Windows directory `%temp%` and start to run. Enter the SQL server name and instance if required. Click **Open/Yes/OK** as appropriate to open files, allow programs and install requirements.
2. If this is a new installation you must enter the filestore location, for example:
`\\<server>\<filestore>`.
3. The Repository Configuration Tool will be installed. Select **OK/Finish** as necessary to complete the installation process.

Install the Application Server

1. Launch the `setup.hta` file in the installation media on the server machine and select the **Install Application Server** option in the contents window. The installation material will be copied to the Windows directory `%temp%` and start to run. Click **Open/Yes/OK** as appropriate to open files, allow programs and install requirements.
2. Prerequisites will be installed automatically as necessary.
3. The Designer installation will then start. Step through the screens to run the installation process.
4. The **Repository Configuration Tool** will be run. Supply the connection details to the SQL server. Enter the server name and instance if required. In the next screen select the required Designer repository and the Designer Default User.

Distributed install Option 2

It is important to note that if you use this install option you will not be able to use the Repository Configuration Tool to backup and restore repositories. You will need to perform these operations using other tools, for example SQL Server Management Studio for the database and 7-Zip for the filestore. When backing up the database, make sure you back up the filestore at the same time.

Configure the SQL Server

1. Locate the following files in the release media sub-directory `windows\work center`

ConfigureSQLServer.SQL	This script configures the SQL Login that the Application Server will use to connect to SQL. It will also configure options required to support upgrading databases from older versions of the software, and to support the import of snapshots created from older versions of the software. You can control these options by setting the optional configuration flags inside the script, so please review the script before proceeding.
Default.dbk	The default database file for a new repository.
Default_Filestore.zip	The default filestore for a new repository.

2. If you are upgrading a repository, skip to step 4. If you are creating a new repository, restore Default.dbk and extract the contents of Default_Filestore.zip into your filestore directory. If you have a database/filestore backup, then now is the time to restore it.
3. Set the restored database to be multi-user and update the Global table in the restored database to reflect the filestore location by executing the following SQL:

```
alter database <database name> set multi_user
use <database name>
update Global set Value = '<filestore location>'
where Name = 'FileStoreRoot'
```

When creating a new repository using the Default database, create a new identifier for it using the following SQL:

```
update Global set Value = NEWID() where Name = 'UniqueGuid'
```

4. 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 [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 must be used when configuring the Application Server.
Using existing SQL Server 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 must be used when configuring the Application Server.
The server principal '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.

Install the Application Server

1. If you are upgrading the software on a machine where the Application Server was installed using Distributed Install Option 1, then you must uninstall the old Designer first.
2. Install Designer, selecting the Client/Server setup type.
3. When the Repository Configuration Tool is launched, use SQL Server Authentication to login with the credentials specified in *Configure SQL Server - step 4* and press Next.
4. Select the Upgrade option and the name of the database to upgrade, and press Next.
5. Select *SQL Server Authentication with specific credentials* and enter the credentials specified in *Configure SQL Server - step 4*.
6. Continue with the installation as normal.

Designer client installation

To install Designer client on to a client machine:

On the client machine browse to where the client installer software was installed on the Application Server machine and from the Client Diskset directory run the setup-designer-client.exe program.

This runs a simple installation wizard that installs the client software and registers the repository location. Note that the default location for the client diskset is:

```
C:\Program Files\PBBI CCM\DOC1\Designer\Client Diskset
```

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 both the filestore and the client diskset folders are available to all users who will be accessing them.

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.

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. **For 64 bit Windows**, on the server machine use the following path to locate the BusObj.config file: C:\Program Files (x86)\PBBI CCM\DOC1\Designer\Server\

For 32 bit Windows:

C:\Program Files\PBBI CCM\DOC1\Designer\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. **For 64 bit Windows**, on the client machine(s) use the following path to locate the Processor.dll.config file: C:\Program Files (x86)\PBBI CCM\DOC1\Designer\Client\

For 32 bit Windows:

C:\Program Files\PBBI CCM\DOC1\Designer\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. ***For 64 bit Windows***, on the client machine(s) use the following path to locate the ResourceAccessService.exe.config file: C:\Program Files (x86)\PBBI CCM\DOC1\Resource Access Service\

For 32 bit Windows:

C:\Program Files\PBBI CCM\DOC1\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.

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 28
- Run the search service installer, refer to [Running the search service installer](#) on page 30
- Optionally customize `searchsettings.json` according to your operational requirements, refer to [Customizing searchsettings.json parameters](#) on page 34

- Use the indexer tool to index assets in your repository, refer to [Maintaining search indexes](#) on page 32

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+

Note that the `JAVA_HOME` environment variable must be assigned to the path of the JDK.

- Microsoft Message Queuing (MSMQ). Refer to [Enabling MSMQ](#) on page 28.
- Microsoft Powershell version 4.0 or higher
- Elastic search 5.2.2.

Note that 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 30.

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:

```
<Designer Installation Download>\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 28 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-5.2.2 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.

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 34 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 38

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 34 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 at:

```
<Designer Installation Download>\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 34 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: C:\Program Files (x86)\PBBI CCM\DOC1\Designer\Server.

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:

C:\Program Files (x86)\PBBi CCM\DOC1\Designer\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.

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:

```
C:\Program Files (x86)\PBBI CCM\DOC1\Designer\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	
LogLevel				
	Default	Log level of the service specific messages captured while the service is running	Warning	Can be Debug, Information, Warning or Error
	System	Log level of the system messages captured while the service is running	Warning	As above

Section	Name	Description	Default value	Comments
	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
SecurityAuthentication				
	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.

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 lose 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 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 reported an error

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

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

Occurs when:	<ul style="list-style-type: none"> • Elastic Search service is down. • Indexing (save, branch, etc.). In this scenario, the index will loose its synchronization.
--------------	---

Troubleshooting informationIn 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
-

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 informationIn 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 32 for further information.

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.
-

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

4 - Start-up & login

In this section

Logging into Designer	45
Resources for samples /confidence tests	45

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

47

Repository selection

In both a Standalone environment and on a machine running the Server component the Repository Configuration Tool is available from the Group1 or PBBI CCM folder within the 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

49

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 Pitney Bowes 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 ©2019 Pitney Bowes, Inc. 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 Pitney Bowes, Inc. (PBS). To the fullest extent permitted by applicable laws PBS 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 PBS is prohibited to the fullest extent permitted by applicable laws.

Nothing in this notice shall limit or exclude PBS 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://logging.apache.org/log4net/license.html>. 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.



3001 Summer Street
Stamford CT 06926-0700
USA

www.pitneybowes.com/us