# Pilot Data Server Administrator Guide

Version 9.6

**Disclaimer**

Vizrt provides this publication “as is” without warranty of any kind, either expressed or implied. This publication may contain technical inaccuracies or typographical errors. While every precaution has been taken in the preparation of this document to ensure that it contains accurate and up-to-date information, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained in this document.

Vizrt’s policy is one of continual development, so the content of this document is periodically subject to be modified without notice. These changes will be incorporated in new editions of the publication. Vizrt may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time. Vizrt may have patents or pending patent applications covering subject matters in this document. The furnishing of this document does not give you any license to these patents.

**Antivirus Considerations**

Vizrt advises customers to use an AV solution that allows for custom exclusions and granular performance tuning to prevent unnecessary interference with our products. If interference is encountered:

- **Real-Time Scanning:** Keep it enabled, but exclude any performance-sensitive operations involving Vizrt-specific folders, files, and processes. For example:
  - C:\Program Files\[Product Name]
  - C:\ProgramData\[Product Name]
  - Any custom directory where [Product Name] stores data, and any specific process related to [Product Name].
- **Risk Acknowledgment:** Excluding certain folders/processes may improve performance, but also create an attack vector.
- **Scan Scheduling:** Run full system scans during off-peak hours.
- **False Positives:** If behavior-based detection flags a false positive, mark that executable as a trusted application.

**Technical Support**

For technical support and the latest news of upgrades, documentation, and related products, visit the Vizrt web site at www.vizrt.com.

**Created on**

2026/05/20

# Contents

# 1 Introduction

Pilot Data Server is installed as an application layer on top of the Viz Pilot database, and acts as an application server for accessing Viz Pilot's database and other services, and enables the use of features including Crop Service, Template Tagging, Update Service, Person Search, search on Viz One, Thumbnail Generator and Timeline Editor in Viz Pilot workflows.

It has a web interface with a REST API, accessible for 3rd party integrations. Pilot Data Server is required when using Graphic Hub as the Viz Pilot database, and for integrating with Viz Story, Viz Pilot Edge and Template Builder.

The full Pilot Data Server system contains the following parts:

| Type | Component |
|---|---|
| Application | Pilot Data Server Launcher |
| Services | Vizrt Pilot Data Server<br>Vizrt Script Runner<br>Vizrt Crop Service<br>Vizrt Thumbnail Generator |

## 1.1 Pilot Data Server Launcher

The Pilot Data Server Launcher is a separate application that can be used to configure, start and stop the Pilot Data Server service.

## 1.2 Pilot Data Server Service

The Pilot Data Server service is the main module of this system. It provides a REST API for web applications to access the Pilot database, and it has a changelog event mechanism for broadcasting changes to templates, elements and concepts inside the database.

## 1.3 Script Runner Service

The Script Runner Service comes packaged with Pilot Data Server and is automatically installed. It can be configured, started and stopped with the Pilot Data Server Launcher. The Script Runner Service provides a simple way for users to use the Media Sequencer's *Live Update* mechanism without needing to create a custom service. The Script Runner acts as an External Update Service for Media Sequencer. See Update Script Editor in the Viz Pilot User Guide.

## 1.4 Crop Service

The Crop Service provides a way to crop images. The Crop Service is provided in a separate installer. It can be installed and run on the same physical host as Pilot Data Server. If installed on the same host, it can be configured, started and stopped through the Pilot Data Server Launcher.

## 1.5 Thumbnail Generator

Thumbnail Generator is a Windows service that automatically generates thumbnails of data elements when they are created or changed. The thumbnails are stored in the Viz Pilot database. Thumbnail Generator requires a Preview Server, which is used to serve the thumbnails in the data element feed.

## 1.6 Related Documents

The Viz Pilot User Guide provides complete documentation of the Viz Pilot system. For more information and documentation for all Vizrt products, please visit the links available under Support.

## 1.7 Feedback and Suggestions

We encourage suggestions and feedback about our products and documentation. To give feedback and/or suggestions, please contact your local Vizrt customer support team at www.vizrt.com.

## 1.8 Support

Support is available at the Vizrt Support Portal.

For more information about all Vizrt products, visit:

- www.vizrt.com
- Vizrt Documentation Center
- Vizrt Training Center
- Vizrt Forum

# 2 Installation

This section covers:



## 2.1 Installation Prerequisites

- Visual C++ Redistributable Packages for Visual Studio 2015

The packages are included in the Pilot Data Server bundle installer, and can also be found on Microsoft's website.

## 2.2 Installing Pilot Data Server

There are two options to install Pilot Data Server:

- Use the bundle installer: *PilotDataServerBundleInstaller-x64-[version].exe*, or
- The MSI installer: *PilotDataServerInstaller-x64-[version].msi*.

| Installer | Content |
|---|---|
| Bundle installer | Pilot Data Server<br>Visual Studio C++ 2015 redist<br>Codemeter runtime |
| MSI | Pilot Data Server<br>Thumbnail Generator Service (optional) |

To install with a UI, double click one of these installers and follow the on-screen instructions.

> **Note:** The default location is *%ProgramFiles%\Vizrt\Pilot Data Server*.

### 2.2.1 ShowEdge Installer

The ShowEdge installer is not bundled into the Pilot Data Server installer, nor into the Viz Pilot Edge installer, it has been available from the Pilot Data Server end point `/showedge`. To still be able to serve out the ShowEdge

installer from this end point, the file `ShowEdge.[version].exe` needs to be downloaded from the FTP, and copied to the `app/showedgeinstaller` folder of the Pilot Data Server installation (for example, to `C:\Program Files\Vizrt\Pilot Data Server\app\showedgeinstaller`).

### 2.2.2 Scripted Installation

For a scripted installation, Pilot Data Server can be installed without using the installer GUI. This can be done with the MSI installer found on the Vizrt's FTP, or extracted from the bundle installer with the *-msi* command line option.

> **Note:** For silent install, the step for configuring the license is skipped. The license configuration of Pilot Data Server is stored at *%ProgramData%\Vizrt\Pilot Data Server\licenses.json*, and the file itself can be copied from one installation to another as a way to apply known license configurations without using the GUI.

In a command shell, run the following command:

```
msiexec /i VizrtPilotDataServer-x64-x.x.x.x-Release.msi /quiet
```

If required to set properties for the silent installer for the base url, use the following MSI properties:

| Property | Example value |
|---|---|
| BASE_URL | https://myserver.com:443/pds |
| BASE_PATH | pds |
| SECURE_PORT | 7373 |
| CONNECTION_METHOD | GH or ORACLE |
| ORACLE_USERNAME | pilot |
| ORACLE_PASSWORD | pilot |
| ORACLE_DATA_SOURCE | localhost/vizrtdb |
| GH_HOST | localhost |
| GH_PORT | 19398 |
| GH_USER_ID | Admin |
| GH_PASSWORD | VizrtDb123 |

| Property | Example value |
| --- | --- |
| INSTALLDIR | "c:\\Program Files\\Vizrt\\Pilot Data Server" |

Examples

```
msiexec /i <application.msi> CONNECTION_METHOD="ORACLE" ORACLE_USERNAME="pilot"
 ORACLE_PASSWORD="pilot" ORACLE_DATA_SOURCE="hostname/vizrtdb" BASE_PATH="pds" /log
install.log
msiexec /i <application.msi> CONNECTION_METHOD="GH" GH_HOST="hostname"
 GH_PORT="19398" GH_USER_ID="admin" GH_PASSWORD="some password123" BASE_URL="https://
something.com/pds" SECURE_PORT="7373" /log install.log
```

## 2.3 Installing the Thumbnail Generator

The Thumbnail Generator (see the Viz Pilot User Guide) can be installed as an optional component during the Pilot Data Server installation. The option for Thumbnail Generator installation is disabled by default. To install it, select the Thumbnail Generator manually when installing Pilot Data Server through the MSI installer.

### 2.3.1 Silent Installation

Since the Thumbnail Generator is not installed by default, the *INSTALLLEVEL=2* needs to be added to the command string for the Thumbnail Generator to be included during a silent Pilot Data Server installation:

```
msiexec /i VizrtPilotDataServer-x64-x.x.x.x-Release.msi INSTALLLEVEL=2 /quiet
```

To upgrade an existing Pilot Data Server to include the Thumbnail Generator without uninstalling it, the following command needs to be run:

```
msiexec /i VizrtPilotDataServer-x64-x.x.x.x-Release.msi
ADDLOCAL=ThumbnailGeneratorService /quiet
```

## 2.4 Alternative Installations

- If **Crop Service** installation is required (see the relevant section in the Installation chapter in the Viz Pilot User Guide), it must be installed separately, after Pilot Data Server.
- If support for **multiple Viz Pilot Database schemas** is required, *separate* Pilot Data Servers must be installed on *separate* machines.

## 2.5 Firewall Exceptions

For Pilot Data Server to be accessible from other machines, Windows firewall exceptions are required. Pilot Data Server adds the following inbound rules to the Windows firewall during installation:

Allow connections on port:

- 8177 - Pilot Data Server HTTP port
- 7373 - Pilot Data Server HTTPS port
- 9876 - STOMP event notifications port
- 1981 - Script Runner HTTP port
- 1982 - for Pilot Script Runner HTTPS port

These ports are the recommended ports for the service. They are removed when Pilot Data Server is uninstalled.

## 2.6 Antivirus

Vizrt does not recommend or test antivirus systems in combination with Vizrt products, as the use of such systems can potentially lead to performance losses. The decision for the use of antivirus software and thus the risk of impairments of the system is solely at the customer's own risk.

There are general best-practice solutions, these include setting the antivirus software to not scan the systems during operating hours and that the Vizrt components, as well as drives on which clips and data are stored, are excluded from their scans (as previously stated, these measures cannot be guaranteed).

A list of services and folders which should be excluded from scanning can be found below:

**Services:**

- vizrt_crop_server
- VizrtMist
- vizrt_pilot_data_server
- vizrt_script_runner
- vizrt_thumbnail_generator

**Folder:**

- *C:\Program Files (x86)\Vizrt*
- *C:\ProgramData\vizrt*
- *C:\Program Files\Vizrt*

**Processes:**

- PilotCropServiceHost.exe
- PilotDataServerHostService.exe
- PilotScriptRunnerHostService.exe
- PilotThumbnailGeneratorHostService.exe
- caddy-service.exe
- caddy.exe

# 3 Default Ports and Protocols

The table below shows all of the default Protocols and Ports used with Pilot Data Server.

## 3.1 HTTP

| Name | Default Value | Comments |
|---|---|---|
| Pilot Data Server | http://pds-host:8177 | The port used for HTTP is configurable in the *DataServerConfig.ini* file on *%ProgramData%\Vizrt\Pilot Data Server*. |
| Script Runner | http://pds-host:1981 | Installed with Pilot Data Server and runs on the same host machine. The port used for HTTP is configurable in the file *PilotScriptRunnerHostService.exe.config* in the Pilot Data Server installation folder. |
| Crop Service | http://pds-host:8178 | Separate installer that normally runs on the same host as Pilot Data Server. The port used for HTTP is configurable in the file *PilotCropServiceHost.exe.config* in the Crop Service installation folder. |
| Graphic Hub REST | http://gh-host:19398 | Graphic Hub uses the same port for HTTP and HTTPS. |
| Viz One | http://vizone-host:80/thirdparty | |
| Preview Server | http://ps-host:21098 | |
| REST VOS / GH Image Extension | http://imex-host:19390/restvos/service | The image extension UI can be accessed on *http://imex-host:19390/imagelibrary/index.html*. |
| Viz Pilot Preview Port | 50008 | Viz Engine preview port used for snapshot previews in Viz Pilot. |

| Name | Default Value | Comments |
|---|---|---|
| Viz Arc MOS Panel | http://arc-host:9004/mos-plugin | |
| PDS changelog notifications | stomp://pds-host:9876/?destination=/changelog&lastKnownId=-1 | |

## 3.2 HTTPS

| Name | Default Value | Comments |
|---|---|---|
| Pilot Data Server | https://pds-host:7373 | The port used for HTTPS is configurable from the Pilot Data Server Launcher. The service needs to restart. |
| Script Runner | https://pds-host:1982 | The port used for HTTPS is configurable from the Pilot Data Server Launcher. The service needs to restart. |
| Crop Service | https://pds-host:8179 | The port used for HTTPS is configurable from the Pilot Data Server Launcher. The service needs to restart. |
| Graphic Hub REST | https://gh-host:19398 | Graphic Hub uses the same port for HTTP and HTTPS. |
| Viz One | https://vizone-host.example.com:443/thirdparty | |
| Preview Server | https://ps-host:4443 | |
| REST VOS / GH Image Extension | | No support for HTTPS. |
| Viz Pilot Preview Port | 50008 | Viz Engine preview port used for snapshot previews in Viz Pilot. |

| Name | Default Value | Comments |
| --- | --- | --- |
| Viz Arc MOS panel | https://arc-host:9005/mos-plugin | |
| PDS changelog notifications | stomp://pds-host:9876/?destination=/changelog&lastKnownId=-1 | |

# 4 Configuration

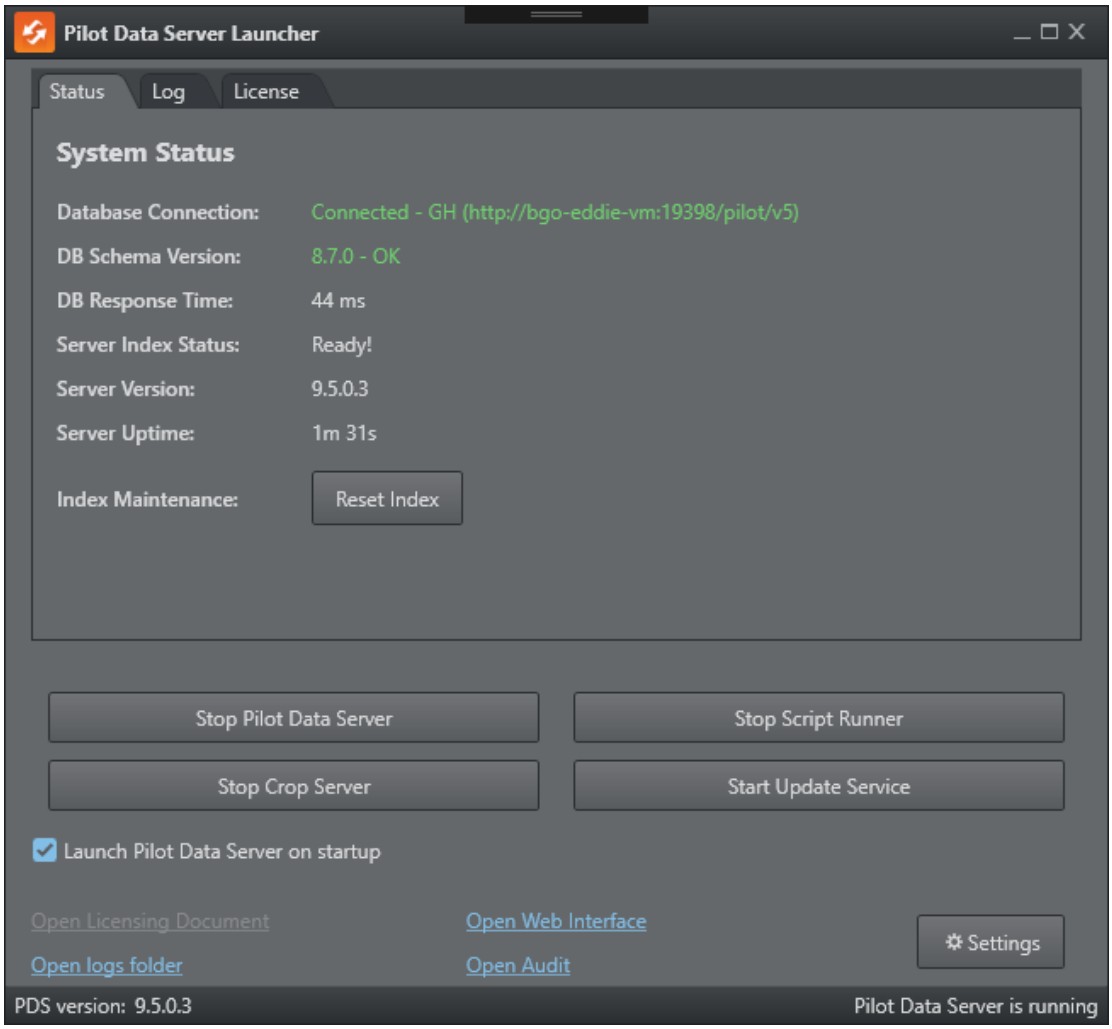
This section covers the following topics:



## 4.1 Launcher

The Pilot Data Server Launcher is a minimal dashboard to start and stop services, and offers a UI to configure the system.

## 4.2 Status Panel




The Status panel provides an overview of the current Pilot Data Server state. The information shown is not taken from a single authoritative status source. Instead, it is derived from two runtime signals: the launcher continuously monitors the **server log** to infer index-building progress, and it periodically queries the **/service/health** endpoint to retrieve health information such as **database connectivity**, **schema compatibility**, **server version**, and **uptime**. Because these values are gathered asynchronously and from different sources, the panel should be understood as an operational summary for troubleshooting and monitoring, not as a strict real-time guarantee of internal service state.

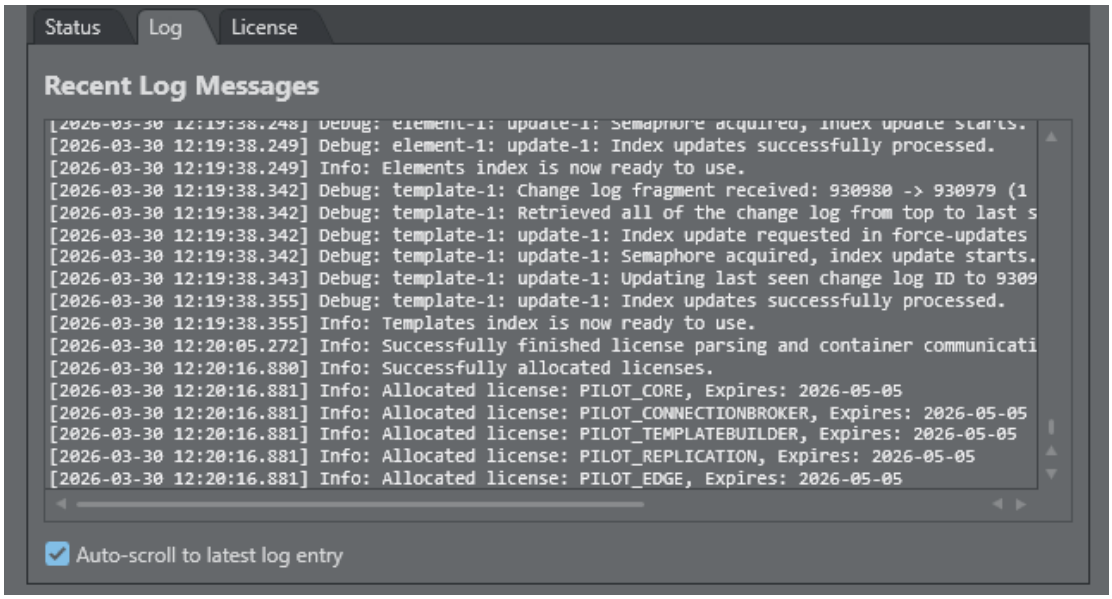
In addition, the following functions are available:

- **Settings:** Opens the settings window, with the main configuration topics:
  - Authentication
  - Script Runner
  - Licensing
  - Pilot Data Server Settings

    - Log Files
    - Crop Service
    - Pilot Update Service
    - Audit Logging
- **Start/Stop:** Start and stop services.
- **Launch Pilot Data Server on startup:** Guarantees the Pilot Data Service and additional services are started when the computer starts.
- **Open Licensing Document:** Opens the local documentation to configure the licenses for Pilot Data Server.
- **Open logs folder:** Opens the logs and ini file folder, normally *%ProgramData%\Vizrt\Pilot Data Server*.
- **Open Web Interface:** Opens the web page of the currently running Pilot Data Server.
- **Open Audit:** Opens the Audit window.

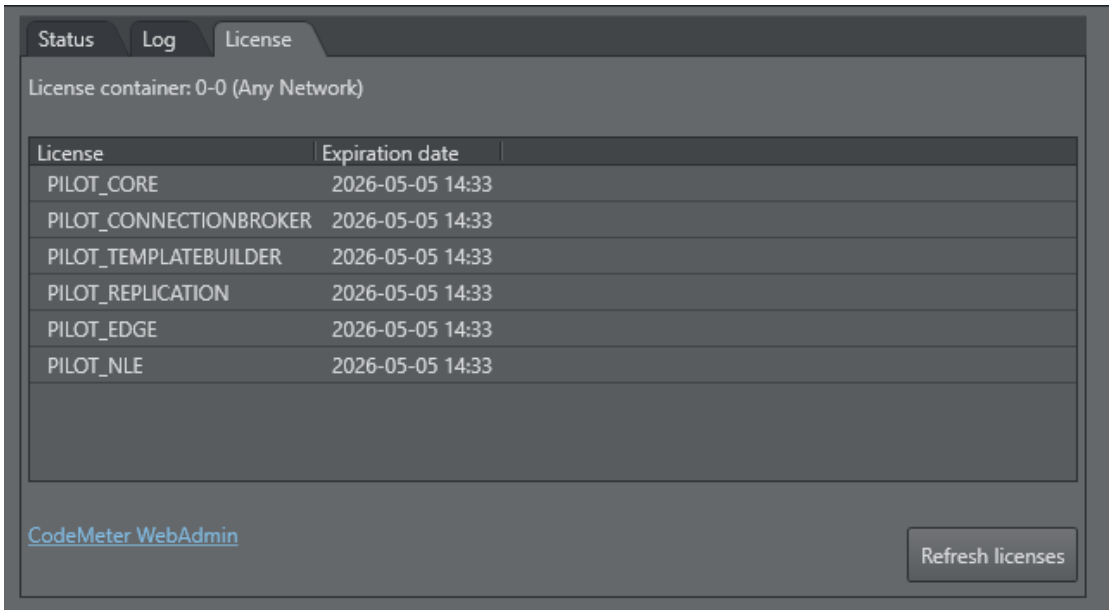
---

## 4.3 Log Panel




The Log panel displays a live view of recent Pilot Data Server log messages, to support troubleshooting and operational monitoring. It reads from the local server log file and shows the latest entries in chronological order, including timestamp, log level and message text. The panel is intended as a convenience view inside the launcher and only shows a limited rolling window of recent messages, not the complete log history. For a full investigation, use the Open logs folder link to inspect the original log files.

## 4.4 License Panel



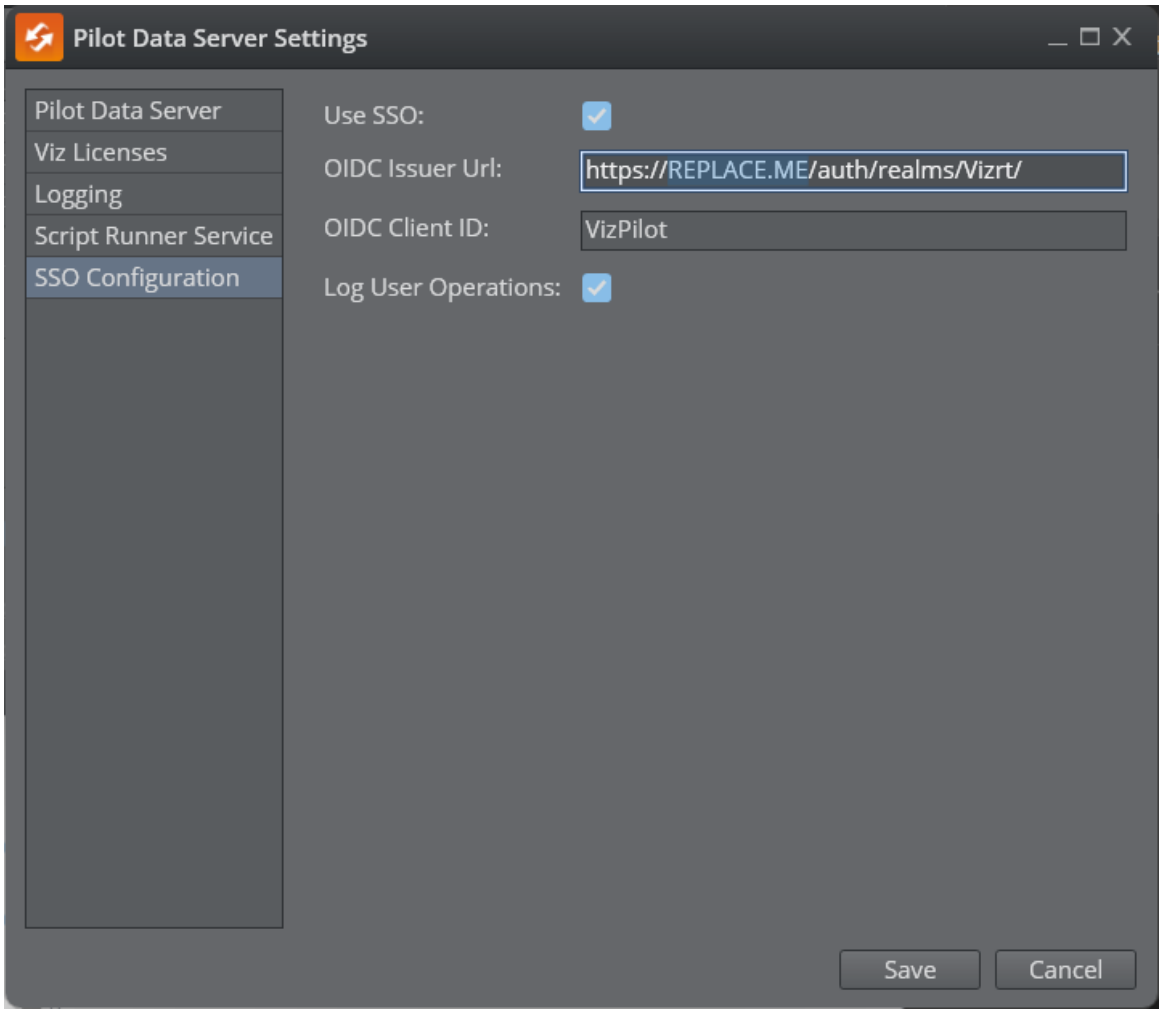

- **Currently activated licenses:** List of the currently activated licenses on the license server/container.
- **CodeMeter WebAdmin:** Opens Wibu license administration in a browser, and requires the Codemeter service to run.
- **Refresh licenses:** Reconnects to the license system and does another attempt to fetch the current licenses.

**Note:** The Codemeter.exe service of the Wibu Licensing system, must be running in order to start Pilot Data Server.

## 4.5 Authentication



### 4.5.1 Configuring Pilot Data Server




- **Use SSO:** Decides whether the system enables authentication or not. It requires TLS/SSL enabled, and the full system needs to communicate over HTTPS.
- **OIDC Issuer URL:** URL to the OIDC compliant identity provider. See below.

- **OIDC Client ID:** Public identifier for the applications. See below for a detailed explanation.
- **Log User Operations:** If enabled, user operations that add, delete or modify templates and data elements, are added to the regular log. The user ID and name or email (depending on the identity provider) is added to the log. For instance, a user modifies a data element:

```
2024-11-06 11:36:23.189 [21] INFO: john (3709b32c-032a-4b78-af83-c82b65c90868)
modified resource at https://host-pilot:7373/dataelements/168 (PUT)
```

### 4.5.2 Roles

Pilot Data Server can be configured to use Authentication to control access to Viz Pilot Edge, Template Builder and DataServerConfig.

There are 4 different roles defined:

- pilot-journalist
- pilot-editor
- graphic-designer
- pilot-admin

It is important to understand the distinction between roles and groups. Roles are what grants a user privileges, and a role can be assigned to a user directly, or to all members of one or more groups depending on the identity provider being used. The role names are fixed and defined by Vizrt, but you are free to define or reuse groups as you like.

These are the access rights for the different roles:

| | Viz Pilot Edge | Template Builder | Data Server Settings |
|---|---|---|---|
| **pilot-journalist** | ✓ | ✗ | ✗ |
| **pilot-editor** | ✓ | ✗ | ✗ |
| **graphic-designer** | ✓ | ✓ | ✗ |
| **pilot-admin** | ✓ | ✓ | ✓ |

In addition, a user's profile or group can determine which Pilot Collections they can access, based on role-to-collection name matching. Roles prefixed with **pilot-collection-** are used to match authenticated users to specific collections. For example:

- A user with the roles “pilot-journalist” and “pilot-collection-news”, only has access to the Pilot Collection named **news.**
- Users can have multiple roles, allowing them to access multiple Pilot Collections.

See the section **Pilot Collections** in the Template Builder user manual.

### 4.5.3 Identity Configuration

Pilot Data Server uses the OpenID Connect Protocol (OIDC) to authenticate users. It can, in theory, be configured to integrate with any OIDC compliant identity provider.

The following providers have been tested and are supported:

- Microsoft Entra ID (formerly Azure Active Directory)
- Vizrt SSO (Keycloak)

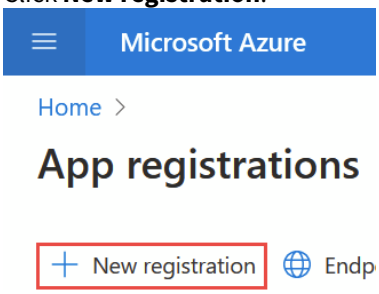
#### Microsoft Entra ID

The Entra ID application configuration consists of two parts: an App Registration for the technical details of the application, and an Enterprise Application that defines user and group access and their role assignments, as well as other aspects of how the app integrates with your tenant.

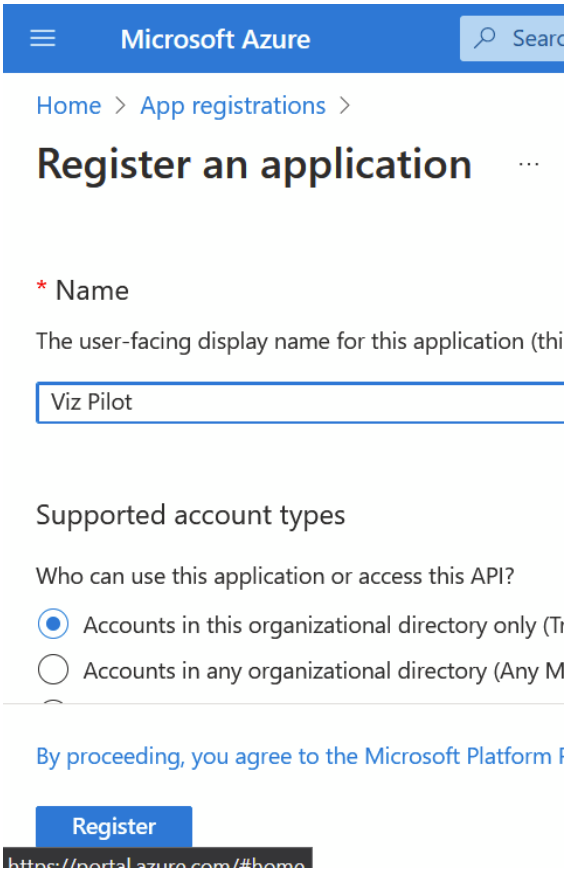
**Warning:** Use the new Microsoft Graph app manifest format. The enclosed configuration file *entra-id-pilot-sso.json* is updated to this format.

To connect Viz Pilot to Microsoft Azure Entra ID (formerly Azure AD), log in to your tenant in the Azure Portal, then perform the following steps:

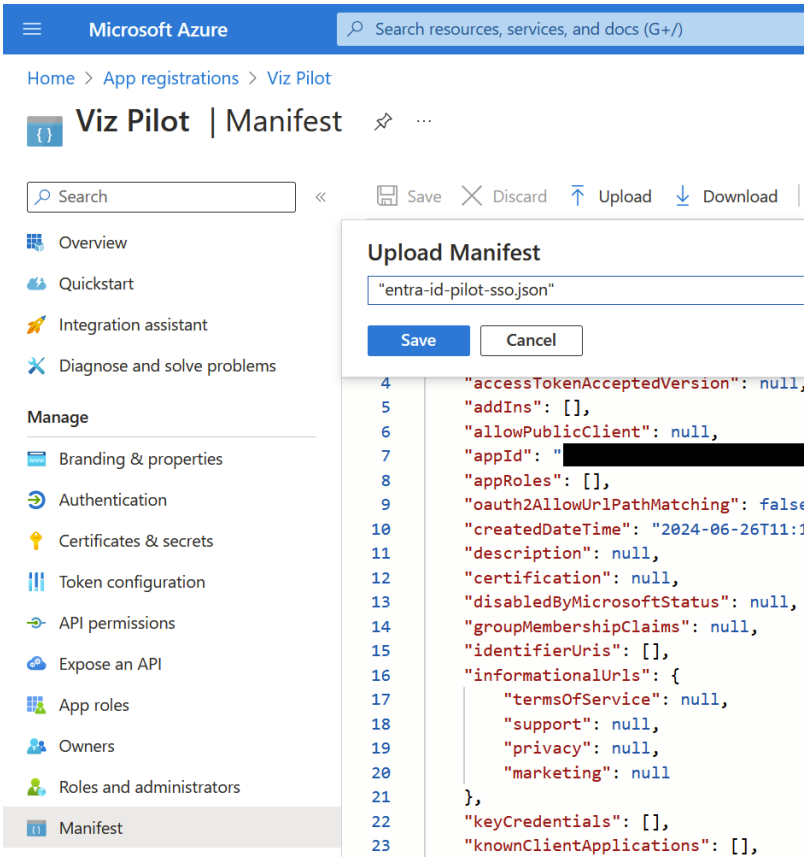
1. Go to the App Registrations blade.
2. Click **New registration**:



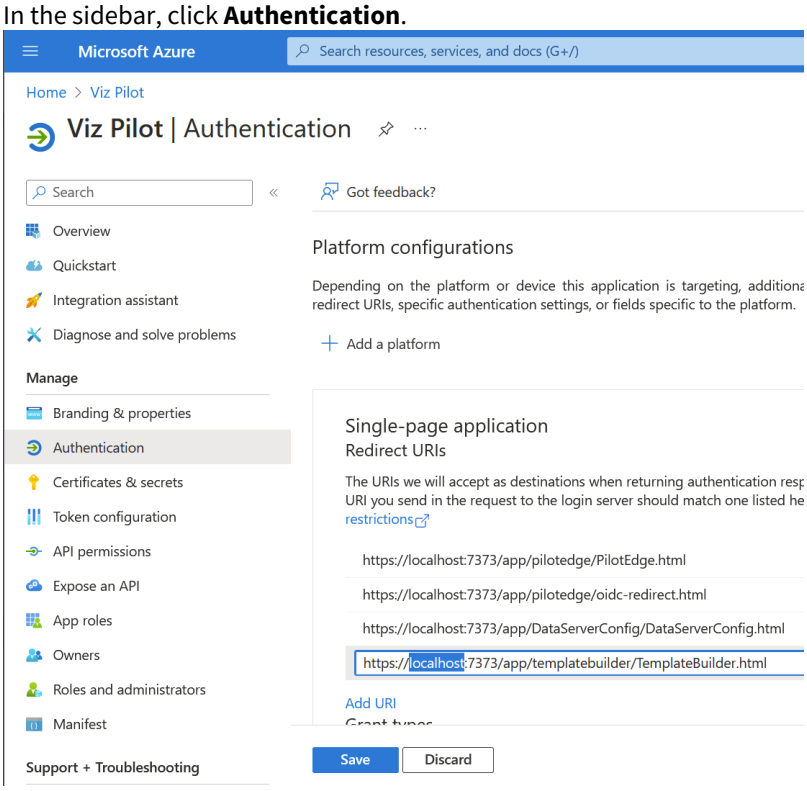

3. Type a name, for example Viz Pilot, and leave the other settings as is. Then click **Register**.



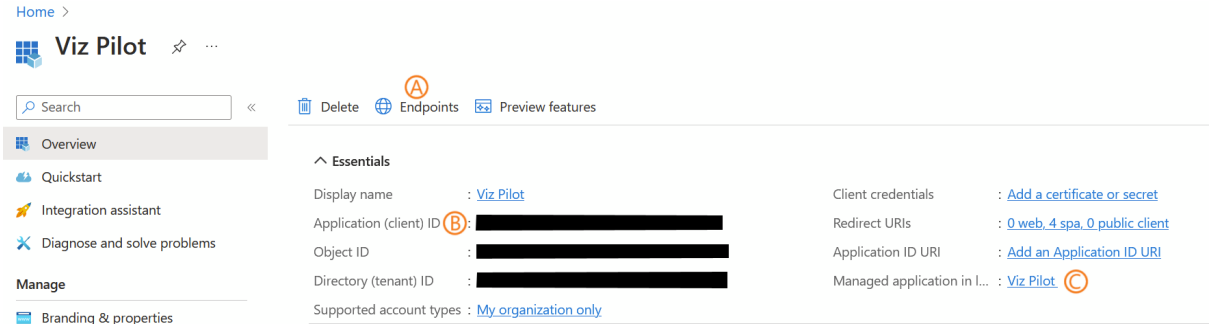

4. In the sidebar, click **Manifest**.



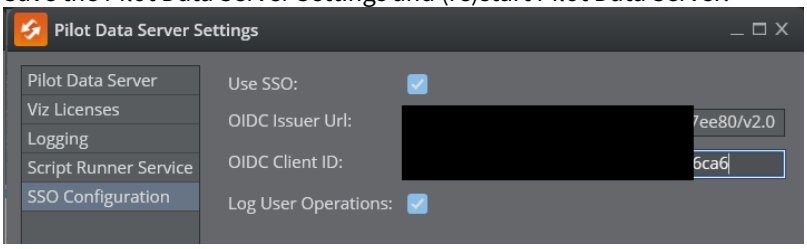

5. Click **Upload**, select the provided JSON file, then click **Save**.
6. In the sidebar, click **Authentication**.



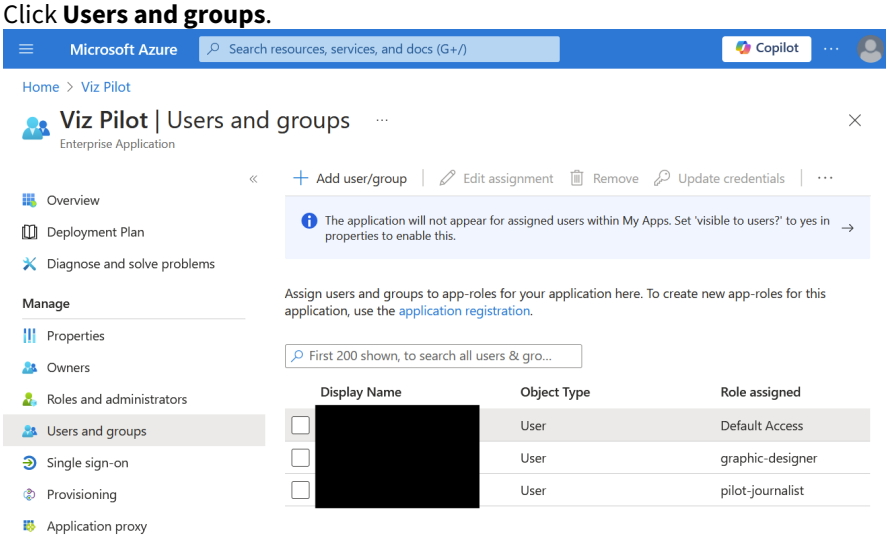
7. Edit the URLs to have the correct hostname, and click **Save**.
   Optional:
   a. In the sidebar, click **API permissions**, then click **Grant Admin Consent for <your tenant name>**.
   b. In the sidebar, click **Branding & properties**, then upload the provided logo.
   c. In the sidebar, click **Owners**, then assign yourself and/or other users that should have access to manage this app registration.
8. In the sidebar, click **Overview**:




9. Open the Pilot Data Server Launcher, click **Settings** and select the SSO page (or use the notepad to collect the following values for a later entry).
10. Click **Endpoints** (A) and find the **OpenID Connect metadata document** field. Copy the left part up (but not including */.well-known/openid-configuration)*, and paste it into the **OIDC Issuer Url** field in Pilot Data Server Settings.
11. Find the **Application (client) ID** (B) value and copy it to the **OIDC Client ID** field in Pilot Data Server Settings.
12. Save the Pilot Data Server Settings and (re)start Pilot Data Server.




13. Click the link with the label **Managed application in local directory** (C), which takes you to the Enterprise Application blade.
14. Click **Users and groups**.

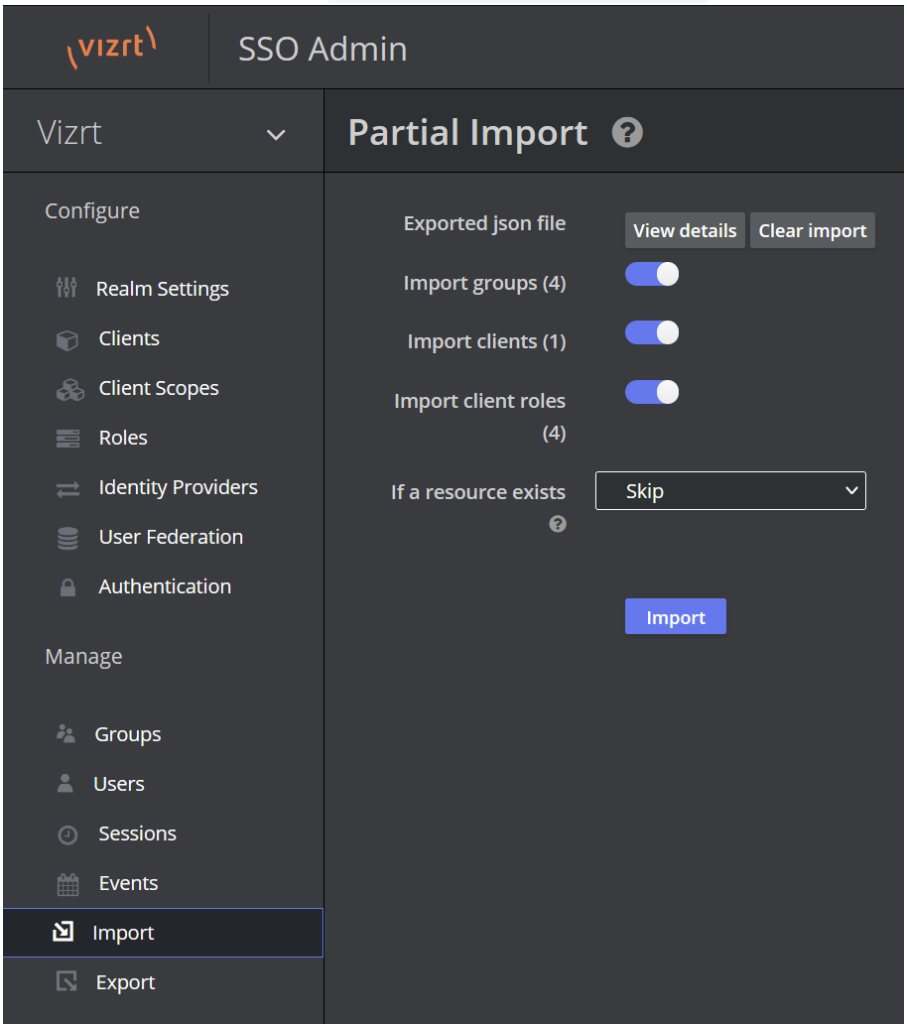


15. Add users and groups to roles as you wish (**Default Access** is added automatically but does not grant access to any Viz Pilot apps, so you may remove it).
16. You may also inspect the other pages in the sidebar to customize behavior and security, and assign other users that should have access to manage this Enterprise Application, as owners . You do not need to perform Provisioning for Viz Pilot.

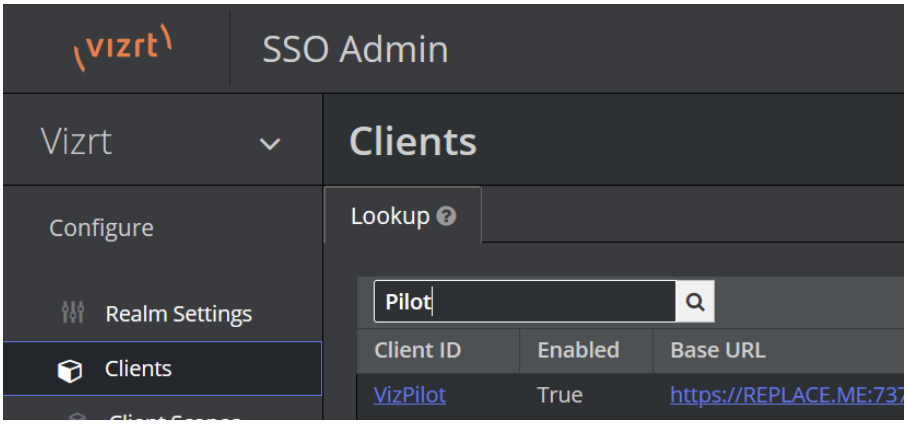
## Vizrt SSO (Keycloak)

To connect Viz Pilot to Vizrt SSO running in an existing Viz One deployment, perform the following steps:

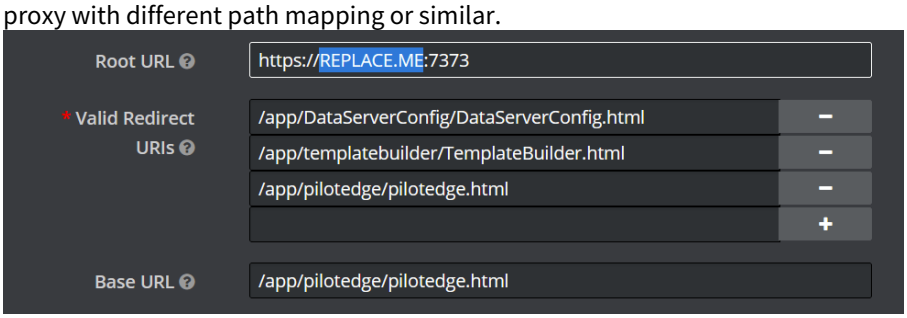
1. Download the `keycloak-pilot-sso.json` file from the same folder that you got the installer on Vizrt's FTP server.
2. Log in to the Viz One SSO Admin pages, and go to the **Import** tab in the left menu.
3. Click **Select file** and pick the `keycloak-pilot-sso.json` file.



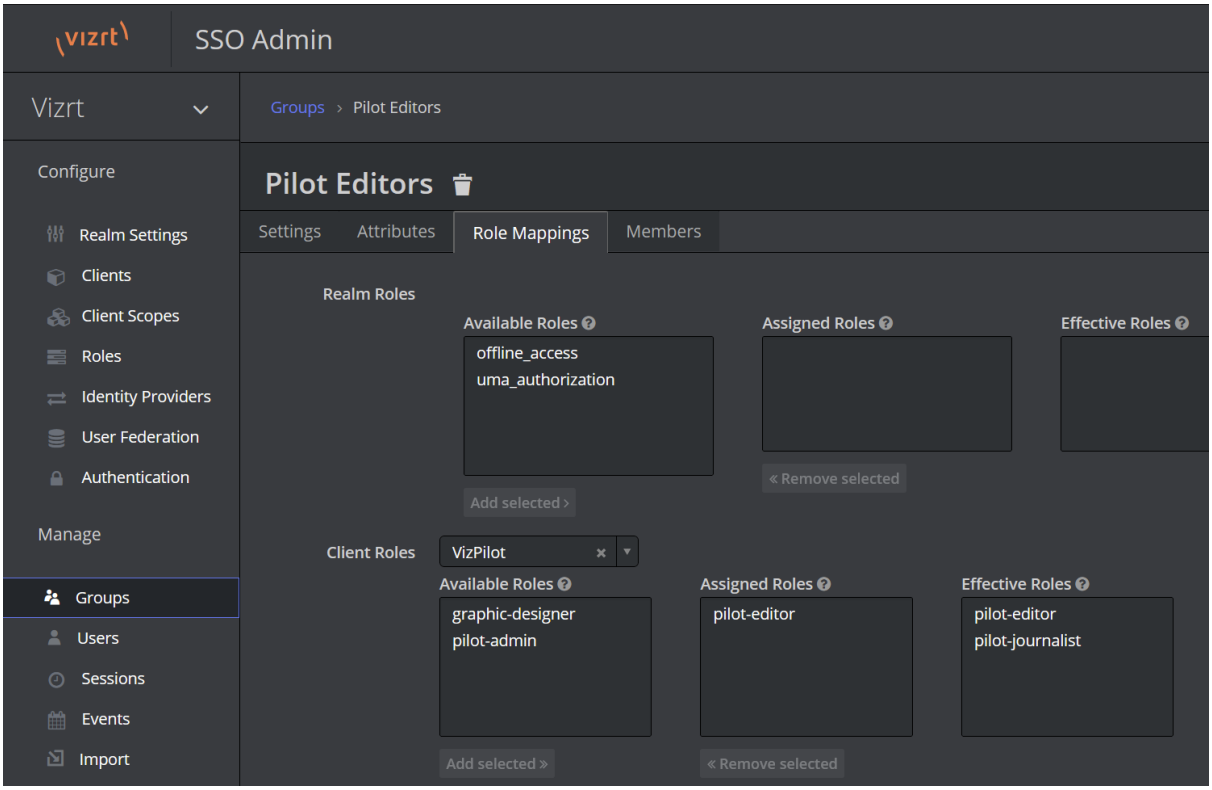

4. Review the options and click **Import** when ready.
5. Go to the **Clients** tab, find and open the **VizPilot** client that was just imported.



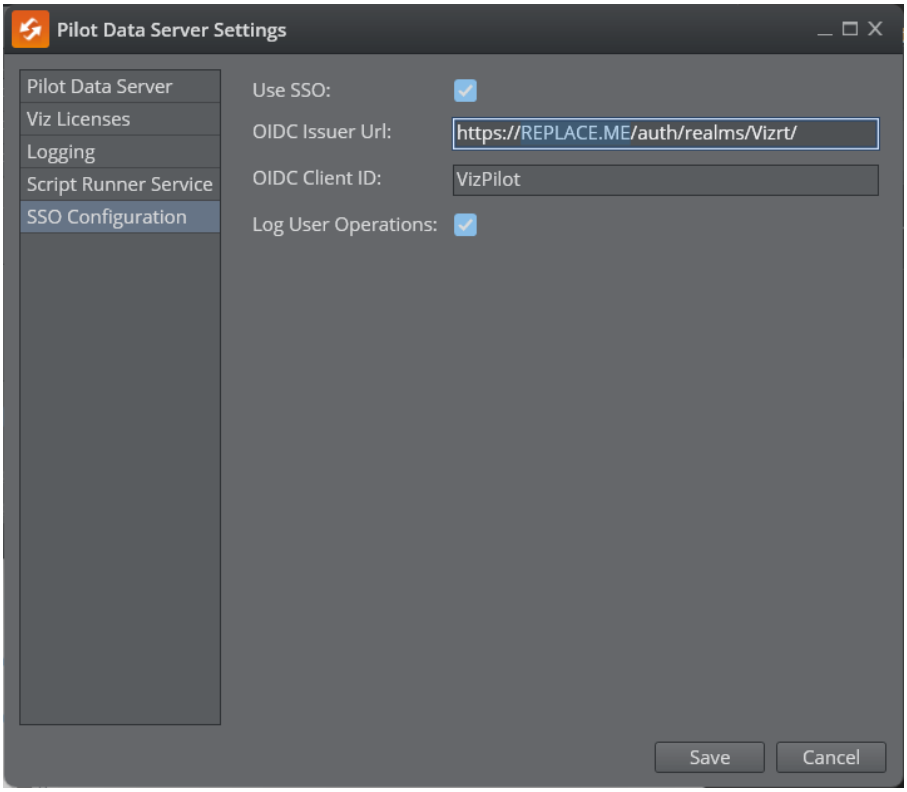

6. Update the Root URL field with the fully-qualified domain name of the Viz Pilot server. Also check and adjust the Valid Redirect URIs in case you have installed multiple versions of the web apps, or are using a reverse proxy with different path mapping or similar.




7. Manage users and groups to assign the four Viz Pilot roles as desired. The roles are defined by the **VizPilot** client, as shown in the screenshot below. The imported config includes example groups which you can use as-is, customize, or replace with your own groups. Remember that these groups don't directly grant any privileges, they just assign the corresponding roles to all group members, and it is the roles that grant a user privileges. The **graphic-designer** and **pilot-editor** roles also implicitly grant the **pilot-journalist** role. The **pilot-admin** role implicitly grants all the other Pilot roles.




8. Use the Pilot Data Server Launcher to configure OIDC settings for the target server. The **Issuer Url** is typically the Viz One server's base url followed by `/auth/realms/Vizrt`. The Client ID should be **VizPilot** when using the imported config.




### 4.5.4 Newsroom Requirements

The Newsroom system hosts MOS plugins like Viz Pilot Edge in their own web browser environment. This environment must be considered secure by the authentication technology we are using in Viz Pilot Edge.

The following requirements must be met:

1. The newsroom web hosting environment for MOS plugins must communicate over secure socket (using the **HTTPS protocol**). This must be configured in the newsroom system.
2. The web browser hosting the MOS plugin guest must support **popup windows** to allow the login dialog to be displayed.
3. The web browser must support the **SubtleCrypto** web API as specified here: https://developer.mozilla.org/en-US/docs/Web/API/SubtleCrypto. All modern browsers support this API, but it may be that the newsroom system still deploys an older browser internally, which can be incompatible with modern authentication technology.

#### Known Limitations

The web browser in **Avid iNEWS** does not support popup windows. To use authentication in iNEWS, the ShowEdge wrapper must be used to host Viz Pilot Edge.

# 4.6 Script Runner



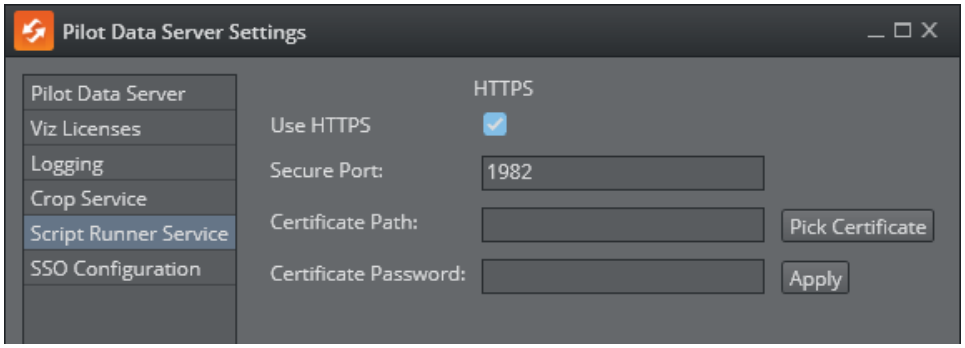
## 4.6.1 Configuring Script Runner

**Script Runner** is Viz Pilot's update service that lets you update template data, such as stock values and player statistics, right before going on air.

> **Note:** You can write Visual Basic In Pilot scripts to modify the data, and the scripts run on the Script Runner. It is also possible to create your own external update service. For more information, see *External Update Service* in the Media Sequencer Manual bundled with the software.

### Use Secure Communication

Script Runner can be configured to use the SSL transport layer when communicating. Click **Settings** > **Script Runner Service** to enter the configuration UI.




To configure Script Runner for SSL communication:

1. Check **Use HTTPS**.
2. The default port suggested is 1982. This can be changed, but it is recommended to use this port.
3. Configure a certificate by either entering a path to the *.pfx file or picking a certificate from the file system.
4. Enter the certificate password.
5. Click **Apply** to register the certificate information.
6. Click **Save**.
7. Restart the Script Runner Service.

### Server Certificates and PFX Files

A server certificate, typically issued by a trusted Certificate Authority (CA), contains the server's public key and identifying information. Adding a certificate for Script Runner or Crop Service is done by importing a PFX file . PFX

(Personal Information Exchange) file, also known as PKCS#12. A PFX file consolidates the server certificate, its private key, and any intermediate and root certificates into a single, secure, password-protected file.

> **Note:** Viz Pilot and Media Sequencer with an Oracle backend, do not support Script Runner on HTTPS.

## Configure the HTTP Port

The Script Runner uses port 1981 by default for HTTP. To change the port:

1. On the Pilot Data Server machine, open the configuration file in a text editor: *%ProgramFiles%\vizrt\Data Server\PilotScriptRunnerHostService.exe.config.*
2. Change the port by modifying the following line:

```
<setting name="DefaultPort" serializeAs="String">
    <value>1981</value>
</setting>
```

3. Save the file.
4. Restart the **Vizrt Script Runner** service from the Windows Services window.

## 4.7 Licensing

Licenses in Pilot Data Server are managed by using WIBU Systems. This manages the use of PDS, Viz Pilot Edge, Template Builder and Viz Pilot Edge through the NLE plugins.

WIBU works through the CodeMeter application, therefore, it is required to be installed in the same machine as Pilot Data Server (it is not necessary to have CodeMeter installed in the same machine as Viz Pilot Edge, Template Builder or the graphic plugins).

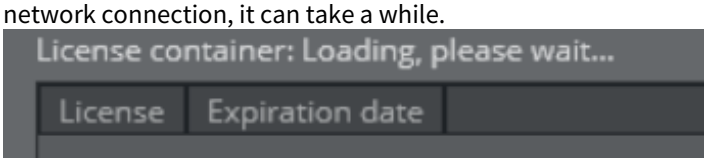
The CodeMeter local installation may have the WIBU containers set up or they can be connected to another CodeMeter license server. Please read the Viz Licensing documentation on how to set up CodeMeter containers.

> **Note:** The Codemeter.exe service of the Wibu Licensing system, must be running in order to start Pilot Data Server.

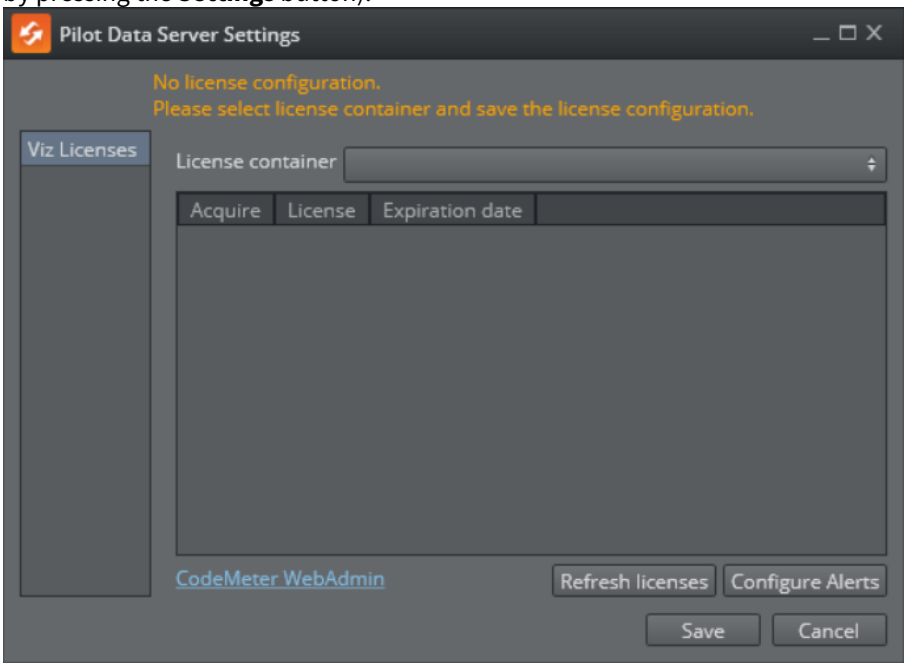


### 4.7.1 Setting Up Licenses

1. After starting Pilot Data Server launcher, it tries to connect to a valid CodeMeter server. Depending on the network connection, it can take a while.

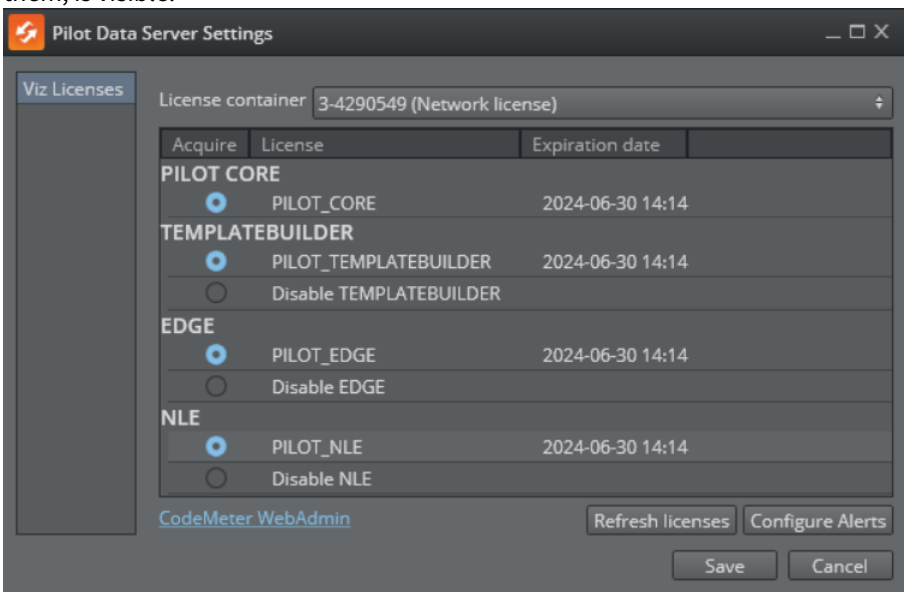



2. If there are no settings saved in the local machine, a new window is opened (this window can also be opened by pressing the **Settings** button).

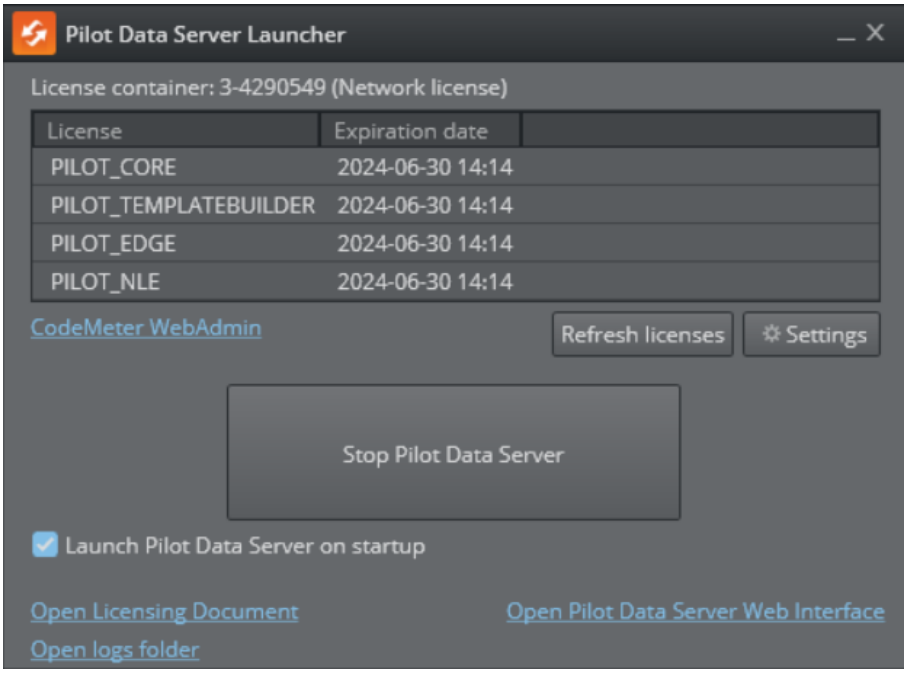


3. Select a container from the drop-down list:


   If no containers are available, press the link for CodeMeter WebAdmin at the bottom left of the window to open the CodeMeter administration page. After solving any issues on CodeMeter, you can press the **Refresh Licenses** button.
4. When a valid container is set, a list of licenses with the available licenses and the expiration date for each of them, is visible:




   The *PILOT_CORE* license is for PDS and it is mandatory. The remaining licenses can be enabled or disabled manually.
5. After selecting the licenses, click **Save** to go back to the initial window. This window shows the state of the selected licenses:



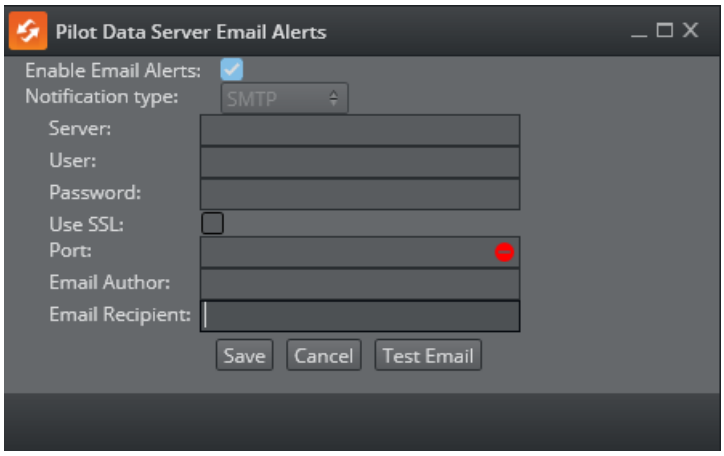

6. Click **Launch as Windows Service** to launch PDS. To start PDS at system startup, tick the **Launch on system startup** box.

## 4.7.2 Configuring Alerts

When PDS is running, alerts about license expiration can be configured by clicking on the **Configure Alerts** button from the MSE launcher. The PDS license alerter can be configured to either send email with SMTP, or to send journal messages via the Graphic Hub. There is a 7 days grace period on expiry of the PDS license.

### Configuring SMTP Alert

To configure email alerts about license expiration, select **SMTP Server** from the **Notification type** dropdown, and configure the SMTP server settings.



**Enable Email Alerts:** Option to use email alerts when license is expiring.
**Notification type:** Only SMTP email configuration is possible.
**Server:** SMTP server to use.
**User:** SMTP server user.
**Password:** SMTP server password.
**Use SSL:** Check if your server uses SSL, otherwise leave unchecked.
**Port:** SMTP server port.
**Email Author:** Email account which sends the email (usually user must have account permissions with this account).
**Email Recipients:** Email account that receives the alert email.

When finished, press the **Save** button.

## 4.8 Pilot Data Server Settings

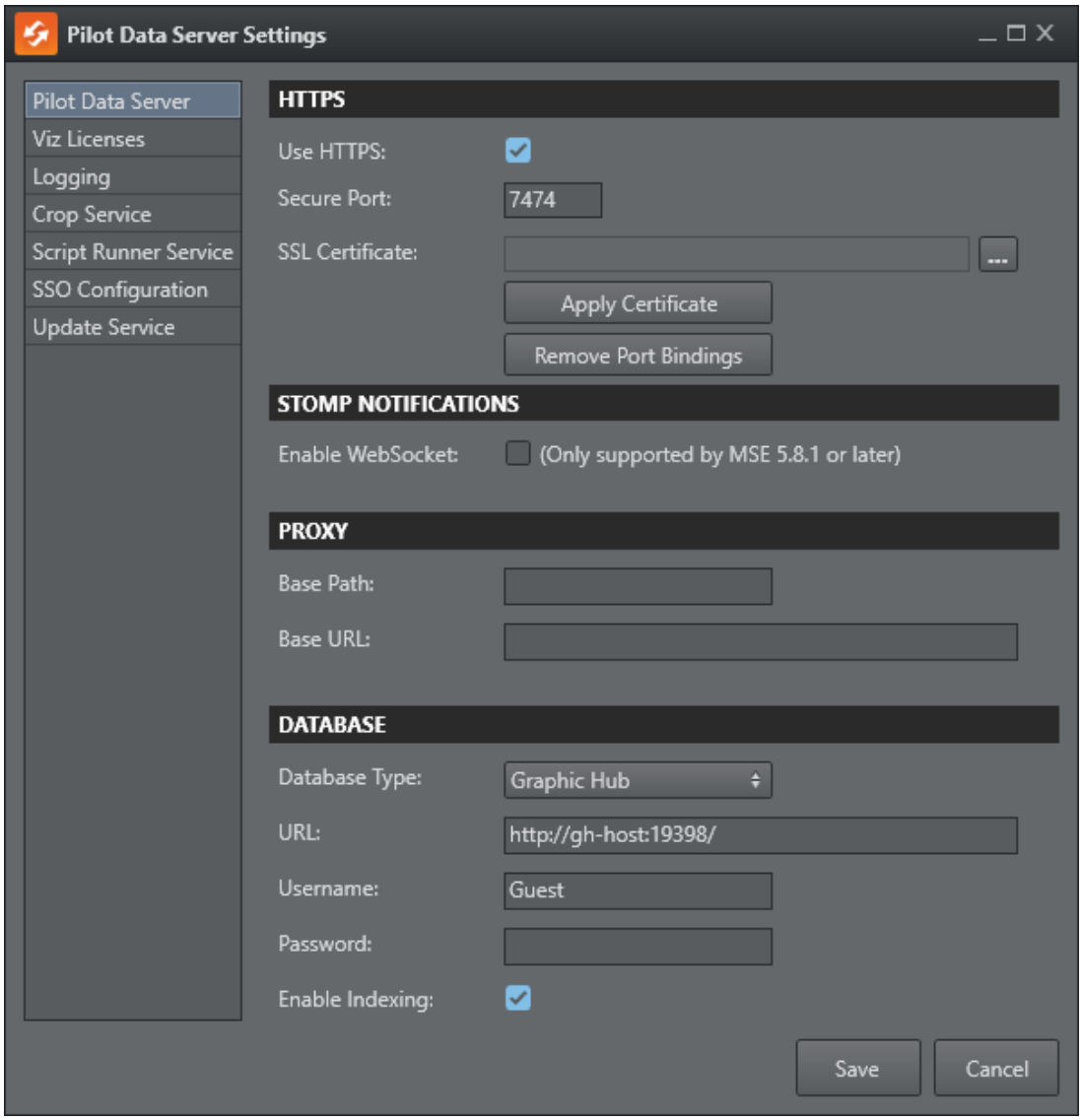


### 4.8.1 The Settings File

Most of the Pilot Data Server settings are stored in a single file: `%ProgramData%\Vizrt\Pilot Data Server\DataServerConfig.ini`.

This file backs most of the settings in the Launcher Settings. This file can be manipulated without the Launcher running, but the Pilot Data Server service needs to restart if the settings are changed in the file. This file can also be copied to a target machine to pre-configure Pilot Data Server without entering the UI.

## 4.8.2 The Settings UI




### HTTPS

Pilot Data Server can be configured to use the SSL transport layer when communicating. If HTTPS is configured, all endpoints of Pilot Data Server can be accessed using the HTTPS protocol.

1. Check **Use HTTPS**.
2. The suggested default port is 7373. Although it can be changed, it is recommended to use this port.

## Configuring SSL Certificates

To enable HTTPS for Pilot Data Server, you must configure an SSL certificate used by the server to secure communications. Select a certificate from the Windows Certificate Store, bind it to the configured port and application ID, and optionally, remove existing port bindings if necessary.

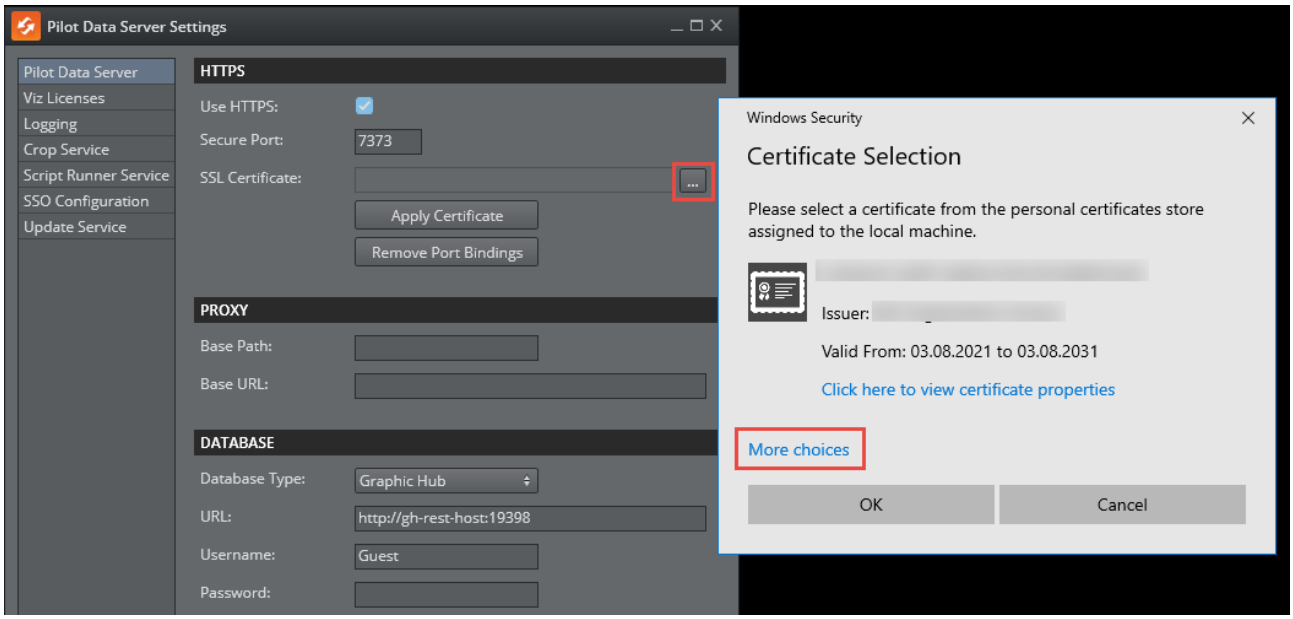
## Understanding SSL Certificates

An SSL certificate is a digital credential that enables a secure communication, by encrypting data transmitted between clients and the server. It verifies the server's identity to clients, ensuring trustworthiness. Certificates are stored in the Windows Certificate Store and can be accessed via specific logical stores and locations. Pilot Data Server uses the following:

- **Store Location**
  - `LocalMachine` : This location stores certificates available to all applications and services on the machine.
  - Certificates in this location typically require administrative privileges to access.
- **Store Name**
  - `My` : Also known as the **Personal** store, it contains certificates intended for server-level use, such as those for HTTPS.

## Configuring the Certificate for Pilot Data Server

To configure a certificate for Pilot Data Server, follow these steps:




1. **Browse for a Certificate**: Click the ... button next to the **SSL Certificate** text box to browse for a certificate.
2. **Select a Certificate**:

- The Windows Certificate Store browser opens, defaulting to `My` store under the `LocalMachine` location.
- It may be necessary to click **More choices** in the dialog box to expand the list of certificates in the store.
- Select a certificate that matches your server's domain or hostname. Ensure the certificate:
  - Is valid (not expired).
  - Contains the private key.
  - Is issued by a trusted Certificate Authority (CA), or is a custom certificate, such as a self-signed certificate, that you trust and intend to use.
- Click **OK** to confirm your selection.

3. **Apply the Certificate**:
   - Once a certificate is selected, its “friendly name” appears in the text box. Hovering over the “friendly name” of the certificate shows more comprehensive info on the selected certificate.
   - Click **Apply Certificate**. This action binds the certificate to the configured port and application ID using the `netsh` command. The command associates the certificate’s thumbprint with the application and ensures HTTPS is enabled.
4. **Remove Existing Port Bindings (Optional)**:
   - If you need to remove existing certificate bindings for the configured port and application ID, click **Remove Port Bindings**. This unregisters any current SSL certificate bindings associated with the application. This operation only locates bindings done by the PDS launcher. Any manual bindings previously done are not detected. See note below.

⚠ **Notes:**

- Search providers and Preview Server must also be in HTTPS environments in order for calls to them to work. To enable support for HTTPS requests for Preview Server, see the Preview Server Administrator Guide.
- If necessary, to manually clear the HTTPS port configured using the Pilot Data Server Launcher (Pilot Data Server, Script Runner, Crop Server), clear from the sslcert and urlacl lists with the following commands:
  - netsh http delete  sslcert ipport=0.0.0.0:7373
  - netsh http delete urlacl https://+:7373/

### STOMP Notifications

- **Enable WebSocket:** This setting enables STOMP notifications over the WebSocket protocol. See below.

### Proxy

Pilot Data Server can be installed along with a proxy/reverse-proxy server. See Proxy Support for more information.

- **Base Path:** This setting modifies the internal URL only when no proxy server is configured. It appends a path segment to the internal URL, defining the endpoint where Pilot Data Server listens for requests. For example, if the internal Pilot Data Server URL is `http://mypds:8177`, setting "pds" as the base path, changes the internal URL to `http://mypds:8177/pds`.

- **Base URL:** This setting rewrites the internal URL to an external-facing URL in every response, making it accessible through the proxy. This ensures clients interact with a consistent URL matching the proxy's external address. The base URL must be provided as an absolute URL string. For example, setting the base URL to `https://example.com/pds` causes the server to rewrite response URLs to match this external address.

> ⚠ **Note:** These two settings cannot be combined. Base URL takes precedence over Base Path.

### Database Connection

Pilot Data Server can connect to either an Oracle or a Graphic Hub database. Select your preferred choice in the dropdown.

### Graphic Hub

- **URL:** Full URL to the Graphic Hub REST service.
- **Username:** Username when Pilot Data Server connects to Graphic Hub.
- **Password:** Password when Pilot Data Server connects to Graphic Hub.

### Oracle

- **Data Source:** A connection string with format *hostname/servicename (SID )* to the Oracle database, for example *localhost/vizrtdb.*
- **Username:** Username when Pilot Data Server connects to the Oracle Database.
- **Password:** Password when Pilot Data Server connects to the Oracle Database.

### Enable Indexing

When enabled, Pilot Data Server creates a local indexing file under:

```
%PROGRAMDATA%\Vizrt\Pilot Data Server\[GUID]\
```

The `[GUID]` is a hash generated from the connection parameters (such as hostname, port and username), used to connect to Graphic Hub or Oracle. As a result, each unique database configuration has its own folder with a corresponding index folder.

The indexing file maintains a local mirror of data element names, template names, and concept names. With indexing enabled, searches across concepts for templates and data elements are significantly faster. However, Pilot Data Server consumes more resources, as it must actively query the database to keep the index in sync.

If indexing is disabled, all client search queries are forwarded directly to the database backend, resulting in slower performance, especially for cross-concept searches.

## 4.8.3 Configuring the Pilot Data Server HTTP Port

The default Pilot Data Server port is 8177. To change the port:

1. On the Pilot Data Server machine, open the ini file in a text editor: *%ProgramData\Pilot Data Server\DataServerConfig.ini*.
2. Replace the following setting: *Port = 8177*.

### 4.8.4 STOMP over TCP

Pilot Data Server sends change notifications to clients through the STOMP protocol (the default STOMP TCP port is 9876). This is used to form a URL to the STOMP server, sending change notifications to Media Sequencer. This URL is served out as a link in the `/changelog` endpoint of Pilot Data Server.

> ⚠ **Note:** This STOMP endpoint cannot be placed behind a HTTP proxy. The host name of the STOMP server must be accessible over TCP from the Media Sequencer host.

On the Pilot Data Server machine, open the ini file in a text editor: *%ProgramData\Pilot Data Server\DataServerConfig.ini*.

The STOMP address consists of the machine's fully qualified domain name (FQDN) and the STOMP port.

#### Changing the STOMP TCP Address

The STOMP TCP server uses the machine's FQDN as its base address (the default port is 9876). The port can be changed by setting `ChangeServerPort` in the ini file:

```
ChangeServerPort = 9988
```

The resulting STOMP URL is *stomp://hostname.domain.com:9988/*.

If the auto-detected FQDN is incorrect or unreachable from client machines (for example, because the machine picks up the wrong DNS suffix), the hostname used in the STOMP URL can be overridden by setting `ChangeServerReportAddress` to the correct hostname.

```
ChangeServerReportAddress = pdshost.server.example.com
```

The resulting STOMP URL becomes *stomp://pdshost.server.example.com:9988/*. If `ChangeServerReportAddress` is left empty (default), the auto-detected FQDN is used.

#### URLs Embedded in STOMP Notifications

The base URL used inside STOMP change messages (for links included in the data of the notifications) is determined separately from the STOMP connection address. It is controlled by the `BaseUrl` setting:

```
BaseUrl = https://baseurl.domain.com
```

If `BaseUrl` is not set, the server constructs a base URL from the configured protocol (HTTP/HTTPS), port, base path, and the machine's FQDN.

**Note:** `ChangeServerReportAddress` does not affect the URLs embedded inside STOMP messages. It only controls the hostname that clients use to connect to the STOMP service.

### 4.8.5 STOMP over WebSocket

Pilot Data Server also provides STOMP over WebSocket for clients that prefer WebSocket transport.

- The WebSocket STOMP endpoint URL is derived from the server's HTTP base URL (not from the STOMP TCP hostname).
- It uses the configured `BaseUrl` (and `BasePath`) to build the endpoint.
  - If `BaseUrl` is set, WebSocket STOMP uses it as the base. Otherwise, the server constructs from protocol (HTTP/HTTPS), port, `BasePath`, and the machine's FQDN.
- `ChangeServerReportAddress` does not affect the WebSocket STOMP endpoint. It only overrides the hostname used in the TCP STOMP URL embedded in the changelog feed.

**Info:** The WebSocket STOMP endpoint can run behind a reverse proxy, provided the proxy supports the WebSocket upgrade (HTTP 101 Switching Protocols). Ensure the proxy forwards the path/stomp, and preserves query parameters.

## 4.9 Log Files

Script Runner, Crop Service and Pilot Data Server services logs to different files.

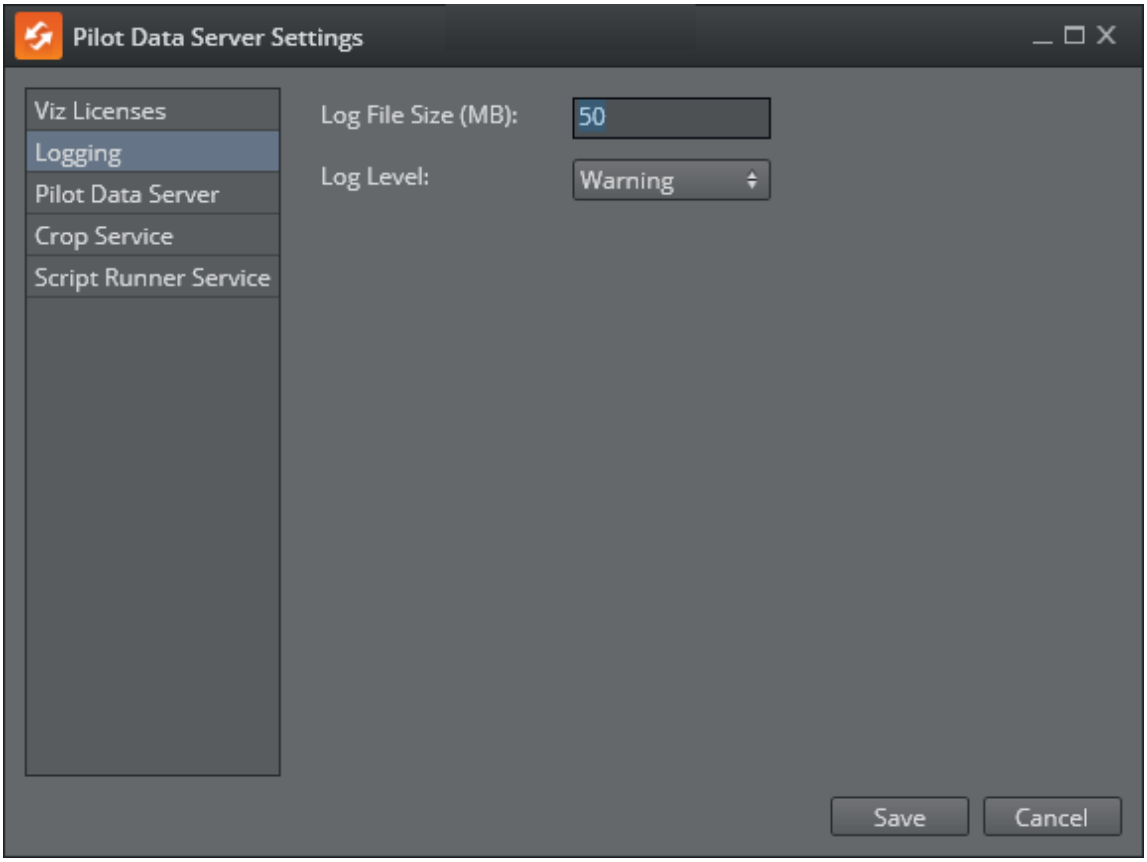
| Service | Log file path | Configurable |
|---|---|---|
| Pilot Data Server | %ProgramData%\Vizrt\Logs\PilotDataServer\server.log | Log file size and log level can be configured in the Launcher UI. |
| Crop Service | %ProgramData%\Vizrt\Logs\CropService\cropservice.log | Log file size and log level can be configured in the Launcer UI if CropServer is installed on the same host as Pilot Data Server. The settings are written to *cropservice.ini*. |
| Script Runner | %ProgramData%\Vizrt\Logs\ScriptRunner\scriptrunner.log | Log file size and log level can be configured in *PilotScriptRunnerHostService.exe.config* in the installation path. |
| Thumbnail Generator | %ProgramData%\Vizrt\Logs\Viz Pilot Thumbnail Generator | Log file size and log level can be configured in *PilotThumbnailGeneratorHostService.exe.config* in the installation path. |

**Note:** The maximum default size of the log is 1000Mb, with one active file and one *.old* file.

### 4.9.1 Logging Configuration

The configuration of Logging for Pilot Data Server is done using the Pilot Data Server Launcher.

1. On the initial window of the launcher, click the **Settings** button.
2. Select **Logging Config** to show the configuration settings.




**Log File Size:** Defines the maximum file size for log files (in mb) for PDS. When the size is exceeded, a new file is created in the same folder.

**Log Level:**

a. **Debug:** Log message for debug purposes.
b. **Info:** Log for general messages, for example "Connected to host", etc.
c. **Warning:** Log message for irregular situations that may cause unexpected results.
d. **Error:** Used for serious and unexpected errors, that may cause situations with unwanted results.

3. After configuring, click **Save**.

**Warning:** Clicking **Save** applies changes to all settings in this window, including logging, licensing, PDS, etc.

**Info:** Each log level, logs all of the messages of the level below. Therefore, debug logs all of the messages while the error level only logs errors.

## 4.10 Crop Service

**Vizrt Crop Service** is a Windows service that enables on-the-fly image cropping, and is delivered as a separate installer. If installed on the same machine as Pilot Data Server, configuration is performed through the **Launcher UI**.

All runtime configuration is stored in `%ProgramData%\Vizrt\CropService\cropservice.ini`.

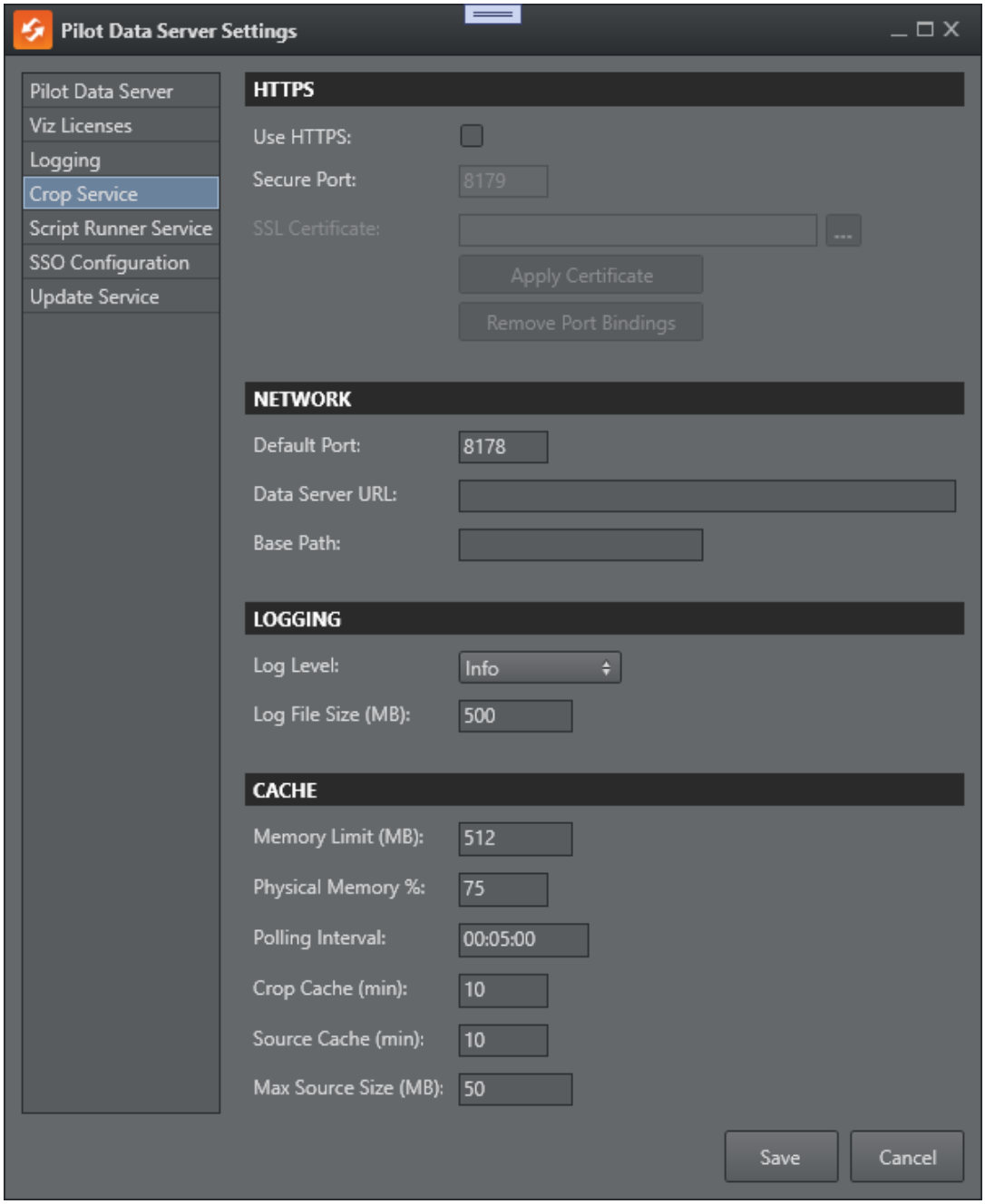
> **Note:** The Crop Service no longer reads port settings from exe.config files. Only **cropservice.ini** is used for runtime configuration.

This page contains the following sections:



### 4.10.1 Configuration via Launcher UI

The Launcher UI is the authoritative way to configure the Crop Service. Each field maps directly to a key in the INI file.




## HTTPS

Used to configure secure communication and certificate binding.

- **Use HTTPS:** Check this option to enable HTTPS.

```
UseHttps=True | False
```

  - If `True` service listens on `SecurePort`.

    - If `False` service listens on `DefaultPort`.
- **Secure Port:** The default port is **8179**.

  ```
  SecurePort=<port>
  ```

  It is recommended to:
    - Keep 8179 unless your environment requires a different port.
    - Ensure the port is open in host and network firewalls.
- **SSL Certificate:** Used to select a certificate from the Windows certificate store.
    - Click **Apply Certificate** to bind the certificate to the HTTPS port.
    - Click **Remove Port Bindings** to remove existing HTTP.SYS bindings.

> **Info:** The Launcher manages HTTP.SYS bindings. Do not use manual `netsh` commands unless explicitly required.

### Network

- **Default Port** (HTTP only)**:** The default port is **8178**. Only used when `UseHttps=False`.

  ```
  DefaultPort=<port>
  ```
- **Data Server URL:** Set the URL to the Pilot Data Server service document (this must be an absolute URL). The Crop Server retrieves the Pilot Data Server settings to extract credentials to pre-authenticated hosts, and reads the *croptool_max_image_area* setting from the database.

  ```
  PilotDataServerUri=<absolute-url>
  ```
- **Base Path** (Reverse Proxy Support)**:** This setting is only used when the Crop Service is published under a sub-path behind a reverse proxy.

  ```
  BasePath=<path>
  ```

  For example, the public URL is https://proxy.company.com/vizrt/cropservice. Therefore, the base path value can be set to *vizrt/cropservice*. In most modern deployments, `BasePath` should NOT be needed.
  If your reverse proxy forwards standard headers and preserves the full request path, the Crop Service can derive the correct public URL from headers.

### Logging

Controls log verbosity and file size.

- **Log Level**

```
LogLevel=<level>
```

  - Debug
  - Info
  - Warning
  - Error
- **Log File Size (MB):** Maximum size before log cycling.

```
LogFileSize=<megabytes>
```

## Cache

These settings control memory usage and caching behavior.

- **Memory Limit (MB):** Total shared memory budget across crop result cache, and source image cache. The default is 512.

```
[Cache]
CacheMemoryLimitMegabytes=<mb>
```

- **Physical Memory %:** Hard upper cap based on system RAM. The default is 75.

```
[Cache]
PhysicalMemoryLimitPercentage=<0-100>
```

  This prevents the service from exhausting system memory even if **Memory Limit** is set high.
- **Polling Interval:** Frequency of cache resource checks (hh:mm:ss). The default is 00:05:00.

```
PollingInterval=00:05:00
```

- **Crop Cache (min):** Retention time for processed crop results. The default is 10 minutes.

```
CropResultCacheMinutes=<minutes>
```

- **Source Cache (min):** Retention time for downloaded source images. The default is 10 minutes.

```
SourceImageCacheMinutes=<minutes>
```

- **Max Source Size (MB):** Do not cache source images larger than this threshold. The default is 50.

```
MaxSourceImageCacheSizeMegabytes=<mb>
```

### 4.10.2 Pilot Database Configuration

In the Pilot Database, configure these settings as follows.

- **crop_service_uri:** Must point to the externally reachable Crop Service endpoint. For example:
  - **HTTP:** `http://hostname:8178/`
  - **HTTPS:** `https://hostname:8179/`

> **Info:** This must match:
> - Protocol (HTTP/HTTPS)
> - Public hostname
> - Public port

- **croptool_max_image_area** (optional)**:** Defines the maximum served image area.

  ```
  width × height (pixels)
  ```

  Images larger than this are scaled down proportionally, even if no crop is applied.

### 4.10.3 Default INI Values

When created initially:

```
[Logging]
LogLevel=Info
LogFileSize=500

[Network]
UseHttps=False
DefaultPort=8178
SecurePort=8179
BasePath=
PilotDataServerUri=

[Cache]
CacheMemoryLimitMegabytes=512
PhysicalMemoryLimitPercentage=75
PollingInterval=00:05:00
CropResultCacheMinutes=10
SourceImageCacheMinutes=10
MaxSourceImageCacheSizeMegabytes=50
```

### 4.10.4 Operational Notes

- Restart the Crop Service after changing settings.

- Ensure configured ports are open in firewalls.
- For HTTPS:
  - Certificate Subject or SAN must match the public hostname.
  - Certificate must be trusted by clients.
- Prefer reverse proxy header forwarding over manual BasePath configuration.
- Do not edit legacy EXE `.config` files.

### 4.10.5 Health endpoint

The Crop Service exposes a lightweight health and diagnostics endpoint that reports basic runtime information, cache statistics, and key configuration paths. It supports content negotiation and returns JSON when requested via the Accept header. Otherwise, it responds with human-readable plain text.

```
Route: /health
```

The purpose is to:
• Verify the service is up and responding.
• Inspect basic process metrics (memory, handles, uptime).
• Inspect cache item counts.
• Discover important configuration paths (for example, INI and log file).

**Example (JSON)**

```
GET /health
Accept: application/json
```

Returns:

```
{
    "cache": {
        "cropResultCacheEntries": 0,
        "sourceImageCacheEntries": 0
    },
    "configFilePath": "C:\\ProgramData\\Vizrt\\CropService\\cropservice.ini",
    "logFilePath": "C:\\ProgramData\\Vizrt\\Vizrt Logging\\CropService\
\cropservice.log",
    "process": {
        "handleCount": 898,
        "memoryWorkingSet": "107 MB",
        "privateMemoryBytes": "74 MB",
        "uptime": "0.00:02:41"
    },
    "timestamp": "2026-02-13T13:33:55.1973069Z",
    "version": "9.5.0.0022"
}
```

**Example (plain text)**

```
GET /health
```

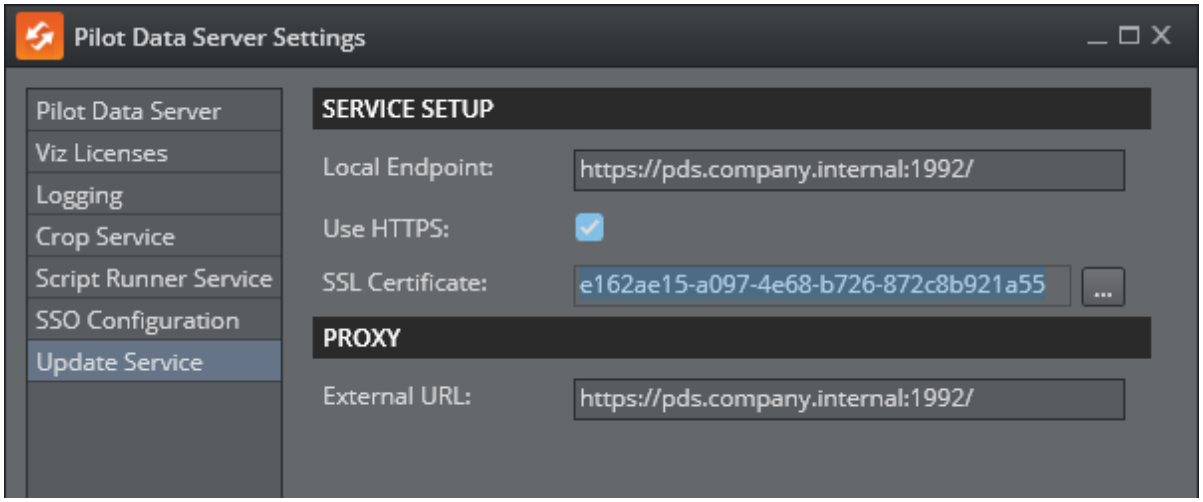
Returns:

```
Timestamp: 2026-02-13T13:35:01.6806058Z
Log file: C:\ProgramData\Vizrt\Vizrt Logging\CropService\cropservice.log

Crop Service version: 9.5.0.0022

Process info:
PrivateMemoryBytes: 74 MB
MemoryWorkingSet: 107 MB
HandleCount: 953
Uptime: 0.00:03:48

Cache runtime stats:
Crop result cache entries: 0
Source image cache entries: 0

Configuration:
Product version: 0.0.0.11569
Config file: C:\ProgramData\Vizrt\CropService\cropservice.ini

[Logging]
LogLevel: Info
LogFileSize: 500 MB

[Network]
UseHttps: False
DefaultPort: 8178
SecurePort: 8179
BasePath: (none)
PilotDataServerUri: (none)

[Cache]
CacheMemoryLimitMegabytes: 511
PhysicalMemoryLimitPercentage: 75
PollingInterval: 00:05:00
CropResultCacheMinutes: 10
SourceImageCacheMinutes: 10
MaxSourceImageCacheSizeMegabytes: 50
```

## 4.11 Pilot Update Service

Pilot Update Service is compatible with scripting in Viz Pilot Edge templates. It runs as a service on the same host as Pilot Data Server, and accepts requests to run the internal template's JavaScript in the **onUpdate** method, or auto-update the data through feed bound fields for the given data element. The service is usually called from Media Sequencer when the data element is being played out in the Viz Engine. The following topics are covered:

- Configuring Update Service
- Configuring Certificate Authorities (CAs)
    - Configuring a Certificate for the Windows Service
    - Configuring a Custom CA for Outgoing Connections
- Logging




### 4.11.1 Configuring Update Service

The configuration of Pilot Update Service, is written to and retrieved from *appsettings.json* in the installation folder, for example `%ProgramFiles%\Vizrt\Pilot Data Server\UpdateService`.

The default ports for the service are **1991** for HTTP and **1992** for HTTPS.

- **Local Endpoint:** URL where the service is running and reachable on the localhost. Use the existing, suggested URL or type in a different URL. The port of this URL is used when the service is starting.
- **Use HTTPS:** Check this box if the service is expected to run in secure mode, communicating with SSL/TLS over the HTTPS protocol.
- **SSL Certificate:** Certificate chosen from the Windows Certificate Store, used by the server to allow clients to connect securely.
- **External URL:** If the service runs behind a proxy, this is the URL the service should be reachable from outside the proxy. Media Sequencer is typically the device that calls Update Service, therefore, this URL has to be reachable from the Media Sequencer host. The URL should be the same as the Local Endpoint if there is no proxy present.

## 4.11.2 Configuring Certificate Authorities (CAs)

The Pilot Update Service runs as a Windows service and wraps a Node.js server internally. While the Windows service can utilize certificates from the Windows Certificate Store to allow clients to connect securely, the Node.js component also needs certificates for its internal functionality. Specifically, the Node.js server connects to a Pilot Data Server to download the script and the VDF model, and the update script itself may fetch resources from servers on HTTPS, which requires properly configured certificate authorities (CAs) to validate those connections.

### Configuring a Certificate for the Windows Service

The **Pilot Update Service** allows secure client connections by binding a selected certificate from the Windows Certificate Store. To configure the certificate:

1. **Access the Certificate Selection Dialog:**
   - Click the button with the three dots ( `...` ) in the service configuration UI. This opens the Windows Certificate Store browser.
2. **Select a Certificate:**
   - In the certificate selection dialog, click **More choices** to view all registered certificates under the **Personal (My)** store.
   - Select the desired certificate from the list. Ensure the certificate is valid and matches the intended domain or service usage.

After selecting the certificate, it is automatically used by the service for HTTPS communication without further manual steps.

### Configuring a Custom CA for Outgoing Connections

Please follow these steps to configure the internal Node.js server to use custom CAs for its internal HTTPS communication:

> ⚠ **Note:** These steps are only required when using **custom Certificate Authorities (CAs)**. If the certificates are issued by an officially recognized CA, no additional configuration is necessary.

### Step 1: Export the Required Certificate(s) from the Windows Certificate Store

1. **Open Certificate Manager:**
   - Press `Win + R` , type `certmgr.msc` , and press `Enter` .
2. **Locate the Certificate:**
   - Navigate to the appropriate certificate store (for example, `Trusted Root Certification Authorities` or `Intermediate Certification Authorities` ).
3. **Export the Certificate:**
   - Right-click on the desired certificate and select **All Tasks > Export**.
   - In the Certificate Export Wizard:
     - Choose **Base-64 encoded X.509 (.CER)** as the export format.
     - Save the exported file to a known location (for example, `C:\certificates\mycert.cer` ).
4. **Repeat for Additional Certificates** (if necessary):

- If multiple certificates are required, export each one individually.

### Step 2: Combine Certificates (Optional)

- If you need to provide multiple certificates, combine them into a single PEM file:
  - Open each exported `.cer` file in a text editor (for example, Notepad).
  - Copy and paste the content of each file into a new file, ensuring they are in the correct order:

```
-----BEGIN CERTIFICATE-----
<Certificate 1 Content>
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
<Certificate 2 Content>
-----END CERTIFICATE-----
```

  - Save the combined file as a `.pem` file (for example, `C:\certificates\custom-ca.pem`).

### Step 3: Configure Node.js to Use the Custom CAs

1. **Set the Environment Variable:**
   - Define the `NODE_EXTRA_CA_CERTS` environment variable and point it to the PEM file:
     - Open the **System Properties** dialog:
       - Press `Win + Pause/Break` > **Advanced system settings** > **Environment Variables**.
     - Under **System variables**, click **New**.
     - Set the following:
       - **Variable name**: `NODE_EXTRA_CA_CERTS`
       - **Variable value**: Path to the PEM file (for example, `C:\certificates\custom-ca.pem`).
2. **Restart the Pilot Update Service:**
   - Restart the service to apply the changes.

> ⚠ **Note:**
> - The Windows service does not require additional configuration for CAs if certificates are already available in the Windows Certificate Store for client connections.
> - The Node.js server requires the custom CA configuration described above, to connect to other HTTPS resources securely.
> - When combining certificates, maintain the order: root certificates should appear last, preceded by intermediate and leaf certificates.
>
> By following these steps, the Pilot Update Service handles both secure client connections via the Windows service and internal HTTPS communication within the Node.js server.

### 4.11.3 Logging

Pilot Update Service uses Serilog, a robust and flexible logging library, to capture and manage application logs. Serilog supports structured logging and can output logs to various sinks, such as files and the console, making it easier to monitor and troubleshoot the service. The system applies the following Serilog settings:

1. **Outputs:**
   - Logs are written to both the console and a file located at `%ProgramData%\Vizrt\logs\Pilot Update Service\service<date>.log`.
2. **File Retention and Size Management:**
   - A maximum of **31 log files** are retained. When the limit is exceeded, the oldest file is deleted.
   - Each log file has a size cap of **1GB**. If this limit is reached, logging to the file stops. This safeguard prevents excessive file creation, particularly in scenarios such as infinite loops, ensuring key log data remains intact.
3. **Log Levels:**
   - The default log level is **Information**.
   - For non-lifetime, Microsoft-based events, the log level is set to **Warning** to reduce noise from routine request processing.

## 4.12 Audit Logging

Pilot Data Server includes an audit feature (available to customers) that records write activity, providing insight into how the system is used over time. The audit is intended for operational overview and transparency.



### 4.12.1 What is Audited

The audit records **write operations** to Pilot Data Server:

- HTTP POST
- HTTP PUT

Typical examples include creating or updating templates and elements.

Read operations (GET) are not audited.

### 4.12.2 What Information is Recorded

Each audit entry represents a single write operation, and includes metadata such as:

- Time of the operation.
- Type of operation (POST / PUT).
- User identity when single sign-on (SSO) is enabled.
- Client IP address when SSO is not enabled.

Template payloads and full data content are not recorded.

### 4.12.3 How Audit Data is Stored

Audit data is stored locally on the machine running Pilot Data Server, in an internal database file.

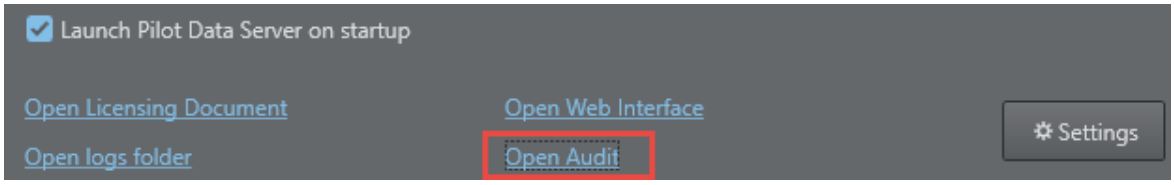
Key characteristics:

- Audit data is stored locally and not exposed through any API.
- Audit data is not transmitted outside the PDS host.
- Access to audit data requires operating system–level access to the server.

The audit log uses an internal integrity mechanism that manually modifies detectable entries.

### 4.12.4 Viewing Audit Data

An Audit Viewer is available from the Pilot Data Server Launcher.




The Audit Viewer can be used to:

- Review audit activity for a selected time period.
- See the number of unique users performing write operations.
- See the total number of write interactions.
- Copy audit summary to clipboard.
- Delete existing audit data and restart audit logging from scratch.

Deleting audit data does not affect normal operation of Pilot Data Server.

### 4.12.5 Configuration

Audit logging is enabled by default, but it can be disabled by setting the following option in the Pilot Data Server configuration file:

```
DisableAudit=True
```

This setting disables all audit logging and is intended as an exception mechanism.

### 4.12.6 Data Protection

Audit entries may contain personal data (as defined in the Vizrt Data Processing Addendum), such as user identifiers or IP addresses, depending on the authentication setup and use of the audit feature. Customers act as controllers in respect to any personal data contained in audit entries, and are solely responsible for ensuring its compliance with Applicable Data Protection Law in connection with its use.

Customers acknowledge and accept that:

- Audit data is stored locally on the Pilot Data Server host, and is controlled by the customer.
- No automatic reporting or external transmission of audit data is performed, unless expressly agreed between Vizrt and the customer.
- Customers are responsible for retention control, access control, and deletion policies.

This feature is intended to provide usage insight and operational transparency, not employee monitoring or compliance enforcement.

# 5 Use Cases

A Viz Pilot system has the following Pilot Data Server use cases:

- Using the Timeline Editor. See the Timeline Editor section under Newsroom Integration in the Viz Pilot User Guide.
- Using the Update Service. See the Update Script Editor under Template Wizard in the Viz Pilot User Guide.
- Searching for and using images and videos from Viz One.
- Using the person search from the Media tab.
- Adding and updating Tag Settings. Tags are assigned to templates in Template Wizard, and then used in Viz Pilot News to organize templates.
- Using the Crop Service. See the Using Crop Service section under Crop Service and Crop Tool in the Viz Pilot User Guide.

  > **Note:** Crop Service requires a separate installer. See Crop Service Installation in the Installation section in the Viz Pilot User Guide.

- Using the Pilot Data Server's REST API to allow third-party systems to read and fill templates and data elements from the Viz Pilot database.
- Configuring Order Management. See the Order Management section in Newsroom Integration in the Viz Pilot User Guide.
- The Pilot Data Server exposes all the data needed by the Media Sequencer through its REST interface. It also provides Change Notifications using the STOMP protocol. It's possible to configure Media Sequencer to connect to the Pilot Data Server instead of the Pilot database (this requires Media Sequencer 4.0 or later).

# 6 Web Interface

The Pilot Data Server web interface offers access to features including the Service Document, Template and Data Element Feeds, Viz Pilot Settings and REST API documentation.

## 6.1 Accessing the Pilot Data Server Web Interface

- Select the **Pilot Data Server Web Interface** from the Windows Start Menu.

- Alternatively, start the **Pilot Data Server** in a browser, using the host name of the machine running the Pilot Data Server, and port `8177`. For example: *http://dataserver.example:8177*

---

## 6.2 Service

The Service Document resource (for example: *http://<dataserver.example>:8177/service*) is an Atom Service Document that can be used by a client to discover the capabilities of a Pilot Data Server and the locations of the available Atom Publishing Protocol collections hosted on it.

---

## 6.3 Template Feed

The Template Feed resource (for example: *http://<dataserver.example>:8177/templates*) is an atom feed that contains entries for each template stored in the Pilot Data Server. Metadata may include details of the template such as description, creation date, link to a thumbnail image and link to the Viz Data Format (VDF) model document describing the template.

---

## 6.4 Data Element Feed

The Data Element Feed resource (for example: *http://<dataserver.example>:8177/dataelements*) is an atom feed that contains entries for each data element stored in Pilot Data Server. Metadata may include details of the data element such as description, creation date, link to thumbnails and link to the Viz Data Format (VDF) payload document describing the data element.

---

## 6.5 Settings

The Settings page (for example: *http://<dataserver.example>:8177/settings*) is used to configure VCP Parameters and Tag Settings. See Settings Parameters for a full description of the Settings tab.

---

## 6.6 Examples

The Examples page (for example: *http://<dataserver.example>:8177/examples*) contains examples on how to interact with the Viz Pilot system.

---

## 6.7 Documentation

The Pilot Data Server Documentation page (for example: *http://<dataserver.example>:8177/help*) describes the REST API provided by Pilot Data Server. The page includes information on the Resource Types and Content Types that are used in the interface. The API itself lets you access template information, perform image search and person search, and provides a programing interface for other systems.

# 7 RestVOS

## 7.1 Comparing RestVOS and Conventional VOS Access

- The fundamental difference is in the protocol used to search for images and deliver the results to the client.
- When accessing an Object Store in the conventional Viz Object Store (VOS) way, the client application goes directly to the database and the network share in order to get the data.
- When using RestVOS, the client accesses the same images and storage, but via the Pilot Data Server. This uses the HTTP protocol, the same protocol used when talking to Viz One systems.
- From the point of view of client applications, accessing RestVOS is no different from accessing a Viz One or any other asset search provider.
- The existing Object Store (see the Object Store section in the Viz Pilot User Guide) application is used to manage the images and storage, whether using standard VOS or RestVOS.

## 7.2 Advantages of RestVOS

- **Flexibility:** HTTP is an easier protocol to use and manage. By using the Pilot Data Server and the OpenSearch protocol, other applications can also perform the same searches, and there is no need to make the network share available on all clients.
- **Extended features:** Certain features such as Crop Service and Filter Media by Person Name (see the Viz Pilot User Guide), are only available when using RestVOS.
- **Future oriented:** As development on RestVOS continues, features such as load balancing and failover capabilities are added, benefiting image transfers. Once direct database access from Viz Pilot News is no longer required, management of Oracle clients is not necessary. Conventional access to VOS is phased out at some stage.

# 8 Change Notifications

Change notifications are available through the REST API and the STOMP protocol on Pilot Data Server.

## 8.1 REST API

A list of database changes ordered from most recent to oldest:

- Located at http://pdshost:8177/changelog.
- Parameters:
  - **startId:** The ID of the least recent change, or -1 for no limit. Default is -1.
  - **lastKnownId:** The ID of the most recent change, or -1 for no limit. Default is -1.

## 8.2 STOMP API

In a Viz Pilot system, the text-based message Streaming Text Oriented Messaging Protocol (STOMP) is used to subscribe to change notifications from Graphic Hub REST and Pilot Data Server. STOMP provides push notifications of new changes.

- Located at: *stomp://pdshost:9876/?destination=/changelog&lastKnownId=-1.*
- Parameters:
  - **lastKnownId:** Mandatory, should be the ID in the STOMP link that is provided by the REST API. For example, when the data stored in the Pilot database changes, a message is delivered to all of the STOMP clients that are subscribed to the server.

**Note:** The default change notification port for the STOMP protocol is `9876`. Other services using this port may suppress the change notifications.

# 9 Proxy Support

When deploying Pilot Data Server behind a proxy server, it is often necessary to rewrite requests, to ensure the application interprets and responds to client requests correctly. This process typically involves adjusting elements of the request to account for modifications introduced by the proxy, such as altered hostnames, schemes, or paths.

There are two primary methods for aligning the application's URL with the external-facing URL used by the proxy.

## 9.1 Using X-Forwarded Headers

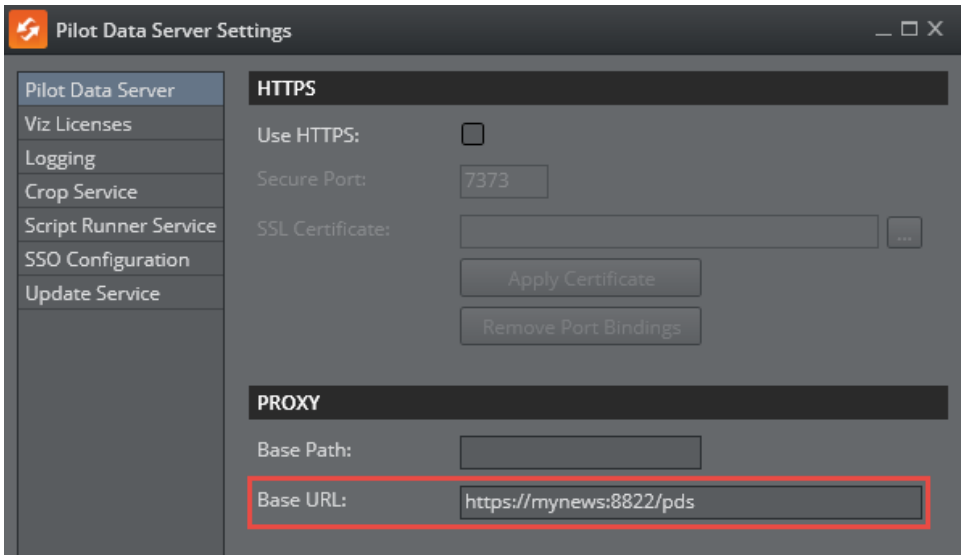
The proxy server must either populate the X-Forwarded headers or the Forwarded header in order for Pilot Data Server to serve the appropriate hostname, port and protocol in its responses:

- `Forwarded`
- `X-Forwarded-Host`
- `X-Forwarded-Port`
- `X-Forwarded-Proto`

Use a proxy server with the above headers configured. If the `Forwarded` or `X-Forwarded` headers are not configured, the proxy server can be configured to pass the `Host` header it received from the client to Pilot Data Server.

## 9.2 Configure Base URL

An alternative method is to explicitly configure the application's Base URL to match the external-facing URL provided by the proxy.




This cannot be combined with the forwarded-headers mentioned above.

> ⚠ **Note:** The **Base Path** cannot be combined with neither forwarded-headers nor the Base URL. The base path is only applied to the internal URL when no proxy is configured.

## 9.3 Redundancy and Failover

Although Pilot Data Server has been designed without built-in redundancy, it does support being used with off-the-shelf third-party HTTP load balancers. As the name suggests, a load balancer's main function is to distribute requests among multiple servers. Multiple servers provide redundancy, and most load balancers come with an option to only route requests to servers that respond, which provides failover.

Both Barracuda Load Balancer ADC and HAProxy have successfully been used in front of Pilot Data Server.

### 9.3.1 Load Balancer Quick Setup

Quickly set up a HAProxy load balancer:

#### Prerequisites

- Two (or more) hosts running Pilot Data Server, connected to the same database. They have host names *pds1.example* and *pds2.example* for the purpose of these instructions. Static IP addresses may be used instead.
- A Linux host that turns into a load balancer. It has the host name *proxy.example* for the purpose of these instructions. This is the host used in the installation steps below.

#### Installation Steps

1. On the Linux host, install HAProxy with the command `sudo apt-get install haproxy`.
2. Edit */etc/haproxy/haproxy.cfg* (for example with `sudo nano /etc/haproxy/haproxy.cfg`) and append the following lines:

```
frontend http-in
    bind :8177
    default_backend servers

backend servers
    server server1 pds1.example:8177 check
    server server2 pds2.example:8177 check

listen stats
    bind :80
    mode http
    stats enable
    stats uri /
```

3. Restart HAProxy with the command `sudo service haproxy reload`.

## Notes

- After following the steps above, it should be possible to use *proxy.example:8177* in place for *pds1.example:8177* and *pds2.example:8177*. Check whether the setup works by opening http://proxy.example:8177/ in a browser. The Pilot Data Server index page should load.
- The above configuration also provides a monitoring page at http://proxy.example/, that shows which Pilot Data Server hosts are responding. This does not tell you whether Pilot Data Server is working correctly, as hosts still respond when they cannot reach the database.

# 10 MOS XML

The MOS (Media Object Server) protocol is a communication standard specifically designed for newsroom computer systems (NRCS) in the broadcast industry. It enables interoperability and seamless integration between various software, devices and plugins used within a newsroom environment. It is important to note that while the MOS protocol provides a standard framework for communication, the specific implementation and feature set may vary between different vendors and systems.

From Viz Pilot Edge 3.0 and Pilot Data Server 9.0, our philosophy in Vizrt is to use these guidelines for how to implement certain parts of the MOS protocol:

1. We want to be as compliant as possible to the MOS protocol version 2.8.5.
2. Where the protocol specification is vague or inadequate, we try to stick to the MOS XML values of existing released Vizrt software.
3. We want to be as internally consistent as possible between Viz Pilot News, Viz Pilot Edge and Pilot Data Server.

The following MOS XML values can be used by newsroom vendors and other consumers as a guide:

**Info:** Please note that the value 50 below depends on the **video_mode** setting in Pilot Data Server, being 50 for PAL and 60 for NTSC.

The value for **<itemEdDur>** and **<itemEdStart>** are given in frames. **<objTB>** is given in frames per second and should be used to calculate the duration in minutes and seconds.

## 10.1 Graphics with No Timing Info

| | Viz Pilot News 8.9 plugin | Viz Pilot Edge 3.0 / PDS 9.0 |
|---|---|---|
| **<objType>** | PILOT | PILOT |
| **<objTB>** | 0 | 50 |
| **<objDur>** | 1 | 1 |

## 10.2 Graphics with Timing Info

| | Viz Pilot News 8.9 plugin | Viz Pilot Edge 3.0 / PDS 9.0 |
|---|---|---|
| **<objType>** | PILOT | PILOT |
| **<objTB>** | 50 | 50 |
| **<objDur>** | 1 | 1 |

| | Viz Pilot News 8.9 plugin | Viz Pilot Edge 3.0 / PDS 9.0 |
|---|---|---|
| **<itemEdDur>** | 250 | 250 |
| **<itemEdStart>** | 50 | 50 |

## 10.3 Videos

| | Viz Pilot News 8.9 plugin | Viz Pilot Edge 3.0 / PDS 9.0 |
|---|---|---|
| **<objType>** | PILOT | VIDEO |
| **<objTB>** | 50 | 50 |
| **<objDur>** | 298 | 298 |

## 10.4 Stillstore Images

| | Viz Pilot News 8.9 plugin | Viz Pilot Edge 3.0 / PDS 9.0 |
|---|---|---|
| **<objType>** | PILOT | STILL |
| **<objTB>** | 0 | 0 |
| **<objDur>** | 1 | 1 |

## 10.5 Overriding MOS XML Values

For backwards compatibility with existing newsroom system implementations, it is possible to override some of these values when served out from Pilot Data Server and using Viz Pilot Edge. This should only be done to overcome potential critical issues when upgrading Viz Pilot Edge or Pilot Data Server, and not be a permanent solution. These are the database parameters that can be overridden:

`mos_override_objtype`

`mos_override_objtb`

`mos_override_objdur`

See Database Parameters on how to add optional database parameters.

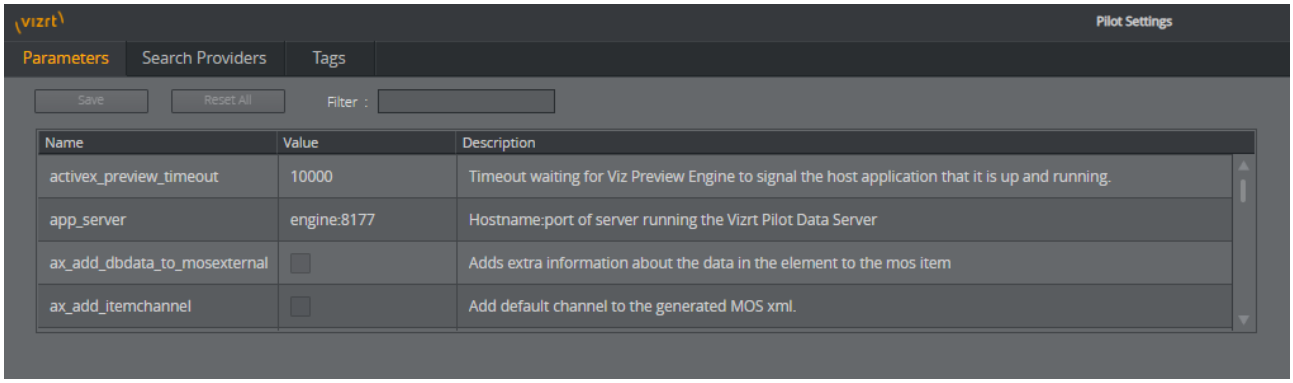
# 11 Setting Parameters

The Settings tab is used to configure the following values:

- Database Parameters
- Search Providers
- Tag Settings

## 11.1 Database Parameters

Configure Viz Pilot system application settings in the VCP Parameters tab. A description of useful database parameters is listed here.




### 11.1.1 Granting Applications Access to Pilot Data Server

1. See how to access the Pilot Data Server Web Interface.
2. Click the **Settings** link.
3. Select the app_server setting and add the parameter for the machine you installed Pilot Data Server on (for example, <DataServer>:8177).
4. Click **Save**.

This grants all applications with a connection to the database access to Pilot Data Server.

**IMPORTANT!** Firewalls must allow inbound communications on port `8177`.

### 11.1.2 Granting Applications Access to Preview Server

1. See how to access the Pilot Data Server Web Interface.
2. Click the **Settings** link.
3. Select the preview_server_uri setting and add the parameter for the machine you installed Preview Server on (for example, http://<hostname>:21098). This machine, with Viz Engine and Preview Server installed, is typically identified as your *frame server*.
4. Click **Save.**

This grants all applications with a connection to the database access to Preview Server.

### 11.1.3 Granting Applications Access to Update Service

**Note:** This setting is required if templates run scripts that use Update Service.

1. See how to access the Pilot Data Server Web Interface.
2. Click the **Settings** link.
3. Select the script_runner_uri setting, and add the parameter for the machine running the Update Service (e.g. http://<hostname>:1981).
4. Click **Save.**

This grants all applications with a connection to the database access to Script Runner.

**IMPORTANT!** If you change the *script_runner_uri* parameter (after setting it the first time), you need to restart Media Sequencer for the changes to take effect.

### 11.1.4 Setting Database Parameters for Crop Service

1. See how to access the Pilot Data Server Web Interface.
2. Click the **Settings** link.
3. Select the crop_service_uri setting and add the parameter for the machine you installed Pilot Data Server on: http://< **cropservicehostname** >: **8178**/.
4. *Optional*: Set the **MediaSearch_ItemPerPage** = Number of items per page to get from MediaSearch.
5. *Optional*: Set the **croptool_max_image_area** to adjust the maximum size of a cropped image that is served by Pilot Data Server. If the image size (*) is larger than croptool_max_image_area, then the image is resized, while still respecting the aspect ratio of the crop. The maximum image size applies even if no cropping is done.
   * Image size = image width x height in pixels.
6. Click **Save**.

### 11.1.5 Enabling Built-in Object Store Search

The built-in Object Store search is disabled by default. To enable or disable it:

1. See how to access the Pilot Data Server Web Interface.
2. Click the **Settings** link.
3. Select the disable_built_in_vos_search setting:
   - To use built-in Object Store search, set **disable_built_in_vos_search** to false.
   - To use RestVOS search, set **disable_built_in_vos_search** to true.
4. Click **Save**.

### 11.1.6 Configuring Order Management

This procedure assumes that you have a Viz One system installed and running the Order Management system, see the Order Management in the Newsroom Integration section in the Viz Pilot User Guide. For more information on how to configure order management on Viz One, see the Tasks and Order Management section in the Viz One Administrator Guide.

1. See how to access the Pilot Data Server Web Interface.
2. Click the **Settings** link.
3. Select the order_mgt_uri_template setting and add the address of the Viz One machine which is hosting the order management system, for example:
   http://<fully-qualified-domain-name-of-viz-one>:8084/tasks/?user={user}
   &locale=en_US&theme=light#create;tl=work_order;form=create_wo_activex;asset-type=ITEM;upload=true;external-hooks=viznewsroom;closewin=false;set-attr=REQ_TYPE-{asset_type}.

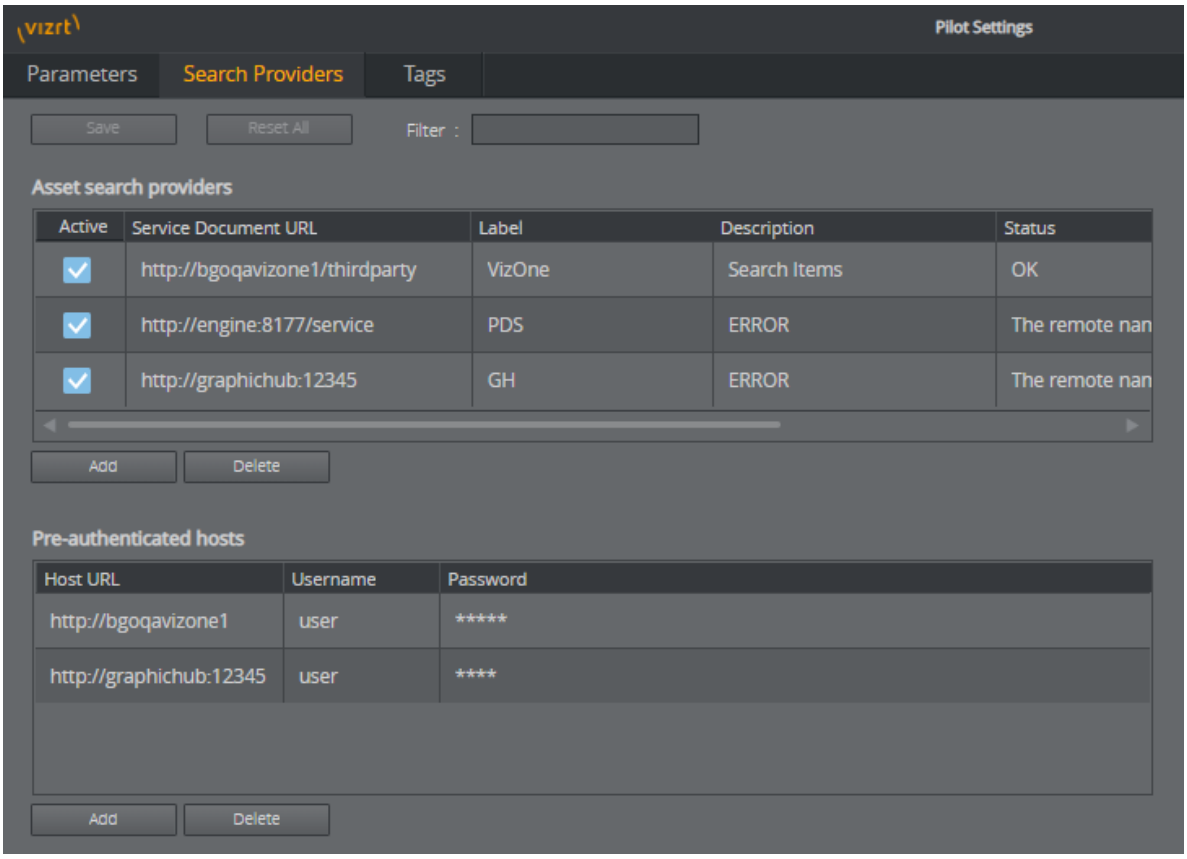
---

## 11.2 Search Providers

Both Director and Viz Pilot News get their Viz One connection parameters from the Viz Pilot database. Media Sequencer must be configured using your control application. Asset search providers can be set to active or inactive. Disabling a search provider means that it is not used when searching for assets in a template.

This section contains the following topics:

- Configuring Search Providers (Viz One)
- Configuring Search Providers (RestVOS)
- Authentication for Feed Browser



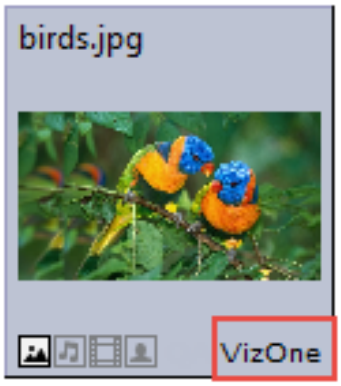

### 11.2.1 Configuring Search Providers (Viz One)

1. See how to access the Pilot Data Server Web Interface.
2. Click the **Settings** link.

3. Click the **Search Providers** link.
4. Under **Asset search providers** click the **Add** button and enter the host's service document **URL**: /war
    - For Viz One 5.4 and later: https://<viz_one>/thirdparty/.
    - For Viz One 5.3: https://<viz_one>/api/.
      Make sure to enter the correct protocol (for example, a Viz One set up on HTTPS requires an **HTTPS** URL).
5. Add a **Short name** such as *VizOne* to help identify the asset's location, for example in search results.

> **Note:** If you change the short name, you need to restart the server for the changes to take effect (see To modify the Pilot Data Server database settings).

The screenshot below shows the Search Provider Short Name as displayed in search results:




6. Under Pre-authenticated hosts click the **Add** button and enter the host's URL, username and password in order to authenticate your search:
    - URL of the host, for example: https://<viz_one>/.
    - Username and password of the pre-configured user on the Viz One system (configured in the Studio admin web interface).
    - This is required for Viz One (not Object Store).

    > **WARNING!** The username and password used here are available in clear text to anyone who has access to the Settings page. The Viz One user entered here should therefore be given as few rights on the Viz One as possible.
7. Click **Save**.

> **Note:** That **Crop Server** also needs a restart after configuring pre-authenticated hosts, as Crop Server uses the settings from Pilot Data Server.

## 11.2.2 Configuring Search Providers (RestVOS)

By default, Viz Pilot 6.0 and later uses the RestVOS search, and a search provider representing the current Pilot Data Server is configured automatically. This requires that the correct value (a public name that points to the current Pilot Data Server) has been stored in the app_server parameter in the database parameters. Changes to this value are picked up during restarts.

The auto-added search provider can be renamed or made active/inactive on the **Search Providers** page. The URI cannot be changed.

> **Note:** Upgrading from Viz Pilot 5.7 to Viz Pilot 6.0 can result in two search providers with the same URL (both the existing and the new provider). Although the new provider is disabled and therefore does not cause an issue, you can delete the old and activate the new provider if you wish to clean up the list.

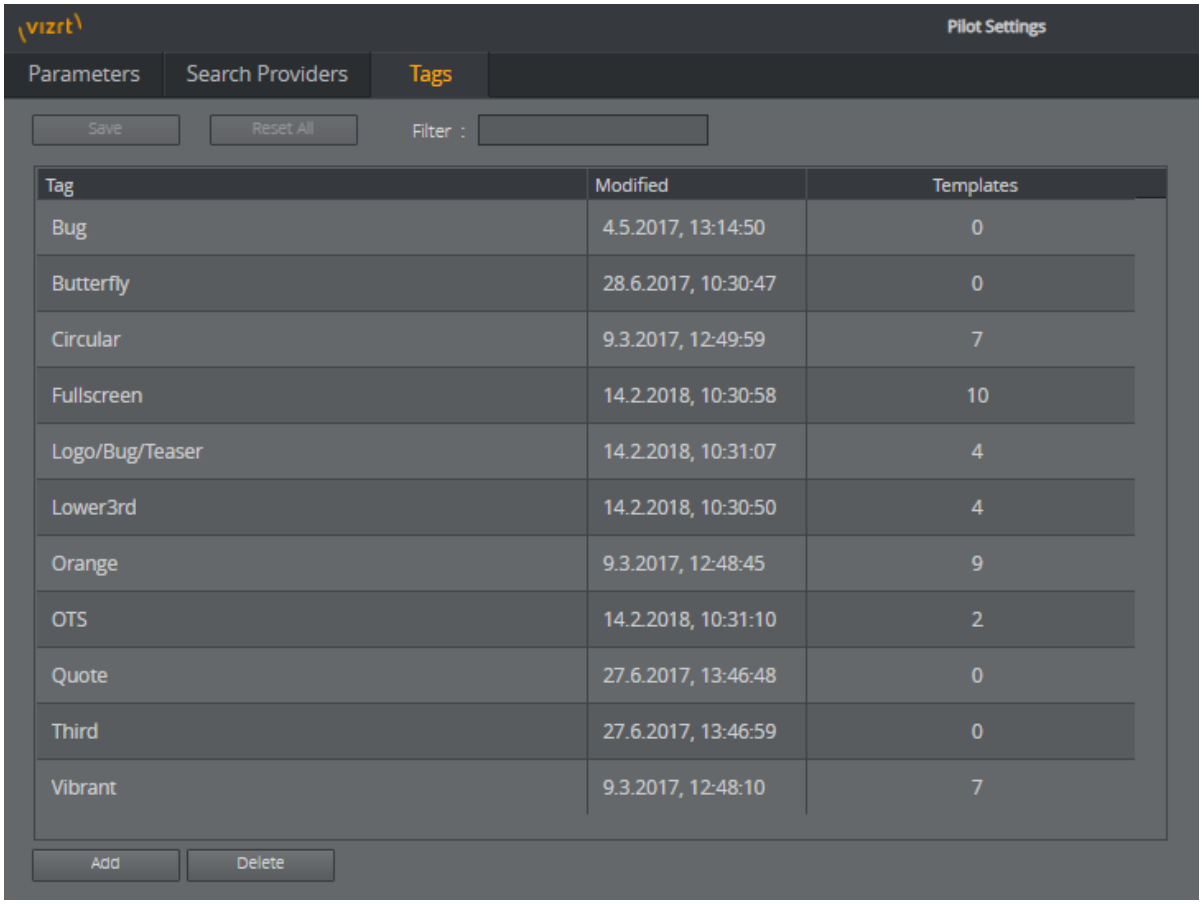
> **Note:** If you change the short name you need to restart the server for the changes to take effect.

### 11.2.3 Authentication for Feed Browser

The Pre-authenticated hosts configured in Pilot Data Server also apply to the Feed Browser when hosted in a Viz Pilot system application (see the Feed Browsing section under Newsroom Integration in the Viz Pilot User Guide).

---

## 11.3 Tags Settings



The Tags Settings page is used for administering tags. The tags are assigned to templates in Template Wizard, and then used in the newsroom component to organize templates.

### 11.3.1 Accessing Tags Settings

1. See how to access the Pilot Data Server Web Interface.
2. Click the **Settings** link.
3. Click the **Tags** tab.

### 11.3.2 Tags Settings Functions

| Function | Description |
|---|---|
| Add tag | Click the **Add** button and enter a name in the **Add a New Tag** dialog. |
| Rename tag | Click a name in the **Tag** list. Enter a new name and click **Save**. |
| Reset changes | Click **Reset All** to reset changes. |
| Delete | Click **Delete** to delete a selected tag.<br>⚠ **Note:** Deleting a tag does not delete the templates that refer to that tag. |
| Filter | Type to filter on tag name. |

## 11.4 Database Parameters

You can access all database parameters from the Pilot Data Server's VCP Parameters page (*http://<hostname>:8177/app/DataServerConfig/DataServerConfig.html*).

**Caution:** Settings should only be changed by administrators.

### 11.4.1 Viz Pilot Database Parameters

- **activex_preview_timeout:** Sets the timeout in milliseconds waiting for the Viz Preview Engine connection before continuing. The default value is 10000. Can be overridden by the local registry setting PreviewTimeOut (see Registry Settings in Viz Pilot User Guide).
- **allow_subfield_in_view:** Used client-side to determine if subfields can be added to custom layouts. Should be set to true when using Preview Server 4.4.1 or above, and set to false otherwise.
- **app_server:** Legacy parameter kept for backwards compatibility. Contains hostname:port of the running Pilot Data Server. The parameter `data_server_url` should be used instead.
- **ax_add_dbdata_to_mosexternal:** Adds extra information about the data in the element to the MOS item.
- **ax_add_itemchannel:** Adds default channel to the generated MOS XML.
- **ax_dataelement_timer_enabled:** Sets the default behavior for showing or hiding the Graphic Event Timing options in the newsroom component when Saving an Element. When enabled it shows the timing editor, and when disabled it hides the timing editor. This setting can be overridden by enabling the *ShowGraphicEventTiming* setting for the Template Information Component in Template Wizard on a template by template basis.
- **ax_disable_clear_layers:** This setting disables the clearing of all layers as the first commands sent when sending to preview.
- **ax_disable_data_overwrite:** When enabled, this setting disables the Save button in the newsroom component, only the **Save As** button is then active. This option also de-selects the *Add to Library* checkbox in the Template Save Dialog Box for all saved elements.
- **ax_disable_dragdrop:** When enabled, this setting disables drag-and-drop in the newsroom component. This option is useful when using a newsroom system that does not facilitate drag-and-drop interaction with plugins.
- **ax_disable_media_drag:** When enabled, this setting disables dragging of media from search results to the rundown. This option is de-selected by default, dragging of media from search results is allowed.
- **ax_disable_overlay_saving:** When enabled, this setting disables saving of overlay timelines for clip assets that have an overlay_timeline link. When a clip is opened from the media tab, there are no previously saved graphics on the timeline. Overlay saving is enabled by default.
- **ax_dont_fetch_thumbnails:** When enabled, the newsroom component does not fetch thumbnails for the templates and data elements when they are in list view mode, only in *show icons* view.
- **ax_enable_refresh_button:** When enabled, the **refresh** button in the ActiveX external preview form is shown. The preview is refreshed without resending the data to the preview server.
- **ax_enableMediaSendToRundown:** When enabled, this setting enables the Add to Rundown option in the media search. This is only useful if the newsroom system does not support drag and drop operations.
- **ax_force_detached_pvw:** This setting forces detached snapshot-preview window in Ax/VCP.
- **ax_hide_data_elements:** This setting hides the data element list in the newsroom component.

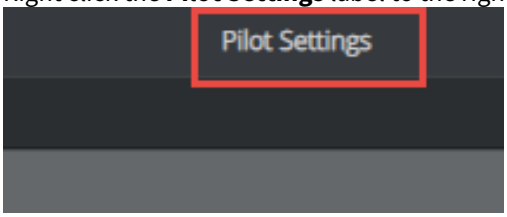
- **ax_hide_media_tab:** This setting hides the media tab in the newsroom component.
- **ax_include_mosobj_tag:** This setting adds a mosobj node to the generated MOS XML. This is required for some (Dalet) newsroom integrations.
- **ax_mos_edit_save_returns:** ActiveX checks this parameter to determine if it should return to the template list after saving (when editing).
- **ax_preview_host:** Hostname of the Viz Engine used to generate snapshot-preview images and fetch icons from Viz resources (images/geoms/materials etc.). It's also used as the fallback vizhost for the ThumbnailGenerator. Multiple renderers can be defined as a comma-separated string. For example <host>, <host>, ... or <host>:<port>,<host>:<port>, ... . If hosts are defined without a trailing port number, it's recommended to set the default port in the Port field. If no port number is set, it defaults to 6100.
- **ax_preview_img_protocol:** Request snapshots in different image formats.
- **ax_preview_port:** Port of the Viz Engine used to generate snapshot-preview images, and fetch icons of Viz resources (images/geoms/materials etc). It is also used as the fallback vizhost for the ThumbnailGenerator.
- **ax_ preview_tl_legacy_support:** When enabled, the preview sends values to legacy transition logic location ($other$object) in addition to normal behavior.
- **ax_preview_tl_show_next_sub:** The `\Toggle*continue_noanim` command is used when previewing transition logic.
- **ax_remove_continue_count:** This setting disables the sending of continueCount in mosExternalMetadata.
- **ax_remove_objSlug:** This setting removes objSlug from the generated MOS XML. This is needed when the newsroom component is hosted in AvidCommand.
- **ax_required_version:** The required version of the Vizrt newsroom component for the ActiveX to work. If the newsroom component version is earlier than this, a warning is displayed.
- **ax_savesas_cancels_edit** : Default behavior is: when editing an item already in a script, then clicking **Save As New**, the item in the script is replaced by the new item. Enabling this setting still creates a new graphic but does not replace the original item in the script.
- **ax_show_hints:** Disable this setting to turn off all tooltips in the ActiveX. Tooltips may cause the Dalet newsroom system to crash.
- **ax_showsavechanges_dlg:** This setting asks users if they want to save the currently open data element or template when opening a new one.
- **ax_use_custom_gui_dlg:** This setting embeds Viz Curious Maps Editor (CME) inside the same window as the newsroom component (and is used for this only).
- **clip_default_itemchannel:** If a value is set, the value is used as the default channel for video clips. The value is used in the itemChannel element in the MOS XML and appears in the Viz Trio playlist and in the NCS. This parameter is empty by default. If a video or timeline element is reopened from the NCS rundown, the itemChannel chosen in the rundown is not reset to default again, even if the element is updated and overwritten.
- **compatible:** Version number
- **country_language:** Set the language to use for the country list. English is supported by default.
- **crop_service_uri:** Sets the URL to the image crop service.
- **croptool_max_image_area:** Sets the maximum area of a cropped image in pixels that is served by Pilot Data Server. Anything bigger is resized, while still respecting the aspect ratio of the crop. The maximum size applies even if no cropping is done. Image size = width x height in pixels. 0 = disabled.
- **data_server_url:** Hosting URL of Vizrt Pilot Data Server (for example: http://exampleserver.com:8177) or the reverse proxy server pointing to Vizrt Pilot Data Server. May contain base path prefix at the end of the base URL (for example: http://exampleserver.com:8177/pds).

- **delete_data_from_activex:** Enable this setting to allow users to delete data elements from the newsroom component.
- **DELETE_DATA_PASSWORD:** Password protection for data element deletion in the Viz Pilot client. If the value is empty, the delete data elements dialog is not password protected.
- **disable_built_in_vos_search:** This setting disables the built-in VOS search. This should be used to prevent duplicate hits when moving to the new RESTful VOS search.
- **graphic_hub_url:** Host or reverse proxy URL of the Vizrt Graphic Hub API used to provide scenes (for example: http://exampleserver.com:19398/ .) If needed, it should contain a base path prefix at the end of the base URL, and an explicit port used, if this differs from the default 19398 port (for example **:** http://exampleserver.com:80/pilot.)
- **image_order_uri_template:** Placeholder URL template for image orders.
- **image_share:** Specifies the path the image Crop Tool (for templates) uses when saving a cropped image ( `<UNC or Windows path>` )
- **live_update_interval:** Update service poll interval in seconds. If the **Update at regular intervals while on air** option is enabled for a template with an update script or external update service, the Media Sequencer invokes the update service repeatedly using the interval specified here. The minimum value is two seconds.
- **MediaSearch_itemPerPage:** The value should correspond to Media Search Items per page.
- **model_uri_template:** URL template for the Pilot App Server to get the model document for a template given by its database ID. It is used by the Maps Preview Server, as well as certain external systems. Do not change this value.
- **moseditor_arc_url (optional):** The URL to a Viz Arc MOS plugin panel. This panel is shown in Viz Pilot Edge if it is configured.
- **moseditor_arc_schema (optional):** The unique qualifier string to identify the Viz Arc MOS plugin editor when reopening Viz Arc MOS items from the newsroom system. The default value is *http://www.vizrt.com/mosObj/vizarc/action*.
- **moseditor_arc_label (optional):** The title of the Viz Arc MOS plugin tab in Viz Pilot Edge. The default value is *Arc Actions*.
- **mos_override_objtb (optional):** Decides the value to be served out for the <objTb> tag in the MOS XML from Pilot Data Server. This value is normally 50 or 60, setting it to 0 breaks the timing info in Playout Instructions.
- **mos_override_objtype (optional):** Decides the value of the <objType> tag in the MOS XML for Pilot graphics elements served out from Pilot Data Server. This value is PILOT per default.
- **mos_override_objdur (optional):** Decides the value of the <objDur> tag in the MOS XML for Pilot graphics elements served out from Pilot Data Server. This value is 1 per default.
- **mse_script_runner:** This setting can be used to set a different script runner for the MSE. Normally not needed. The default value is blank.
- **order_mgt_uri_template:** URL template used to invoke the order management form.
- **payload_uri_template:** URL template for use with the Pilot App Server to get the payload document for a data element given by its database ID. It is used by the timeline editor in the newsroom ActiveX as well as certain external systems. Do not change this value.
- **preview_server_uri:** Base URL to the Preview Server used when requesting preview images in the timeline editor (for example: http://exampleserver.com:54000 ). This value is read by the Pilot App Server and published in its service document.
- **schema_build:** Minor build version of the database schema. Do not modify.
- **schema_version:** Major version of the database schema. Do not modify.

- **script_runner_uri:** The URL to the script runner that executes *update scripts* to *data/payloads*.
- **shared_curious_server:** Sets the Viz World Server IP or hostname for maps. In a multi-server setup, use comma-separated IP/hostnames.
- **show_playlist_element_index:** This setting shows a column in the VCP playlist that displays each item index in the list.
- **slim_mos_xml:** This setting reduces the MOS XML by removing optional tags.
- **spellcheck_dict_filename:** Sets the filename for the dictionary file (*.dic*) and affixation file (*.aff*) for the spell checker. This parameter is mandatory and is *en_US* by default. For example, if *spellcheck_dict_filename* is set to *my-special-en_US*, then the spell checker looks for the two files *my-special-en_US.dic* and *my-special-en_US.aff*. Several spell checking dictionaries are included in the Viz Pilot system installation. The dictionaries are installed in the default location: *%ProgramFiles(x86)%\vizrt\Common\dicts*. Other dictionaries can be downloaded from OpenOffice, however, these must be UTF-8 formatted. Contact your local Vizrt representative if you need to convert and use another dictionary for spell checking purposes.
- **spellcheck_dict_path:** Sets the path to the dictionary files for the spell checker. By default, this parameter is empty, which means the default location is used (*%ProgramFiles(x86)%\vizrt\Common\dicts*). If the files are not located under the default location, use a full path, mapped drive or UNC path.
- **timeline_update_service:** The service document URL for the Timeline Update Service.
- **vcp_schema_name:** For example, PILOT.
- **video_mode:** Video mode for channel. PAL or NTSC.
- **VOS_PASSWORD:** Shows the encrypted version of the VOS password. As this setting contains the encrypted version, it should not be set here, but rather be set using the VOS change password dialog. Setting this to blank disables password protection for VOS settings.
- **vtw_disable_unique_name_check:** Used if you need to save different templates in different concepts with the same name - if you are not using the Template Manager to link several scenes to the same template, but are instead creating several templates. This could be due to incompatible scene structure or similar.
- **vos_allow_edit_from_dll (optional):** Enables or disables the toolbar in Object Store. When disabled, it prohibits registering and editing of images and person information when Object Store is used with Viz Pilot News and Template Wizard. This option can be added to the database. Values are *Y* for allowing edits and *N* for prohibiting edits.

## Add Optional Database Parameters

1. Right click the **Pilot Settings** label to the right of the Settings window:




2. Click **Add New Optional Parameter**.
3. Enter the key as specified above for optional parameters, a value and a description.