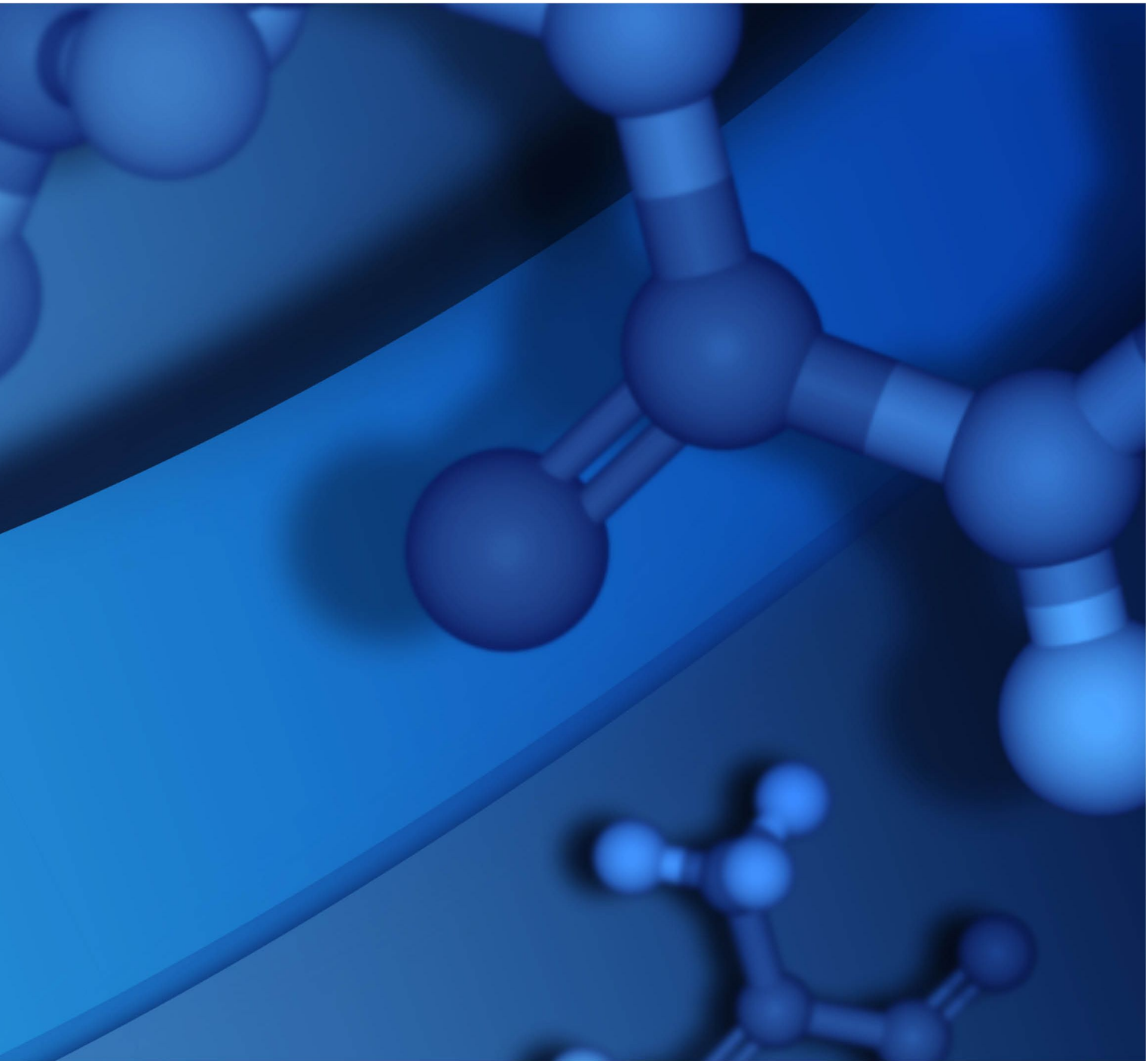
DASSAULT SYSTEMES

BIOVIA

MESSAGING SERVICE ADMINISTRATION GUIDE

DESKTOP CONNECTOR 2020



Copyright Notice

Acknowledgments and References

To print photographs or files of computational results (figures and/or data) obtained by using Dassault Systèmes software, acknowledge the source in an appropriate format. For example:

> "Computational results were obtained by using Dassault Systèmes BIOVIA software programs. BIOVIA Desktop Connector was used to perform the calculations and to generate the graphical results."

Dassault Systèmes may grant permission to republish or reprint its copyrighted materials. Requests should be submitted to Dassault Systèmes Customer Support, either by visiting https://www.3ds.com/support/ and clicking **Call us** or **Submit a request**, or by writing to:

Dassault Systèmes Customer Support
10, Rue Marcel Dassault
78140 Vélizy-Villacoublay
FRANCE

Contents

Chapter 1:
Introduction

This document is a guide to setting up and administering the **BIOVIA Desktop Connector Messaging Service** on a server computer.

What is the BIOVIA Desktop Connector?

The BIOVIA Desktop Connector is a desktop application for Windows and macOS that enables communication between BIOVIA applications running in users' web browsers, and local desktop applications like BIOVIA Draw, ChemDraw, and Microsoft Office. It also provides access to the client computer's file system and clipboard.

The Desktop Connector supersedes the BIOVIA Plugin application that was available up until the BIOVIA 2018 product releases. It provides a limited set of operations, such as opening documents, getting files, and getting clipboard objects. These are defined in application-specific plugin libraries.

The Desktop Connector can run in **Legacy modes** or **Messaging Service mode**:

- **Legacy modes:** These are the communication modes that were formerly provided by the BIOVIA Plugin. Depending on the browser being used, BIOVIA applications communicate with the Desktop Connector via ActiveX or a WebSocket service on the local server.
- **Messaging Service mode:** BIOVIA web applications communicate with the BIOVIA Desktop Connector via a Messaging Service that is installed on the application's server computer, or on a "standalone" server.

Applications That Use the Desktop Connector

Applications that use the Desktop Connector include:

- **BIOVIA Notebook:** Uses the Desktop Connector to interact with Microsoft Office (Word, Excel), and molecular structure drawing applications (BIOVIA Draw, ChemDraw, Marvin), the local file system, and the computer's clipboard.
- **BIOVIA Experiment:** Uses the Desktop Connector to take and store screen cuttings from BIOVIA Experiment and other applications. It also uses it to share cuttings with BIOVIA Notebook.
- **BIOVIA Chemical Registration** and **Biological Registration:** Use the Desktop Connector to launch and receive data from structure drawing applications such as BIOVIA Draw and ChemDraw.

What is the Messaging Service?

The Desktop Connector Messaging Service is a service on the server computer that mediates between the Desktop Connector and applications on the client computer. It must be installed in order for the Desktop Connector to run in Messaging Service mode (see Communication Modes of the Desktop Connector), and to serve the web page from which users download the Desktop Connector client.

Do I Need to Install the Messaging Service?

You must install the Messaging Service if either of the following is true:

- The Desktop Connector that you provide to users does not run in the Legacy modes, *and* those users do not use Internet Explorer to access BIOVIA applications.
- The Desktop Connector is not supplied to users of BIOVIA applications from an application-specific download page (as is the case with Notebook).

Chapter 2:
Communication Modes of the Desktop Connector

The BIOVIA Desktop Connector supports the "Legacy" communication modes that were provided by the BIOVIA Plugin until the 2018 product releases. It adds support for a new "Messaging Service" communication mode.

Each BIOVIA web application uses the communication mode appropriate to its requirements. The mode is configured in the web application.

For use in Messaging Service mode, the Desktop Connector requires the Messaging Service to be set up on the server computer. This procedure is described in Installing the Messaging Service.

Legacy Modes

The Legacy modes are the communication modes that were provided by the BIOVIA Plugin application until the 2018 BIOVIA product releases. The Desktop Connector continues to support these modes.

If a BIOVIA web application is configured to use the Legacy modes, the Desktop Connector chooses a mode according to which browser is being used:

- **Microsoft Internet Explorer (Windows):** The web application communicates with the Desktop Connector using an ActiveX browser plugin.
- **Google Chrome and Mozilla Firefox (Windows):** The web application communicates with the Desktop Connector using a WebSocket service on the local network (localhost). The WebSocket service is provided by the Desktop Connector. This communication channel is likely to be blocked by major browsers in the near future. It is not supported by Microsoft Edge.

Note: The Legacy modes are no longer supported on macOS. This is due to the discontinuation in Mac browsers of support for NPAPI, which was used by the Legacy modes.

Messaging Service Mode

In Messaging Service mode, the BIOVIA web application communicates with the Desktop Connector via a Messaging Service. This is a web service specifically designed for relaying messages between a BIOVIA web application and the BIOVIA Desktop Connector. It provides a REST API for handling messages, and a SignalR service for sending notifications to clients.

The Messaging Service is installed on Microsoft IIS.

The Messaging Service mode can be used with major browsers on Windows and Apple macOS computers. It is not currently available for Opera.

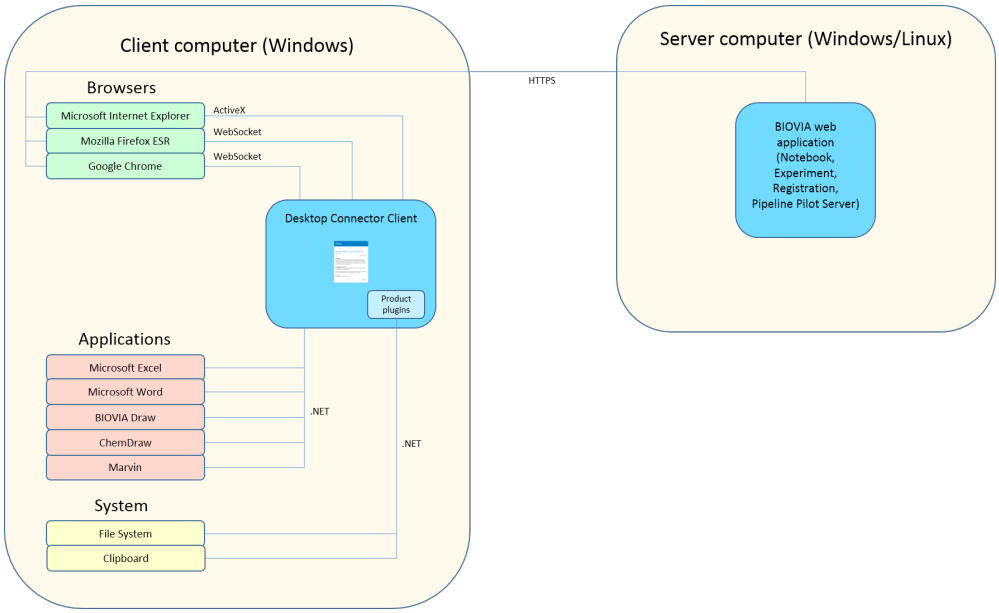
Notes:

- If the application that uses the Messaging Service is installed on Linux (for example, Pipeline Pilot Server), the Messaging Service must be installed on a separate Windows Server computer.
- Because the Desktop Connector is always used in the context of a BIOVIA web application, users should consult the documentation for that application for full details of browser compatibility.

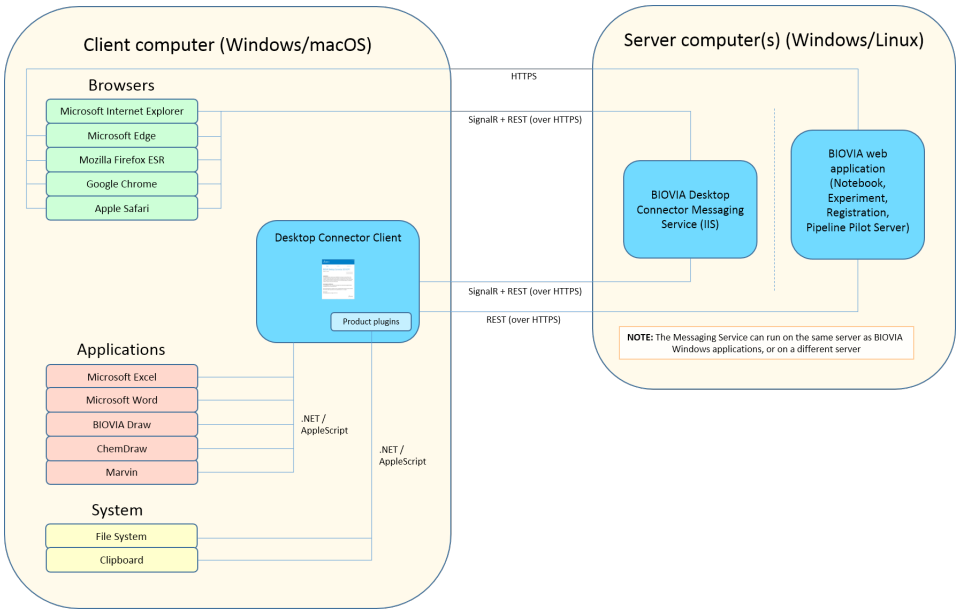
Chapter 3:
Architectural Overview

The following diagrams show the communications channels in the Legacy modes and the Messaging Service mode.

Legacy Modes



Messaging Service Mode



Chapter 4:
Requirements

This chapter lists the requirements for the Messaging Service, and compatibility of the Desktop Connector with desktop applications.

Messaging Service Requirements

- **Windows Server 2012 R2 or 2016 or 2019.** Only 64-bit variants are supported.
- **Internet Information Services (IIS) 7.0 or later.**

  Some extra configuration of IIS is required. See Configuring IIS.
- **.NET 4.6.1 or 4.7.**
- **HTTPS.** All BIOVIA web applications that use the same instance of the Messaging Service must employ the same transfer protocol (HTTP or HTTPS) to access it. For security reasons, **this should always be HTTPS.**

Configuring IIS

Before you install the Messaging Service on your server, you must define your server as a Web Server, and configure IIS, as described below.

> **Note:** The procedure for configuring IIS on your computer might differ slightly from that shown below, depending on how the Server Manager is set up. If you are unsure how to adapt the instructions for your environment, consult your system administrator.

1. Start the Server Manager.
2. Select **Manage > Add Roles and Features**, and work your way through the wizard, following the instructions in the steps below. Click **Next** after each step:
   a. In the **Installation Type** section, select **Role-based or feature-based installation**.
   b. In the **Server Selection** section, select **Select a server from the server pool**, and choose the appropriate server from the list.
   c. In the **Server Roles** section, expand **Web Server (IIS)**, and select the following features:
      - Web Server
        - Common HTTP Features
          - Default Document
          - Directory Browsing
          - HTTP Errors
          - Static Content
          - HTTP Redirection
        - Health and Diagnostics
          - HTTP Logging
          - Request Monitor

- Performance
  - Static Content Compression
  - Dynamic Content Compression
- Security
  - Request Filtering
  - Basic Authentication
  - Windows Authentication
- Application Development
  - .NET Extensibility 3.5
  - .NET Extensibility 4.5
  - ASP.NET 4.5
  - ISAPI Extensions
  - ISAPI Filters
  - Server Side Includes
- Management Tools
  - IIS Management Console
  - IIS Management Compatibility
    - IIS 6 Metabase Compatibility
    - IIS 6 Management Console

d. In the **Features** section, select the following features:

- .NET Framework 3.5 Features
  - .NET Framework 3.5
  - HTTP Activation
  - Non-HTTP Activation
- .NET Framework 4.5 Features
  - .NET Framework 4.5
  - ASP.NET 4.5
  - WCF Services
    - HTTP Activation
    - TCP Activation
    - TCP Port Sharing

e. In the **Confirmation** section, click **Install**.

Desktop Connector Client: Compatibility

Integration with Desktop Applications

The Desktop Connector client supports integration with the local desktop applications listed below.

Although all these applications are supported by the Desktop Connector, individual BIOVIA applications that use the Desktop Connector typically only support a subset of them. For example, the Pipeline Pilot Sketcher Integration Collection does not integrate with Microsoft Office applications or MarvinSketch.

For full information on requirements and compatibility of individual BIOVIA applications, see the application-specific documentation.

Windows:

Software	Versions	Comments
Microsoft Excel	2019 (Office 365) 2016 (Office 365)	■ BIOVIA currently only supports the desktop versions of the Office applications that are released as part of an Office 365 subscription or a "standalone" installation of Office. ■ BIOVIA does not support the mobile Office applications which are delivered as part of the Office 365 subscription. ■ BIOVIA applications support Office applications when installed by the Click-to-Run installers.
Microsoft Word	2019 (Office 365) 2016 (Office 365)	
BIOVIA Draw	2020 2019	Enterprise Edition (EE) only.
ChemAxon MarvinSketch	19 18	Windows only.
PerkinElmer ChemDraw	18.2 18 17	-

Mac:

Software	Versions	Comments
Microsoft Excel for Mac	2019 2016	-
Microsoft Word for Mac	2019 2016	-
PerkinElmer ChemDraw	18.2 18 16	-

Compatible Browsers

The Desktop Connector can be used with BIOVIA applications in the following web browsers:

Browser	Version	Operating System
Microsoft Internet Explorer	11	Windows 10
Microsoft Edge	41 or later	Windows 10
Mozilla Firefox	60 ESR or later	Windows 10 OSX 10.12 OSX 10.13 OSX 10.14
Google Chrome	74 or later	Windows 10 OSX 10.12 OSX 10.13 OSX 10.14
Apple Safari (Messaging Service mode only)	12 or later	OSX 10.12 OSX 10.13 OSX 10.14

Chapter 5:
Installing the Messaging Service

When the BIOVIA Desktop Connector is used in Messaging Service mode, it communicates with BIOVIA web applications via its own Messaging Service. The Messaging Service can be installed on the same server as the BIOVIA web application, or on a separate "standalone" server. Using a standalone Messaging Service can improve performance by reducing the number of persistent HTTPS connections to the BIOVIA web application domain. (Use of non-secure HTTP is not recommended.) Because the Messaging Service runs on Windows, a standalone Messaging Service is mandatory if its "consumer" application runs on Linux (for example, a Linux installation of Pipeline Pilot Server).

If you are an administrator of a BIOVIA application that uses the Desktop Connector in Messaging Service mode (see Communication Modes of the Desktop Connector), you must install the Desktop Connector Messaging Service. An instance of the Messaging Service can be shared between multiple BIOVIA web applications.

To install the Desktop Connector Messaging Service:

1. Download the installer `BIOVIA Desktop Connector Service.msi`.
2. Open the command prompt **as an administrator**.
3. Navigate to the folder to which you downloaded the installer.
4. Run the following command:

   ```
   msiexec /i BIOVIA Desktop Connector Service.msi
   ```

5. Follow the instructions in the installation wizard.

   At the appropriate stages during installation:

   a. Select an installation type:

   - **BIOVIA existing site:** Install the Messaging Service on the same IIS site as an existing BIOVIA web application.
   - **Create web site:** Create a new site in IIS. This creates a standalone Messaging Service on a new site. This configuration may improve performance by reducing the number of persistent HTTP connections to the BIOVIA web application's domain.

     Note: If the Messaging Service will use HTTPS, you must manually create an HTTPS binding after the Messaging Service is installed. See Configuring the Messaging Service to Run Over HTTPS.

   b. Choose an installation folder.

   c. If you selected **BIOVIA existing site**, choose an existing web site from the list that opens.

   Note: If no existing sites are listed, check that IIS is running and that a site is configured. If there are no existing sites, click **Back** in the installer until you return to the page where you select the installation type. There, select **Create web site**, and proceed with the installation.

   d. Click **Finish** in the last stage of the wizard to complete installation.

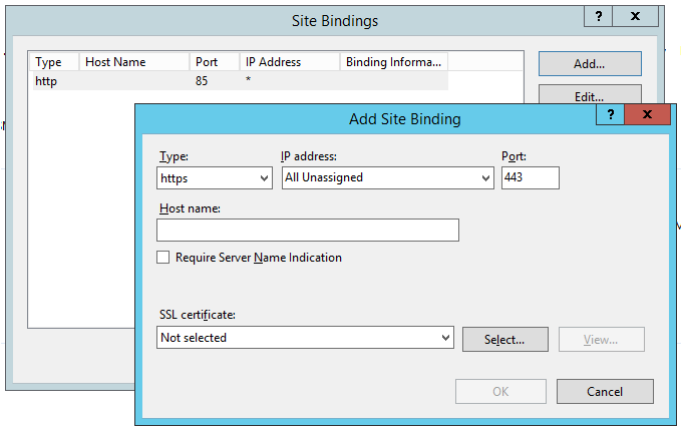
6. If you will use WebSocket as the SignalR transport protocol, enable WebSocket as described in Enabling WebSocket Transport for SignalR Connections.

After you install the Messaging Service, you must configure it. See Configuring the Messaging Service.

Configuring the Messaging Service to Run Over HTTPS

If you selected **Create web site** when you installed the Messaging Service, the new site is created with an HTTP binding on port 85. It is strongly recommended that you change the site to run over HTTPS. To do this, you must manually configure an HTTPS binding in IIS:

1. Install an SSL certificate.
2. In IIS Manager, add an HTTPS Site Binding for your site, using the following settings:
   - **Type:** `https`
   - **IP address:** `All Unassigned`
   - **Host name:** Leave blank
   - **Require Server Name Indication:** Deselect
   - **SSL certificate:** Select an installed SSL certificate



For full instructions on configuring IIS, see the documentation provided by Microsoft for your system.

Enabling WebSocket Transport for SignalR Connections

The Desktop Connector Messaging Service uses SignalR to send notifications to the web client and to the Desktop Connector on the client computer.

SignalR uses one of several available transport protocols. Which transport protocol is used is determined by a negotiation between the client and the server.

BIOVIA has found that optimum performance and stability are achieved using the WebSocket transport protocol. This protocol can be used by all supported client applications except the Desktop Connector running on Windows 7.

To use WebSocket for SignalR communications, you must enable WebSocket on the web server computer, as follows:

1. Open the Windows Server Manager.
2. Select **Manage > Add Roles and Features**.
3. Select **Role-based or Feature-based Installation**, and click **Next**.
4. Select the appropriate server (your local server is selected by default), and click **Next**.
5. In the **Roles** tree, expand **Web Server (IIS) > Web Server > Application Development**.
6. Select **WebSocket Protocol**, and click **Next**.

7. If you do not need to install any other features, click **Next**.
8. Click **Install**.
9. When the installation completes, click **Close** to exit the wizard.

Disabling and Removing WebDAV

WebDAV should not be enabled in the same IIS instance as the Messaging Service, because it can disrupt the Desktop Connector's access to desktop applications.

If WebDAV is enabled in IIS, disable it or remove it as described below. If you remove it, you should disable it first.

If WebDAV is not installed, you do not need to make any changes.

Checking Whether WebDAV is Installed

To check whether WebDAV is installed in IIS:

1. Open the **Internet Information Services (IIS) Manager** from Administrative Tools.
2. Select the Messaging Service web site (from which WebDAV should be removed).
3. Look to see whether an option **WebDAV Authoring Rules** is available.

   If the option is available, proceed to Disabling WebDAV.

   If the option is not available, WebDAV is not installed, and you do not need to take any further action.

Disabling WebDAV

To disable WebDAV in IIS:

1. Perform the steps in Checking Whether WebDAV is Installed.
2. Double-click **WebDAV Authoring Rules**.
3. Click **Disable WebDAV** in the list of options.

Removing WebDAV Handler Mappings and Modules

To remove WebDAV handler mappings and modules from IIS:

1. Open the **Internet Information Services (IIS) Manager** from Administrative Tools.
2. Select the **Eln** website (from which the WebDAV elements should be removed).
3. Double-click **Handler Mappings**.
4. Select **WebDAV** from the list, and click **Remove**.
5. Double-click **Modules**.
6. Select **WebDAV module** from the list, and click **Remove**.

Chapter 6:
Configuring the Messaging Service

After you install the Desktop Connector Messaging Service, you must configure it for the BIOVIA applications that will use it.

To configure the Desktop Connector Messaging Service:

1. On the Messaging Service server computer, open the following file:

   `C:\Program Files (x86)\BIOVIA\DesktopConnectorService\Api\Web.config`

   If you installed the Messaging Service in a non-default location, use the appropriate file path.
2. Edit settings as applicable, as described in the sections that follow:
   - Cross-Origin Access
   - Server Log Settings
   - Desktop ID Cookie Lifetime
   - Overriding BIOVIA Notebook Database Connection Settings
   - Providing a Custom Desktop Connector Installer (optional)
3. If you are implementing load balancing, follow the instructions in Setting Up a Load-Balanced Environment.
4. Save and close the `Web.config` file.
5. Perform addition configuration of the BIOVIA applications that will use the Messaging Service (for example, Pipeline Pilot Server, Notebook, Experiment). For instructions, see the application-specific documentation.

   Note also the guidelines in the **IMPORTANT!** box under Cross-Origin Access.

Cross-Origin Access

If the "consumer" BIOVIA web application and the Messaging Service do not use the same host address, the Messaging Service must be configured to allow Cross-Origin Resource Sharing (CORS). To do this, add a `CorsAllowOrigins` application setting in the `appSettings` section of the `web.config` file for the Messaging Service. The value of this setting is a comma-delimited list of other origins to which access will be granted.

If an origin does not use the default port (80 for HTTP, 443 for HTTPS), you must include the port number in the origin specification.

Example:

```
<add key="CorsAllowOrigins" value="https://server2.company.com,
https://server3.company.com:9443"/>
```

Here, `server2.company.com` uses the default HTTPS port 443, and `server3.company.com` uses the non-default port 9443.

IMPORTANT!

- Each origin reference must include the fully qualified domain name. For example, specify `https://server2.company.com`, **not** `https://server2`.
- The consumer application must be configured to refer to the Messaging Service using its full qualified domain name. For details, see the consumer application's documentation.
- Users of the consumer application must gain acces to the consumer application using its fully qualified domain name.

Server Log Settings

For server logs, supply values for the keys shown in the example configuration below.

```
<add key="LogDirectory" value="C:\temp"/>
<add key="LogMaxFiles" value="5"/>
<add key="LogMaxFileSizeInKilobytes" value="5120"/>
<add key="LogLevel" value="3"/>
<!-- 0:ERROR, 1:WARNING, 2:INFO/NETWORK, 3:VERBOSE -->
```

The keys are:

- `LogDirectory`: The path of the folder in which log files are created. This can be an absolute path, or a path relative to the Messaging Service installation folder.

  The default log folder is the `Log` subfolder of the installation folder.

  If the configured folder or the default folder are inaccessible, log files are created in the `BIOVIA\Log` subfolder of the user's `temp` folder, or in `C:\temp\BIOVIA\Log`.
- `LogMaxFiles`: The maximum number of Messaging Service log files to be retained. If the current number of Messaging Service log files is equal to or larger than the specified number, the oldest file is deleted when a new file is created.

  The default value is 5.
- `LogMaxFileSizeInKilobytes`: The maximum size of a Messaging Service log file. When this limit is reached, a new log file is created.

  The default value is 5120.
- `LogLevel`: The degree of verbosity of log files. This must be an integer between `-1` and 4.

  Valid log levels are as follows:
  - `-1`: No logging
  - 0: ERROR (ERROR messages only)
  - 1: WARNING (ERROR and WARNING messages)
  - 2: INFO (ERROR, WARNING, and INFO messages)
  - 3: VERBOSE (All messages)
  - 4: DEBUG (All messages, plus debugging information)

  The default value is 1 (ERROR and WARNING messages).

  Note: Log Level 4 produces a huge number of log records, and should only be used for troubleshooting.
- `EnvtID`: A string that is inserted into log file names, before the timestamp. The string is preceded and followed by underscores. There is no default value.

This key can be useful if the Messaging Service is operating in a load-balanced environment.

Example:

```
<add key="EnvtID" value="LB1"/>
```

This produces file names with this format:

```
DesktopConnectorService_Computer123_LB1_20190725_2052500504.log
```

Desktop ID Cookie Lifetime

The `desktop_id` browser cookie created by the Messaging Service is required for the browser to address the correct Desktop Connector instance. This cookie has a default lifetime of 100 days. You can change its lifetime by editing the `DesktopIdCookieMaxAgeSeconds` configuration setting in `Web.config`:

```
<!-- Maximum Lifetime of DesktopId cookies.
Must be specified number of seconds.
Default is 8640000, corresponding to 100 Days -->
<add key="DesktopIdCookieMaxAgeSeconds" value="8640000" />
```

Overriding BIOVIA Notebook Database Connection Settings

If the Messaging Service is installed on the same server and the same web site as BIOVIA Notebook, and load balancing is *not* being used, you must add the following lines to the `appSettings` section of `Web.config`:

```
<!-- Prevent inheriting the DATABASETYPE and DATABASESERVERNAME
configuration values from the Notebook application -->
<add key="DATABASETYPE" value="" />
<add key="DATABASESERVERNAME" value=""/>
```

These settings force the Messaging Service to run in standalone mode without attempting to connect to the database, even if BIOVIA Notebook has been reconfigured to use the common settings file.

Providing a Custom Desktop Connector Installer

You can create a custom self-extracting installer for the Desktop Connector with its own default configuration settings, and make this the installer that users download.

You can make the custom installer downloadable from the same location as the default installer, or from a different location.

Note: If you are a Notebook administrator, and you want to create a custom Desktop Connector that does not use the Messaging Service and is downloadable from Notebook's **Download Now** button, follow the instructions in *Notebook Installation Guide* > BIOVIA Desktop Connector Service Configuration > "Providing a Custom Desktop Connector Installer".

Creating the Custom Desktop Connector Installer

IMPORTANT! If the custom installer will be used on 32-bit Windows computers, you must create it on a 32-bit Windows computer.

To create a custom Desktop Connector installer:

1. In a web browser, download the Desktop Connector zip file from its path on the Messaging Service. For example:

   `https://server.company.com/dc/api/download/BIOVIADesktopConnector.zip`

2. Extract the contents of the zip file to a temporary folder; for example, `E:\temp\BIOVIADesktopConnector`.
3. If necessary, edit the default settings in the file `BIOVIA_Desktop_Connector_IExpress_Template.SED` as follows:
   a. Set `SourceFiles0` to match the temporary folder that you created in Step 2. Example:

      `SourceFiles0=E:\temp\BIOVIADesktopConnector\`

      Note: The default setting is `C:\temp\BIOVIADesktopConnector\`.

   b. Set `TargetName` to the full path (folder and file name) of the new installer that will be created. Example:

      `TargetName=E:\temp\BIOVIADesktopConnector\BIOVIA_Desktop_Connector_Setup.exe`

      Note: The default location is `c:\temp\BIOVIADesktopConnector\BIOVIA_Desktop_Connector_Setup.exe`.

4. Configure the behavior of the Desktop Connector for the custom installer, by editing the following two files:
   - `DesktopClient.exe.config`: Configures the Desktop Connector.
   - `DesktopClientMonitor.exe.config`: Configures the Desktop Connector Monitor application, which manages the Desktop Connector (starts it and provides the system tray icon, among other things).

   These files are in the temporary folder that you created in Step 2. Add keys to the `appSettings` sections of the files from the table below.

Key	Description
`HostProcessPort`	The port used by the local WebSocket server provided by the Desktop Connector, unless disabled (see below). This WebSocket server is used only by the Legacy modes. The valid range is 5000-5011. The default value is 5000. If the port specified by `HostProcessPort` is being used by another application, the Desktop Connector starts the WebSocket server using the first available port found within a range of eleven ports, starting with the specified `HostProcessPort` value. If the Desktop Connector is already running, and then another application attempts to start using the same WebSocket port, this application will fail to run. To avoid this situation, you can set `HostProcessPort` so that lower-numbered ports are reserved for other applications. For example, if another application uses Port 5003, you can set `HostProcessPort` to 5004. An alternative way of avoiding WebSocket port conflicts is to use `DisableLocalWebsocketServer` to disable the Desktop Connector's local WebSocket service completely. Example: `<add key="HostProcessPort" value="5004" />`
`DisableLocalWebsocketServer`	If set to `true`, specifies that the local WebSocket server provided by the Desktop Connector is disabled. This means that the Legacy modes are disabled for all browsers except Internet Explorer. The default value is `false`. You can set `DisableLocalWebsocketServer` to `true` to avoid WebSocket port conflicts (see `HostProcessPort`). If you do this, ensure: ■ *Either* that all BIOVIA applications that use the Desktop Connector are configured to use the Messaging Service mode; ■ *Or* that if an application uses the Legacy modes, all users gain access to the application using Internet Explorer. Example: `<add key="DisableLocalWebsocketServer" value="true" />`

Key	Description
LogDirectory	The folder to which Desktop Connector log files are written. The path can be absolute, or relative to the folder %USERPROFILE%\AppData\Local\BIOVIA\BIOVIA Desktop Connector. The default value is Log. If the Desktop Connector cannot write to, or create, the configured folder, it attempts to save the log file in an alternative location. The list of locations that it tries is as follows: 1. The folder specified by LogDirectory 2. %USERPROFILE%\AppData\Local\BIOVIA\BIOVIA Desktop Connector\Log 3. The Log subfolder of the Desktop Connector installation folder 4. %TEMP%\BIOVIA\Log 5. C:\temp\BIOVIA\Log Example: <add key="LogDirectory value="Temp" />
LogLevel	The degree of verbosity of log files. Valid log levels are as follows: ■ -1: No logging ■ 0: ERROR ■ 1: WARNING ■ 2: INFO/NET/NOTIFY ■ 3: VERBOSE ■ 4: DEBUG The default setting is 1. **Note:** Log Level 4 produces a huge number of log records, and should only be used for troubleshooting. Example: <add key="LogLevel" value="3" />
LogMaxFiles	The maximum number of Desktop Connector log files to be retained. If the current number of log files is equal to or larger than the specified number, the oldest file is deleted when a new file is created. The default value is 5. Example: <add key="LogMaxFiles" value="10" />

Key	Description
LogMaxFileSizeInKilobytes	The maximum size of a Desktop Connector log file. When this limit is reached, a new log file is created. The default value is 5120. Example: <add key="LogMaxFileSizeInKilobytes" value="2560" />

5. Open a command window, and change directory to the temporary folder. Example:
   ```
   cd E:\temp\BIOVIADesktopConnector
   ```
6. Create the self-extracting installer EXE file by executing the following command:
   ```
   iexpress /N BIOVIA_Desktop_Connector_IExpress_Template.SED
   ```
7. Digitally sign the installer EXE file if this is required by the company or site policy.

Publishing the Installer to the Default Location

To publish the custom Desktop Connector installer to the default location:

1. Copy the custom installer EXE file to the server, in the folder
   ```
   DesktopConnectorSevice\api\download
   ```
2. In the file `DesktopConnectorSevice\api\download\Web.config`, set `DesktopClientWindowsDownloadUri` to the name of the custom EXE installer file. Example:
   ```
   <appSettings>
     ...
     <add key="DesktopClientWindowsDownloadUri" value="BIOVIA_Desktop_
   Connector_Setup.exe"/>
     ...
   </appSettings>
   ```

Publishing the Installer to a Non-Default Location

To publish the Desktop Connector installer (custom or default) to a non-default location:

1. Copy the installer to the server folder from which you want users to download it.
2. In the file `DesktopConnectorSevice\api\download\Web.config`, set `DesktopClientWindowsDownloadUri` to the absolute URL of the custom EXE installer file. Example:
   ```
   <appSettings>
     ...
     <add key="DesktopClientWindowsDownloadUri"
   value="https://server.company.com/download/BIOVIA_Desktop_Connector_
   Setup.exe"/>
     ...
   </appSettings>
   ```

Chapter 7:
Setting Up a Load-Balanced Environment

When the Desktop Connector Messaging Service is run in a load-balanced environment, runtime data is shared between the individual nodes in the environment using a database. Currently, this database must be an Oracle database.

If you are using Desktop Connector in a load-balanced configuration, you must configure storage of messages sent between the web client and users' desktops in the database. The database configuration is specified in the Messaging Service's `Web.config` file. The sections that you must edit are `configSections`, `oracle.manageddataaccess.client`, and `appSettings`.

Note: When using load balancing, ensure that individual Messaging Service nodes are only accessed via the load balancer, and that individual nodes are never accessed directly.

To enable database storage of the messages sent between the client and the desktop in a load-balanced environment:

1. On the Messaging Service server computer, open the following file:

   `C:\Program Files (x86)\BIOVIA\DesktopConnectorService\Api\Web.config`

   If you installed the Messaging Service in a non-default location, use the appropriate file path.

2. In the `configSections` section, add details of the `oracle.manageddataaccess.client` section. For guidelines on configuring the Oracle .NET managed driver, see http://www.oracle.com/technetwork/topics/dotnet/downloads/odpnet-managed-nuget-121021-2405792.txt

   Example:

```
<configSections>
  <!-- See http://www.oracle.com/technetwork/topics/dotnet/downloads/
odpnet-managed-nuget-121021-2405792.txt -->
  <section name="oracle.manageddataaccess.client" type=
"OracleInternal.Common.ODPMSectionHandler, Oracle.ManagedDataAccess,
Version=4.121.2.0, Culture=neutral, PublicKeyToken=89b483f429c47342"/>
</configSections>
```

3. In the `oracle.manageddataaccess.client` section, add the database connection details. For guidelines on configuring the Oracle .NET managed driver, see http://www.oracle.com/technetwork/topics/dotnet/downloads/odpnet-managed-nuget-121021-2405792.txt

   Example:

```
<oracle.manageddataaccess.client>
  <version number="*">
    <dataSources>
      <!-- Customize these connection alias settings to connect to
Oracle DB -->
      <dataSource alias="MyDataSource" descriptor="(DESCRIPTION=
(ADDRESS=(PROTOCOL=tcp)(HOST=localhost)(PORT=1521))(CONNECT_DATA=
(SERVICE_NAME=ORCL))) " />
    </dataSources>
```

```
  </version>
</oracle.manageddataaccess.client>
```

4. In the appSettings section, supply values for the keys shown in the example configuration below.

```
<appSettings>
  ...
  <add key="UseDatabaseNotification" value="true"/>
  <add key="DatabaseNotificationPort" value="1005"/>
  <add key="DATABASESERVERNAME" value="my_oracle_alias"/>
  <add key="DATABASEUSERNAME" value="jsmith"/>
  <add key="DATABASEPASSWORD" value="P4$$vv0r6"/>
  <add key="DefaultSchema" value="my_messaging_service_schema"/>
  <add key="DATABASETYPE" value="ORACLE"/>
  <add key="Catalog" value="my_catalog"/>
  ...
<appSettings>
```

The keys are:

- UseDatabaseNotification: A Boolean flag (true or false) indicating whether the individual Messaging Service instance is notified of changes to stored data.

  IMPORTANT! This setting must be true in a load-balanced environment.

- DatabaseNotificationPort: The database port number for notifications.

  This setting must be configured if there is a firewall separating your application server and your database server. You must open this port in your firewall (see Step 6).

- DATABASESERVERNAME: The Oracle server alias, as defined in the oracle.manageddataaccess.client section.
- DATABASEUSERNAME: The name of a user with privileges to read and write data.
- DATABASEPASSWORD: The password of the database user.
- DefaultSchema: The database schema that holds the Messaging Service data tables.
- DATABASETYPE: The database type.

  IMPORTANT! Currently, the only allowed value is ORACLE.

- Catalog: The database catalog that is used to hold the server.

Note: Another key is available, EnvtID. This specifies a string to insert into log file names, and can be useful in load-balanced environments.

5. In an Oracle client application, grant the change notification Oracle privilege to the database user, using the following command:

   ```
   grant change notification to USERNAME;
   ```

   Example:

   ```
   grant change notification to jsmith;
   ```

6. If there is a firewall separating your application server and your database server, open the database notification port in your firewall. (In the example from Step 4, this is Port 1005.)
7. Save and close the web.config file.

Chapter 8:
Using the Messaging Service Behind a Reverse Proxy Server that Rewrites URLs

The Messaging Service can be installed behind a reverse proxy server that rewrites URLs received from client computers. The proxy server replaces the public URL used by the client computer with the equivalent internal URL for the computer hosting the Messaging Service.

The Messaging Service requires configuration for this type of environment, to convert URLs that it sends to client computers to their publicly valid format. This configuration is described in this chapter.

Note: The software library that provides the functionality described in this chapter is common to the following BIOVIA applications and components:

- Notebook
- Notebook Data Archive
- NotebookLoginService
- Desktop Connector Messaging Service

For instructions on configuring the Notebook-related items, see the *BIOVIA Notebook Installation Guide* and the *BIOVIA Notebook Data Archive Installation Guide*. The configuration required is almost identical to that for the Messaging Service. If Notebook and the Messaging Service are hosted on the same IIS site, a single configuration file can be used that is inherited by both the Notebook items and the Messaging Service. See Configuring the Messaging Service.

Why Reverse Proxy Servers Rewrite URLs

A reverse proxy server might rewrite URLs for any of the following reasons:

- The server computer hosting the Messaging Service is protected by a firewall, and is only directly accessible through an internal network address that is not visible to client computers.
- Secure SSL traffic terminates at the firewall, and simple HTTP requests are used within the private network.
- A load balancer is configured in such a way that a different URL is used for each web node.
- Request paths are rewritten to fit the internal architecture.

Why the Messaging Service Must be Configured for Rewritten URLs

The Messaging Service can include public URLs in its responses to client computers that the clients use in further requests to the Messaging Service. Such URLs are used for various purposes, for example:

- In location response headers (i.e. responses to POST/PUT requests. or redirect responses).
- In Hypermedia as the Engine of Application State (HATEOAS) links.
- In notification messages.

If the Messaging Service is situated behind a reverse proxy server that rewrites URLs, the default URLs that the Messaging Service creates for clients will point to the Messaging Service's own internal network address, and not to the public address of the proxy server. Client requests using such URLs will fail. For

this reason, functionality is provided to configure the Messaging Service to convert internal URLs to public URLs. You can configure the following URL modifications:

- Host name substitution
- Protocol substitution (HTTP/HTTPS)
- Local path conversion

Summary of the Configuration Process

The following are the main steps to set up the Messaging Service for rewritten URLs:

1. Configure the proxy server to add forwarding headers to requests before sending the requests on to the Messaging Service server. The forwarding headers provide the protocol, host name, and port of the public URL to the Messaging Service.
2. Add settings to the IIS web configuration file for the Messaging Service to generate public URLs.

Configuring the Reverse Proxy Server

You must configure the reverse proxy server to add forwarding headers to requests received from client computers, before it sends them on to the Messaging Service. The Messaging Service supports two formats of forwarding headers:

- The `Forwarded` header. This is specified in RFC-7239. The Messaging Service uses the `host` and `proto` parameters in this header.

  Example:

  ```
  Forwarded: for=10.220.244.129;by=proxy.company.com;host=public.company.
  com;proto=https
  ```

- The `X-Forwarded-Host` and `X-Forwarded-Proto` headers. These are de facto standard headers, and not part of any current specification.

  Example:

  ```
  X-Forwarded-Host: public.company.com
  X-Forwarded-Proto: https
  ```

Notes:

- Whichever headers you use, the host value must include any non-default port number, as specified in Section 5.4 of RFC-7230. If no port number is provided, the default port for the protocol is assumed.
- If there are intermediate forwarding steps between the proxy server and the Messaging Service, only the forwarding headers supplied by the proxy server in the first step are used in rewriting of URLs.

Configuring the Messaging Service

Note: Examples are provided in the next section.

To configure public URL substitutions for the Messaging Service:

1. On the Messaging Service server computer, open the following file:

   ```
   C:\Program Files (x86)\BIOVIA\DesktopConnectorService\Api\Web.config
   ```

If you installed the Messaging Service in a non-default location, use the appropriate file path.

2. Add the `section` element shown in bold below as a child of the `configSections` element:

```
<configSections>
  ...
  <section name="PublicUrlHelper"
type="Biovia.DesktopConnector.WebUtils.Server.PublicUrlHelperConfigSecti
on" />
  ...
</configSections>
```

3. Create a new element `PublicUrlHelper` as a child of the `configuration` element.

   Include attributes from the table below to implement URL replacement rules (all the attributes are optional; if none are set, public URLs are not modified):

Attribute	Description	Default Value
`UseForwardHeaders`	If `true`, the Messaging Service uses the `Forwarded` header to specify the protocol and host segments of the URL. If the `Forwarded` header reflects several forwarding steps, the value from the first step is used. It is recommended to set `UseForwardHeaders` to `true`.	`false`
`UseXForwardHeaders`	If `true`, the Messaging Service uses the `X-Forwarded-Host` and `X-Forwarded-Proto` headers to specify the protocol and host segments of the URL. If the `Forwarded` header reflects several forwarding steps, the value from the first step is used. If `UseForwardHeaders` is `true` and the request contains a `Forwarded` header, the `Forwarded` header is used, and `UseXForwardHeaders` is ignored.	`false`
`PathRegex`	A regular expression that can be used to match a string pattern in the path segment of internal URLs. The matching string is replaced with the value of `PathReplacement`. The regular expression search is executed on the URL after the `UseForwardHeaders` and `UseXForwardHeaders` replacement rules have been applied to it. `PathRegex` can be used in combination with either of the two attributes above.	None
`PathReplacement`	A string to use as the replacement of matches for `PathRegex`. The replacement string can consist of any combination of literal text and substitution placeholders. If `PathRegex` is not specified, `PathReplacement` is ignored.	Empty string

Note: The functionality for rewriting public URLs is common to the following BIOVIA applications and tools:

- Notebook
- Notebook Data Archive
- NotebookLoginService
- Desktop Connector Messaging Service

When more than one of these items are served from the same site, you do not need to add the `PublicUrlHelper` element to the `web.config` file of each item. You can define it once in a common `web.config` file that is inherited by each item.

Examples

Example 1: The Proxy Server Adds the Forwarded Header

The public URL to the Messaging Service is `https://public.company.com/dcservice`. The proxy server forwards the request to the internal address `http://internal.company.com:8080/dcservice` after adding the `Forwarded` header:

```
Forwarded: host=public.company.com;proto=https
```

The corresponding configuration in the `web.config` file for the Messaging Service is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <configSections>
    <section name="PublicUrlHelper"
type="Biovia.DesktopConnector.WebUtils.Server.PublicUrlHelperConfigSection"
/>
  </configSections>

  <PublicUrlHelper
    UseForwardedHeaders="true"
  />
```

Example 2: The Proxy Server Adds the X-Forwarded-Host and X-Forwarded-Proto Headers

The public URL to the Messaging Service is `https://public.company.com/dcservice`. The proxy server forwards the request to the internal address `http://internal.company.com:8080/dcservice` after adding the `X-Forwarded-Host` and `X-Forwarded-Proto` headers:

```
X-Forwarded-Host: public.company.com
X-Forwarded-Proto: https
```

The corresponding configuration in the web.config file should be:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <configSections>
    <section name="PublicUrlHelper"
type="Biovia.DesktopConnector.WebUtils.Server.PublicUrlHelperConfigSection"
/>
```

```
  </configSections>

  <PublicUrlHelper
    UseXForwardedHeaders="true"
  />
```

Example 3: The proxy Adds the Forwarded Header and Adds a Segment to the Application Root Path

The public URL to the Messaging Service is `https://public.company.com/dcservice`. The proxy server forwards the request to the internal address `http://internal.company.com:8080/notebookroot/dcservice` after adding the `Forwarded` header:

```
Forwarded: host=public.company.com;proto=https
```

The corresponding configuration in the web.config file should be:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <configSections>
    <section name="PublicUrlHelper"
type="Biovia.DesktopConnector.WebUtils.Server.PublicUrlHelperConfigSection"
/>
  </configSections>

  <PublicUrlHelper
    UseForwardedHeaders="true"
    PathRegex="^/notebookroot/"
    PathReplacement="/"
  />
```

Testing the configuration

You can test the connection against anonymous-access endpoints of the Desktop Connector Messaging Service REST API.

Using a browser or a REST client, send a request to the `GET v1/config` public endpoint of the Desktop Connector REST API. For example:

```
GET https://public.company.com/dc/api/v1/config
```

This returns a JSON or XML response body, depending on the default `Accept` header values of the client.

Check that the response contains the expected public URLs (partial response shown):

```
{
  ...
  "VersionUri": "https://public.company.com/dc/api/v1/version",
  ...
  "BrowserConnectPageUri":
"https://public.company.com/dc/api/BrowserConnect.aspx",
  ...
}
```

To double-check, verify that the `VersionUri` URL returns the current Messaging Service version, and that the `BrowserConnectPageUri` URL returns the "browser connect" web page when requested in a browser.

Appendix A:
Security Considerations

When you install and configure the Messaging Service, you should consider what settings are appropriate to your security requirements. The list below provides some guidance on Messaging Service settings and configuration options that relate to security.

Security Consideration	References	Notes
Server configuration	Configuring IIS	Observe the security options listed when you configure IIS.
Firewall, SSL, proxy servers	Using the Messaging Service Behind a Reverse Proxy Server that Rewrites URLs	When installing the Messaging Service behind a reverse proxy server, follow the instructions for configuring the Messaging Service appropriately.
SSL	Configuring the Messaging Service to Run Over HTTPS	If the Messaging Service runs on a new site, you should modify the site to run over HTTPS.
Proxy servers	Setting Up a Load-Balanced Environment	When using load balancing, ensure that individual Messaging Service nodes are only accessed via the load balancer, and that individual nodes are never accessed directly.
Roles and privileges	Configuring IIS	Configure server roles as appropriate.
Client security	Compatible Browsers lists the browsers supported by the Desktop Connector. No additional security configuration is required.	

If you have any questions about Desktop Connector and Messaging Service security, or if you encounter problems, contact Dassault Systèmes Customer Support by visiting https://www.3ds.com/support/.