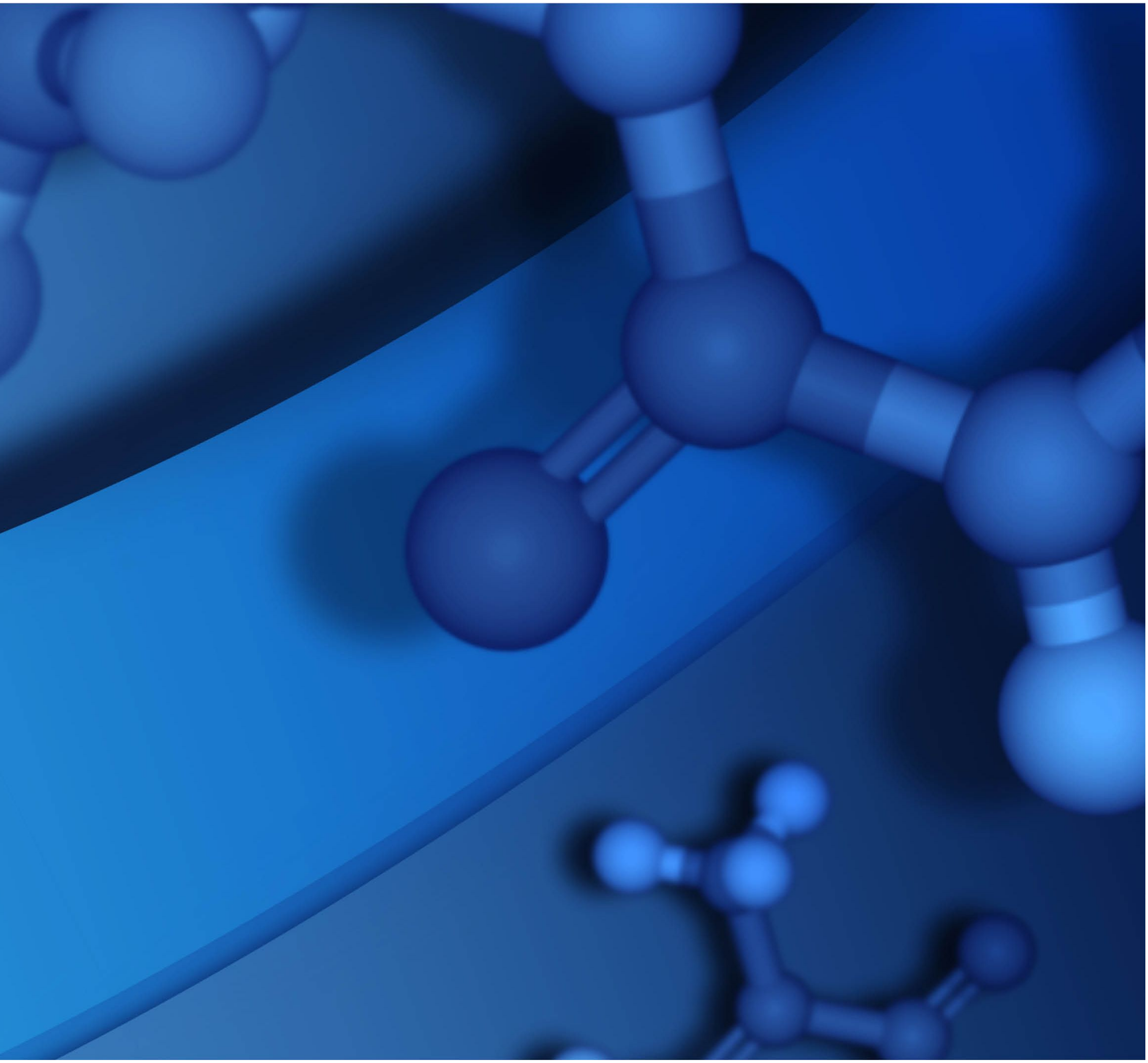
MESSAGING SERVICE ADMINISTRATION GUIDE

DESKTOP CONNECTOR 2019



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 BIOVIA Support, either by visiting https://www.3ds.com/support/ and clicking **Call us** or **Submit a request**, or by writing to:

BIOVIA 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.

About 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, getting clipboard objects, and so on. 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, a WebSocket service on the local server, or an NPAPI plugin.
- **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.

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.

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.

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 chooses the communication mode according to its requirements.

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 Desktop Connector 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.
- **Apple Safari (macOS):** The web application communicates with the Desktop Connector via an NPAPI plugin. Safari is likely to drop support for NPAPI plugins in the near future.

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.

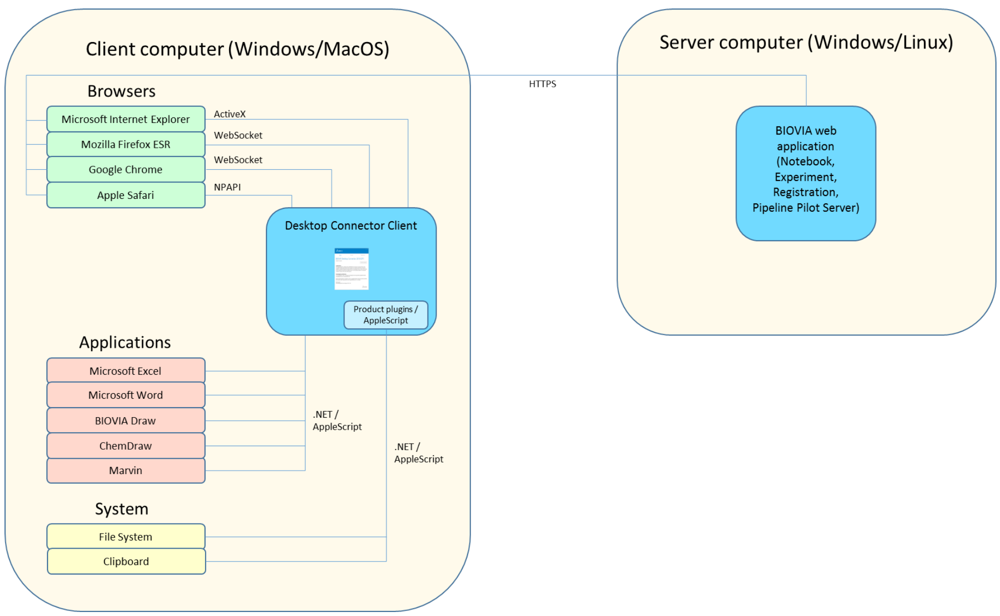
Notes:

- 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.

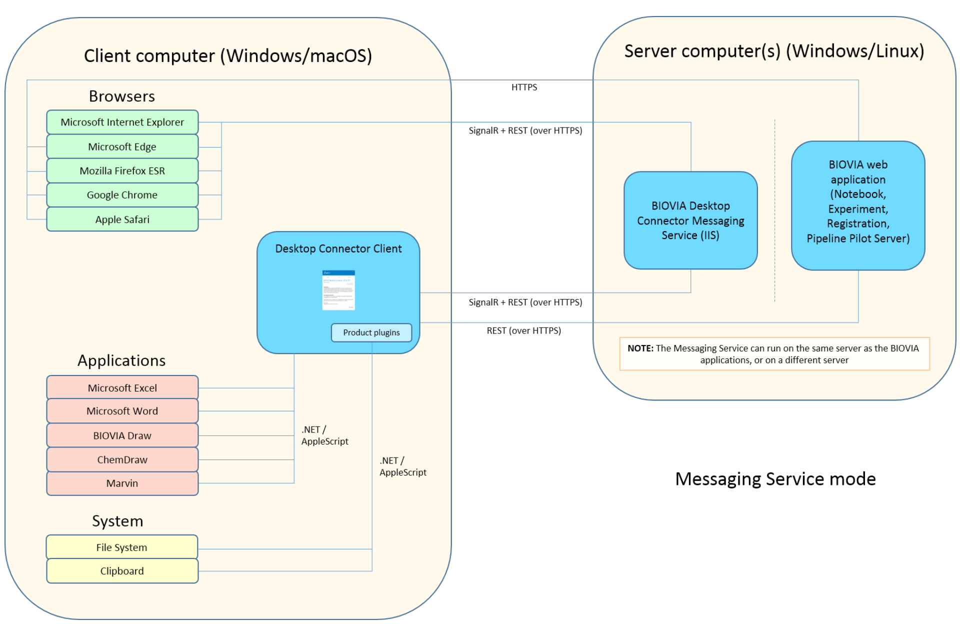
Chapter 3:
Architectural Overview

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

Legacy Modes



Messaging Service Mode



Messaging Service mode

Chapter 4:
Requirements

The Desktop Connector Messaging Service has the following requirements:

- **Windows Server 2012 R2 or 2016.** 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.**
- **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 Desktop Connector 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: Supported 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:

<table>
<tr><th>Software</th><th>Versions</th><th>Comments</th></tr>
<tr><td>Microsoft Excel</td><td>2016 (Office 365)
2013</td><td rowspan="2">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 applications only support the desktop versions of the Office applications when installed using the MSI installer.
BIOVIA does not support the mobile Office applications which are delivered as part of the Office 365 subscription, or the desktop versions of Office applications installed by the Click-to-Run installers unless otherwise stated.</td></tr>
<tr><td>Microsoft Word</td><td>2016 (Office 365)
2013</td></tr>
<tr><td>BIOVIA Draw</td><td>2019
2018</td><td>Enterprise Edition (EE) only.</td></tr>
<tr><td>ChemAxon MarvinSketch</td><td>18
17</td><td>Windows only.</td></tr>
<tr><td>PerkinElmer ChemDraw</td><td>17
16</td><td>-</td></tr>
</table>

Mac:

Software	Versions	Comments
Microsoft Excel for Mac	2016	-
Microsoft Word for Mac	2016	-
PerkinElmer ChemDraw	17 16	-

Chapter 5:
Installing the Desktop Connector 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. Run the installer `BIOVIA Desktop Connector Service.msi`.

   > **Note:** You **must** run the installer as an administrator.

2. 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.

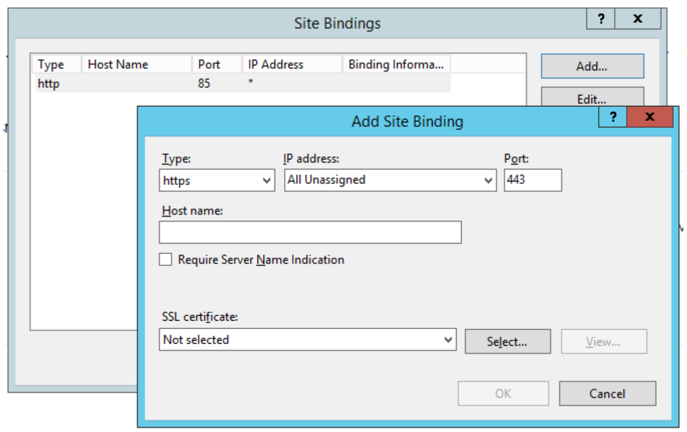
   d. Click **Finish** in the last stage of the wizard to complete installation.

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

Chapter 6:
Configuring the Desktop Connector 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
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 `3`.

  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)

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

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.

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.

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.