Web Gateway Guide

Version 2024.1
2024-05-02

InterSystems Corporation 1 Memorial Drive Cambridge MA 02142 www.intersystems.com

Web Gateway Guide
InterSystems IRIS Data Platform Version 2024.1 2024-05-02

For Support questions about any InterSystems products, contact:

InterSystems Worldwide Response Center (WRC)
Tel: +1-617-621-0700
Tel: +44 (0) 844 854 2917
Email: support@InterSystems.com

Table of Contents

1

The Web Gateway: Serve InterSystems Web Applications and REST APIs to a Web Client

An InterSystems IRIS® *web application* consists of code which provides content dynamically to a web client (usually a web browser) in response to a request. The InterSystems *Web Gateway* makes this possible: it is a software utility that mediates the connection between your web server and the InterSystems IRIS instance or instances which host your web application code. The Web Gateway supports HTTP, HTTPS, and WebSocket protocols, and it provides capabilities such as load balancing and failover for your application traffic.

The Web Gateway:

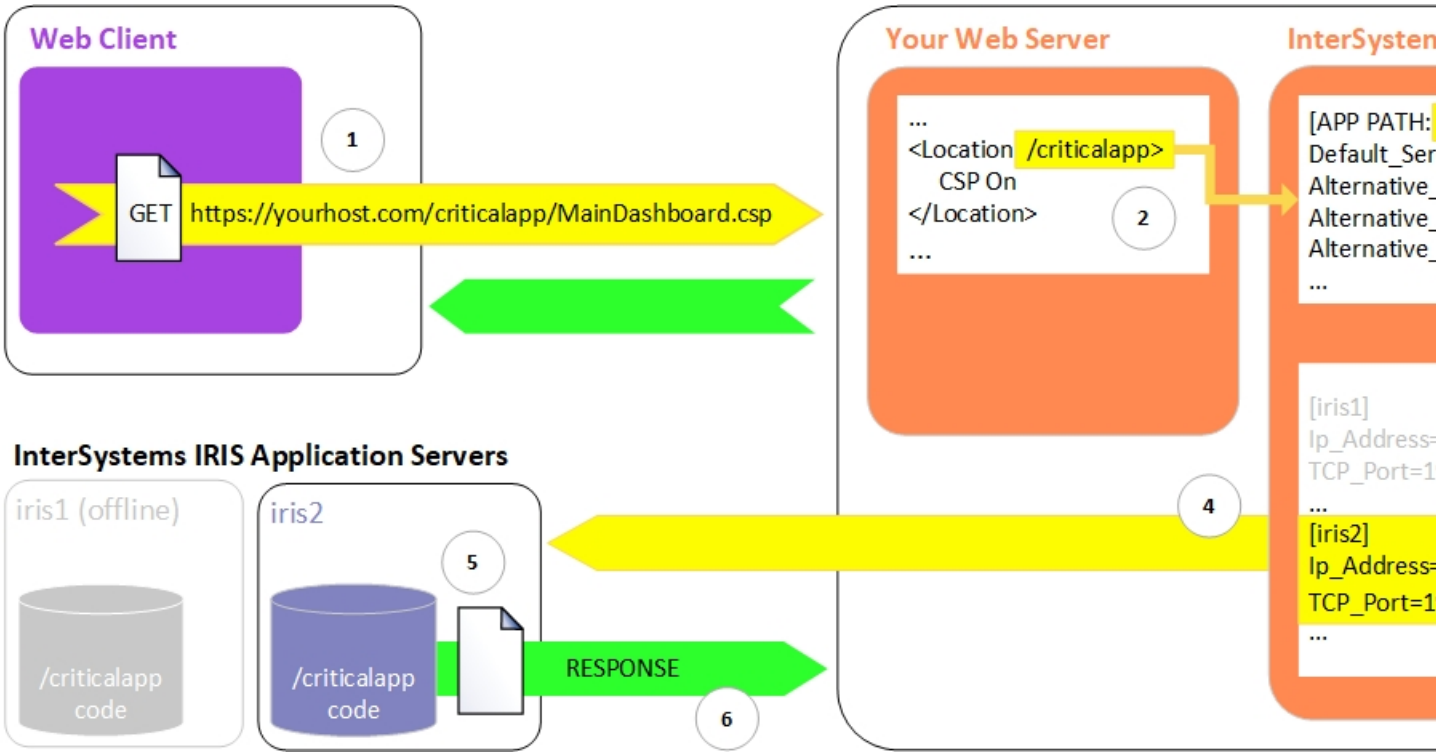
- extends the functionality of your *web server* so that the web server can recognize and handle requests for InterSystems web applications.
- manages server access profiles to connect with the *application server* processes within each of your InterSystems IRIS instances.
- routes a request for an application to the InterSystems IRIS instance which hosts the application, based on application access profiles between certain request paths and certain InterSystems IRIS application servers.

Note: The process within an InterSystems IRIS instance which invokes application code in response to incoming requests is sometimes referred to as the CSP Server or CSP Engine, in reference to the legacy InterSystems web application framework. However, this process responds to requests for all InterSystems IRIS web applications—not merely CSP applications. For this reason, the Web Gateway documentation uses the term “application server” when it is necessary to refer to this process specifically. Elsewhere, the documentation treats the entire InterSystems IRIS instance as the effective target of a Web Gateway connection.

Web applications provide access to several important InterSystems IRIS system utilities, such as the Management Portal. To use these, you must connect InterSystems IRIS to a web server through a Web Gateway. See Serve the Management Portal (and Other System Applications) Using Your Web Server.

1.1 How the Web Gateway Routes InterSystems Web Application Requests

The following diagram visualizes the way a request for a web application travels from a client (such as a web browser) to an InterSystems IRIS instance where the web application is running.



The process can be summarized as follows:

1. The client sends a request to a certain URL path. In the example, the client sends an HTTP GET request to `https://yourhost.com/criticalapp/MainDashboard.csp`.
2. The web server receives the request and checks its configuration to determine whether it should invoke the Web Gateway for that request. The diagram demonstrates this with an excerpt from the configuration file for a hypothetical Apache web server. In the example, the web server configuration instructs the web server to invoke the Web Gateway to handle any request within the `/criticalapp` path.
3. The Web Gateway checks its configuration to determine which InterSystems IRIS application server should receive the request. The diagram demonstrates this with excerpts from the Web Gateway's hypothetical CSP.ini configuration file. In our example, the application access profile for the `/criticalapp` path instructs the Web Gateway to route requests within that path to the application server process for the InterSystems IRIS instance `iris1` by default, or (if `iris1` is unavailable) to the alternative application server for the instance `iris2`.
4. The Web Gateway transmits the request over a secured TCP connection to the host and superserver port for the appropriate application server. As presented in the diagram, the Web Gateway stores these connection details in a server access profile for each application server. In the example, `iris1` is currently offline, so the Web Gateway sends the request to the alternative server `iris2`.
5. The application server process for the InterSystems IRIS instance calls the appropriate application code based on the settings for the web application within the InterSystems IRIS instance. The application code produces a response.

6. The response travels back through the Web Gateway to the web server. The web server transmits the response to the client.

1.1.1 Structure of an InterSystems Web Application URL

The Web Gateway allows you to serve InterSystems IRIS web applications at URLs which follow the following format:

[protocol]://[hostname]:[port]/[instancePrefix]/[appPath]/[fileOrQuery]

URL Part	Description
[protocol]	`http` or `https`, depending on whether or not you have configured your web server to use TLS. InterSystems strongly recommends the use of TLS.
[hostname]	The IP address or DNS name where your web server is available. When the web server is on the same machine as the client, this is usually `localhost`.
[port]	The port number over which your web server is listening for requests. You do not need to specify a port number unless your web server is listening for requests on a port other than the standard ports for HTTP (80) or HTTPS (443).
[instancePrefix]	A string which uniquely identifies one of your InterSystems IRIS application servers. If you do not need to discriminate between multiple InterSystems IRIS instances, you can configure the Web Gateway to serve web applications at URLs which omit this portion of the path, as in the preceding example. However, if you serve multiple InterSystems IRIS instances using a single web server and you must access a web application unique to one instance (such as an instance's Management Portal), you must configure your web server and Web Gateway to route requests to the instance using this identifier as a prefix to the application path. This identifier is the *CSPConfigName* for the instance. By default, an instance's *CSPConfigName* is its instance name, in all lowercase characters. However, it can be configured. If the application in the preceding example were exclusively available on the InterSystems IRIS application server named `iris2`, then you could configure the /iris2 path within the web server and Web Gateway configurations to serve the application at the following URL: https://yourhost.com/iris2/criticalapp/MainDashboard.csp
[appPath]	The relative path unique to the application within each InterSystems IRIS application server. When you define an application within an InterSystems IRIS application server, this is the application's **Name**.
[fileOrQuery]	*Optional*. Any combination of subordinate path, file name, and query parameters which the application may use to process the request.

1.2 Set Up a Web Gateway for Your System

You can set up a Web Gateway connection in several ways. As the preceding diagram suggests, you can deploy the Web Gateway alongside your web server on a separate machine, remote from your clients and your InterSystems IRIS instances. Alternatively, the web client, web server, and Web Gateway can all reside on the same machine. You can install the Web

Gateway on-premises; alternatively, you can deploy a `webgateway` container from a Docker image available in the InterSystems Container Registry. Each of these `webgateway` container images contain a web server (Apache or nginx) which is pre-configured with the Web Gateway extension.

Set Up a Web Gateway for Your System provides an overview of the entire setup process, regardless of your use case.

1.3 Manage a Web Gateway Connection

Regardless of how you deploy the Web Gateway, it provides a standard set of interfaces for managing connections between web clients and the InterSystems IRIS instances which host your web applications.

The Web Gateway management pages provide a graphic user interface for configuring default parameters, defining connections to your InterSystems IRIS instances, monitoring and testing those connections, defining routing behavior for your applications, and more. Web Gateway configuration information is stored in the CSP.ini. In the majority of cases, InterSystems recommends the use of the Web Gateway management pages or Web Gateway Registry methods to configure the Web Gateway. However, for containerized deployments, you can automate and synchronize the configuration of one or more `webgateway` containers upon deployment by modifying CSP.ini files directly using the CSP.ini merge feature.

The InterSystems IRIS Web Gateway Registry enables an InterSystems IRIS instance to monitor and manage its connection to the Web Gateway programmatically. All web server installations and Web Gateway installations are registered with InterSystems IRIS as they connect. Using the Gateway Registry, InterSystems IRIS code can interact with the Web Gateways to which the instance is connected, enabling it to read and write configuration details, monitor the system status, and audit the Web Gateway Event Log.

2

Access the Management Portal and Other Built–in Web Applications Using Your Web Server

InterSystems IRIS® provides access to several important system utilities—such as the Management Portal—through built-in web applications. If you want to use these web-based utilities, you must connect InterSystems IRIS to a web server through a Web Gateway.

In many cases, the InterSystems IRIS installer can automatically configure a new or upgraded instance to serve its built-in web applications using your web server. If you want to serve your own custom web applications as well, the installer's automatic procedure provides a basic configuration that you can easily amend to suit your needs. See the section Connect Your Web Server Automatically for details.

If you want to manually set up your web server to route requests for an InterSystems IRIS instance's built-in web applications, you must follow the same procedure as you would for any other InterSystems IRIS web application. The page Set Up a Web Gateway for Your System provides a general outline for this procedure. On this page, the section Connect Your Web Server Manually provides supplementary guidance specific to these built-in utilities.

This page concludes with instructions for disabling and removing the Private Web Server (PWS), which InterSystems included alongside versions of InterSystems IRIS prior to 2023.2.

2.1 The Management Portal URL

The Management Portal is one of the most important web application utilities built into the InterSystems IRIS data platform. Because of its importance and its ubiquity, the Management Portal is the subject of several examples on this page.

When your system is configured as specified on this page, the URL for an instance's Management Portal home page has the following form, using the <baseURL> for the instance:

```
https://<baseURL>/csp/sys/UtilHome.csp
```

2.2 For New Installations

If you are installing a new instance of InterSystems IRIS and your system meets the installation conditions for automatic configuration, the installer asks you if you want to configure your web server automatically as part of the installation process. To automatically configure your web server, simply select the option to do so when prompted.

On Windows, the stand-alone Web Gateway installation methods and the **Custom** setup type do not ask you if you want to configure your web server automatically; these methods perform the automatic configuration silently, by default. If you want to opt out of automatic configuration during a **Custom** installation, clear the **Web Gateway** > **CSP for IIS** installation item.

The automatic procedure configures the web server and the Web Gateway to route requests to your new InterSystems IRIS instance as described in Automatic Configuration Behavior. Review this behavior; under certain circumstances, you must perform some configuration steps manually.

If you complete an installation without automatically configuring your web server and then wish to do so subsequently, use the installer to modify the installation. On Windows, be sure to include the **Web Gateway** > **CSP for IIS** installation item. As noted in Automatic Configuration Behavior, in this case you must manually set the CSPSystem credentials for the instance within the Web Gateway's server access profile for the instance and then restart IIS.

For stand-alone Web Gateway installations, the installer's automatic configuration procedure can add the Web Gateway module to your web server configuration automatically. However, you must configure connections to your InterSystems IRIS instances and route requests for your web applications manually.

If your system does not meet the installation conditions for automatic configuration, connect your web server manually.

Note: After the installation is complete, restart your web server to ensure that all configuration changes take effect.

2.3 For Upgrades

Prior to version 2023.2, all InterSystems IRIS installations included a Private Web Server (PWS), a minimal build of Apache httpd which was configured to handle requests for the instance's Management Portal and other built-in system web applications.

If you are upgrading from a version of InterSystems IRIS prior to 2023.2 and your system meets the requirements for automatic configuration, the installer provides you the option to automatically configure your instance to connect to a web server which is external to your InterSystems IRIS installation. Simply select the option to do so when prompted.

The automatic procedure configures the web server and the Web Gateway to route requests for built-in system web applications to your upgraded InterSystems IRIS instance as described in Automatic Configuration Behavior. Review this behavior; under certain circumstances, you must perform some configuration steps manually.

Videos which demonstrate how to perform this procedure are available here: https://learning.intersystems.com/course/view.php?id=2333

Important: The PWS is not suitable for serving web applications in production, or for use outside of a secured environment. Beginning with version 2023.2, InterSystems stopped installing a PWS with new installations of InterSystems IRIS data platform products (except Community Editions and other evaluation distributions). However, the PWS files remain in place when you upgrade InterSystems IRIS to version 2023.2; you can remove these files manually. Effective with the first EM release of 2026, upgrading any InterSystems IRIS instance will remove all PWS data for the instance.

Though the installer's automatic configuration procedure can set up your InterSystems IRIS instance to use an external web server to serve its built-in applications, it cannot migrate custom configurations from the PWS and its associated Web Gateway to the external web server and its associated Web Gateway. You must apply these customizations to the external web server and its Web Gateway manually. For example, to serve a custom application using the external web server, you must manually invoke the Web Gateway for requests to that application and you must create a Web Gateway application access profile to direct those requests to the appropriate InterSystems IRIS instance or instances.

CAUTION: When you migrate an instance of InterSystems Caché® or InterSystems Ensemble® to InterSystems IRIS on Windows, the conversion process removes the Windows service for the instance's Apache PWS. If the installer cannot configure IIS during the conversion, the InterSystems IRIS instance will not be connected to a web server and built-in web applications such as the Management Portal will be unavailable.

If you complete an installation without automatically configuring your web server and then wish to do so subsequently, use the installer to modify the installation. On Windows, be sure to include the **Web Gateway** > **CSP for IIS** installation item. As noted in Automatic Configuration Behavior, in this case you must manually set the CSPSystem credentials for the instance within the Web Gateway's server access profile for the instance and then restart IIS.

After the automatic configuration procedure, the PWS is disabled. However, it is not uninstalled; the PWS configuration retains any customizations you have made. Once you have migrated any custom configurations and ensured that your web server is successfully routing requests for all desired web applications, you can delete the PWS for that instance.

If you complete an installation without automatically configuring your web server and then wish to do so subsequently, use the installer to modify the installation. On Windows, be sure to include the **Web Gateway** > **CSP for IIS** installation item. As noted in Automatic Configuration Behavior, in this case you must manually set the CSPSystem credentials for the instance within the Web Gateway's server access profile for the instance and then restart IIS.

After the automatic configuration procedure, the PWS is disabled. However, it is not uninstalled; the PWS configuration retains any customizations you have made. Once you have migrated any custom configurations and ensured that your web server is successfully routing requests for all desired web applications, you can delete the PWS for that instance.

Note: After you have completed the upgrade and any post-upgrade modifications to the web server and Web Gateway, restart your web server to ensure that all configuration changes take effect.

2.4 Connect Your Web Server Automatically

2.4.1 When Is Automatic Configuration Possible?

If you are installing an InterSystems IRIS instance and you choose a Setup Type that includes the Web Gateway, you can automatically configure your web server to serve requests through a Web Gateway to your instance under the conditions described in the following list. All methods for installing the Web Gateway as a stand-alone component can also configure the web server to include the Web Gateway, provided these conditions apply. However, you must configure the connection to each InterSystems IRIS application server manually.

Automatic web server configuration is possible if:

1. You are using:
   - Apache httpd on UNIX®, Linux, or macOS
   - Microsoft Internet Information Services (IIS) on Microsoft Windows.
2. You have already installed the web server within the local file system prior to performing the installation. (Or in the case of IIS, you have enabled it.) For guidance installing Apache on your UNIX® or Linux system, refer to the following

Developer Community article: https://community.intersystems.com/post/how-install-apache-iris-supported-operating-systems

Note: On macOS systems, you must have installed a copy of Apache httpd using the Homebrew package manager (https://brew.sh/). The installer does not support automatic configuration for the Apache httpd installation which is distributed with macOS. For all other platforms, the installer supports the web server provided by the platform.

3. You have installed the web server in its default installation location.
4. The web server is running.

2.4.2 Automatic Configuration Behavior

Important: The installer's automatic configuration procedure cannot modify the port over which the web server is listening, even when a custom port is specified during installation. Specifying a custom port during installation only modifies the WebServerPort parameter for the InterSystems IRIS instance. In all other cases, the installer assumes that the web server is listening over the default HTTP port, 80 (8080 for macOS).

To access InterSystems IRIS web applications, you must use the port which is specified by your web server configuration.

Except where noted, the installer's automatic configuration procedure does the following:

1. If the Web Gateway module is not part of your web server configuration, the installer adds it and sets appropriate permissions for the directory which contains the Web Gateway binaries.
2. In the web server configuration, it enables the Web Gateway as the handler for the following relative URL path:
   - /<instanceName>

   where <*instanceName*> is the name of the instance you are installing or upgrading, in all lowercase characters.

   Important: If the installer performs the automatic configuration procedure on an Apache web server for an instance which has the same name as an instance which was formerly configured to use the web server, it overwrites any manual edits to the `ApacheCSP-SECTION-<instanceName>` section which it adds to the configuration file (httpd.conf), restoring the section's original contents and deleting any customizations. Comments mark the start and end of this section.
3. In the web server configuration, it enables the handler for the following relative URL paths:
   - /api
   - /csp
   - /isc
   - /oauth2
   - /ui

   Important: Each time the installer performs the automatic configuration procedure on an Apache web server, it overwrites any manual edits to the `ApacheCSP-SECTION` which it adds to the configuration file (httpd.conf), restoring the section's original contents. Comments mark the start and end of this section; if you want to customize your Apache configuration, you must do so by adding directives elsewhere in the file.

Note: For Windows systems, you must perform additional IIS configuration to enable a fully functional connection to VS Code (which is facilitated by the instance's `/api/atelier` application).

4. For **Custom** installations on UNIX®, Linux, and macOS systems: it updates the port number listed in the Apache web server configuration to the port number you specified in the installation wizard.
5. On UNIX®, Linux, and macOS systems: it adds a directive to the Apache web server configuration that redirects requests within the relative path /csp/docbook/ to the corresponding page on the InterSystems documentation site. This is required for links to documentation from within the Management Portal.

   Note: On Windows systems, you must configure IIS to redirect requests from /csp/docbook/ to the InterSystems documentation site manually.

6. On RHEL systems with SELinux: it configures appropriate read and write access to the Web Gateway configuration and log files and allows outgoing network connections from the Apache web server.
7. In the Web Gateway configuration, it creates a server access profile for the InterSystems IRIS instance you are installing or upgrading, using the superserver port and the CSPSystem account credentials you specified. This server access profile allows the Web Gateway to connect to your new instance. The server access profile **Server Name** is the name of the instance, in all lowercase characters.

   Note: If you previously installed an instance of InterSystems IRIS 2023.2 without configuring a web server and then allow the installer to run the automatic configuration procedure by running the installer's modify option, you must set the CSPSystem account credentials for your instance within its server access profile manually. Then, restart IIS.

8. In the Web Gateway configuration, it creates an application access profile for the following relative application path:

   - /<instanceName>

   where <*instanceName*> is the name you specified for the instance you are installing or upgrading, in all lowercase characters.

   The application access profile specifies the new server access profile as the default server for all requests sent to this application path.
9. In the Web Gateway configuration, it creates application access profiles for the following relative application paths, if they do not already exist:

   - / (root)
   - /csp

   These application access profiles specify the new server access profile as the default server for all requests sent to these application paths.

   If these application access profiles already exist within the Web Gateway configuration, the installer updates them so that they specify the new server access profile as the default server for all requests to these application paths.
10. It sets the value of the InterSystems IRIS instance's WebServerPort parameter to 80 (8080 on macOS) and it sets the value of the WebServerURLPrefix to the instance name in all lowercase characters. (If you specify a custom port number during installation, the installer sets the WebServerPort parameter to that number instead.) An InterSystems IRIS instance uses these parameters to connect to InterSystems Studio.

Note: If you are installing the Community Edition, you must update the WebServerPort parameter manually to match your web server's port number.

You must also update the WebServerPort parameter manually if you are upgrading or modifying an existing InterSystems IRIS instance and the instance previously served its web applications using a different port (or did not serve web applications at all).

11. It sets the value of the InterSystems IRIS instance's WebServer parameter to false (`0`), preventing the instance from starting its Private Web Server upon instance startup. This change effectively disables the Private Web Server.

    Important: If you are installing the Community Edition, the installer does not set the WebServer parameter. You must disable the instance's Private Web Server manually.

12. On Windows systems, it configures the relevant InterSystems IRIS Server Manager parameters which the InterSystems IRIS launcher requires to construct valid URLs for web application utilities using the launcher—that is, the Management Portal, the class reference, and documentation. These parameters are also necessary to connect to InterSystems Studio. The installer sets the Server Manager's **Web Server Port** parameter to 80 (8080 on macOS).

    Note: If you are upgrading or modifying an existing InterSystems IRIS instance and the instance previously served its web applications using a different port (or did not serve web applications at all), you must update the Server Manager's **Web Server Port** parameter to match your web server's port number manually.

The automatic configuration procedure ensures that you can access the built-in applications for a specific InterSystems IRIS instance at URLs which include the instance name as a prefix before the application path, in all lowercase characters. You can easily configure the prefix you use to specify an instance.

In addition, the automatic configuration procedure makes built-in applications for the most recently configured InterSystems IRIS instance available at URLs which omit the instance prefix. This means that if only one instance is configured, that instance's Management Portal is available at the simplified URL`<protocol>://<hostname>/csp/sys/UtilHome.csp`.

Note: If you have a system with multiple instances configured and you uninstall the most recently configured instance, application access profiles for / (root) and /csp remain associated with the server access profile for the uninstalled instance. To serve requests at those paths, you must manually update the profiles for the / (root) and /csp paths to direct requests to a different default application. You can then remove the obsolete server access profile for that instance.

2.4.2.1 Automatic Configuration Example

If you installed a single InterSystems IRIS instance named IRISserv1 on a Linux system running Apache httpd, the automatic configuration procedure would add the following directives to the Apache httpd.conf file:

```
#### BEGIN-ApacheCSP-SECTION ####
LoadModule csp_module_sa "/opt/webgateway/bin/CSPa24.so"
CSPModulePath "/opt/webgateway/bin/"
CSPConfigPath "/opt/webgateway/bin/"

<Location "/csp/">
     CSP On
</Location>
<Location "/api/">
     CSP On
</Location>
<Location "/oauth2/">
     CSP On
</Location>
<Location "/isc/">
     CSP On
</Location>
<Location "/ui/">
     CSP On
```

```
</Location>

<Directory "/opt/webgateway/bin/">
     AllowOverride None
     Options None
     Require all granted
     <FilesMatch "\.(log|ini|pid|exe)$">
          Require all denied
     </FilesMatch>
</Directory>

Redirect /csp/docbook/ http://docs.intersystems.com/irislatest/csp/docbook/
#### END-ApacheCSP-SECTION ####
#### BEGIN-ApacheCSP-SECTION-IRISSERV1 ####
# Note: IRISSERV1 reinstallation or upgrade may replace this section.
<Location /irisserv1>
     CSP On
</Location>
Redirect /irisserv1/csp/docbook/ http://docs.intersystems.com/irislatest/csp/docbook/
#### END-ApacheCSP-SECTION-IRISSERV1 ####
```

The Web Gateway configuration would feature an IRISSERV1 server access profile, and application access profiles for the paths / (root), /csp, and /irisserv1. All three application access profiles would specify IRISSERV1 as the default application server.

As a result of this configuration, the Management Portal home page for IRISserv1 would be available at either `http://localhost/irisserv1/csp/sys/UtilHome.csp` or the simpler `http://localhost/csp/sys/UtilHome.csp`.

If you subsequently installed a second InterSystems IRIS instance named IRISserv2 to the same system, the installer would restore the content of the httpd.conf file's `ApacheCSP-SECTION`, overwriting any manual changes. It would then append the following directives, after the `ApacheCSP-SECTION-IRISSERV1` code block:

```
#### BEGIN-ApacheCSP-SECTION-IRISSERV2 ####
# Note: IRISSERV2 reinstallation or upgrade may replace this section.
<Location /irisserv2>
     CSP On
</Location>
Redirect /irisserv2/csp/docbook/ http://docs.intersystems.com/irislatest/csp/docbook/
#### END-ApacheCSP-SECTION-IRISSERV2 ####
```

The automatic configuration procedure would add an IRISSERV2 server access profile to the Web Gateway configuration. It would also add an application access profile for the path /irisserv2. The application access profiles for /irisserv2, / (root), and /csp would specify IRISSERV2 as the default application server; the application access profile for the path /irisserv1 would remain unchanged.

As a result of these changes, the Management Portal home page for IRISserv1 would only be available at `http://localhost/irisserv1/csp/sys/UtilHome.csp`. The Management Portal home page for IRISserv2 would be available at `http://localhost/irisserv2/csp/sys/UtilHome.csp` or `http://localhost/csp/sys/UtilHome.csp`.

2.5 Connect Your Web Server Manually

The procedure for manually configuring a web server to route requests for the web-based utilities built into InterSystems IRIS (such as the Management Portal) is essentially the same as the procedure for configuring a custom InterSystems IRIS web application. This procedure is outlined in Set Up a Web Gateway for Your System.

This section provides supplemental guidance specific to these built-in applications.

If you are upgrading or re-configuring a version of InterSystems IRIS prior to 2023.2, you can disable and delete the private web server once you have finished configuring your web server to replace it.

2.5.1 Specify the Application Paths You Need

For any InterSystems IRIS system applications you wish to enable, you must do the following:

1. Configure your web server to invoke the Web Gateway to handle all expected requests for the corresponding application path, as well as for any subordinate paths within the application path.
2. Configure a Web Gateway server access profile for your InterSystems IRIS instance
3. Configure a Web Gateway application access profile which specifies that server access profile as the default application server for requests sent to the application path (and subordinate paths).

For a list of the system web applications installed with InterSystems IRIS, see Built-In Applications. InterSystems strongly recommends configuring your web server and Web Gateway to handle requests for all these built-in web utilities. These web applications use endpoints with the following base paths:

- /api
- /csp
- /isc
- /ui
- /oauth2

Because a web server configuration and an application access configuration both apply rules of inheritance, the most expedient approach would be to configure requests to these five base paths; this is the approach taken by the installer's automatic configuration procedure.

If you are serving multiple InterSystems IRIS instances from one Web Gateway and you wish to access built-in utilities for instances independently, you must configure your web server and your Web Gateway so that requests sent to paths which begin with each instance's *CSPConfigName* route to that instance. You can configure this parameter for an instance, enabling a custom string to serve as this instance prefix. By default, this parameter is the instance's name in all lowercase characters. Therefore, the most expedient approach would configure requests to the path /<instanceName> for each instance, where <*instanceName*> is the instance's name in all lowercase characters. This is the approach taken by the installer's automatic configuration procedure.

2.5.2 Route All Necessary Requests for Each Path

You can configure your web server to invoke the Web Gateway to handle all requests sent to a given path, or only requests for files with certain file type extensions.

You can serve the Management Portal by routing requests exclusively for the following file types:

```
.csp .cls .zen .cxw .jpg .gif .png .svg .css .js
```

However, REST APIs must support receiving requests at endpoints which do not specify a file (and therefore do not specify a file type). For such applications, it is not possible to invoke the Web Gateway for requests based on the file type, and you must enable the Web Gateway as the handler for all requests at the path level.

2.5.3 Redirect Documentation Links

When pages within the Management Portal include links to relevant documentation, those links refer to resources within the path <instancePrefix>/csp/docbook for the instance, where <*instancePrefix*> is the instance name in all lowercase characters. On Windows, the InterSystems IRIS Launcher's **Documentation** link also routes to this path.

For these links to function, your web server must redirect requests to this path, sending them to the equivalent URL on the InterSystems documentation web site.

Important: For all installations on Windows systems, you must configure IIS to redirect documentation links manually as described in For Microsoft IIS

The destination URL for documentation links varies depending on the InterSystems product you are using. To find the correct product identifier for your instance, access any documentation page for the InterSystems product and version you are using and check your browser's address bar. URLs which specify the `latest` version of a product (`irislatest`, `healthconnectlatest`, and so on) always provide documentation for the most recently released version of the product.

The following sections describe methods to redirect documentation links for Apache httpd and Microsoft IIS. The best method for your system may vary; refer to your web server documentation to explore the options available to you.

2.5.3.1 For Apache

For an Apache web server, add a `Redirect` directive to the web server's configuration for each instance you want to configure which redirects the <instancePrefix>/csp/docbook path for the instance to the analogous URL on the InterSystems documentation website. (By default, <*instancePrefix*> is the instance name in lowercase characters.)

For example, you can redirect documentation links for a InterSystems IRIS instance named IRISinst2 by adding the following `Redirect` directive to your web server's configuration (in the httpd.conf file, or files included therein):

```
Redirect /irisinst2/csp/docbook/ http://docs.intersystems.com/irislatest/csp/docbook/
```

2.5.3.2 For Microsoft IIS

For a Microsoft IIS web server, perform the following steps:

1. Ensure that **HTTP Redirection** for IIS is enabled:
   a. Open the **Windows Features** manager by searching for **Turn Windows features on or off** or by opening the **Control Panel** and selecting **Programs > Programs and Features > Turn Windows features on or off**.
   b. Select **Internet Information Services** > **World Wide Web Service** > **Common HTTP Features** > **HTTP Redirection**, if it is not already selected.
   c. Select **OK**.
2. Within the installation directory for your instance, create a /csp/csp/docbook directory.

   IIS requires that each application path correspond with a physical path, regardless of whether the web server serves static files. When the installer automatically configures IIS, the path for your InterSystems IRIS instance is mapped to the instance's /csp directory, and not instance's base installation directory. If this is the case, create an <installDir>/csp/csp/docbook directory, where <*installDir*> is the installation directory for your instance. The application will associate requests for /csp/docbook with this directory.
3. In the **Connections** panel, expand your localhost connection, then **Sites**, then the application which corresponds with your instance. Within that application, select **csp** > **docbook**.
4. On the **docbook Home** page, select (double-click) **HTTP Redirect**.
5. On the **HTTP Redirect** page:
   - For the field captioned **Redirect requests to this destination:** provide the URL for the online documentation page, appending the string `$S$Q` (to retain all suffixes and query parameters at the end of the URL). For example:

     ```
     https://docs.intersystems.com/irislatest/csp/docbook$S$Q
     ```
   - Select **Redirect all requests to exact destination (instead of relative to destination)**.

6. In the **Actions** panel, select **Apply**.
7. Restart IIS for changes to take effect.

2.5.4 Windows Only: Update InterSystems IRIS Server Manager

On Windows systems, the InterSystems IRIS launcher uses the server connection details specified within the InterSystems IRIS Server Manager to direct users to the instance's web-based utilities. To enable the InterSystems IRIS launcher for an instance, modify the server connection details in the InterSystems IRIS Server Manager to match your new web server configuration. See Define a Remote Server Connection.

2.5.5 Windows Only: Configure IIS to Enable VS Code

On Windows systems which use IIS, you must configure IIS to allow VS Code to connect with your InterSystems IRIS instance. The necessary IIS configurations are described in the sections which follow. For detailed instructions on configuring IIS, refer to the IIS documentation (https://learn.microsoft.com/en-us/iis/get-started/introduction-to-iis/iis-web-server-overview).

2.5.5.1 Disable the WebDAV Module

The IIS WebDAV module (if it is installed) interferes with the Web Gateway when both are enabled to handle communication for an InterSystems IRIS instance's `/api/atelier` application. This application provides the connection between the instance and VS Code.

To use VS Code with an InterSystems IRIS instance, you must remove the WebDAV **Handler Mapping** and disable the WebDAV **Module** for the relevant path. Depending on your web server configuration, this relevant path may be the instance prefix path, /<instancePrefix>/api, or /<instancePrefix>/api/atelier.

You can perform these configurations using the Internet Information Services Manager, or by editing the applicationHost.config file to include `<remove>` directives within the `<location>` directive block for the relevant path (*{path}*) as in the following example:

```
<location path="{path}">
  <system.webServer>
    <modules>
      <remove name="WebDAVModule" />
    </modules>
    <handlers>
      <remove name="WebDAV" />
    </handlers>
  </system.webServer>
</location>
```

After making changes to the IIS configuration, restart IIS to ensure they take effect.

2.5.5.2 Enable the WebSockets Feature (for Debugging)

The debugging tool in VS Code requires a WebSockets connection to the InterSystems IRIS instance. Ensure that the IIS **WebSocket Protocol** feature is enabled by performing the following steps:

1. Open the **Windows Features** manager by searching for **Turn Windows features on or off**, or by opening the **Control Panel** and selecting **Programs** > **Programs and Features** > **Turn Windows Features on or off**.
2. Select **Internet Information Services** > **World Wide Web Service** > **Application Development Features**.
3. Select **WebSocket Protocol**, if it is not already selected.
4. Select **OK**.
5. Restart IIS to ensure all changes take effect.

2.5.5.3 Allow Double Escaping (to Access Certain Packages)

For any file with a name which begins with the `%` (percent) character followed by two hexadecimal digits (that is, numerals between `0` and `9` or letters between `a` and `f`), IIS interprets these first three characters as an encoded hexadecimal character by default. As a result, VS Code cannot view or edit such a file.

If you must view or edit such a file, you must configure IIS **Request Filtering** to **Allow double escaping**, either globally or for the specific IIS application location which corresponds with your instance. You can modify this setting by using the IIS Manager or the command line interface (see https://learn.microsoft.com/en-us/iis/manage/configuring-security/configure-request-filtering-in-iis) or by setting the attribute `allowDoubleEscaping="true"` for the appropriate `<requestFiltering>` element within the IIS configuration file (see https://learn.microsoft.com/en-us/iis/manage/configuring-security/use-request-filtering).

2.6 For Upgrades from Versions Prior to 2023.2: Disable and Remove the Private Web Server

If you have upgraded to this version of InterSystems IRIS from InterSystems IRIS 2023.2 and you have re-configured your instance to use an external web server, you can disable and (optionally) remove the instance's private web server (PWS). To do so, perform the following steps:

1. Ensure that:
   - Your external web server and Web Gateway successfully route requests for all desired web applications.
   - Requests to the /csp/docbook application redirect to the documentation web site.
   - On Windows: the InterSystems IRIS Server Manager connection details for the instance match your web server and Web Gateway configuration. (This is necessary to enable the InterSystems IRIS launcher.)
2. Prevent the PWS from starting when the instance starts by setting the instance's WebServer parameter equal to `0`. You can edit this parameter within the Management Portal by navigating to **System Administration** > **Configuration** > **Additional Settings** > **Startup**, or by editing the CPF.ini file directly. (If the installer automatically configured the instance to use the external web server, this step should already be completed.)
3. Restart the instance to stop the PWS. Alternatively, you can stop the PWS without restarting the instance by issuing the following command from the command prompt:

   #### Unix®/Linux/macOS

   ```
   kill `cat <irisInstallDir>/httpd/logs/httpd.pid`
   ```

   #### Windows

   ```
   <irisInstallDir>\httpd\bin\httpd -k stop -n <instanceName>httpd
   ```

   Where <*irisInstallDir*> is the installation directory for the instance, and <*instanceName*> is the name of the instance.

Note: If you must re-enable an instance's PWS for any reason, you can do so by resetting the WebServer parameter to `1` in the CPF.ini file and then restarting the instance. Alternatively, you can start the PWS without restarting the instance by issuing the following command from the command prompt:

Unix®/Linux/macOS

```
<irisInstallDir>/httpd/bin/httpd -d <irisInstallDir>/httpd
  -c "Listen <port>"
```

Windows

```
<irisInstallDir>\httpd\bin\httpd -k start -n <instanceName>httpd
  -c "Listen <port>"
```

Where <*irisInstallDir*> is the installation directory for the instance, <*instanceName*> is the name of the instance, and <*port*> is the port number for the PWS.

Important: Ensure that you have successfully configured your web server and Web Gateway to route requests for all desired web applications before you irreversibly remove the PWS, as described in the next step.

4. Optional: permanently remove the PWS for the instance by deleting the <irisInstallDir>/httpd/ directory, where <*irisInstallDir*> is the installation directory for the instance.

3

Overview: Set Up a Web Gateway for Your System

To serve an InterSystems web application, you must configure a web server to route requests from a web client (such as a web browser) to an InterSystems IRIS® application server through the InterSystems Web Gateway.

Note: In many cases, installers for both InterSystems IRIS and a Web Gateway can perform most of this configuration automatically. See Access Built-In Web Applications Using Your Web Server.

For users upgrading an InterSystems IRIS instance from an instance which uses the Private Web Server (PWS), videos which demonstrate the ease of automatically configuring a web server using the InterSystems IRIS installer are available here: https://learning.intersystems.com/course/view.php?id=2333

The process of manually setting up a connection between your web server and one or more InterSystems IRIS instances consists of the following procedures:

1. Install the Web Gateway files, or acquire the Docker image for the `webgateway` container.
2. Add the Web Gateway to your web server configuration.
3. Within your web server configuration, specify which requests your web server should route through the Web Gateway.
4. Within your Web Gateway configuration, register a server access profile for each InterSystems IRIS application server you want to serve web applications.
5. Within your Web Gateway configuration, register an application access profile to associate an application path with the InterSystems IRIS application server (or servers) which serve the application (or set of applications) available at that path.
6. Secure the connections between the client and the web server, between the Web Gateway and your InterSystems IRIS instances, and between a client and the Web Gateway management pages.

This page provides an overview of these procedures. For each procedure, it provides links to more specific instructions as needed. It concludes with a summary of how to decommission a Web Gateway.

3.1 Install the Web Gateway Files

Install a copy of the Web Gateway for each web server in your system. Depending on your needs, you can deploy the Web Gateway as part of an InterSystems IRIS installation, as a stand-alone component, or as a `webgateway` Docker container.

In many cases, the InterSystems IRIS installer and the stand-alone Web Gateway installer can automatically configure the Web Gateway to connect to your web server. This minimizes the need to perform the setup steps on this page, as follows:

- For the stand-alone Web Gateway, the installer can automatically add the Web Gateway to the web server configuration. You must manually configure your web server to invoke the Web Gateway as a handler for your InterSystems web applications, including each InterSystems IRIS instance's built-in system applications.
- For an InterSystems IRIS instance, the installer can automatically add the Web Gateway to the web server configuration and automatically configure your web server to invoke the Web Gateway as a handler for the instance's built-in system applications. Depending on your system, certain tasks must be performed manually to complete the configuration, as noted in Automatic Configuration Behavior. Once this configuration process is complete, you can easily extend it to suit the needs of your custom applications.
- In all cases, you must configure connection security manually.

3.1.1 Install as Part of an InterSystems IRIS Installation

InterSystems IRIS provides access to several important system functions through web applications such as the Management Portal. For this reason, the InterSystems installer includes the necessary Web Gateway files as part of the following setup types:

- **Development**
- **Server**
- **Web Server** (on Windows)
- **Custom**, if the **Web Server Gateway** component is selected

3.1.2 Install as a Stand-Alone Component

If you would like to deploy your web server on a separate machine from the machines that host your InterSystems IRIS instances, install the Web Gateway on your web server machine as a stand-alone component in one of the following ways:

- Run the stand-alone Web Gateway installer. This installer is available on the WRC download page (https://wrc.intersystems.com/wrc/coDistGen.csp). Simply type `web gateway` in the **Name** column and locate the kit for your system. As with the InterSystems IRIS installer, InterSystems implements the stand-alone Web Gateway installer as an executable file for Windows and as a script (`GatewayInstall`) for all other systems.
- Run the InterSystems IRIS installer using one of the following setup types:
  - **Web Server** (on Windows)
  - **Custom**, selecting only the **Web Server Gateway** component
- Pull the Docker image for a `webgateway` container from the InterSystems Container Registry. See Deploy a `webgateway` Container for more information.

In many cases, the stand-alone installer provides you the option to automatically configure your web server to use the Web Gateway. After automatic configuration, no further action is needed to add the Web Gateway to your web server configuration. However, you must manually configure your web server to invoke the Web Gateway as a handler for your InterSystems web applications, including the built-in system applications for each of the InterSystems IRIS instances in your system. You must also manually configure connection security.

3.1.3 Deploy a `webgateway` Container

The InterSystems Container Registry provides Docker images for `webgateway` containers. These containers include a web server (Apache or nginx) that is already pre-configured with a Web Gateway extension.

To configure a Web Gateway connection using a `webgateway` container, your configuration must satisfy the same conditions described in the sections which follow. However, in a containerized deployment, the configuration methods vary considerably from the methods referred to on this page. When you use the `webgateway` container:

- No further action is required to add the Web Gateway to the web server configuration.
- To automate web server configuration upon deployment, you should configure the web server in the `webgateway` container to invoke the Web Gateway for requests sent to particular paths by amending the configuration file programmatically within the image's Dockerfile or by describing the configuration in the YAML file for the container cluster.
- To automate Web Gateway configuration upon deployment, you can leverage the CSP.ini merge feature to define server access profiles and application access profiles for `webgateway` containers directly within their CSP.ini.

Refer to Using the InterSystems Web Gateway Container for specific instructions.

3.2 Extend the Functionality of Your Web Server with the Web Gateway

When a web client requests static content (plain text, images, JavaScript), the web server's role is straightforward: it simply serves content from the file system location that corresponds to the request's URL path. The web server's configuration defines mappings between URL paths and the file system locations which it has permission to access.

However, to serve dynamic content from a web application, you must extend a web server's functionality with a library module or an external program that can interpret an HTTP request as a call to invoke web application code.

This is the purpose provided by the InterSystems Web Gateway: it extends a web server's functionality to recognize and serve requests for InterSystems web applications. For a web server to serve an InterSystems IRIS web application, you must configure the web server to include the Web Gateway extension and then configure it to recognize which requests the Web Gateway extension should handle, usually based on the URL path specified by the request.

3.2.1 Add the Web Gateway to Your Web Server Configuration

InterSystems Supported Technologies page lists the web servers compatible with this version of the product. In many cases, the InterSystems IRIS installer can add the Web Gateway to your web server automatically.

The procedures for adding the Web Gateway to your web server manually vary considerably depending upon your operating system, the web server you use, and your use case. Some initial considerations:

- In general, InterSystems implements the Web Gateway extension for a web server as a pair of binaries: one binary is responsible for core runtime functionality; the other is responsible for the functionality of the Web Gateway management pages. The management binary is distinguished from the runtime binary by the addition of `Sys` to the end of the filename (for example, CSPa24.so and CSPa24Sys.so). Both binaries must be in the same directory.
- For Apache (UNIX®/Linux/macOS) and Microsoft IIS (Windows), the most straightforward way to deploy the Web Gateway is to leverage the web server's proprietary API for adding dynamically-loaded extension modules—.so files for Apache, .dll files for IIS. Nginx (UNIX®/Linux/macOS or Windows) only supports adding the Web Gateway extension by building the extension's source code into the web server at compilation time.

- The Web Gateway extension for Nginx invokes an external process that performs the Web Gateway's functions. This process is called the *Network Service Daemon (NSD)*. If you must detach the operation of the Web Gateway from your web server, InterSystems also provides extensions for deploying an NSD Web Gateway with Apache and Microsoft IIS, using their respective proprietary APIs.

  Important: For security reasons, the NSD files should not be accessible by your web server's processes. InterSystems recommends installing the NSD files in a file system location outside of the directories which your web server processes can access.

- If you must use Common Gateway Interface (CGI) extensions, InterSystems also provides CGI executables for configuring an NSD Web Gateway for Apache and Microsoft IIS.
- Regardless of the deployment method you choose, you must configure your system so that your web server worker processes have adequate permissions to access the following directories (and their contents):
  - the installation directory for the Web Gateway binaries. By default, these locations are /opt/webgateway on UNIX®/Linux/macOS and C:\Inetpub\CSPGateway on Windows.
  - the directory which contains the Web Gateway configuration file (CSP.ini).
  - the /temp subdirectory within the Web Gateway installation directory, which records the contents of the Web Gateway cache in the form of .dat files.
  - the directory for any static files which your web application may serve. Static files associated with CSP applications are stored in corresponding directories within the **<IRISinstallDir>/csp/** path, where **<IRISinstallDir>** is the installation path for the InterSystems IRIS instance which hosts the application. The <IRISinstallDir>/csp/broker path contains static files for several built-in system applications.

The following table provides links to specific instructions for adding the Web Gateway to your web server configuration, based on your operating system and your web server. For further information specific to your web server, refer to the web server's product documentation.

Operating System	Web Server	Deployment Instructions
UNIX®/Linux/macOS	Apache	• Recommended: Configure Apache to Work with the Web Gateway • For locked-down versions of Apache such as SELinux: Add the Web Gateway to a Locked-Down Apache Installation (along with other relevant pages on this list) • For NSD deployments: Alternative Options for Apache and Use the Network Service Daemon (UNIX®/Linux/macOS) • Atypical and legacy configurations: Alternative Options for Apache • Additional considerations: Apache Web Server Considerations (UNIX®/Linux/macOS)
UNIX®/Linux/macOS	nginx	• Build and Configure nginx (UNIX®/Linux/macOS) and Use the Network Service Daemon (UNIX®/Linux/macOS)
Windows	IIS	• Recommended: Configure IIS to Work with the Web Gateway • For NSD deployments: Alternative Options for IIS 7 or Later and Use the Network Service Daemon (Windows) • Atypical and legacy configurations: Alternative Options for IIS 7 or Later
Windows	nginx	• Build and Configure nginx (Windows) and Use the Network Service Daemon (Windows)

3.2.2 Specify Which Requests the Web Server Routes through the Web Gateway

To serve an InterSystems web application, the web server must invoke its Web Gateway extension to handle HTTP requests which it receives at URL paths designated for that application. This configuration procedure varies depending on the web server.

Regardless of your web server, InterSystems provides two methods for specifying which requests the Web Gateway should handle:

1. Invoke the Web Gateway when the client requests certain file types at a given path. Requests for many InterSystems web applications end with one of the following InterSystems file type extensions:

   ```
   .csp .cls .zen .cxw
   ```

   The first three file types indicate different kinds of code which can run within InterSystems IRIS. The last file type (.cxw) is reserved for use by the Web Gateway management pages exclusively. If your web application serves static files as part of its response (.jpg, .js, and so on), you must configure the web server to invoke the Web Gateway in response to requests for those static file types as well.

2. Invoke the Web Gateway for any request at a given path.

Most REST APIs expose endpoint paths without specifying a file or file type (for example, the endpoint for collecting InterSystems IRIS performance data: /api/monitor/metrics). To serve such an application, you must invoke the Web Gateway for all requests at the application's URL paths.

The following table briefly summarizes both methods of invoking the Web Gateway for each of the supported web servers. For more detailed instructions on configuring your web server, refer to your web server's product documentation.

Web Server	Route all Requests at Path	Route for Certain File Types

Apache	Edit the Apache configuration file (usually httpd.conf) using the directives `CSP On` and `CSP Off` within a `<Location>` block. For example:
	
	`<Location {/path}>`
	`  CSP On`
	`</Location>`
	
	where *{/path}* represents the relative URL path.
	
	Note: You cannot use this web server directive within a `<VirtualHost>` block
		Edit the Apache configuration file (usually httpd.conf) using the directive `CSPFileTypes` within a `<Location>` block. For example:
		
		`<Location {/path}>`
		`  CSPFileTypes {xxx yyy ...}`
		`</Location>`
		
		where *{/path}* represents the relative URL path and *{xxx yyy ...}* represents a list of file type extensions separated by spaces.
		
		Note: You cannot use this web server directive within a `<VirtualHost>` block

IIS	Add an unrestricted * (wildcard)**Handler Mapping** invoking the Web Gateway Native Module. See Set Handler Mappings for Application Requests for details.	Add file-restricted **Handler Mapping**s for the desired file extensions. See Set Handler Mappings for Application Requests for details.

nginx	Edit the nginx configuration file (nginx.conf) using the directives `CSP on` and `CSP off`. For example:
	
	`location </path> {`
	`  CSP on;`
	`}`
	
	where *</path>* represents a relative URL path. See Configure Nginx to Invoke the NSD for details.
		Edit the nginx configuration file (nginx.conf) using the directive `CSPFileTypes`. For example:
		
		`location </path> {`
		`  CSPFileTypes <xxx yyy ...>`
		`}`
		
		where *</path>* represents a URL path and *<xxx yyy ...>* represents a list of file type extensions separated by spaces. See Configure Nginx to Invoke the NSD for details.

InterSystems recommends that you consider the full process of routing application requests from your web server to the InterSystems IRIS instance which serves your application before you choose the URL paths for which your web server will invoke the Web Gateway. This is especially true if your system features multiple InterSystems IRIS application servers or if your organization has specific URL naming conventions. See Choose the Paths Which Route Requests Through the Web Gateway for guidance.

3.3 Direct Requests from the Web Gateway to Your InterSystems IRIS Instances

After the web server passes a request for a web application to the Web Gateway, the Web Gateway routes it to an InterSystems IRIS instance which hosts the code for that application.

To do this, you must provide the following information to the Web Gateway:

- connection information necessary to communicate with the application server process for each instance. The Web Gateway maintains this information in a server access profile.
- mappings between a web application path (or a base path for a set of web applications) and the server access profile for the instance which hosts the application code. The Web Gateway maintains the routing behavior for a path in an application access profile.

Note: The application server process for an InterSystems IRIS instance is sometimes referred to as a "CSP server," in reference to the legacy InterSystems web application framework.

The Web Gateway management pages for configuring and monitoring connections to instances' application server processes refer to these processes simply as "servers." (The Web Gateway's connection to the *web* server is determined by the web server configuration.)

To minimize confusion, the Web Gateway documentation refers to an InterSystems IRIS instance as the effective target of a Web Gateway connection, except where it is necessary to refer to the application server process specifically.

Regardless of the deployment, every Web Gateway provides the same convenient web interface for configuring server access profiles and application access profiles: the *Web Gateway management pages*. The Web Gateway management pages also allow you to configure global default parameters, monitor Web Gateway connections, test connections, and more.

The Web Gateway maintains configuration information in the CSP.ini.

Important: Except in containerized deployments where it may be necessary to edit the CSP.ini file directly, InterSystems recommends restricting access to the CSP.ini file and performing all Web Gateway configuration using the Web Gateway management pages or using Web Gateway Registry methods.

3.3.1 Connect InterSystems IRIS Instances to the Web Gateway

To allow the Web Gateway to establish and maintain connections to the application server process for an InterSystems IRIS instance, you must define a server access profile which provides connection information for each instance—IP address, superserver port number, connection security credentials, and so on. See Define a Server Access Profile for Your InterSystems IRIS Instance.

3.3.2 Associate Instances with an Application

A single Web Gateway can route requests to multiple InterSystems IRIS instances for applications specific to those instances (for example, the Management Portal for each instance). In addition, the Web Gateway can associate requests for a single application with multiple instances which host the same application code, allowing for load balancing and failover.

The Web Gateway determines the destination instance based on an application access profile. An application access profile defines a relationship between the URL path specified by a request and one or more InterSystems IRIS instances (as repre-

sented by their server access profiles). The application access profile also includes other details about how the Web Gateway should handle traffic for that path. See Define an Application Access Profile for Your Web Application Path.

InterSystems recommends that you consider the full process of routing application requests from your web server to the InterSystems IRIS instance which serves your application before you choose the URL paths for which you will define application access profiles. This is especially true if your system features multiple InterSystems IRIS application servers or if your organization has specific URL naming conventions. See Choose the Paths Which Route Requests Through the Web Gateway for guidance.

3.4 Secure All Connections

A request passes through two TCP connections on its way from the web client to an InterSystems IRIS application server:

1. The connection between the web client and your web server. To secure the connection between the web client and the web server—that is, to use HTTPS—you must configure SSL/TLS for your web server. Refer to the documentation for your web server for further guidance.
2. The connection between the Web Gateway and the InterSystems IRIS application server. InterSystems supports multiple ways to secure this connection; see Protect Web Gateway Connections to InterSystems IRIS for details.

In addition, you should secure access to the Web Gateway management pages by defining the IP addresses which can access them and by requiring authentication to access them.

3.5 Decommission a Web Gateway Connection

If you have uninstalled an InterSystems IRIS instance from your system, you should decommission the Web Gateway connection to that instance by removing all configuration elements related to it. To do so:

1. Delete the Web Gateway application access profiles for any application paths you no longer require. For example, if the instance was named IRISserv1, you would remove the application access profile for the /irisserv1 path (the instance prefix), as well as application access profiles for descendant paths such as /irisserv1/csp.
2. If the instance was serving any applications which you would like to continue serving using other InterSystems IRIS instances, update the fields for the corresponding application access profiles so that they identify the server access profiles for those other instances. (This may involve adding server access profiles for any new InterSystems IRIS application servers.)
3. Delete the Web Gateway server access profile which corresponded to the InterSystems IRIS instance you have uninstalled. Note that you must remove all references to a server access profile from your application access profiles before the Web Gateway allows you to delete a server access profile.
4. Remove web server configuration directives for the paths you no longer require. For example, if the instance was named IRISserv1, you would remove the `<Location /irisserv1>` block from an Apache web server's configuration file, along with the `<Location>` blocks for descendant paths such as /irisserv1/csp. If you chose the automatic web server configuration option when you installed the instance and you are using an Apache web server, comments mark the beginning and end of the configuration directives specific to an instance, as demonstrated in this example.

If you are completely discontinuing use of the Web Gateway by your web server, remove all elements related to the Web Gateway from your web server configuration. Then, delete the Web Gateway files. By default, these files are stored in /opt/webgateway/bin (UNIX®/Linux/macOS) or C:\Inetpub\CSPGateway (Windows).

4

Install a Stand-Alone Web Gateway

If your web server is located on a separate machine from the InterSystems IRIS® instances which host your web applications, you can install the InterSystems Web Gateway on the web server machine as a stand-alone component.

This page describes the procedure for installing a stand-alone Web Gateway using the installation kit available on the WRC **Components** download page (https://wrc.intersystems.com/wrc/coDistGen.csp).

Note: There are alternative methods for deploying a stand-alone Web Gateway on the web server machine in such a scenario, described elsewhere:

- Use the InterSystems IRIS installer to install only the Web Gateway, selecting either a Custom setup type or (on Windows) a Web Server setup type.
- Deploy a container.

In many cases, the stand-alone Web Gateway installer can configure your web server to use the Web Gateway automatically. Otherwise, the installation media provide the files necessary to add the Web Gateway to your web server configuration manually.

The stand-alone Web Gateway installer also allows you to specify connection information for an InterSystems IRIS instance. The stand-alone Web Gateway installer uses this information to create a server access profile for this instance as well as an application access profile for the instance prefix URL path. If the installer succeeds in connecting the Web Gateway to your web server automatically, it configures the web server to invoke the Web Gateway for all URL paths associated with the instance's web applications, including the instance prefix URL path.

4.1 Step 1: Install a Supported Web Server

The Web Gateway installation kit can automatically configure a supported web server to use the Web Gateway, provided that it is installed and running prior to installing the Web Gateway. This is the recommended approach for most use cases.

The web server for which automatic configuration is supported varies depending on your system:

- For UNIX®/Linux/macOS: install the Apache httpd web server in the default installation location. For further guidance installing Apache httpd, refer to this Developer Community article: https://community.intersystems.com/post/how-install-apache-iris-supported-operating-systems
- For Windows: enable Microsoft Internet Information Services (IIS). See Enable IIS for instructions.

Important: If you install a web server other than the one specified for your operating system, or if do not install the specified web server until after you have completed the Web Gateway installation, you will need to configure the web server manually.

Depending on your system, your web server installation may need to satisfy additional conditions to enable automatic configuration. See When is Automatic Configuration Possible?

4.2 Step 2: Download the Installation Kit

Stand-alone Web Gateway installation kits are distributed through the WRC **Components** download page (https://wrc.intersystems.com/wrc/coDistGen.csp).

Type `web gateway` in the **Name** column and then use the **Os** and **Arch** columns to locate the correct Web Gateway installation kit for your system.

For UNIX®, Linux, and macOS systems, InterSystems distributes the Web Gateway installation kit as a compressed tarball (.tar.gz) which contains an installation script (`GatewayInstall`); for Windows systems, InterSystems provides the installer as an executable file.

4.3 Step 3: (UNIX®/Linux/macOS Only) Extract the Installation Kit Files

Uncompress and extract the contents of the tarball into a temporary filesystem location. To do this while preserving the original permissions, you can issue the command below (replace <*WebGatewayKit*> with the name of the tarball you have downloaded):

```
tar zpxvf <WebGatewayKit>.tar.gz
```

The files extract into a directory with the same name, <WebGatewayKit>/. The contents of <WebGatewayKit>/ include:

- An install/ subdirectory, which contains:
  - The installation script, GatewayInstall. The subsequent instructions on this page describe how to use this script to install the Web Gateway.
  - The nginx/ subdirectory. This subdirectory contains the file necessary to manually configure a build of Nginx which includes the Web Gateway.
- A subdirectory with a name corresponding to your operating system and architecture, /<platformCode>. This subdirectory contains files necessary to manually configure an Apache httpd web server to use the Web Gateway.

  Note: The GatewayInstall script can automatically configure an existing Apache httpd web server to use the Web Gateway. Therefore, manual configuration is only necessary in atypical deployment scenarios, such as if you want to deploy a Web Gateway which uses the NSD.

4.4 Step 4: (UNIX®/Linux/macOS Only) Log in as `root`

To run the GatewayInstall script, you must be running a command line session as a user with root privileges. (It is sufficient to `su` as `root` from another user account.)

4.5 Step 5: Run the Installer

4.5.1 UNIX®/Linux/macOS

Run the GatewayInstall script to begin the installation process. You can do this by navigating to the <WebGatewayKit>/install/ directory and then issuing the following command:

```
./GatewayInstall
```

GatewayInstall provides a series of interactive prompts. As prompted, perform the following steps:

1. Specify the type of web server you have installed:
   - Select `Apache` if you have installed Apache httpd and you would like the installer to automatically configure Apache to use the Web Gateway. The modified web server configuration will include directives invoking the Web Gateway for requests which are sent to relative URL paths associated with built-in InterSystems IRIS web applications.
   - Select `None` if you have installed any other supported web server, or if no web server is installed. When you select this option, the installer copies the files necessary to configure a supported web server manually into the directory you specify in the next step. It also configures the Web Gateway using the connection information you provide in a subsequent step. It does not attempt to configure a web server.

   Note: On some platforms, GatewayInstall may also prompt you to provide a string representing your system platform name. This string is provided at the end of the filename for the installation kit—for example: if your installation kit has the form WebGateway-<version>-lnxrh8x64.tar.gz, then `lnxrh8x64` is the platform name. Alternatively, you can retrieve this value using the cplatname script. To do so, navigate to the installation kit's /install subdirectory and issue the following command:

   ```
   # ./cplatname identify
   ```

2. Specify your desired destination directory for the Web Gateway files, or accept the default (usually /opt/webgateway/). If the directory does not yet exist, the installer asks if you want to create it.
3. Provide information about the InterSystems IRIS instance you would like the Web Gateway to connect to:
   a. The hostname for the machine where the instance is located.
   b. The superserver port where the instance accepts Web Gateway connections.
   c. The configuration name () for the instance. (By default, this is the instance name in lowercase characters.) The configuration name serves as a prefix URL path which can be used to route requests to that instance specifically.
   d. The security settings for the instance:
      - Select `Minimal` if the instance allows unsecured Web Gateway connections.

- Select `Normal` if the instance requires Web Gateway connections to be authenticated using the instance's CSPSystem user account credentials.
- Select `Locked Down` if the instance requires Web Gateway connections to be authenticated using the instance's CSPSystem user account credentials, and the security settings of the instance are Locked Down.

If you select `Normal` or `Locked Down`, you must also provide the password for the instance's CSPSystem user account.

The installer uses this information to edit the Web Gateway's configuration file (CSP.ini), creating a server access profile to connect to the instance and application access profiles for the instance's applications (including one based on the instance's configuration name).

If the installer successfully configures your web server to use the Web Gateway automatically, it also uses this information to configure the web server. The web server configuration will include directives to invoke the Web Gateway for requests which are sent to the instance prefix URL path, based on the configuration name.

Note: The installer requires responses for these prompts. If you are installing a Web Gateway before you have installed any InterSystems IRIS instance that it will serve, accept the default values or provide placeholder values. After the installation, you can remove the unneeded configuration items or modify them to connect to an existing instance.

4. Confirm the configuration details to begin the installation.

4.5.2 Windows

Run the executable file to launch the standalone Web Gateway installer wizard. The wizard provides a series of interactive prompts. As prompted, perform the following steps, selecting **Next >** to advance to the next prompt:

1. Select a setup type:
   - **Complete** installs all Web Gateway files in their default locations. In other words, it has the same effect as a **Custom** setup type (described next) wherein all installation features are included and no default location is overridden. If IIS is enabled, the installer configures IIS to use the Web Gateway.
   - **Custom** allows you to include or exclude installation features which install particular subsets of files, and select the directories where they will be installed:
     - Select **Web Gateway CGI modules** to install the files which implement the Web Gateway as a CGI extension. These files are generally not necessary unless your organization requires the exclusive use of CGI extensions.
     - Select **Web Gateway for IIS** to install the files which are necessary to configure an IIS web server to use the Web Gateway. If IIS is enabled and this item is selected, the installer configures IIS to use the Web Gateway. When configured, IIS will invoke the Web Gateway to handle requests which are sent to relative URL paths associated with built-in InterSystems IRIS web applications.

       Note: The installer always installs **Web Gateway for IIS** files within the default IIS directory (C:\inetpub\), whether or not IIS is enabled.

     - Select **Static files for Web Gateway** to install static files which are used by built-in InterSystems IRIS applications within the Web Gateway installation directory. This provides you with the option to configure the web server to serve these static files from a central filesystem location on the web server machine. (By default, each InterSystems instance serves files associated for its web application from its own directory.)

– Select **Web Gateway modules for Apache HTTPD servers** to install the files which implement the Web Gateway as an extension for the Apache httpd web server.

CAUTION: InterSystems no longer supports the use of the Apache httpd web server on Windows systems.

2. Select **Configure Web Gateway to connect to a server** if you want the installer to configure the Web Gateway to connect to an InterSystems IRIS instance. When this option is selected, you can provide the following connection information about the instance:

   - **Application name**: a custom name which can be used as a URL path to route requests for the application or applications which the instance serves. In most situations, InterSystems recommends providing the instance's . (By default, this is the instance name in lowercase characters.)
   - **IRIS server address**: the hostname for the machine where the instance is located.
   - **IRIS server port number**: the superserver port where the instance accepts Web Gateway connections.
   - **Connection password (optional)**: the password for the instance's CSPSystem user account. (The CSPSystem user account is used to authenticate Web Gateway for instances installed with **Normal** or **Locked Down** initial security settings.)

   The installer uses this information to edit the Web Gateway's configuration file (CSP.ini), creating a server access profile to connect to the instance and application access profiles for the instance's applications (including one based on the **Application name** you specify).

   If the installer successfully configures your web server to use the Web Gateway automatically, it also uses this instance information to configure the web server. The web server configuration will include directives to invoke the Web Gateway for requests which are sent to the instance prefix URL path, based on the configuration name.

3. Select **Install** to begin the installation.

4.5.2.1 Modify, Repair, or Remove a Web Gateway Installation

If there is already a stand-alone Web Gateway installed on your system, the stand-alone Web Gateway installation executable allows you to **Modify**, **Repair**, or **Remove** that installation.

- **Modify** allows you to customize your installation, including or excluding installation features which represent particular subsets of Web Gateway files. These installation features are equivalent to those available for a new installation's **Custom** setup type. **Modify** mode also provides you the opportunity to configure (or reconfigure) connection details for an InterSystems IRIS instance.

  If IIS is enabled and you **Modify** your Web Gateway installation to include the **Web Gateway for IIS** feature, the installer configures IIS to use the Web Gateway, as it does in a new installation. If you specify connection details for an InterSystems IRIS instance, the installer configures the Web Gateway and IIS to serve requests for that instance, as it does in a new installation.

- **Repair** allows you to repair your stand-alone Web Gateway using its original installation settings, restoring any files which were deleted or corrupted. If your original installation included the **Web Gateway for IIS** feature but IIS was not yet enabled, **Repair** mode again attempts to configure IIS to use the Web Gateway. If your original installation specified connection details for an InterSystems IRIS instance but IIS was not yet enabled, **Repair** mode again attempts to configure the web server to serve requests for the instance.

- **Remove** allows you to uninstall the stand-alone Web Gateway.

5

Extend Your Web Server Configuration with the Web Gateway

After you install the files for the InterSystems Web Gateway, you must add the Web Gateway to your web server configuration as an extension. Once you have done so, your web server can invoke the Web Gateway to handle requests which are intended for your InterSystems IRIS® web applications.

This page describes the recommended way to add the Web Gateway extension to a web server configuration for each supported web server. For each web server, it also summarizes the method InterSystems has implemented for specifying which requests should be routed through the Web Gateway. (Refer to Choose Which URL Paths Route Requests Through the Web Gateway as you consider how you should route application requests.)

In many cases, the installer can perform the configuration steps on this page automatically for Apache (on UNIX®/Linux/macOS) or IIS (on Windows). You can then simply customize the resulting configuration to suit the needs of your own system.

Once you have configured the web server, use the Web Gateway management pages to connect your InterSystems IRIS instances and associate application paths with them.

5.1 Files to Consider

5.1.1 Web Gateway Files

The Web Gateway files are installed in one of the following locations:

- /opt/webgateway/bin on UNIX®/Linux/macOS. InterSystems recommends using Web Gateway files in this common location instead of within an InterSystems IRIS installation directory, because they will not be affected by modifications to an InterSystems IRIS installation.
- C:\Inetpub\ on Windows when Internet Information Services (IIS) is configured. InterSystems recommends using Web Gateway files in this common location instead of within an InterSystems IRIS installation directory, because they will not be affected by modifications to an InterSystems IRIS installation.
- <installDir>/csp/bin or <installDir>/bin when:
  - You have installed an InterSystems IRIS instance in *<installDir>*
  - The web server is not configured.

- <installDir> or <installDir>/bin:
  - You have installed a stand-alone Web Gateway in <*installDir*>
  - The web server is not configured.

In general, InterSystems implements the Web Gateway as a pair of binary files: one file implements the runtime functionality and the other implements the Web Gateway management pages. The management binary is distinguished from the runtime binary by the addition of `Sys` to the end of the filename.

5.1.2 Web Server Files

The installation location of a web server varies depending on the web server and the operating system; see your web server documentation for help finding where your web server is installed.

- Apache: https://httpd.apache.org/docs/2.4/
- IIS: https://learn.microsoft.com/en-us/iis/get-started/introduction-to-iis/iis-web-server-overview
- Nginx: http://nginx.org/en/docs/

Note: The Web Gateway documentation sometimes specifies a web server installation location—usually the default—in its examples, for maximum clarity. This location may not match its location on your system; substitute the location of the file on your system instead.

5.1.3 Static Files

An InterSystems IRIS instance can serve static files as part of its web application responses, provided that you configure the web server to allow the Web Gateway to handle requests which are sent to the application's relative URL path for all the required file types.

The static files which an InterSystems IRIS CSP web application serves are located in the <installDir>/csp/ directory which corresponds with the web application, where <*installDir*> is the installation directory for the InterSystems IRIS instance.

For example:

- <installDir>/csp/broker contains files shared by several built-in system applications.
- <installDir>/csp/sys contains files used by the Management Portal

By default, an InterSystems IRIS CSP web application is configured to serve static files itself, from its associated directory.

However, the web application can allow the web server to serve these static files instead. For example, if you have a system where a single web server serves multiple remote InterSystems IRIS instances, it may be expedient to serve these files from a common location local to the web server machine. Note, however, that this may cause problems: for example, if the common web server serves different versions of InterSystems IRIS, the web server may encounter a conflict between two different versions of the same file.

To configure your web server to serve and cache static files for an InterSystems IRIS web application, you must create a mapping between the path for an application (to which requests for the application's static files are directed) and a local file system location that the web server has adequate permission to access, where the static files are stored.

Here and elsewhere, the Web Gateway documentation describes ways of creating this optional mapping for each web server. For further instructions on how to configure your web server to serve static files from a local file system location it manages itself, refer to your web server's documentation.

5.2 Apache for UNIX®/Linux/macOS

The Apache HTTP web server is supplied by the Apache Group and can be downloaded free of charge from http://www.apache.org. Many systems are shipped with Apache preinstalled, configured and ready to go. For guidance installing Apache on a supported UNIX® or Linux system, refer to this Developer Community article: https://docs.intersystems.com/irislatest/csp/docbook/DocBook.UI.Page.cls?KEY=GCGI_webserver#GCGI_ux_apache.

Note: On macOS, the InterSystems IRIS installer cannot automatically configure the preinstalled version of Apache. However, it can automatically configure the version of Apache installed using Homebrew (https://brew.sh/).

This page describes how to deploy the Web Gateway in Apache with the dynamic module `csp_module_sa`, which InterSystems implements using Apache's native API.

The `csp_module_sa` module implements the Web Gateway using two dynamic shared object (.so) files:

- CSPa24.so — the runtime binary.
- CSPa24Sys.so — the Web Gateway management binary.

This is the recommended deployment option for Apache; it is effective for the vast majority of use cases.

Note: State-aware connectivity (preserve mode 1) should not be used with this module.

If you are configuring the Web Gateway on a system which uses a security-restricted version of Apache (such as RHEL with SELinux), see Add the Web Gateway to a Locked-Down Apache Installation for supplementary instructions. Alternative Options for Apache describes other options for deploying the Web Gateway with Apache, including options which use the Network Service Daemon (NSD).

Apache Considerations provides some guidance for administering an Apache web server connected to a Web Gateway. For detailed guidance, consult the Apache documentation (https://httpd.apache.org/docs/).

5.2.1 Verify That Apache Can Manage Shared Object Modules

Before you attempt to add the Web Gateway to your Apache configuration using Apache's proprietary API, check that your build of Apache includes the built-in module for managing shared objects: mod_so.

To perform this check on Red Hat Linux, issue the following command from the command line (select the command appropriate for your operating system):

RHEL

```
httpd -l
```

Ubuntu/SUSE

```
apache2 -l
```

This command displays the list of modules included in your current Apache installation. If mod_so is not included in this list, follow the instructions provided within the Apache documentation (https://httpd.apache.org/docs/2.4/) to compile a new build of Apache that includes this module.

5.2.2 Add the Web Gateway Modules to Your Web Server Configuration

1. Note the file system location of the Web Gateway module files. These instructions use the most common location, /opt/webgateway/bin.

2. Open the Apache web server configuration file (httpd.conf) in a text editor. This file is located in your Apache installation directory, within the /conf subdirectory. The most common installation locations are:

   RHEL

   ```
   /etc/httpd/conf
   ```

   Ubuntu/SUSE

   ```
   /usr/apache2/conf
   ```

3. Append directives to the end of the file which add the Web Gateway module. These directives should accomplish the following:

   - Load the module using the `LoadModule` directive, providing the name of the module and the location of the .so file for the Web Gateway's runtime functions.
   - Issue the `CSPModulePath` directive, providing the full path for the directory that contains the .so file for the Web Gateway's runtime functions.
   - Issue the `CSPConfigPath` directive, providing the full path for the directory that contains the .so file for the Web Gateway management functions.
   - Configure access settings for the directory that contains the Web Gateway binaries. To do so, add a `<Directory>` directive block for that directory as follows (the example assumes the most common installation location):

     ```
     <Directory "/opt/webgateway/bin">
        AllowOverride None
        Options MultiViews FollowSymLinks ExecCGI
        Require all granted
        <FilesMatch "\.(log|ini|pid|exe)$">
           Require all denied
        </FilesMatch>
     </Directory>
     ```

4. Optional: if you want the web server to serve static files for your web applications, append directives to map the appropriate /csp paths to file system locations accessible to your web server. The Apache `Alias` directive provides one such method to accomplish this.

   For example: if the application /dashboard is served by a remote InterSystems IRIS instance but you want to serve static files for the application from the directory C:\iris\csp\dashboard on the web server's machine, you could add the following directive:

   ```
   Alias /dashboard/ "C:/iris/csp/dashboard"
   ```

5. Issue directives which invoke the Web Gateway to handle requests intended for your InterSystems IRIS applications. InterSystems provides the following directives for this purpose:

   - `CSPFileTypes`, which invokes the Web Gateway for requests for the set of file types you specify. (Because REST API endpoints are often paths and not files, this directive is not sufficient to invoke requests for REST applications.)
   - `CSP On` and `CSP Off`, which enable or disable the Web Gateway as the handler for all requests.

   By issuing these directives within `<Location>` blocks, you can configure your web server in a granular way, invoking the Web Gateway only for those relative URL paths which correspond with your InterSystems IRIS web applications.

   The following location block would invoke the Web Gateway exclusively for requests directed to the Management Portal for an InterSystems IRIS instance named iris3:

   ```
   <Location "/iris3/csp/sys/">
      CSP On
   </Location>
   ```

Note: To access the management pages for a Web Gateway without access to an InterSystems IRIS instance, enable the Web Gateway for the `/csp` path.

6. Save the httpd.conf file.
7. Restart Apache to allow the configuration changes to take effect.

The installer employs this procedure when it automatically configures an Apache web server. See Automatic Configuration Example for an example.

5.3 Microsoft Internet Information Services (IIS) for Windows

Microsoft IIS (https://www.iis.net/) is preinstalled with many distributions of Windows. However, it is usually disabled by default. See Enable IIS.

This page describes how to deploy the Web Gateway as a Native Module using an IIS proprietary API.

The Web Gateway Native Module implements the Web Gateway using two dynamic linked library (.dll) files:

- CSPms.dll — the runtime binary
- CSPmsSys.dll — the Web Gateway management binary

This is the recommended deployment option; it is effective for the vast majority of use cases.

Alternative Options for IIS describes other options for deploying the Web Gateway with IIS, including options which use ISAPI and options which use the Network Service Daemon (NSD).

5.3.1 Enable IIS

To enable IIS:

1. Open the **Windows Features** manager by searching for **Turn Windows features on or off**, or by opening the **Control Panel** and selecting **Programs** > **Programs and Features** > **Turn Windows Features on or off**.
2. Select **Internet Information Services**.
3. To enable redirection of documentation links, **HTTP Redirection** must also be enabled within IIS. To enable it, ensure that **Internet Information Services** > **World Wide Web Service** > **Common HTTP Features** > **HTTP Redirection** is selected.
4. To enable debugging sessions in a supported IDE, the **WebSocket Protocol** must also be enabled within IIS. To enable it, ensure that **Internet Information Services** > **World Wide Web Service** > **Application Development Features** > **WebSocket Protocol** is selected.
5. Select **OK**.

If you would like the installer to automatically configure a new or upgraded InterSystems IRIS instance to serve its web applications over IIS, IIS must be enabled prior to installation. However, the **HTTP Redirection** and **WebSocket Protocol** features of IIS can be enabled at any time.

5.3.2 Set Permissions for the Web Gateway Components

By default, IIS does not allow the user of a web application access to anything outside of the web document root directory (usually C:\Inetpub\wwwroot). To provide access to web application resources located in a directory outside of the web document root, the following user groups must possess Read, Write, and Execute permissions for the directory:

- **[machine_name]\IIS_IUSRS**, the user group under which IIS worker processes and applications controlled through IIS (such as the Web Gateway) operate.
- **[machine_name]\Users**

You must manually set these permissions for the Web Gateway directory (usually C:\Inetpub\CSPGateway) and for any directory containing static files which a web application serves. To do so:

1. In the Windows File Explorer, navigate to the directory's parent directory. For example: if you are configuring the C:\Inetpub\CSPGateway directory, navigate to navigate to C:\Inetpub.
2. Right-click the directory name and select **Properties**.
3. Select the **Security** tab.
4. Select **Edit**.
5. Select **Add**.
6. In the **Enter the object names to select** text box enter:

   `[machine_name]\IIS_IUSRS`
7. Select **Check Names** and **OK**.
8. Select **[machine_name]\IIS_IUSRS** in the **Group or Usernames** window, then:
9. Assign **Read & Execute** and **Write** permissions in the **Permissions** window.
10. Select **Apply** and **OK**.
11. Repeat the above process for the **[machine_name]\Users** user group.

If the Web Gateway configuration file (CSP.ini) or the Web Gateway log file (CSP.log) is located elsewhere within your file system, you must also ensure that the `IIS_IUSRS` group has full read and write permissions for these files.

5.3.3 Register the Native Modules

Before you can use the Web Gateway Native Modules, you must register them within IIS. To do so:

1. Open the **Internet Information Services (IIS) Manager**.
2. Select your localhost connection from the **Connections** panel.
3. Within the **Features View** for the connection's **Home** page, open (double-click) the **Modules** feature configuration item.
4. Select the **Configure Native Modules...** action from the **Actions** panel.
5. Select **Register** and enter the following in the **Register Native Module** dialog:

   Name: `CSPms` (or similar)

   Path: C:\Inetpub\CSPGateway\CSPms.dll
6. Select **OK**.
7. Within the **Connections** panel, expand the contents of your localhost connection to select **Sites** > **Default Web Site**.

8. Ensure that the Web Gateway module is enabled:
    a. Within the **Features View** for the **Default Web Site Home** page, open (double-click) the **Modules** feature configuration item.
    b. Select the **Configure Native Modules...** action from the **Actions** panel.
    c. Within the **Configure Native Modules** dialog, if **CSPms** appears in the list, select it and then select **OK**.

5.3.4 Configuring the Web Application Path

For each InterSystems IRIS web application you want to serve, configure its relative path (/csp or /irisinstance2) as an IIS **Application**.

Note that configuring an IIS **Application** creates the path mapping which is required to allow the web server to serve static files, provided that the web server has adequate permissions to access the physical path.

For each application path you want to configure, perform the following steps:

1. Open the **Internet Information Services (IIS) Manager**.
2. Within the **Connections** panel, expand the contents of your localhost connection to select **Sites** > **Default Web Site**.
3. Select **View Applications** from the **Actions** panel.
4. Select **Add Application...** from the **Actions** panel.
5. In the **Add Application** dialog, provide the following information:
    - **Alias**: the relative base URL path for the application. For example: /irisinstance2 for applications hosted by the instance IRISinstance2, or csp for the application /csp/sys and its sibling applications.
    - **Physical path**: the directory which contains the static files associated with the application. If you want IIS to serve static files from a different directory for applications with paths subordinate to the application path you are currently configuring, define an IIS **Virtual Directory** for the child directory.
6. Select **OK**.
7. If you are finished configuring IIS, restart it to allow configuration changes to take effect.

Note: To access the management pages for a Web Gateway without access to an InterSystems IRIS instance, enable the Web Gateway for the `/csp` path.

If you are using a Web Gateway solution based on an alternative option, set up an application called /bin under the /csp application. Map this to the physical directory holding the Web Gateway binaries. In most cases, this would be a mapping between the application path /csp/bin and the physical path C:\Inetpub\CSPGateway

5.3.4.1 Define a Virtual Directory

IIS Virtual Directories enable users to serve some static content from a file system location outside of the file system location specified for the application. For example, the administrator of a static web site which hosts its images at `domain.com/images` can store most of the web site's content in C:\Inetpub\wwwroot but store images in C:\siteimg by mapping the Virtual Directory `images` to that physical path.

You can use IIS Virtual Directories if you maintain different static file storage locations for different InterSystems IRIS applications within one parent path directory (for example, /csp).

To configure an IIS Virtual Directory:

1. Within the **Internet Information Services (IIS) Manager**, locate the IIS Application for which you want to configure a subordinate **Virtual Directory** in the **Connections** panel, and select it.
2. Select **View Virtual Directories** from the **Actions** panel.
3. Select **Add Virtual Directories** from the **Actions** panel.
4. In the **Add Virtual Directory** dialog, provide the following information:
   - **Alias**: the subordinate part of the application URL path. For example: sys for the path **/csp/sys**
   - **Physical path**: the alternative location of static files for this path.
5. Select **OK**.
6. If you are finished configuring IIS, restart it to allow configuration changes to take effect.

5.3.5 Set Handler Mappings for Application Requests

For each IIS **Application** you have configured, direct IIS to invoke the Web Gateway as a handler for some (or all) requests sent to the application path by creating **Handler Mappings**.

To invoke the Web Gateway as a handler for all requests sent to an application path: follow the steps below to create a Handler Mapping for the **Request Path** * (wildcard).

To invoke the Web Gateway exclusively for certain file types: follow the steps below for each file type, specifying *.xxx as the **Request Path**, where *xxx* is the file extension. To serve InterSystems specific file types, create Handler Mappings for the following request paths: *.csp, *.cls, *.zen, and *.cxw. You must also create Handler Mappings for any static file types you want your application to serve. (Because REST API endpoints are often paths and not files, this file type-specific approach is not sufficient to invoke requests for REST applications.)

Perform the following steps to create a **Handler Mapping** for an IIS Application:

1. Open the **Internet Information Services (IIS) Manager**.
2. Within the **Connections** panel, expand the contents of your localhost connection and select the name of your **Application** from the list available within **Sites** > **Default Web Site**.
3. Within the **Features View** for the connection's **Home** page, open (double-click) the **Handler Mappings** feature configuration item.
4. Within the actions panel, select **Add Module Mapping...**.

   Note: Do not use the **Add Wildcard Script Mapping...** action to set the Web Gateway as the handler for all requests at an application path. Attempting to do so yields an error. Instead, add a module mapping for the wildcard (*) character.

5. In the **Add Module Mappings** dialog, specify the following:
   - **Request Path** — provide an expression that specifies the set of requests which the Web Gateway should handle. (See the beginning of this section for examples of valid expressions.)
   - **Module** — from the drop-down menu, select the name you provided for the Web Gateway module (**CSPms**).
   - **Name** — provide a descriptive name for this mapping, such as `WebGateway_*`.
6. Select **Request Restrictions** and ensure that the box next to **Invoke handler only if request is mapped to** is cleared (*not* selected). This action sets the value of **Path Type** to **Unspecified**, as shown in the **Handler Mappings** table.
7. Select **OK** to return to the **Add Module Mappings** dialog, and then select **OK** again.
8. Repeat the above process to add all desired **Handler Mappings** for the IIS Application.

9. If you are finished configuring IIS, restart it to allow configuration changes to take effect.

5.3.6 Enable URLs with /bin

By default, IIS does not serve resources from request paths which contain the directory /bin. You must disable this filter to serve the Web Gateway management pages.

If the installer configured the Web Gateway automatically, this step was done automatically for you. If you are installing the Web Gateway manually, you must perform this configuration manually.

The following procedure enables IIS to serve the Web Gateway management pages specifically:

1. Open the applicationHost.config file in a text editor. The file is usually located in the C:\Windows\System32\Inetsrv\Config directory. To find the location of this file on your system:
   a. Select (double-click) the **Configuration Editor** from the **Home** page for your localhost connection or any site item within that connection in the **Connections** panel.
   b. Select **Search Configuration...** from the **Actions** panel.
   c. In the **Hierarchy View**, select **ApplicationHost.config** (or any of its child items). When you do so, the **Configuration Search** dialog displays the location of the ApplicationHost.config file.
2. At the end of the `<configuration>` block (but within it), append the following `<Location>` tag:

```
<location path="localhost/csp/bin/Systems/Module.cxw">
    <system.webServer>
        <security>
           <requestFiltering>
              <hiddenSegments>
                 <remove segment="bin" />
              </hiddenSegments>
           </requestFiltering>
        </security>
     </system.webServer>
 </location>
```

3. Save the file.
4. If you are finished configuring IIS, restart it to allow configuration changes to take effect.

5.3.7 Configure the Launcher for Remote Web Server Connections

To enable the InterSystems IRIS launcher to construct valid URLs for an InterSystems IRIS instance's built-in web, you must specify the connection details for the instance within the InterSystems IRIS Server Manager. These connection details should match your web server configuration. See Define a Remote Server Connection in *System Administration Guide*.

5.3.8 Configuring IIS to Return SOAP Fault Details

An InterSystems IRIS web service that encounters an error may return an `HTTP 500` error without the associated SOAP fault details. By default, IIS returns extended error information only to local clients. However, you can modify this behavior in the `<httpErrors>` element within the configuration file web.config. To do so, add the following section to instruct IIS to dispatch detailed error information to all clients.

```
<configuration>
    <system.webServer>
       <httpErrors errorMode="Detailed" />
    </system.webServer>
</configuration>
```

Use caution with this approach as sensitive information about the hosting environment may be revealed to clients. An alternative approach that avoids the security concerns of using `errorMode="Detailed”` is to instead use the `existingResponse="PassThrough"` directive.

```
<configuration>
    <system.webServer>
       <httpErrors existingResponse="PassThrough" />
    </system.webServer>
</configuration>
```

Restart IIS after making changes to the configuration.

You can make these changes manually to the IIS web.config file. Or, for a better, less error prone, approach, use the **Configuration Editor** built into the **IIS Manager**.

1. In the IIS Manager, from the **Connections** panel on the left, select the path which corresponds to the web service. For example: **Default Web Site**, then csp.
2. In the middle panel, below the **Management** heading at the bottom, double-click on **Configuration Editor**.
3. In the **Configuration Editor** dropdown at the top labeled Section, expand system.webServer and click httpErrors.
4. Click on the value next to `existingResponse` and use the dropdown to view the options. Select `PassThrough`.
5. In the **Actions** pane on the right, click **Apply**.
6. If you are finished configuring IIS, restart it to allow configuration changes to take effect.

Further information about error handling in IIS can be found at:

https://docs.microsoft.com/en-us/iis/configuration/system.webServer/httpErrors/

5.3.9 Restarting IIS

This section describes what happens when IIS is restarted via the various control panels.

Most configuration changes can be made in real-time to an active IIS installation. However, the **Internet Information Services (IIS) Manager** control panel provides stop, start, and restart options. These are useful for the refreshing the web server configuration. However, it does not reinitialize a Web Gateway installation.

When you have made modifications to a Web Gateway installation, restart IIS in one of the following ways:

- Open the Windows **Services** manager and restart the **World Wide Web Publishing** service.
- From the command line, run the following command to stop IIS:

  ```
  sc stop W3SVC
  ```

  After the web server stops, run the following command to restart it:

  ```
  sc start W3SVC
  ```

5.3.10 Troubleshooting

This section describes problems that commonly occur in configuring third-party modules (both Native and ISAPI) to work with IIS.

The most common problem likely to be encountered is that, after reconfiguring, requests to IIS fail with the following error:

```
Service Unavailable

HTTP Error 503. The service is unavailable.
```

This usually indicates that the default **Application Pool** has terminated.

1. Open the **Internet Information Services (IIS) Manager** window.
2. In the left panel expand the top level to reveal the **Application Pools** section.

   [MACHINE_NAME] ([machine_name]\[user_name])

   Application Pools
3. Check that the Default Application Pool (`DefaultAppPool`), or whatever application pool your server is configured to use, is marked with a Status of **Started**.
4. Restart the application pool if necessary (using the options in the right panel).
5. If problems persist, look for clues in the main Windows Event Log (in the **Applications** section). In particular, check for the following error message:

   `Failed to find the RegisterModule entrypoint in the module DLL` C:\Inetpub\CSPGateway\CSPms.dll`. The data is the error.`

   This, for example, indicates that the version of Web Gateway DLLs that you are using do not implement the Native Modules interface. Either obtain later DLLs from InterSystems or configure the Web Gateway to work through the conventional ISAPI interface.

As with all software, restarting often clears transient problems: To completely restart IIS, restart the **World Wide Web Publishing** service via the main Windows Services control panel.

Do not use the **Add Wildcard Script Map** utility to map file extensions. If you do, you may see this error: `The specified module required by the handler is not in the modules list. If you are adding a script map handler mapping, the IsapiModule or the CgiModule must in the modules list.` Instead use **Add Module Mapping for *** to map file extensions using a wildcard.

If URLs with /bin in them do not work, see Manual Step for Enabling URLs with /bin

5.4 Nginx for UNIX®/Linux/macOS

5.4.1 Introduction

This page describes how to build and configure an Nginx web server for use with the InterSystems Web Gateway on UNIX®, Linux, or macOS.

Nginx is an Open Source product and the source code can be downloaded free of charge from: http://nginx.org/

Some prebuilt kits are available for Linux which are, generally, a few builds behind the latest Nginx build. However, given that extensions must be compiled into the Nginx core, it is necessary to build the web server locally from the source code in order to include support for CSP.

After the steps in this page, you can use the Web Gateway management pages to further configure the Web Gateway.

5.4.2 Assumptions

This page assumes that:

- CSP/Web Gateway web server components are installed in /opt/webgateway/bin/
- InterSystems IRIS, if installed locally, is in /opt/iris/

- The web server is installed under /opt/nginx/

If the layout is different on your system, modify the configuration directives as appropriate.

5.4.3 Installation

The Web Gateway components and the CSP static files should be installed as follows:

1. Web Gateway Network Service Daemon (NSD)

   - CSPnsd

   The default location for this binary is /opt/webgateway/bin/

2. HyperEvents Components

   CSPBroker.js

   CSPxmlhttp.js

   The default location for these files is /opt/iris/csp/broker

   If these files are to be served as static components directly by the web server, copy them to /opt/nginx/html/csp/broker

3. Miscellaneous static resources used by the Management Portal

   A number of static web resources (such as image files) are required by the Management Portal. The default location for these files is /opt/iris/csp/sys

   If these files are to be served as static components directly by the web server, copy them to /opt/nginx/html/csp/sys

5.4.4 Building the Nginx Web Server for CSP

Most of the Web Gateway functionality is provided by the NSD (CSPnsd). For CSP access, Nginx can be built and configured to communicate with the NSD through a small compiled-in module, ngx_http_csp_module.c. For convenience, all Web Gateway installations include this source file.

The build instructions given here are based on the official documentation for building Nginx under UNIX® systems:

http://nginx.org/en/docs/configure.html

The Nginx documentation stipulates that the following third party add-ons are also required:

- PCRE

  http://www.pcre.org/

- OpenSSL (for SSL/TLS)

  https://www.openssl.org/

- Zlib

  http://zlib.net/

Important: The OpenSSL toolkit you use should be compatible with your version of InterSystems IRIS; see Which TLS Versions Does My Instance of InterSystems IRIS Support?

However, it is possible to create a fully functional server without these components, provided the final installation doesn't require the functionality that would otherwise be provided by them.

A typical configuration script for building Nginx, including all optional modules listed above, is as follows:

```
./configure --prefix=/opt/nginx --with-http_ssl_module
```

This results in a default Nginx build installed under: /opt/nginx

The build process can be modified to exclude optional modules:

- OpenSSL - Remove SSL/TLS capability:

  Remove directive: `--with-http_ssl_module`

- Zlib - Remove GZIP capability:

  Add directive: `--without-http_gzip_module`

- PCRE - Remove HTTP rewrite capability:

  Add directive: `--without-http_rewrite_module`

5.4.4.1 Procedure for building Nginx for CSP

1. Unpack the source distribution under a location of your choice. For example:

   /opt/

   After unpacking, if you specify /opt/, the source code distribution is under:

   /opt/nginx-n.n.n/

2. Create a directory for the CSP extension:

   /opt/nginx-n.n.n/csp/

3. Copy the module source code (ngx_http_csp_module.c) to the directory created above.

4. In that same directory, create a configuration file called config. This file should contain the following lines:

   ```
   ngx_addon_name=ngx_http_csp_module
   HTTP_MODULES="$HTTP_MODULES ngx_http_csp_module"
   NGX_ADDON_SRCS="$NGX_ADDON_SRCS $ngx_addon_dir/ngx_http_csp_module.c"
   CORE_LIBS="$CORE_LIBS -ldl"
   ```

5. Working in /opt/nginx-n.n.n/, configure the Nginx build environment:

   ```
   ./configure --prefix=/opt/nginx
               --with-http_ssl_module
               --add-module=/opt/nginx-n.n.n/csp
   ```

   Alternatively, without the optional functionality provided by OpenSSL, ZLIB and PCRE:

   ```
    ./configure --prefix=/opt/nginx
                --without-http_rewrite_module
                --without-http_gzip_module
                --add-module=/opt/nginx-n.n.n/csp
   ```

   Note the final line containing the instructions to include the CSP module.

6. Compile Nginx:

   ```
   make
   ```

7. Install Nginx:

   ```
   make install
   ```

If successful, you should find the complete server installation under:

/opt/nginx/

5.4.5 Using the NSD with Nginx

You must configure the web server so that it recognizes requests your InterSystems IRIS application must serve and then passes those requests to the NSD.

To do so, edit the web server configuration file (nginx.conf) which is found in /opt/nginx/conf

This section describes the server configuration directives which the CSP extension module provides for configuring the web server. Issuing any of these directives within the context of a `location` block applies the directive to traffic at the path specified.

`CSPNSD_pass` *`hostname`*`:`*`portNum`*`;`

(Required.) Specifies the address (*hostname* and *port*) where the NSD is listening.

If you do not specify an NSD address for a particular path, the NSD listens at the address `127.0.0.1:7038` by default.

`CSP on;` and `CSP off;`

Enables or disables routing to CSP servers through the Web Gateway for all requests.

If you do not issue a CSP directive which applies to a particular path, no requests sent to that path are routed through the `CSP off`, and the web server does not route any requests sent to that path through the Web Gateway.

`CSPFileTypes` *`filetype1`*`[` *`filetype2`*`...];`

Enables the routing of requests for particular file types (*filetype1*, *filetype2*, and so on) to CSP servers through the Web Gateway.

For example, if you want the Web Gateway to route requests sent to the /demo/app path if (and only if) they are requesting .csp or .cls files, issue the following directive block:

```
location /demo/app {
  CSPFileTypes csp cls;
}
```

Issuing the directive `CSPFileTypes *` enables the routing of requests for all file types.

This directive does not route requests for URLs which do not request files. Therefore, this directive is inappropriate for most APIs, which often specify paths at endpoints.

`CSPNSD_response_headers_maxsize` *`size`*`;`

Specifies the maximum size of headers for an HTTP response. If response headers exceed this size, the web client receives an error.

By default, the CSP extension module applies the directive `CSPNSD_response_headers_maxsize 8k`.

`CSPNSD_connect_timeout` *`time`*`;`

Specifies the timeout for connecting to the NSD when a request is received from the web client.

By default, the CSP extension module applies the directive `CSPNSD_connect_timeout 300s`.

CSPNSD_send_timeout *time*;

Specifies the timeout for a single send operation request (such as **POST** or **PUT**). The timeout is applied only between successive send operations; it does not apply to the completion of a single transmission once it has begun.

By default, the CSP extension module applies the directive `CSP_send_timeout 300s`.

CSPNSD_read_timeout *time*;

Specifies the timeout for a single read operation (such as **GET**) upon delivery of a response. The timeout is applied only between successive read operations; it does not apply to the completion of a transmission once it has begun.

By default, the CSP extension module applies the directive `CSP_read_timeout 300s`.

5.4.5.1 Example: Enable CSP Routing for All Traffic at a Particular Path

Place the following section within the appropriate **server** configuration block to route all traffic sent to the /csp path to the Web Gateway:

```
location /csp {
CSP On;
CSPNSD_pass localhost:7038;
}
```

5.4.5.2 Example: Route Requests for InterSystems IRIS File Types to the Web Gateway

Place the following section within the appropriate **server** configuration block to enable CSP routing for requests sent to the /csp path for the InterSystems IRIS file types (.csp, .cls, .zen, and .cxw) as well as file types required to serve static elements of the Management Portal interface:

```
location /csp {
CSPFileTypes  csp cls zen cxw .jpg .gif .png .svg .css .js;
CSPNSD_pass localhost:7038;
}
```

5.4.6 Additional Configuration Required to Use IDEs

To use VS Code with the InterSystems ObjectScript extensions (or any IDE which uses the /api/atelier web application) as an ObjectScript IDE, you must perform the following additional modifications to your Nginx configuration:

1. The /api/atelier web application uses HTTP headers which include underscores. By default, Nginx silently drops these headers. You must configure your Nginx server to include them. See https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/#missing-disappearing-http-headers.
2. Features such as the debugger require a WebSocket connection. By default, Nginx is not configured to support WebSockets. For information on enabling WebSockets, refer to the Nginx documentation: (http://nginx.org/en/docs/http/websocket.html).

5.4.7 Start and Stop Nginx and the NSD

To start Nginx:

```
/opt/nginx/sbin/nginx
```

To stop Nginx:

```
/opt/nginx/sbin/nginx –s stop
```

See Operating the NSD for instructions on how to operate the NSD.

5.4.8 Deprecated: Building Nginx to Work with the Universal Modules

Important: Use of the Universal Modules with Nginx has been deprecated due to stability issues. Deployments of the Web Gateway which connect to Nginx using the NSD fully support all features, including WebSockets.

If you are currently using the Universal Modules with Nginx, InterSystems recommends upgrading to the most recent version of the Web Gateway and rebuilding your Nginx server to work with the NSD. Be sure to remove the `CSPModulePath` directive from your server configuration when you edit the server configuration file.

The following instructions serve as a reference for existing installations only.

Nginx can be built to work with the dynamically-linked Universal Modules CSPx.so (runtime) and CSPxSys.so (Web Gateway systems management). The procedure for building and configuring Nginx to work with the Universal Modules varies from the NSD-based deployment as follows:

- In step 3, copy the module source code ngx_http_csp_module_sa.c, cspapi.h, and ngx_http_csp_common.h to the specified directory, instead of ngx_http_csp_module.c.
- In step 4, the configuration file for CSP (/opt/nginx-n.n.n/csp/config) should read as follows:

```
ngx_addon_name=ngx_http_csp_module_sa
HTTP_MODULES="$HTTP_MODULES ngx_http_csp_module_sa"
NGX_ADDON_SRCS="$NGX_ADDON_SRCS $ngx_addon_dir/ngx_http_csp_module_sa.c"
```

- Add the **CSPModulePath** directive from the **http** configuration block to specify the path to the Universal Gateway Modules.

```
CSPModulePath /opt/webgateway/bin;
```

- The following directives are not supported:
  - **CSPNSD_pass**
  - **CSPNSD_response_headers_maxsize**
  - **CSPNSD_connect_timeout**
  - **CSPNSD_send_timeout**
  - **CSPNSD_read_timeout**
- The following directives are supported:
  - **CSP**
  - **CSPFileTypes**

5.5 Nginx for Windows

5.5.1 Introduction

This page describes how to build and configure an Nginx web server for use with the InterSystems Web Gateway on Windows.

Nginx is an Open Source product. The source code can be downloaded free of charge from: http://nginx.org/

Some prebuilt kits are available for Windows which are, generally, a few builds behind the latest Nginx build. However, given that extensions must be compiled into the Nginx core, it is necessary to build the web server locally from the source code in order to include support for CSP.

After the steps in this page, you can use the Web Gateway management pages to further configure the Web Gateway.

5.5.2 Assumptions

This page assumes that

- The CSP/Web Gateway components are installed in *install-dir*\csp\
- The web server is installed in C:\nginx\

If the layout is different on your system, modify the configuration directives as appropriate.

5.5.3 Installation

The Web Gateway components and the CSP static files should be installed as follows:

1. Web Gateway Network Service Daemon (NSD)
   - CSPnsd.exe (main binary)
   - CSPnsdSv.exe (Windows service)

   The default location for these files is *install-dir*\bin

   The configuration file and log file are written in this directory.
2. HyperEvents Components
   - CSPBroker.js
   - CSPxmlhttp.js

   The default location for these files is *install-dir*\csp\broker

   If these files are to be served as static components directly by the web server, copy them to C:\nginx\html\csp\broker
3. Miscellaneous static resources used by the Management Portal

   A number of static web resources (such as image files) are required by the Management Portal. The default location for these files is *install-dir*\csp\sys

   If these files are to be served directly by the web server, copy these to C:\nginx\html\csp\sys

5.5.4 Building the Nginx Web Server for CSP

Most of the Web Gateway functionality is provided by the NSD (CSPnsd[Sv].exe). For CSP access, Nginx can be built and configured to communicate with the NSD through a small compiled-in module: ngx_http_csp_module.c. For convenience, all Web Gateway installations include this source file.

Prerequisites for building Nginx:

- Microsoft Visual Studio (version 10 or higher): http://www.microsoft.com
- MSYS2 (from MinGW): https://www.msys2.org/
- Perl (preferably ActivePerl): https://www.activestate.com/products/perl/

- Mercurial source control client: https://www.mercurial-scm.org/

The build instructions given here are based on the official documentation for building Nginx under Windows:

http://nginx.org/en/docs/howto_build_on_win32.html

The Nginx documentation stipulates that the following third party add-ons are also required:

- PCRE: http://www.pcre.org/
- OpenSSL (for SSL/TLS) https://www.openssl.org/
- Zlib: http://zlib.net/

Important: The OpenSSL toolkit you use should be compatible with your version of InterSystems IRIS; see Which TLS Versions Does My Instance of InterSystems IRIS Support?

However, it is possible to create a fully functional server without these components, provided the final installation doesn't require the functionality that would otherwise be provided by them.

The default configuration script for building Nginx, including all optional modules listed above, is as follows:

```
auto/configure \
 --with-cc=cl \
 --with-debug \
 --prefix= \
 --conf-path=conf/nginx.conf \
 --pid-path=logs/nginx.pid \
 --http-log-path=logs/access.log \
 --error-log-path=logs/error.log \
 --sbin-path=nginx.exe \
 --http-client-body-temp-path=temp/client_body_temp \
 --http-proxy-temp-path=temp/proxy_temp \
 --http-fastcgi-temp-path=temp/fastcgi_temp \
 --http-scgi-temp-path=temp/scgi_temp \
 --http-uwsgi-temp-path=temp/uwsgi_temp \
 --with-cc-opt=-DFD_SETSIZE=1024 \
 --with-pcre=objs/lib/pcre-8.44 \
 --with-zlib=objs/lib/zlib-1.2.12 \
 --with-openssl=objs/lib/openssl-1.1.1k \
 --with-openssl-opt=no-asm \
 --with-http_ssl_module
```

The build process can be modified to exclude optional modules:

- OpenSSL - Remove SSL/TLS capability:

  Remove directive: `--with-http_ssl_module`

- Zlib - Remove GZIP capability:

  Add directive: `--without-http_gzip_module`

- PCRE - Remove HTTP rewrite capability:

  Add directive: `--without-http_rewrite_module`

5.5.4.1 Procedure for Building Nginx for CSP

1. Working in a MSYS2 shell, create the working directory structure suggested in the Nginx documentation:

   /opt/

2. Working in /opt, check-out the Nginx source code using the following command:

   ```
   hg clone http://hg.nginx.org/nginx
   ```

   This places the Nginx source code under: /opt/nginx/

3. Create a directory for the CSP extension:

   ```
   mkdir /opt/nginx/objs/lib/csp/
   ```

4. Copy the module source code (ngx_http_csp_module.c) to the directory created in the previous step.

5. In the same directory, create a configuration file called config. This file should contain the following lines:

   ```
   ngx_addon_name=ngx_http_csp_module
   HTTP_MODULES="$HTTP_MODULES ngx_http_csp_module"
   NGX_ADDON_SRCS="$NGX_ADDON_SRCS $ngx_addon_dir/ngx_http_csp_module.c"
   ```

6. Working in /opt/nginx/, configure the Nginx build environment:

   ```
   auto/configure --with-cc=cl --builddir=objs --prefix=
                  --conf-path=conf/nginx.conf --pid-path=logs/nginx.pid
                  --http-log-path=logs/access.log --error-log-path=logs/error.log
                  --sbin-path=nginx.exe
                  --http-client-body-temp-path=temp/client_body_temp
                  --http-proxy-temp-path=temp/proxy_temp
                  --http-fastcgi-temp-path=temp/fastcgi_temp
                  --with-cc-opt=-DFD_SETSIZE=1024 --without-http_rewrite_module
                  --without-http_gzip_module
                  --with-select_module --with-ipv6
                  --add-module=objs/lib/csp
   ```

   Note the final line containing the instructions to include the CSP module.

7. Compile Nginx. This can be done in either the current MSYS2 shell or a Visual Studio developer command prompt.

   To use the MSYS2 shell, locate the vcvarsall.bat script corresponding to your desired Visual Studio build environment and compile Nginx.

   ```
   cd /c/path/to/vcvarsall
   vcvarsall.bat
   cd -
   nmake -f objs/Makefile
   ```

   Alternatively, if you do not know where to find vcvarsall.bat, you can open a Visual Studio developer command prompt, which will set up the build environment for you. First, convert the MSYS2 path to an equivalent Windows path in the current MSYS2 shell.

   ```
   cygpath –m $(pwd)
   ```

   Then, open a Visual Studio command prompt for your desired build environment and navigate to that Windows path. Compile Nginx.

   ```
   nmake -f objs/Makefile
   ```

   If successful, you should find the server (nginx.exe) in: /opt/nginx/objs/

8. Install Nginx: The easiest way to do this is to first download and install a pre-built version of Nginx for Windows to obtain the directory structure (usually under C:\nginx\) then replace the nginx.exe file in that installation with the one created locally.

The typical directory structure for an operational Nginx installation is as follows:

```
Directory of C:\nginx

03/07/2017  09:09    <DIR>          .
03/07/2017  09:09    <DIR>          ..
26/06/2017  10:14    <DIR>          conf
26/06/2017  10:14    <DIR>          contrib
10/05/2018  12:53    <DIR>          csp
26/06/2017  10:14    <DIR>          docs
26/06/2017  10:14    <DIR>          html
10/05/2018  15:57    <DIR>          logs
04/07/2017  15:52           715,264 nginx.exe
26/06/2017  10:17    <DIR>          scgi_temp
26/06/2017  10:17    <DIR>          temp
26/06/2017  10:17    <DIR>          uwsgi_temp
```

Replace the copy of *nginx.exe* in this directory with the version created by the build procedure.

5.5.5 Configure Nginx to Invoke the NSD

You must configure the web server so that it recognizes requests for InterSystems file types and passes those requests (as well as requests for any other static files your InterSystems IRIS application serves) to the NSD for processing.

To do so, edit the web server configuration file (nginx.conf), which is found in C:\nginx\conf

This section describes the server configuration directives which the CSP extension module provides for configuring the web server. Issuing any of these directives within the context of a `location` block applies the directive to traffic at the path specified.

`CSPNSD_pass` *`hostname`*`:`*`portNum`*`;`

(Required.) Specifies the address (*hostname* and *port*) where the NSD is listening.

If you do not specify an NSD address for a particular path, the NSD listens at the address `127.0.0.1:7038` by default.

`CSP on;` and `CSP off;`

Enables or disables routing to CSP servers through the Web Gateway for all requests.

If you do not issue a CSP directive which applies to a particular path, no requests sent to that path are routed through the `CSP off`, and the web server does not route any requests sent to that path through the Web Gateway.

`CSPFileTypes` *`filetype1`*`[` *`filetype2`*`...];`

Enables the routing of requests for particular file types (*filetype1*, *filetype2*, and so on) to CSP servers through the Web Gateway.

For example, if you want the Web Gateway to route requests sent to the /demo/app path if (and only if) they are requesting .csp or .cls files, issue the following directive block:

```
location /demo/app {
  CSPFileTypes csp cls;
}
```

Issuing the directive `CSPFileTypes *` enables the routing of requests for all file types. However, REST APIs must support receiving requests at endpoints which do not specify a file (and therefore do not specify a file type). For such applications, it is not possible to invoke the Web Gateway for requests based on the file type, and you must enable the Web Gateway as the handler for all requests at the path (using `CSP on;`).

CSPNSD_response_headers_maxsize *size*;

Specifies the maximum size of headers for an HTTP response. If response headers exceed this size, the web client receives an error.

By default, the CSP extension module applies the directive `CSPNSD_response_headers_maxsize 8k`.

CSPNSD_connect_timeout *time*;

Specifies the timeout for connecting to the NSD when a request is received from the web client.

By default, the CSP extension module applies the directive `CSPNSD_connect_timeout 300s`.

CSPNSD_send_timeout *time*;

Specifies the timeout for a single send operation request (such as **POST** or **PUT**). The timeout is applied only between successive send operations; it does not apply to the completion of a single transmission once it has begun.

By default, the CSP extension module applies the directive `CSP_send_timeout 300s`.

CSPNSD_read_timeout *time*;

Specifies the timeout for a single read operation (such as **GET**) upon delivery of a response. The timeout is applied only between successive read operations; it does not apply to the completion of a transmission once it has begun.

By default, the CSP extension module applies the directive `CSP_read_timeout 300s`.

5.5.5.1 Example: Enable CSP Routing for All Traffic at a Particular Path

Place the following section within the appropriate **server** configuration block to route all traffic sent to the /csp path to the Web Gateway:

```
location /csp {
CSP On;
CSPNSD_pass localhost:7038;
}
```

5.5.5.2 Example: Route Requests for InterSystems IRIS File Types to the Web Gateway

Place the following section within the appropriate **server** configuration block to enable CSP routing for requests sent to the /csp path for the InterSystems IRIS file types (.csp, .cls, .zen, and .cxw) as well as file types for static files which the Management Portal interface uses:

```
location /csp {
CSPFileTypes  csp cls zen cxw jpg gif png svg css js;
CSPNSD_pass localhost:7038;
}
```

5.5.6 Start and Stop Nginx and the NSD

To start Nginx:

```
C:\nginx\nginx
```

To stop Nginx:

```
C:\nginx\nginx –s stop
```

See Operating the NSD for instructions on how to operate the NSD.

5.5.7 For VS Code Users: Additional Configuration Needed

If you want to connect VS Code with the InterSystems ObjectScript Extensions to an InterSystems IRIS instance which uses an Nginx web server, you must configure Nginx as follows:

1. Supported ObjectScript IDEs connect to InterSystems IRIS using the /api/atelier web application. This web application sends requests which include HTTP headers with underscores. By default, Nginx silently drops these headers. Enable Nginx to retain headers with underscores. See https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/#missing-disappearing-http-headers.
2. Features such as the debugger require a WebSocket connection. By default, Nginx is not configured to support WebSockets. Enable Nginx to support WebSockets; see http://nginx.org/en/docs/http/websocket.html

5.5.8 Deprecated: Building Nginx to Work with the Universal Modules

Important: Use of the Universal Modules with Nginx has been deprecated due to stability issues. Deployments of the Web Gateway which connect to Nginx using the NSD fully support all features, including WebSockets.

If you are currently using the Universal Modules with Nginx, InterSystems recommends upgrading to the most recent version of the Web Gateway and rebuilding your Nginx server to work with the NSD. Be sure to remove the `CSPModulePath` directive from your server configuration when you edit the server configuration file.

The following instructions serve as a reference for existing installations only.

Nginx can be built to work with the dynamically-linked Universal Modules CSPx.dll (runtime) and CSPxSys.dll (Web Gateway systems management) instead of the NSD. The procedure for building and configuring Nginx to work with the Universal Modules varies from the NSD-based deployment as follows:

- In step 4, copy the module source code ngx_http_csp_module_sa.c and ngx_http_csp_common.h to the specified directory, instead of ngx_http_csp_module.c.
- In step 5, the configuration file for CSP (/opt/nginx/objs/lib/csp/config) reads as follows:

```
ngx_addon_name=ngx_http_csp_module_sa
HTTP_MODULES="$HTTP_MODULES ngx_http_csp_module_sa"
NGX_ADDON_SRCS="$NGX_ADDON_SRCS $ngx_addon_dir/ngx_http_csp_module_sa.c"
```

- Add the **CSPModulePath** directive to the **http** configuration block to specify the path to the Universal Gateway Modules.

```
CSPModulePath install-dir/bin;
```

- For Windows, the thread stack size must be increased to 2MB. Add the following directive to the top of the Nginx configuration file (before the **http** section).

```
thread_stack_size 2000000;
```

- The following directives are not supported:
  - **CSPNSD_pass**
  - **CSPNSD_response_headers_maxsize**
  - **CSPNSD_connect_timeout**
  - **CSPNSD_send_timeout**
  - **CSPNSD_read_timeout**

- The following directives are supported:
  - **CSP**
  - **CSPFileTypes**

6

Choose Which URL Paths Route Requests Through the Web Gateway

After you extend your web server configuration with the InterSystems Web Gateway module, the next step in the process of setting up a Web Gateway connection for your InterSystems web applications is to specify which requests your web server passes along to the Web Gateway to handle, based on the request's URL's relative path (that is, omitting components such as the protocol, hostname, and port number).

However, the paths you choose to configure are also important for other stages of the routing process:

- The Web Gateway uses the request's relative path (omitting URL components such as the protocol, hostname, and port number) to determine which InterSystems IRIS application server receives the request, based on the application access profiles you define.
- The InterSystems IRIS application server determines the web application it should invoke for a given request by matching the request's relative path with the name of one of its web applications.
- To target a particular InterSystems IRIS instance on a system with multiple instances, the relative paths for which you choose to route requests must include a prefix which uniquely identifies each instance, but which does not prevent the application server from matching the request's path with one of its web applications.

This page details the relevant criteria used to determine how a request is routed at each stage, based on the relative path specified in the request's URL. Consider these criteria as you decide the URL paths you use to host your web applications.

This page also includes information about routing requests in the following situations:

- Multiple InterSystems IRIS instances hosted by a single web server.
- Multiple InterSystems IRIS instances hosted by a single web server, where the instance prefixes used to address each instance must be customized.
- Using Apache virtual hosts to route application requests using different host names.

6.1 From the Web Server

Generally, a web server applies rules of inheritance when it determines how to handle requests sent to a URL path; if no it applies the configuration directives for the most specific path which matches the given request path.

For example, if the web client sends a request to the relative URL path /Accounts/Invoices, an Apache web server checks its configuration for `<Location>` blocks in following order, applying the directives within the most specific block it finds.

1. `<Location /Accounts/Invoices>`
2. `<Location /Accounts>`
3. `<Location />`

Refer to your web server's documentation for details about its routing behavior. How you configure your web server to invoke the Web Gateway for requests at a particular URL path also varies depending on the web server you are using. See this section of the Setup Overview page for a summary; see the appropriate section of Extend Your Web Server Configuration for details.

6.2 Through the Web Gateway

The Web Gateway also applies rules of inheritance for application paths; it uses the application access profile for the most specific path which matches to a given request path.

Given the example request from the preceding section (/Accounts/Invoices), the Web Gateway checks whether valid application access profiles are defined for paths in the following order, applying the profile for the most specific path it finds:

- /Accounts/Invoices
- /Accounts
- / (root)

The Web Gateway automatically defines application access profiles for the /csp and / (root) paths.

Note: Each time you allow the installer to automatically configure your web server to serve the web applications for a new or upgraded instance, the installer overwrites the existing /csp and / (root) application access profiles so that they route requests to your most recently installed or upgraded instance. To avoid inconvenience, when you manually configure an application within the /csp path (such as /csp/foo/bar), define an application access profile for a path subordinate to /csp (such as /csp/foo or /csp/foo/bar).

Application paths are case-sensitive. Application paths cannot contain any dots (periods), because these lead to ambiguity. For example, given the path /csp/samples/menu.csp/csp/foo/bar/baz.cls, the Web Gateway could either interpret this as a request for /csp/samples/menu.csp/csp/foo/bar/baz.cls or as a REST request for /csp/samples/menu.csp (where PATH_INFO is /csp/foo/bar/baz.cls).

6.3 To an InterSystems IRIS Application Server

When an InterSystems IRIS application server receives a request from the Web Gateway, it attempts to match the request's relative URL path to the name of a web application configured on that InterSystems IRIS instance. It then calls the appropriate code for the web application. As with the web server and the Web Gateway, the application server applies rules of inheritance. Given a request for /Accounts/Invoices, the application server would attempt to call application code for an /Accounts/Invoices application first. If no such application is defined, it would attempt to call application code for an /Accounts application.

You may want to host multiple InterSystems IRIS instances on a single web server, and target certain application requests to a certain instance. Taking the previous example, perhaps your organization has a Payable instance (for tracking payments owed to vendors) and a Receivable instance (for tracking payments owed from customers). Both instances host code for the same /Accounts/Invoices application, but the data is different. To make /Accounts/Invoices data available from both

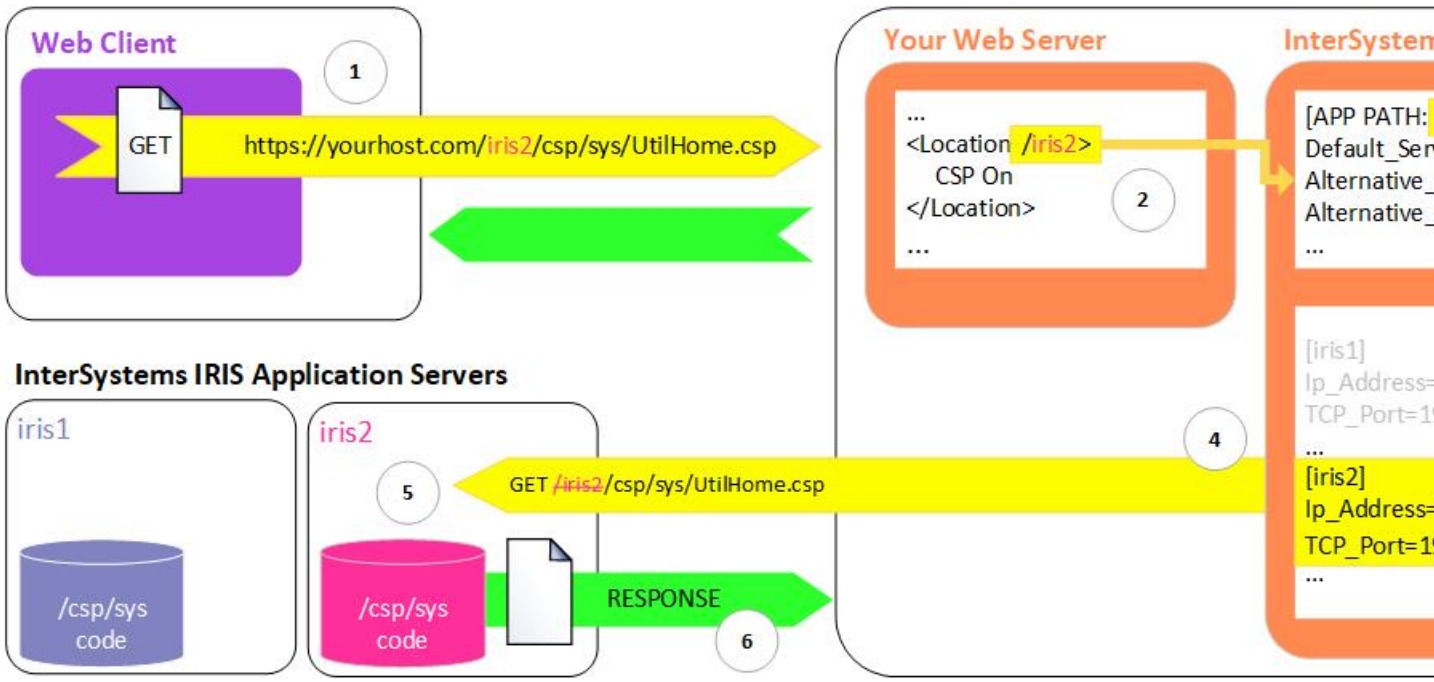
Payable and Receivable with a single web server, the request URL must identify the target instance in a way which does not interfere with the ability of the target application server to match the request to the application. The remainder of this section describes how to accomplish this.

6.3.1 Target Applications on Multiple InterSystems IRIS Servers

Some InterSystems IRIS web applications serve a function which is specific to a single InterSystems IRIS instance. For example, every instance includes the code to serve the Management Portal web application. However, the Management Portal provides an interface to administer one instance in particular.

In a system with multiple InterSystems IRIS instances, you can route requests addressed to a specific instance by configuring application access profiles for paths which include the instance's *CSPConfigName* parameter as a prefix to the application path. By default, this parameter is the name of the instance, in all lowercase characters. Before an InterSystems IRIS application server determines what application code to invoke in response to a request, it discards the value of its *CSPConfigName* from the beginning of the application path associated with the request. This allows the web server and the Web Gateway to target an instance uniquely using *CSPConfigName* as a prefix without requiring any customization to the behavior of that instance's applications.

Consider the system illustrated in the following diagram:



1. The web client requests the application path for the iris2 instance's Management Portal home page. The application path is prefixed with the default *CSPConfigName* for the target instance (`iris2`).
2. The configuration directives for the hypothetical Apache web server invoke the Web Gateway for requests sent to all relative URL paths within the /iris2 path. This includes the client's current request.
3. Within the Web Gateway configuration, an application access profile associates requests within the /iris2 path with the server access profile named `iris2`.
4. The Web Gateway sends the request to the InterSystems IRIS application server at the IP address and port specified by the server access profile `iris2`.
5. The InterSystems IRIS application server for the iris2 instance discards the first part of the request's relative path because it matches the default value for the instance's *CSPConfigName*: the instance name. Based on the remaining portion of the path (/csp/sys/UtilHome.csp) it calls the appropriate application code for the /csp/sys application.

6. The /csp/sys application code returns the response to the Web Gateway, which relays the request back to the client through the web server.

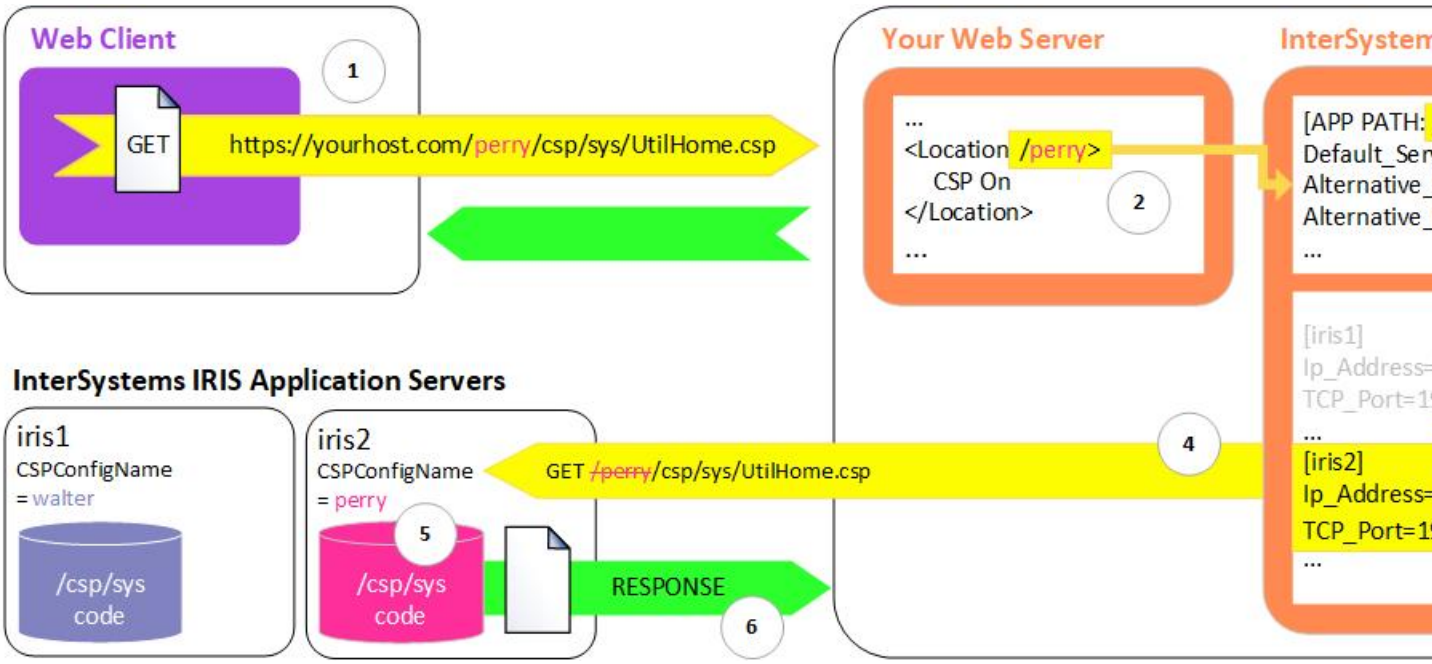
When an InterSystems IRIS installer automatically configures an instance to connect to the web server during an installation or upgrade, it adds a web server configuration directive and a Web Gateway application access profile corresponding to the instance's default *CSPConfigName* automatically. If you are configuring the Web Gateway for an instance manually, you must configure your web server and add this application access profile manually. (If you have not already done so, you may need to add a server access profile for the instance first.)

On Windows systems, additional configuration is necessary to enable the InterSystems IRIS launcher for an InterSystems IRIS instance with modified web connection details. See Define a Remote Server Connection in *System Administration Guide*.

6.3.2 Address Each InterSystems IRIS Server Using a Custom Instance Prefix

If you do not want the instance name for an InterSystems IRIS application server to appear in the web application URL, follow the procedure described in this section, substituting your own names and application access requirements for those provided.

For example: assume you want to serve the Management Portal for two instances named iris1 and iris2, but that you would like to use `walter` as the substitute name for iris1 and `perry` as the substitute name for an instance named iris2, as depicted in the following image:



To serve the Management Portal for each instance using these substitute names, you would perform the following steps:

1. Ensure that both of the following conditions are met:

   a. Your web server is invoking the Web Gateway for requests that are sent to /walter/csp/sys and /perry/csp/sys (or parent directories).

   b. The Web Gateway configuration includes server access profiles for both `iris1` and `iris2`, named accordingly.

2. In an ObjectScript shell on the `iris1` instance, run the following command:

```
d $System.CSP.SetConfig("CSPConfigName","walter")
```

3. In an ObjectScript shell on the `iris2` instance, run:

```
d $System.CSP.SetConfig("CSPConfigName","perry")
```

4. Using the Web Gateway management pages, add an application access profile for the application path /walter/csp/sys (or /walter/csp, or /walter) with a **Default Server** of `iris1`.
5. Add an application access profile for the application path /perry/csp/sys (or /perry/csp, or /walter) with a **Default Server** of `iris2`.

`CSPConfigName` also accepts a comma delineated list for CSP configuration names. This allows you to have multiple configuration names instead of a single one. For example:

```
d $System.CSP.SetConfig("CSPConfigName","perry,perry1,perry2,perry3")
```

To see other CSP global parameters, enter `%SYS>d $system.CSP.DisplayConfig()`.

On Windows systems, additional configuration is necessary to enable the InterSystems IRIS launcher for an InterSystems IRIS instance with modified web connection details. See Define a Remote Server Connection in *System Administration Guide*.

6.3.3 Configuring Apache Virtual Hosts

As an alternative to the instance prefix mechanism described in the previous sections, the Web Gateway supports using Virtual Host names to serve applications on multiple InterSystems IRIS instances. Apache Virtual Hosts provide a way of transparently serving different applications at different host names using a single web server. Refer to the Apache documentation for details about Virtual Hosts: https://httpd.apache.org/docs/2.4/vhosts/

For example, you can configure a single Apache web server to host InterSystems IRIS applications at two distinct web sites: `www.virtualhost1.com` and `www.virtualhost2.com`.

If you have configured the web server to invoke the Web Gateway for requests to a virtual host name within the web server's global configuration, the Web Gateway can route requests for a virtual host name to a specific InterSystems IRIS instance. Simply create an application access profile for an application path beginning with two forward slashes (`//`).

The following configuration procedure allows the Web Gateway to send requests for `www.virtualhost1.com/csp/sys/UtilHome.csp` can to the Management Portal home page for an instance named IRISserv1 and requests for `www.virtualhost2.com/csp/sys/UtilHome.csp` to the Management Portal home page for an instance named IRISserv2:

1. Navigate to the Web Gateway management pages main menu.
2. Create server access profiles for `irisserv1` and `irisserv2`.
3. Create application access profiles for //virtualhost1.com/csp/sys/ and //virtualhost2.com/csp/sys/. Configure them as follows:
   - Set the **Server 0** for path //virtualhost1.com/csp/sys/ to the server access profile `irisserv1`.
   - Set the **Server 0** for path //virtualhost2.com/csp/sys/ to the server access profile `irisserv2`.

Note: Although the Web Gateway supports the use of virtual host names in application access profiles, issuing Apache configuration directives to invoke the Web Gateway (that is, `CSPFileTypes` and `CSP On/Off`) within a `<VirtualHost>` directive block is not supported and will yield an error. In other words, you cannot enable the Web Gateway for the desired Virtual Hosts alone; you must enable the Web Gateway within the web server's global configuration.

On Windows systems, additional configuration is necessary to enable the InterSystems IRIS launcher for an InterSystems IRIS instance with modified web connection details. See Define a Remote Server Connection in *System Administration Guide*.

7

Overview of the Web Gateway Management Pages

The Web Gateway provides a set of management pages that you can use to configure and monitor the Web Gateway. This page describes how to access these pages and how to localize them, and provides an overview of the options in them.

7.1 Accessing the Web Gateway Management Pages

By default, only clients local to the Web Gateway's hosting computer can access the Web Gateway management pages, so that the browser you use to access these pages must be running on the same machine as the web server and the Web Gateway.

When this is the case, the URL for the Web Gateway management pages has the following form:

```
http://localhost/csp/bin/Systems/Module.cxw
```

If you have configured your web server to listen over non-standard port, append `:<portNum>` after the hostname, where *<portNum>* is the port number (for example `localhost:11555`).

Bookmark this URL for easy access.

Note: For Apache web servers, URL paths and file names are case-sensitive.

When you try to access the Web Gateway management pages, you are asked for credentials for the CSP system user. Look for the username in the configuration file. The password is the one that you entered during the InterSystems IRIS installation. If you forget the password, see Security.

Tip: Note that you can also access these pages from the Management Portal for any InterSystems IRIS instance connected to the Web Gateway. Simply navigate to **System Administration** > **Configuration** > **Web Gateway Management**. The same considerations apply with respect to client addresses.

7.2 Enabling Access from Additional Client Addresses

You can add additional clients to the list of authorized administrators by adding the client IP addresses to the `System_Manager` parameter in the SYSTEM section in the configuration file. This parameter represents a comma- or

plus-separated list of clients (by IP address) who may access the Web Gateway management pages. The directive shown below grants access to three remote clients in addition to the default local access.

```
[SYSTEM]
System_Manager=190.8.7.6, 190.8.7.5, 190.8.7.4
```

For new Gateway installations, for which there is no local browser available, manually edit the configuration file and add the *System_Manager* parameter, which is equivalent to the **Systems Manager Machines** setting, found under the **Default Parameters** section of the Web Gateway management pages. You can specify wildcard and numeric ranges in the entries for this parameter.

Note: If you attempt to load the Web Gateway management pages, and the browser fails to load the page, giving an error **You are not authorized to use this facility**, this is likely due to the *System_Manager* setting blocking access to your IP address.

The following example indicates that the last part of the IP address can take the value of a number between 4 and 6 inclusive.

```
[SYSTEM]
System_Manager=190.8.7.4-6
```

The previous example is a more convenient way of writing:

```
[SYSTEM]
System_Manager=190.8.7.6, 190.8.7.5, 190.8.7.4
```

You can also use wildcards, such as, in this example:

```
[SYSTEM]
System_Manager=190.8.7.*
```

The following directive grants access to all clients:

```
[SYSTEM]
System_Manager=*.*.*.*
```

However, it is not recommended to use such a directive on operational systems; this approach does not provide strong security, because client IP addresses can be spoofed.

The use of a proxy between the client and the web server/Gateway installation effectively translates all client IP addresses to that of the proxy. In this scenario, you would have to either specify the proxy's IP address as a Gateway Systems Manager (which would effectively grant access to all web users coming in through the proxy) or, preferably, enable the designated systems managers to bypass the proxy layer altogether.

The IP-based scheme, while useful as a first line of defense, should not be relied upon as the sole means through which access to the Web Gateway management pages is controlled – certainly not for CSP installations that are available over the internet. For production systems, it is recommended that you use the hosting web server configuration to control access to the Web Gateway systems management modules.

7.3 Available Options

The following table shows the options available on the Web Gateway Management Main Menu page.

Menu Item	Action
About Web Gateway	Shows information about the Web Gateway, including the version of the InterSystems IRIS distribution, the Web Gateway build number, the version of OpenSSL that is loaded, the version of the hosting web server, the active interface, the name and location of the Web Gateway configuration file (CSP.ini), and the name and location of the event log (CSP.log). The Web Gateway build number is made up of two numeric components. The first number indicates the version of InterSystems IRIS. The second number is the internal Web Gateway build number.
System Status	Displays the status of active server connections. Also allows you to close connections and clear the Web Gateway cache.
Test Server Connection	Tests the connection to an InterSystems IRIS server by opening a stateless session.
View Event Log	Allows you to view information in the Web Gateway Event Log, as well as clear its contents. This log is a file maintained on the web server host.
View HTTP Trace	Provides an interactive view of the HTTP requests and responses processed by the Web Gateway.
Default Parameters	Allows you to configure the Web Gateway on a specific web server. Also, it allows you to customize CSP responses to errors and other conditions.
Server Access	Configures Web Gateway access to a specific InterSystems IRIS server.
Application Access	Configures the access to an application according to the application path. Path, in this context, refers to the path contained within the application URLs.
Back to Management Portal	Returns to the InterSystems IRIS Management Portal page.

These pages include a **Help** button.

7.4 Localization

Localization of the Web Gateway management pages is based on the contents of the file CSPres.xml, if it is installed. If no localization file is present, the Web Gateway management pages use the embedded English text. The language settings of the browser have no influence on this mechanism.

You can support alternative languages by installing the appropriate text resource file as a file named CSPres.xml in the Web Gateway's home directory. When the Web Gateway starts or restarts, it loads the text resources found in CSPres.xml and the Management forms then appear in the chosen language.

To create a CSPres.xml file, rename the appropriate CSPres_xx.xml file in the InterSystems IRIS bin directory to CSPres.xml.

For example, to convert to Spanish:

1. Rename CSPres_es.xml to CSPres.xml
2. Restart the web server. You must restart because the language text affects the CSP module for the given web server.

To convert back to English:

1. Rename CSPres.xml back to CSPres_es.xml
2. Restart the web server.

8

Define a Server Access Profile for Your InterSystems IRIS Instance

This page describes how to configure server access profiles for the InterSystems Web Gateway. Server access profiles enable the InterSystems Web Gateway to establish and maintain a connection with the InterSystems IRIS® application server which hosts your web application.

Each InterSystems IRIS instance accessed by the Web Gateway must be defined in a server access profile. Any unspecified optional parameters or custom system forms are automatically inherited from the Web Gateway default settings. After you have defined server access profiles for your InterSystems IRIS application servers, you can define application access profiles to associate application paths with their corresponding application servers.

For these and other configuration tasks, use the Web Gateway management pages or Web Gateway Registry methods. The Web Gateway maintains this configuration information in the CSP.ini.

Important: Except in containerized deployments where it may be necessary to edit the CSP.ini file directly, InterSystems recommends restricting access to the CSP.ini file and performing all Web Gateway configuration using the Web Gateway management pages.

8.1 Add a Server Access Profile

To allow the Web Gateway to connect to an InterSystems IRIS application server, define a server access profile to identify the InterSystems IRIS server within the Web Gateway configuration. To do so:

1. From the Web Gateway management pages main menu, select **Server Access**.
2. Select **Add Server**. The second configuration screen appears. Note that many parameter fields have default settings.
3. In the **Server Name** text box, enter a unique, descriptive name for the server. This logical name is used to identify the server configuration in the CSP configuration file.
4. Enter the system parameters (described below) for this server access profile.
5. Select **Save Configuration**.

8.1.1 Server Access Parameters

The set of base server configuration parameters are as follows:

Server Configuration Parameter	Function
Server Name	Logical name to identify this server access profile in the CSP configuration file.
Service Status	Allows you to enable and disable this server within your Web Gateway configuration(default is Enabled).
IP Address	The DNS host name or IP address (physical or virtual) of the InterSystems IRIS server to connect to.
Superserver TCP Port	The TCP port number on which the InterSystems IRIS server is listening for incoming connections. This is the TCP port number of the InterSystems IRIS superserver which is 1972 by default, but may be different if multiple instances are deployed on the same system.
Configuration is Mirror Aware	Configures a mirror primary as a server to access mirrored databases. In a failover or disaster recovery, the connection is redirected. By default, not selected. *Note*: If you have configured a mirror VIP, do not configure a mirror aware Web Gateway, which causes the Web Gateway to ignore the VIP. Instead, simply configure the Web Gateway to connect to the VIP like any other client. In general, use of a mirror aware Web Gateway is the appropriate choice only in unusual circumstances. To configure, enter the IP address of one of the failover members. From this failover member, the Web Gateway obtains a list of the failover and disaster recovery (DR) async members in the mirror and connects to the current primary based on this list (and not the VIP even if one is configured). The CSP connection fails until a primary is found. Once the connection is established, if the mirror fails over, the Web Gateway changes the connection to the new primary. If no primary can be found among the failover members, the Web Gateway attempts to find one among the DR asyncs in the list, which enables it to reestablish the connection when a DR async is promoted to primary in a disaster recovery situation. For details, see Redirecting Application Connections Following Failover or Disaster Recovery in Mirroring in the *High Availability Guide*.

8.1.2 Stateless Parameters

The set of parameters relevant to stateless connections are as follows:

Stateless Parameter	Function
Minimum Server Connections	The Web Gateway implements process affinity. This means that it always attempts to reconnect sessions to the same InterSystems IRIS process that serviced its previous request if possible. This parameter specifies the minimum number of connections that the Web Gateway should make to the InterSystems IRIS server before starting to share the connections among many clients. The higher this number, the more effective process affinity is. The default value is 3.

Stateless Parameter	Function
Maximum Server Connections	This is the absolute maximum number of connections that the Web Gateway is allowed to make to the InterSystems IRIS server. If concurrent usage exceeds this number, the Web Gateway starts to queue requests. Requests remain in the queue until an InterSystems IRIS connection becomes available to service the request or the *Queued Request Timeout* is exceeded. This is unspecified by default, indicating that the only hard maximum is the number of maximum connections for the Web Gateway, which is 1024 by default.
Maximum Connections per Session	This represents the maximum number of connections to InterSystems IRIS that can be concurrently used by an individual session. The default value is 3.

8.1.3 Connection Security Parameters

Connection Security settings are required by the Web Gateway to access the InterSystems IRIS application server. These parameters are discussed in Protecting Web Gateway Connections to InterSystems IRIS. The set of parameters relevant to connection security are as follows:

Connection Security Parameter	Function
Connection Security Level	Level of security required for connecting to the InterSystems IRIS server. Select one of the options: • Password • Kerberos • Kerberos with Packet Integrity • Kerberos with Encryption • SSL/TLS
Username	Username required by the Web Gateway for connecting to the InterSystems IRIS server.
Password	Password required by the Web Gateway for connecting to the InterSystems IRIS server. Alternatively, on UNIX®/Linux/macOS systems, this field can specify an operating system command to retrieve the password programmatically, within braces (`{}`).
Password (Confirm)	When you create a new password, confirm the new password by entering it again.
Product	Product being connected to (InterSystems IRIS).
Service Principal Name	Service principal name. A Generate button is provided for creating a default name with respect to the target InterSystems IRIS server.
Key Table	Full path to the Key Table file.

8.1.4 SSL/TLS Parameters

The following parameters are relevant only to installations using SSL/TLS to secure connections between the Web Gateway and InterSystems IRIS.

SSL/TLS Parameter	Function

SSL/TLS Parameter	Function
Minimum SSL/TLS Protocol Version	Minimum version of the SSL/TLS protocol to use. The following options are provided: • TLSv1.0 • TLSv1.1 • TLSv1.2 • TLSv1.3 (on platforms where it is supported) On platforms where TLSv1.3 is supported, the default value is `TLSv1.2`. Otherwise, the default value is `TLSv1.1`.
Maximum SSL/TLS Protocol Version	Maximum version of the SSL/TLS protocol to use. The following options are provided: • TLSv1.0 • TLSv1.1 • TLSv1.2 • TLSv1.3 (on platforms where it is supported) On platforms where TLSv1.3 is supported, the default value is `TLSv1.3`. Otherwise, the default value is `TLSv1.2`.
SSL/TLS Key Type	The type of SSL/TLS key file (based on the algorithm used to generate it). The following options are provided: • DSA — Digital Signature Algorithm • RSA — Rivest, Shamir, and Adelman (inventors of the algorithm) The default is RSA.
Require Peer Certificate Verification	If checked, requires peer certificate verification for this installation.
SSL/TLS Cipher Suites (TLSv1.2 and below)	Cipher suites for TLSv1.2 and below. The default is `ALL:!aNULL:!eNULL:!EXP:!SSLv2`.
SSL/TLS Cipher Suites (TLSv1.3)	Cipher suites for TLSv1.3. The default is `TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256`. Available only on platforms where TLSv1.3 is supported.
SSL/TLS Certificate File	The full path to the SSL/TLS certificate file for the Web Gateway. Supported file formats for certificate files are the same as those supported for InterSystems IRIS TLS Configurations. `Example: C:\InterSystems\certificates\clicert.pem`
SSL/TLS Private Key File	The full path to the private key associated with the Web Gateway's SSL/TLS certificate. Supported file formats for certificate files are the same as those supported for InterSystems IRIS TLS Configurations. `Example: C:\InterSystems\certificates\clikey.pem`

SSL/TLS Parameter	Function
SSL/TLS CA Certificate File	The full path to the certificate for Certificate Authority (CA) for the Web Gateway's certificate. Supported file formats for certificate files are the same as those supported for InterSystems IRIS TLS Configurations. `Example: C:\InterSystems\certificates\cacert.pem`
SSL/TLS Private Key Password	The password to the SSL/TLS Private Key. Alternatively, on UNIX®/Linux/macOS systems, this field can specify an operating system command to retrieve the password programmatically, within braces (`{}`).

8.1.5 Optional Parameters

The descriptions of the Optional Parameters are given in Configuring Default Parameters, If any of these parameters is blank, its value is inherited from the Web Gateway global configuration described in Connections to InterSystems IRIS.

8.1.6 Error Pages

The Error Pages parameters let you customize the Web Gateway responses. If not specified, the parameters are inherited from the global configuration. For a description of each parameter, see Custom Error Pages.

8.2 Copy a Server Access Profile

You can quickly configure a new server access profile by copying an existing server access profile. Having done this, both configuration entries are identical, except for the server name. You can then edit the second configuration and make changes to it (such as changing the IP address).

Tip: This feature is also useful for fine-tuning a configuration. By creating a second (temporary) configuration for a server, you can test parameter changes without worrying about losing the original configuration.

To copy an existing server access profile:

1. From the Web Gateway management pages main menu, select **Server Access**.
2. At the **Server Access** screen, select an existing server name.
3. Select the **Copy Server** option.
4. Select **Submit**. The second configuration screen appears.
5. In the **Server Name** text box, enter a unique, descriptive name for the new server.
6. Select **Save Configuration**.

8.3 Disable Access to an InterSystems IRIS Server

Use this facility to prevent users from accessing a configured InterSystems IRIS server through this Gateway installation.

To disable access to a server:

1. From the Web Gateway management pages main menu, select **Server Access**.
2. At the **Server Access** screen, select an existing server name.
3. Select the **Edit Server** option.
4. Select **Submit**. The Server configuration screen appears.
5. For the **Server Status** parameter, select **Disabled**.
6. Select **Save Configuration**.

To re-enable access, repeat the procedure and select **Enabled** at Step 5.

8.4 Delete a Server Access Profile

To delete a server access profile:

1. From the Web Gateway management pages main menu, select **Server Access**.
2. At the **Server Access** screen, select a server name.
3. Select the **Delete Server** option.
4. Select **Submit**.
5. Confirm by selecting **YES : DELETE**.

9

Define an Application Access Profile for Your Web Application Path

This page describes how to configure application access profiles for the InterSystems Web Gateway. Application access profiles determine where the Web Gateway routes requests for a certain web application path. An application access profile can specify the server access profile for one target application server, or it can specify a series of server access profiles as part of a load-balancing or failover scheme.

To route a request for a web application, you must configure an application access profile for a path which contains the request's URL path. For a given request, it may be possible to route requests to a given web application in multiple ways. See this page for guidance.

For these and other configuration tasks, use the Web Gateway management pages or Web Gateway Registry methods. The Web Gateway maintains configuration information in the CSP.ini.

Important: Except in containerized deployments where it may be necessary to edit the CSP.ini file directly, InterSystems recommends restricting access to the CSP.ini file and performing all Web Gateway configuration using the Web Gateway management pages.

9.1 Add an Application Access Profile

To specify where and how the Web Gateway should route requests for a certain application path, define an application access profile for that path within the Web Gateway configuration. To do so:

1. On the Web Gateway management pages main menu, select **Application Access**.
2. Select **Add Application**. Note that many parameters have default settings.
3. In the **Application Path** text box enter a unique path for the application. This path is the path which appears in the application URLs.
4. Enter the other configuration path and server parameters (described in the tables below) for this application.
5. When you have finished, select **Save Configuration**. Changes you make to the application configuration take effect as new user sessions are created for that application path. Existing users are unaffected.

9.1.1 Application Access Profile Configuration Parameters

The set of base parameters are as follows:

Parameter	Function
Service Status	Enable and disable access to an application via the application path (default is Enabled).
Web Server Physical Path	Path to the corresponding directory on the web server. This setting is particularly important for Microsoft IIS systems where each path configured must be set up as a virtual directory under the web server configuration. Each virtual directory defined within IIS must have a physical path associated with it. The purpose of this additional configuration procedure for IIS is to allow the paths used by InterSystems IRIS to be defined with execute permissions. The default is for execute permissions to be denied.
Extra CGI Environment Variables	Comma-separated list of additional CGI environment variables to be returned to the InterSystems IRIS environment with each and every request. The commonly-used CGI environment variables are automatically sent with each request. Enter the wildcard character (*) to instruct the Web Gateway to send all environment variables supplied by the web server to the InterSystems IRIS server with each request.
Process with this class	Process files in this path with the specified class. This allows you to build your own request handlers in CSP.
GZIP Compression	Enable or disable GZIP compression for all CSP pages returned in this path (default is Disabled).
GZIP Minimum File Size	Minimum response size, in bytes, for which GZIP compression is invoked. Default is 500 bytes.
GZIP Exclude File Types	This is a list of file types to be excluded from GZIP compression. Files to be excluded can be listed by MIME type (such as image/jpeg) or by common extension (such as jpeg). By default, these common (natively compressed) image files are excluded: `GZIP Exclude File Types: jpeg gif ico png gz zip mp3 mp4 tiff` Separate additional types or extensions with a space.

Parameter	Function
Response Size Notification	This parameter provides configurable control over the method used by the Web Gateway to notify clients of the amount of data contained in each response. Web clients typically require some form of response size notification if HTTP KeepAlive connectivity is used. Under these circumstances, the Web Gateway defaults to using chunked transfer encoding, provided HTTP v1.1 is in use. If an earlier HTTP protocol is in force it buffers the response data received from InterSystems IRIS and generate a content-length header instead. Also, in cases where the entire response fits into one output buffer a content-length header is generated instead of using chunked transfer. There are scenarios in which it is desirable to instruct the Web Gateway to specifically use one method or the other. For example, in cases where HTTP v1.1 is used but some intermediary (such as a proxy) is unable to properly support chunked transfer. Also, while not sending any form of size notification (such as, where the *close connection* event is used as the response terminator) should be supported by all web clients, it is nevertheless recommended as 'good practice' that all responses should be accompanied by some form of size notification. Indeed, some clients require this. The following options are provided: • Chunked Transfer Encoding and Content Length (the default) • Chunked Transfer Encoding • Content Length This parameter is supplemented with a check box to instruct the Web Gateway to always generate a size notification for all requests regardless of whether or not KeepAlive is used. There is a 500 kilobyte limit on the size of HTTP responses that specify a content-length header, as opposed to chunked responses. If you exceed this limit, a warning message appears in the CSP log: `WARNING: Unable to generate a 'Content-Length' header directive for this oversize response (Current size: size; Maximum buffer size allowed: 500000)`
KeepAlive	Enable or disable HTTP **KeepAlive** connectivity for this path. Default is `No Action` in which case the **KeepAlive** status is determined by the HTTP response headers for each request.
Non-Parsed Headers	Enable or disable **Non-Parsed Headers** protocol for this path. Default is **Enabled** in which case HTTP response headers are streamed directly back to the client. If this property is disabled, the response headers are submitted back to the hosting web server. This gives the web server the opportunity to parse the headers and invoke any output filters that may be indicated. For example, the Apache Group's mod_deflate facility. Note that for the Apache web server, if keep-alive is enabled, then the response headers are submitted back to Apache regardless of the Non-Parsed Headers setting.

9.1.2 Server Parameters

You can define a list of InterSystems IRIS servers to use for an application and the purpose for which they are to be used.

Parameter	Function
Use Alternate Servers For	The first server listed, **Server 0**, is the default InterSystems IRIS server. It is used first. Other listed servers can be used for load balancing or failover, depending on the option checked. • **Fail-Over** If the first server fails (becomes unavailable), use an alternative. • **Load-Balancing and Fail-Over** . If the first server fails, use a server that is configured as either failover or load-balancing.
Server #:	List of servers. The configuration screen initially shows only three server slots, but additional slots appear that enable you to define any number of alternative servers. Each server can be checked as **Enabled** or **Disabled**. Default is always **Enabled**. See Load Balancing and Failover Between Multiple InterSystems IRIS Server Instances for more information.

9.2 Copy an Application Access Profile

You can quickly configure a new application path by copying the application access profile for an existing path and editing it.

Tip: This feature is also useful for fine-tuning a configuration. By creating a second (temporary) application access profile for an application path, you can test parameter changes without worrying about losing the original configuration.

To copy an existing application access profile:

1. From the Web Gateway management pages main menu, select **Application Access**.
2. On the **Application Access** screen, select an existing application path.
3. Select **Copy Application**.
4. Select **Submit**.
5. In the **Application Path** text box, enter a new and unique application path.
6. Select **Save Configuration**. The new application configuration takes effect as new user sessions are created for the new application path. Existing users are unaffected.

9.3 Disable Access via an Application Path

Use this facility to prevent users accessing a configured application through this Web Gateway installation.

To disable access via an application path:

1. From the Web Gateway management pages main menu, select **Application Access**.

2. At the **Application Access** screen, select an application path.
3. Select **Edit Application**.
4. Select **Submit**. The configuration screen for the application path appears.
5. For the **Application Status** parameter, select **Disabled**.
6. Select **Save Configuration**.

To re-enable access, repeat the procedure and select **Enabled** at Step 5.

9.4 Delete an Application Access Profile

To delete an existing application access profile:

1. From the Web Gateway management pages main menu, select **Application Access**.
2. At the **Application Access** screen, select an application path.
3. Select the **Delete Application** option.
4. Select **Submit**.
5. Force a restart of all the applications by restarting the web server.

10

Configure System–Wide Parameters for the Web Gateway

When you configure the InterSystems Web Gateway to access a particular InterSystems IRIS® instance, any optional parameters and custom system forms you do not specify within the instance's server access profile are automatically inherited from the set of default parameters specified globally (system-wide) for the Web Gateway. For example, if you do not set a **Server Response Timeout** parameter within an instance's server access profile, that instance inherits the global **Server Response Timeout** setting.

This page describes how to configure the global (system-wide) configuration parameters for the Web Gateway. Other articles describe how to configure server access profiles (for InterSystems IRIS instances) and application access profiles (for the web applications available on those instances).

10.1 Ways to Configure Web Gateway Parameters

Within the Web Gateway management pages, the **Default Parameters** page allows you to modify all the global configuration parameters for the Web Gateway. (Note that you must be a system manager to use this option.) In general, InterSystems recommends configuring the Web Gateway's global parameters using this interface. The rest of this page assumes that this is the method you are using.

Alternatively, the Web Gateway Registry's %CSP.Mgr.GatewayMgr class provides a method, SetDefaultParms, which allows you to configure the global parameters for a Web Gateway programmatically from any connected InterSystems IRIS instance.

The Web Gateway maintains these system-wide default parameters in the CSP.ini file. The CSP.ini provides analogous names for the parameters described below; use the analogous parameter names to configure the Web Gateway using the Web Gateway Registry and the CSP.ini file.

Important: Except in containerized deployments where it may be necessary to edit the CSP.ini file directly, InterSystems recommends restricting access to the CSP.ini file and performing all Web Gateway configuration using the Web Gateway management pages or Web Gateway Registry methods.

If you modify the Web Gateway's global parameters by using the Web Gateway Registry or by editing the CSP.ini file, you must force the Web Gateway to reload its configuration for changes to take effect.

10.2 Web Gateway (General Settings)

This section of the Web Gateway **Default Parameters** management page contains the following parameters for configuring the Web Gateway's general settings:

Instance Host Name

This is the network host name for this particular instance of the Web Gateway. The Web Gateway generates a default value which is shown beneath the text box. The value of this parameter is transmitted to InterSystems IRIS with the request data as system variable CSPIHN. Your application can use the value to access management services provided by the Web Gateway over the network.

The format for this parameter is: `server_name:port`

Maximum Connections

Maximum number of connections to InterSystems IRIS that can be created from this Gateway instance. The default value is set to 1024. Increasing this value allows better application responsiveness if an application uses more connections but may also result in heavier server resource utilization.

Changes to the Maximum Connections parameter only take effect after a restart of the Web Gateway (or the hosting web server).

Maximum Cache Size

Maximum amount of shared memory to be reserved for the purpose of caching CSP response data.

The cache size may be specified as a number with no suffix for bytes, a number followed by **K** for kilobytes, or a number followed by **M** for megabytes.

The default value for this parameter is 256K. This value can be raised or lowered as required.

Changes to the Maximum Cache Size parameter only take effect after a Gateway (or hosting web server) restart.

Web Server ID Cookie

Suppresses the Web Server ID Cookie (*CSPWSERVERID*). It can be set to:

- Enabled (Default)
- Disabled

The Web Server ID Cookie is used to enable load balancers to implement passive cookie affinity for web applications. However, there are situations where it is desirable to suppress the automatic generation of this cookie. For example, in proxy applications where the web request is transparently passed to other servers for processing.

The Web Server ID Cookie is not dispatched when returning resources that are deemed to be static (i.e. images and JS files). In this context, static files include all responses generated by InterSystems IRIS that are not accompanied by a Web Server ID Cookie. An exception to this rule is made for cases where the application is configured to `Never` use session cookies. In this case the Web Server ID Cookie is included with all responses (as before).

10.3 Security

If a username and password are defined in this section of the Web Gateway **Default Parameters** management page, then all system managers must provide this username and password to access the Web Gateway management pages.

If you forget the password, use the following procedure to set a new password:

1. Edit the CSP.ini to specify a new value for the **Username** and **Password**; make the changes in the SYSTEM of the file. You can specify the value of the password in plaintext.
2. Restart the web server. The web server reloads the configuration and updates the file to hold the password hash instead of the plaintext password.

You can now log in to the web server with the updated username and password.

```
[SYSTEM]
Username=cm
Password=1Bx4tt88mttAWaf7isJg3Urqc2zE
```

You can configure the following Web Gateway security parameters from the management page:

Access to these forms

Enable or disable access to the Web Gateway management pages option. The default is **Enabled**. When access is **Disabled** you cannot re-enable access using the Web Gateway management pages. To re-enable access, manually edit the configuration file. Set the `SM_Forms` parameter to `Enabled` in the `SYSTEM` section of this file.

```
[SYSTEM]
SM_Forms=Enabled
```

Username

Username required to access the Web Gateway management pages.

Password

Password required to access the Web Gateway management pages.

Password (Confirm)

When the password is modified, confirm the new value here.

Session Timeout

The amount of idle time (in seconds) that an active Systems Management session remains logged on. After this time has expired, the management session expires and the manager is automatically logged out of the Web Gateway management pages.

System Manager Machine/s

Defines a list of client machines (by IP address) through which you can access these Systems Management options. Any client with System Manager access can add or delete access to any CSP system, change any setting in the configuration file, and close down any active sessions. The addresses are separated by either a comma or a plus sign. In this example, two clients have System Manager access:

```
127.0.0.1, 45.123.231.12
```

If this field is undefined, only a client operating on the same machine as the Web Gateway (that is, the web server host) can configure CSP. See Enabling Access from Additional Client Addresses for more information.

This field is supplemented with a check box (**Override Username and Password**) which, if checked, allows listed client machines to be exempt from entering a username and password to gain access to the Management Forms.

Custom Login Form

Defines a custom login form that controls access to the Web Gateway management pages. This parameter can be either the full path to a physical file or a link which enables the hosting web server to serve the form.

Examples:

```
C:\Inetpub\wwwroot\login.html
/login.html
```

If a physical file name is specified then the Web Gateway retrieves and dispatches the form to the client. Otherwise, it sends an 'HTTP Redirect' response header to enable the client to request the form directly from the hosting web server. The custom form must implement an HTTP POST request to login Gateway Administrators.

The essential form fields are shown below:

```
<FORM METHOD=POST ACTION="/csp/bin/Systems/Module.cxw">
<INPUT TYPE=HIDDEN NAME="CSPSYS" VALUE="17">
<INPUT TYPE=HIDDEN NAME="CSPSYSsmSection" VALUE="SYSTEM">
<INPUT TYPE=TEXT NAME="CSPUNM" SIZE='20' VALUE="">
<INPUT TYPE=PASSWORD NAME="CSPPWD" SIZE='20' VALUE="">
<INPUT TYPE=SUBMIT NAME="CSPSYSbOK" VALUE="Login">
```

Where CSPUNM is the username and CSPPWD the password. The text assigned to the Login (submit) button (shown as 'Login' above) can be changed.

A simple but complete example is shown below:

```
<html>
<head>
<title>Web Gateway Management</title>
</head>
<h2>Web Gateway Management</h2>
<FORM METHOD=POST ACTION="/csp/bin/Systems/Module.cxw">
<INPUT TYPE=HIDDEN NAME="CSPSYS" VALUE="17">
<INPUT TYPE=HIDDEN NAME="CSPSYSsmSection" VALUE="SYSTEM">
<BR>
Username:
<INPUT TYPE=TEXT NAME="CSPUNM" SIZE='20' VALUE="">
<BR>
Password:
<INPUT TYPE=PASSWORD NAME="CSPPWD" SIZE='20' VALUE="">
<BR>
<INPUT TYPE=SUBMIT NAME="CSPSYSbOK" VALUE="Login">
</form>
</html>
```

10.4 Connections to InterSystems IRIS

This section describes parameters related to maintaining connections to InterSystems IRIS.

Server Response Timeout

The maximum number of seconds allowed for the target InterSystems IRIS server to respond to a request from the web server. The timeout refers to a period of no activity, so, for example, sending a line of HTML data every second for 10 hours does not cause a timeout. The minimum allowable value for this field is 5 seconds.

The value set here is the default for the system. If an Inherited Value is specified, the value came from the Default Parameters page. You may, however, set a different value on individual server-specific configurations or within the application itself.

Note that if you have an Apache server, you can also set this value using *Timeout* in the Apache httpd.conf file. The lower of these two values is triggered first.

Queued Request Timeout

This is the maximum number of seconds that a request can remain in a queue waiting for an available connection to the appropriate InterSystems IRIS system. The minimum allowable value is 5 seconds. If an Inherited Value is specified, the value came from the Default Parameters page.

No Activity Timeout

This parameter is relevant to stateless connections only. The parameter indicates the maximum amount of time (in seconds) that a stateless connection remains open in an idle state before closing. If this timeout is exceeded, the session automatically closes. This facility prevents stateless sessions accumulating on your InterSystems IRIS server, particularly after periods of high activity where a large number of connections were opened to cope with the increased workload. If a value is not specified, stateless connections remain open until they are manually closed. If an Inherited Value is specified, the value came from the Default Parameters page.

Note that a process may remain up to a few minutes beyond the configured timeout. By design, the Web Gateway only checks for connection timeouts periodically; it is not immediately notified when a timeout occurs. Depending on the timing of this check, a process can linger for up to 420 additional seconds.

Apply timeout to all Connections

Applies the **No Activity Timeout** option to all connections (including those making up the minimal connection pool). If this option is not checked, the Web Gateway does not apply the **No Activity Timeout** to the minimal connection pool (as defined by the **Minimum Server Connections** parameter). If this option is checked the Web Gateway applies the timeout to all connections in the pool. This option is used by installations that have a very low level of CSP usage and, as a result, have a preference for all CSP processes to time out. If an Inherited Value is specified, the value came from the Default Parameters page.

Event Log Level

Controls what information is written to the Web Gateway Event Log. See Event Logging Parameters for details.

Event Log File

Specifies a location and filename for the Web Gateway Event Log. If not specified, it is written to the directory hosting the Web Gateway installation. For example:

To specify an alternative location:

/opt/logfiles/cspgateway/

To specify an alternative location and file name:

/opt/logfiles/cspgateway/event_log_01012006.log

Retain All Log Files

If **Event Log Rotation Size** is blank (the default), the Web Gateway Event Log grows until the administrator manually clears it. If the capacity of the file is specified by **Event Log Rotation Size**, InterSystems IRIS copies the log file to a file named filename.old, where *filename* is the full original filename. Subsequent log rotations overwrite filename.old with the current contents of the log. To retain all log files, select **Retain All Log Files**. Each log is named with the date and time when the copy took place.

Event Log Rotation Size

This defines the size after which log rotation should take place. The default value is blank which means that the Web Gateway maintains one log file which grows until the administrator manually clears it.

If rotation is required, the size may be specified as a number with no suffix for bytes, a number followed by **K** for kilobytes, or a number followed by **M** for megabytes.

The minimum size that can be specified is 100K. This value is automatically set if the administrator attempts to set a lower value in the maintenance suite.

Rotated copies of the log file, if retained, are named according to the date and time of rotation as follows:

```
CSP_YYYYMMDD_hhmm.log
```

where `YYYY` is year, `MM` is month, `DD` is date, `hh` is hour, and `mm` is minutes past the hour, for example:

```
CSP_20090109_1830.log (Log rotated at 18:30 on 9th January 2009)
```

If more than one log file rotation takes place in the space of one minute, a serial number is appended to the file name to prevent duplicates, for example:

```
03/12/2015  17:02         106,660 CSP_20151203_1702.log
03/12/2015  17:02         124,752 CSP_20151203_1702.log.0001
03/12/2015  17:02         124,752 CSP_20151203_1702.log.0002
```

The rotated log file that is not to be retained is named: filename.old, where *filename* is the full original filename.

In order for this facility to work, the Web Gateway must have create/write access to the directory hosting the Web Gateway binaries (i.e. the location where the main log file is kept). If the Web Gateway is unable to perform a successful rotation it continues writing to the current log file.

This field is supplemented with a check box labeled **Retain All Log Files**. If selected, this option instructs the Web Gateway to keep all log files according to the naming scheme outlined above.

Maximum Logged Request Size

If you have enabled logging for HTTP requests by specifying an Event Log Level of V9 (or a variant such as V9b), this parameter specifies how much of the HTTP request is included in the log. Any request which exceeds this maximum is truncated.

The default value for this parameter is 256K, and the minimum value is 40K. If you leave the field empty, it is set to the default (256K). The minimum is enforced; if you attempt to assign a value lower than the minimum, it is set to 40K.

SSL/TLS Library Path

Specifies the path to the OpenSSL libraries. On UNIX®, these files are libssl.so and libcrypto.so, and on Windows, the files are libcrypto-1_1-x64.dll and libssl-1_1-x64.dll. The default is for the Web Gateway to source these libraries locally in its home directory. See Overriding the Library Path If You Use SSL/TLS in Kerberos Library for more information.

Preserve Mode Exclude File Types

Allows static files to be served asynchronously in state-aware applications. In stateless applications, statics (files other than csp, cls, csr, and zen) are processed asynchronously with respect to the main session. In other words, requests for these files bypass the session lock and can be processed concurrently outside the main processing stream for the application.

This parameter allows this scheme to be extended to state-aware applications. State-aware applications are not only subject to the conventional session lock but are also subject to the connection lock in the Web Gateway. The connection lock is responsible for ensuring that all requests for the user/session are routed to the same InterSystems IRIS process. For applications that rely on static components being served from InterSystems IRIS, this leads to excessive request queuing which, in turn, can lead to browser instability (such as hangs).

Use this parameter to define a list of (space separated) file types (by extension) to process asynchronously and therefore exempt from connection/session locking in the Web Gateway and InterSystems IRIS. If the list is prefixed with `*-` (asterisk hyphen) then all files are processed asynchronously EXCEPT those defined in the following list.

Examples

```
Preserve Mode Exclude File Types=gif jpg jpeg
```

Process files of type GIF, JPG and JPEG asynchronously with respect to the state-aware session:

```
Preserve Mode Exclude File Types=*- csp cls csr zen
```

Process all files asynchronously with respect to the state-aware session EXCEPT those of type CSP, CLS, CSR and ZEN. This, incidentally, is the rule applied in the CSP engine for stateless applications.

This mechanism can be monitored using log level v4. When invoked for a request, a record similar to the one shown below is added to the log.

```
 >>> Time: Fri Oct 04 14:56:40 2017 ...GET /csp/samples/zenutils.js
     State-Aware Session (preserve == 1)
     Process this request concurrently in the pool of stateless connections (File Type=js)
```

10.5 ASP Redirect

Web Document Root

This is the full physical path to the document root directory of the web server. For example, for Microsoft IIS Web Servers, this path is usually c:\InetPub\wwwroot. This parameter is only required if you plan to use the facility within CSP to send the CSP output through the Microsoft ASP engine to render the final page.

Temp ASP Directory

This is the full physical path to a directory where the Web Gateway can temporarily store Microsoft ASP content. This parameter is only required if you plan to use the facility within CSP to send the CSP output through the Microsoft ASP engine to render the final page.

10.6 Internal HTTP Server

This section is only relevant to the NSD. This section contains the following parameters:

Service Status

The HTTP server can be Enabled or Disabled. Select either:

- Enabled
- Disabled

The default is `Enabled`.

In the interests of security, it is best to disable this facility, unless it is intended that the NSD should be able to respond to raw HTTP requests.

NSD Document Root

For cases where the NSD is intended to be used as a stand-alone web server in its own right, this parameter defines the full physical path to the web documents root. For example:

```
/opt/webgateway/home/
```

If the server is used to serve web applications, then the broker components should be installed under:

```
/opt/webgateway/home/broker/
```

The static files used to support the CSP samples:

```
/opt/webgateway/home/samples/
```

The static files used to support the Management Portal:

```
/opt/webgateway/home/sys/
```

10.7 Custom Error Pages

The **Error Pages** section of the global configuration screen allows you to customize Web Gateway error messages and system responses. These can be set on a global or per-InterSystems IRIS server basis. To customize the default CSP responses, perform the following:

1. From the Web Gateway management pages main menu, select **Default Parameters**.
2. In the **Error Pages** section, enter the name of the CSP page that you wish to replace the corresponding Gateway page with. Enter the full physical path to your CSP page, or enter a path relative to that of the Web Gateway.
3. Select **Save Configuration**.

You can customize the following Web Gateway system responses:

Server Error

Page to display when the Web Gateway encounters an internal error. For example, an error occurs if there is a problem communicating with an InterSystems IRIS server. The specific error is always recorded in the Web Gateway Event Log.

Server Busy

Page to display when all available CSP connections are in use.

Server Unavailable

Page to display when the InterSystems IRIS server (or application) has been deliberately disabled from within the configuration.

Server Timeout

Page to display when the request has timed out.

Connection Closed

Page to display when you log out of a state-aware session.

10.8 Event Logging Parameters

The **Event Log Level** field specifies the information that Web Gateway writes to the Web Gateway Event Log. Logging options are defined as a string of characters, each character representing a logging command. The value set here for the log

level is the default for the system (that is, all InterSystems IRIS servers). Except where noted, you can set a different value for individual InterSystems IRIS servers.

You can view or clear the log from the CSP Web Management page menu. The logging parameters, shown below, are used mainly for troubleshooting:

Logging Option	Function
E	Record all errors. This option allows you to monitor connection failures.
V	Verbose: Record the basic connection dialog between the Web Gateway and an InterSystems IRIS system. Use this option to record the strategic points of communication between the Web Gateway and an InterSystems IRIS server. There are 7 levels to this command (1 to 7). Each successive level records more detailed information. The levels are accumulative. For example, level V3 includes all log information specified for V1 and V2.
EV	Enter EV to turn on basic event logging. The higher log levels generate a large volume of data in the log file and should only be used for diagnosing problems. For production systems it is recommended that the log level should be set to no higher than EV.
V1	Same as V.
V2	In addition to the information specified for previous levels, this level records: • Information regarding basic connection management between the Web Gateway and InterSystems IRIS (Start and Close points for each connection). • Transmission interrupts received from the browser. • Cases where connections to InterSystems IRIS are forcefully closed (due to a lack of response from InterSystems IRIS or other errors where the connection cannot be recovered). • Access violations in state-aware (preserve mode 1) sessions (For example, Invalid Session ID).
V3	In addition to the information specified for previous levels, this level records: InterSystems IRIS headers and HTTP headers. *Note:* When this logging level is specified for an individual server, request headers are not logged, but response headers and other data will be.
V4	In addition to the information specified for previous levels, this level records: Information regarding the serialization of state-aware sessions. *Note:* When this logging level is specified for an individual server, request headers are not logged, but response headers and other data will be.

Logging Option	Function
V5	In addition to the information specified for previous levels, this level records the contents of data buffers received from, and sent to, InterSystems IRIS via the WebSocket protocol. All data framing (where applicable) is also recorded. Finally, further information about the nature of the WebSocket created is also recorded at initial connection time. For example: • WebSocket Connection • WebSocket Connection Accepted by InterSystems IRIS: WSClassProtocolVersion=2; SharedConnection=0; NoDataFraming=2; BinaryData=1; *Note:* When this logging level is specified for an individual server, request headers are not logged, but response headers and other data will be.
V6	In addition to the information specified for previous levels, this level records: • Headers to the data blocks sent to InterSystems IRIS. • Request Data from the web server (except multipart attachments). • Headers to the data blocks received from InterSystems IRIS. *Note:* When this logging level is specified for an individual server, request headers are not logged, but response headers and other data will be.
V7	In addition to the information specified for previous levels, this level records: The full content returned from InterSystems IRIS. *Note:* When this logging level is specified for an individual server, request headers are not logged, but response headers and other data will be.
V9	Record incoming HTTP request data. The full bodies of all HTTP requests are recorded. This log directive can be further extended and refined. • v9r: In addition to logging all HTTP requests, record all HTTP responses. • v9a: Record all HTTP requests to http.log in the Web Gateway home directory. • v9b: Record all HTTP requests on a per-session basis. Log files of the form http[session_id].log is created in the Web Gateway home directory, where *session_id* is the 10-Byte session ID. • v9m: Log all multi-part posts in the Web Gateway home directory. The raw incoming HTTP request are recorded together with the individual components in both their encoded and decoded form. *Note:* The forms V9, V9r, V9a, and V9b have no effect when specified for individual servers. These forms of logging can be enabled only at the default level.

Logging Option	Function
s	Sessions: Record information about the management of session tokens: • The point at which new session IDs are allocated. • For existing sessions: an indication as to whether the session token was extracted from a cookie or the form/URL variable CSPCHD. • For all requests: the final session ID transmitted to InterSystems IRIS. *Note:* This logging option has no effect when specified for individual servers. This option can be enabled only at the default level.
c	Connections: Record information about connections made using the Kerberos Library (IRISCONNECT). Include a log level of lowercase c to instruct the Web Gateway to record a complete audit of all IRISCONNECT functions called, together with the input parameters supplied and the result returned. For the sake of brevity, the content of the input and output buffers to and from InterSystems IRIS are not recorded at this level. Set a log level of uppercase C to record, in addition to the IRISCONNECT function calls, the contents of the input and output buffers. In addition to the logging facilities provided by the Web Gateway, it is possible to instruct the IRISCONNECT library to generate a detailed trace recording its internal processes. To additionally request that a IRISCONNECT trace be generated, add a digit to the c directive to indicate the type of trace required. For example, a log level of c3, in addition to the standard Gateway log entries, generates a level 3 IRISCONNECT trace. Valid IRISCONNECT trace levels are 1 to 6 and are defined as follows: • 6 — errors • 5 — warnings • 4 — informational message • 3 — output data • 2 — input data • 1 — normal events Unlike the Web Gateway log levels, the IRISCONNECT trace is less verbose at the higher log levels. Log level 1, therefore, provides the most detailed trace file. The Web Gateway instructs the IRISCONNECT library to maintain its trace in a file called *irisconnect.log* located in the Web Gateway home directory. The security considerations and permissions for this file are the same as those for the Web Gateway Event Log. *Note:* An IRISCONNECT trace can only be activated on a per-process basis, so it cannot be truly isolated to a server. Once configured, trace log generation is not triggered until a new SSL connection is attempted.

Logging Option	Function
t	Transmission: Record the raw data buffers received by and dispatched by the Web Gateway. The format for this option is: `t[x][y]`. The value of `x` instructs the Web Gateway to record data buffers transmitted between the Web Gateway and InterSystems IRIS and the value of `y` instructs the Web Gateway to record data buffers transmitted between the Web Gateway and the client, via the hosting web server. `x` and `y` can take the following values: • 0: No transmission data to be recorded. • 1: Record request data only. • 2: Record response data only. • 3: Record request and response data. Using lowercase `t` results in the Web Gateway recording just the first 256 Bytes of transmitted data for each buffer. Using uppercase `T` results in the Web Gateway recording the full data buffer. All non-printable characters are recorded in their escaped form. *Note:* When this logging level is specified for an individual server, `y` options record response buffers sent to the client, but not incoming request buffers from the client.
p[n]	Performance: Instructs Gateway to capture information to assess the performance of the CSP installation. *n* is the number of seconds (total service time) below which data is not recorded for a request. For example, a directive of `p` records data for all requests, `p2` records data for requests taking longer than 2 seconds to service. The following information is recorded. • Total time to service request: The total time spent in servicing the request (from the time it reaches the Web Gateway to the time at which the last Byte of response data leaves the Web Gateway environment. • Obtain [NEW] connection to InterSystems IRIS: Time taken between the request reaching the Web Gateway and a connection to InterSystems IRIS being reserved for the purpose of servicing the request. The message recorded indicates if a new connection is created during this time (as opposed to an existing one being reused). • Send request to InterSystems IRIS: Time taken between the first and last Byte of request data being read from the web server and dispatched to InterSystems IRIS. • Processing request in InterSystems IRIS: Time taken between the last Byte of request data being dispatched to InterSystems IRIS and the first Byte of response data being received by the Web Gateway. • Receive response from InterSystems IRIS: Time taken between the first and last Byte of response data being received from InterSystems IRIS and dispatched to the web server.

Logging Option	Function
p[n]([v])	Provides the capability to conditionally activate verbose logging based on the results of the performance monitor. Useful in situations where you want to record further information about requests that take more than a certain time to process. *n* is the optional lower time-to-service threshold (in seconds) for which performance data is recorded and *v* is the verbose log level required. This mechanism applies to verbose Event Log and HTTP logging settings. A request to record error information, *e* is always applied to all requests regardless of whether or not they are recorded by the performance monitor. For example: `ep5(v9)` This option records any errors encountered while processing requests for all requests (*e*). In addition, it records the HTTP request message (*v9*) but only for requests that take longer than 5 seconds to process (*p5*). Gateway event logging is designed to have a minimal impact on performance and to occupy a small footprint in terms of system resources consumed. Therefore, the following limitations apply: • Only one verbose log level can be specified per individual setting. In other words it is not possible to specify a *v9* level for requests recorded by the performance monitor and a *v2* level for all other requests. For example, if `v2p5(v9)` is specified then only the conditionally applied *v9* level is honored. • The Web Gateway configuration allows you to specify an Event Log level both globally and on a per server basis. When verbose logging is in force, some records are written before the target InterSystems IRIS server has been identified so, for best results, it is best to specify conditional logging at the global level under Default Parameters.

Logging Option	Function
pp[n]	Provides detailed timing information as follows: • Pre-processing of request: Time taken to identify the target InterSystems IRIS server; includes the initial handover from the web server and basic request processing to identify the server. • Obtain [NEW] connection to InterSystems IRIS: Time taken to allocate a connection to the appropriate InterSystems IRIS server. Indicates whether a new connection is created (instead of an existing one reused). • Format request: Time taken to parse and format the request message for transmission to InterSystems IRIS. • Send request to InterSystems IRIS: Time taken between the first and last byte of request data read from the web server and dispatched to InterSystems IRIS. • Processing request in InterSystems IRIS: Time taken between the last byte of request data dispatched to InterSystems IRIS and the first byte of response data received by the Web Gateway. • Post-processing of response(b): When a content-length header is required, this reports the time taken for the dispatch of the response data back to the client via the web server. • Post-processing of response(c): Time taken between the dispatch of the response and the Web Gateway being ready to read the response footer data from InterSystems IRIS. The footer data is part of the internal communication protocol between the Web Gateway and InterSystems IRIS and includes control information (For example: instructions to change the preserve setting for the session). • Receive footers from InterSystems IRIS: Time taken to receive the response footer data from InterSystems IRIS. • Post-processing of footers: Time taken to process footer data and respond to instructions received. • Release connection to InterSystems IRIS: Time taken to release the active connection to InterSystems IRIS. • Cleanup: Time taken to release resources used in servicing the request and return control back to the hosting web server.

Logging Option	Function
W (or w)	On Windows, generates a memory dump if a crash occurs. This option is case insensitive. On AIX, generates a core file using the `gencore` utility. This option is case insensitive. On Linux or MacOS, this option is case sensitive. Specifying `w` generates a standard core dump using `gcore`. Specifying `W` dumps all memory mappings (including shared memory) into the core file by executing `gcore -a`. On Unix systems, the following preconditions must hold: • The `gcore` (Linux or MaxOS) or `gencore` (AIX) is present on the machine and is available through the PATH environment variable. On Linux and MacOS systems, the version of `gcore` must support the `-a` command line option. • Web server worker processes need write permission to the directory where the Web Gateway modules are located. In default installations this directory is `/opt/webgateway/bin`. • A non-root process needs permission to produce a core dump of another process running under the same user ID. On MacOS, System Integrity Protection must be disabled. On Linux, if the Yama security module is present (as on RHEL and Ubuntu systems), execute the following command to grant the required permission until the next reboot: `echo 0 | sudo tee /proc/sys/kernel/yama/ptrace_scope`. To permanently grant this permission, create or edit the file `/etc/sysctl.d/10-ptrace.conf`. If there is a line starting with “kernel.yama.ptrace_scope”, change it to “kernel.yama.ptrace_scope = 0”. If no such line exists, add “kernel.yama.ptrace_scope = 0”, then execute `sysctl –p`. **Note:** For security reasons, it is recommended that such permission be granted only temporarily.

11

Protecting Web Gateway Connections to InterSystems IRIS

This page describes options for protecting connections from the Web Gateway to InterSystems IRIS®. For more details on CSP authentication, see the Authentication Guide. Web Gateway connections to InterSystems IRIS can be protected according to the following levels of security:

1. Minimal connection security (not recommended)
2. Simple username- and password-based authentication
3. Kerberos-based authentication and data protection
4. SSL/TLS-based authentication and data protection (including mutual TLS authentication)

Remember that security applied here is solely for the purpose of authenticating the Web Gateway host to the InterSystems IRIS server. It protects against the unauthorized creation of connections between the Web Gateway and an InterSystems IRIS application server (`%cspServer`). It does not, however, identify an individual *user* of a web application. A user of a web application can only be positively identified by whatever user login facility is provided by the application itself. For example, a Systems Manager logging on to the Management Portal can only be identified by the username and password supplied to the Management Portal login form.

The stateless nature of the Web should also be borne in mind. There is no fixed relationship between a Web Gateway connection to InterSystems IRIS and an individual user of a web application. Many users share the same connection.

Authenticating the Web Gateway to InterSystems IRIS at connection time is important. If an attacker can impersonate a Web Gateway, it can redirect traffic through a system under his control (by technical means and/or social engineering) and read and/or modify data at will. This is distinct from authenticating individual users to a web application. The Web Gateway's InterSystems IRIS username and password, Windows network credentials, or UNIX® Kerberos key table should never be used by ordinary users.

11.1 Configuring Connection Security for the Web Gateway

To configure the connection security for the Web Gateway, in all cases you use the Web Gateway management pages. The relevant options are in the **Configuration** > **Server Access** > **Connection Security** section, which provides the following settings:

- **Connection Security Level** — Choice of:
  - Password
  - Kerberos
  - Kerberos with packet integrity
  - Kerberos with encryption
  - SSL/TLS
- **Username**
- **Password**
- **Product**
- **Service Principal Name**
- **Key Table**

11.2 Minimal Connection Security (Not Recommended)

To use minimal connection security, the **Connection Security Level** is set to **Password** and the **Username** and **Password** fields are left empty.

In this mode, there is a minimal level of security applied to the connection between the Web Gateway and InterSystems IRIS.

In this mode of operation, ensure that the Web Gateway service (`%Service_WebGateway`) together with the username under which it operates (for example, `CSPSystem`) is not expecting any form of authentication.

11.3 Simple Username/Password Authentication

Username/password authentication is the simplest form of authentication that can be applied between the Web Gateway and InterSystems IRIS. The installation process for an instance creates the CSPSystem user to authenticate Web Gateway access for the instance. This user (CSPSystem or any other) should have no expiration date; that is, its Expiration Date property should have a value of 0.

In all cases, the default username and password used for the Web Gateway is as follows:

```
Username: CSPSystem
Password: SYS
```

To configure simple username and password authentication for an instance using the Web Gateway management pages, modify the server access profile by setting the **Connection Security Level** to `Password` and then supplying values for the **Username** and **Password**.

For any InterSystems IRIS application server, you can change the credentials for the user account which grants access to the Web Gateway as you wish. However, you must also update the corresponding server access profile in the Web Gateway with these new credentials. Otherwise, the Web Gateway is unable to establish new connections to that application server. Keep in mind that passwords cannot begin with `{` and end with `}`, because the Web Gateway attempts to interpret strings between these braces as a password retrieval command.

Remember: passwords are a weak form of authentication since they must be sent over the network as plain text for authentication in InterSystems IRIS. Network sniffing is easy to do and can be used to reveal these passwords. Passwords used in this configuration option must be held in the Web Gateway configuration file in accordance with the following guidelines.

For UNIX®, Linux, and macOS, passwords in the CSP.ini file are stored as base64 hashes. For Windows, passwords are encrypted in the Web Gateway configuration file using functionality provided by Microsoft's Data Protection API (DPAPI). The Web Gateway Management **Default Parameters** page handles the encryption of passwords.

Note: This password encryption is used for Windows because ordinary Windows user accounts are occasionally granted membership in the Administrators Group, although this is not recommended practice for production systems. Encrypting the password offers a higher level of protection for all Windows installations.

11.3.1 Passwords Introduced from Outside

Occasionally, you need to introduce a password outside the context of the Web Gateway Management pages, for example, if the Web Gateway configuration is set up by custom configuration scripts. In this case, the password should be filed as plain text and the Web Gateway encrypts it when it is started for the first time. If you use this method, the password to the Gateway Management forms cannot start with '1' or 'PBKDF2|', and the password to IRIS cannot start with ']]]'. If you need passwords that start with these characters, use the `CSPpwd` utility at the OS command prompt instead. This utility encodes passwords held in the Web Gateway configuration file. The general form is:

`CSPpwd` *<path to the CSPx.so|dll library> <context> <clear text password>*

Where:

- context = 0: Password to the Web Gateway management pages
- context = 2: Password to Cache/IRIS servers

The encoded password is written to the standard output.

Examples (Windows):

```
CSPpwd C:\Inetpub\CSPGateway\CSPx.dll 0 MyGatewayManagementPassword
CSPpwd C:\Inetpub\CSPGateway\CSPx.dll 2 MyIRISServerPassword
```

Examples (UNIX®):

```
CSPpwd /opt/webgateway/bin/CSPx.so 0 MyGatewayManagementPassword
CSPpwd /opt/webgateway/bin/CSPx.so 2 MyIRISServerPassword
```

11.3.2 Passwords Encrypted on Other Computers

On Windows, the web server uses the machine store rather than a user store. Consequently, DPAPI password encryption is machine-specific; it is not possible to decrypt a Web Gateway password that was encrypted on another computer. This means it is impossible for machines in a clustered environment to share information within a CSP.ini file.

Here are some possible solutions to this problem:

- Use a machine outside of the cluster as the web server.
- Each time you fail over, reset the same password in the Web Gateway.
- Configure each computer participating in the cluster so that it has its own copy of the Web Gateway configuration file on a disk that does not belong to the cluster. InterSystems IRIS maintains the file in the directory hosting the Web Gateway DLLs. Save and encrypt the password on each individual computer before introducing the node to the cluster.

For example, where *Disk C* from each machine does not belong to the cluster and InterSystems IRIS is installed on *Disk S*, you may have the following:

CLUNODE-1: A copy of CSP.ini with password XXX encrypted by CLUNODE-1

CLUNODE-2: A copy of CSP.ini with password XXX encrypted by CLUNODE-2

- Disable password encryption by manually adding the following directive to the configuration file before starting the Web Gateway and adding the passwords:

```
[SYSTEM]
DPAPI=Disabled
```

11.3.3 Retrieve Passwords Programmatically (UNIX®/Linux/macOS)

On UNIX®/Linux/macOS systems, you can instruct the Web Gateway to execute an operating system command which retrieves the password from a secure storage solution (such as a vault application or a mounted secret file) instead of providing the password itself as plain text. To do so, provide the command within braces (`{}`) in the **Password** field of an instance's server access profile using the Web Gateway management pages. Alternatively, you can set the corresponding CSP.ini parameter (`Password`) directly`Password` within the section of the CSP.ini that corresponds to the instance. For example, a CSP.ini file may contain the following line:

```
Password={sh /tmp/PWretrieve.sh}
```

The string is stored as a base64 hash value within the CSP.ini file, just like a plain text password would be.

The Web Gateway executes the command you provide when you save the server access profile using the Web Gateway management pages or when the `RELOAD=1` is found in the CSP.ini file's `[SYSTEM]` section. The output of the command is then stored as the password for the instance within memory. The command that you provide must be sufficient to retrieve the password without further action: the command line interface for your secure storage solution must be fully configured on your system and any credentials needed must be available in environment variables or configuration files.

Note: You can also use this method to retrieve an **SSL/TLS Private Key Password** programmatically within a server access profile or within the CSP.ini file.

11.4 Kerberos-based Authentication and Data Protection

To use Kerberos-based Authentication and Data Protection, three levels of authentication (and data protection) are provided through the **Connection Security Level** parameter.

1. Kerberos. This option provides initial authentication only for the connection.
2. Kerberos with Packet Integrity. This option provides initial authentication and guarantees data packet integrity.
3. Kerberos with Encryption. This is the highest level of security and provides initial authentication, guaranteed data packet integrity, and, finally, encryption for all transmitted messages.

11.4.1 Kerberos Library

To use any of the Kerberos-based modes, the Web Gateway must be able to load the InterSystems Kerberos client library:

- Windows DLL: irisconnect.dll
- UNIX® Shared Object: irisconnect.so

Install the appropriate library in a location specified in the PATH environment variable for the operating system or at one of the following locations relative to the Web Gateway installation.

- . (that is, local to the Web Gateway)
- ./bin
- ../bin
- ../../bin

The Web Gateway attempts to load the library at the time it is first required. If successful, the following status message is written to the Web Gateway Event Log:

Web Gateway Initialization The IRISCONNECT library is loaded - Version: 5.3.0.175.0.

(This library is used for the optional Kerberos-based security between the Web Gateway and InterSystems IRIS.)

If the Web Gateway is unable to locate or link to the `IRISCONNECT` library, a suitable statement of failure and error message is written to the Web Gateway Event Log.

For Kerberized communications between the Web Gateway and InterSystems IRIS, the Web Gateway is the Kerberos client.

The procedure for configuring the Web Gateway to use Kerberos is in the Windows section.

11.4.1.1 Overriding the Library Path If You Use SSL/TLS

By default, the Web Gateway expects dependent security libraries (shared objects) to be installed in its home directory (that is, the directory with the Web Gateway binaries).

If you use SSL/TLS connectivity between the Web Gateway and InterSystems IRIS, these libraries include the IRISCONNECT library and SSL/TLS libraries (on UNIX®: libssl.so and libcrypto.so, on Windows: libcrypto-1_1-x64.dll and libssl-1_1-x64.dll).

When the Web Gateway and IRISCONNECT libraries, loaded in the web server's process space, load a copy of the SSL/TLS libraries, there is a conflict between different versions of the same libraries that were previously loaded by the hosting web server. To ensure that only one copy of the SSL/TLS libraries are loaded in the web server process space, the Web Gateway must instruct the IRISCONNECT library to source the SSL/TLS libraries from the same location as those used by the hosting web server.

The Web Gateway Management **Default Parameters** page provides the parameter **SSL/TLS Library Path** to allow you to use an alternative set of OpenSSL libraries. For example:

```
SSL/TLS Library Path = /usr/bin/
```

Important: It is possible to create an Apache installation that does not permit OpenSSL usage, and it is possible to configure Apache to disable OpenSSL. In this situation, the Web Gateway loads the libraries it was shipped with unless **SSL/TLS Library Path** is set to a different location.

If a library version mismatch occurs between Apache and the Web Gateway, TLS connections between the Web Gateway and an InterSystems IRIS instance can fail. A TLS error can occur when a connection is attempted, or the Web Gateway might crash with SIGSEGV when calling an OpenSSL function.

11.4.2 Windows

Kerberos key tables are not implemented for Windows. Therefore, authentication uses network credentials that are either obtained when the hosting service starts in a named account or from the Trusted Computing Base (TCB) when the hosting service runs in the System Logon Session (that is, as `LOCAL SYSTEM`).

Windows domain accounts use a permanent key derived from a password to acquire a Kerberos Ticket Granting Ticket (TGT) and service ticket for the local machine. The local machine must also have a permanent Kerberos key, shared with the Key Distribution Centre (KDC) component of the domain controller. That key can be used to acquire a TGT and service ticket to authenticate to another Kerberos principal such as InterSystems IRIS.

For practical purposes the Web Gateway, operating within the context of a Windows-based web server is operating through either the Network Service logon session or the System logon session. The account used must have `Log on as a batch job` rights assigned.

The built-in Network Service logon session has access to the machine's credentials and is designed for services that need network credentials to authenticate to other machines. However, the Network Service logon session is not always present. The System logon session can also be used for the purpose of authenticating the Web Gateway to InterSystems IRIS.

For IIS installations, and ISAPI extensions in particular, using the Network Service login session is the preferred means through which both databases (local and remote) and remote computers should be accessed.

11.4.2.1 Windows Web Gateway Configuration for Kerberos

- Set the **Service Principal Name** to that of the target InterSystems IRIS server that the Web Gateway is connecting to.
- Leave the **Username**, **Password**, and **Key Table** fields empty.
- The client principal name (or client username) is that of the Web Gateway host. This is the Kerberos name representing the Web Gateway hosts' network service session: *<computer_name>$*
- Assign this principal the necessary privileges in the InterSystems IRIS server to allow the Web Gateway's service to operate.

11.4.3 UNIX® Web Gateway Configuration for Kerberos

These operating systems support Kerberos Key Tables.

11.4.3.1 UNIX® Web Gateway Configuration for Kerberos

The Web Gateway configuration is conceptually more straightforward for these systems.

- Set the **Service Principal Name** to that of the target InterSystems IRIS server that the Web Gateway is connecting to.
- Enter the name of the key table file (including the full path) in the **Key Table** field.
- Set the **Username** field to the name of the appropriate key in the key table file.
- Leave the **Password** field empty.
- The client principal name (or client username) is that of the Web Gateway host. This is the name used to identify the key in the Kerberos Key Table. Assign this principal the necessary privileges in the InterSystems IRIS server to allow the service of the Web Gateway to operate.

11.5 SSL/TLS-Based Authentication and Data Protection

You can also use the SSL/TLS protocol to secure communications between the Web Gateway and InterSystems IRIS.

In this mode, the SSL/TLS transport, as configured for this host, secures connections to InterSystems IRIS. The **SSL/TLS Configuration Name** field should be set to the appropriate value for the target server. The **Service Principal Name** and **Key Table** fields are not relevant and should be left empty.

For more information on creating SSL/TLS client configurations for InterSystems IRIS systems, see Configuring the Web Gateway to Connect to InterSystems IRIS Using TLS.

11.5.1 Mutual TLS

Mutual TLS is a form of authentication that you can apply between the Web Gateway and InterSystems IRIS, as an alternative to password authentication. See Mutual TLS for more details.

12

Managing and Monitoring the Web Gateway

This page describes how to manage and monitor the InterSystems Web Gateway via the Web Gateway management pages.

12.1 Checking System Status

The **System Status** option displays the status of all active connections. You must be a system manager to use this feature. In each of the tables below, click a column head to sort by that column.

12.1.1 Connections to InterSystems IRIS

The first status table (Connections to InterSystems IRIS) displays information on connections to InterSystems IRIS®.

Item	Function
Connection Number	Number that the Web Gateway assigns to the connection. Your InterSystems IRIS license determines the number of possible connections.
Gateway PID	The Web Gateway (or hosting web server) process ID for the connection.
Server Name	Name of the InterSystems IRIS system connected to. Mirror members show current configuration name with mirror member name appended.
InterSystems IRIS PID	Process ID on the InterSystems IRIS server.
Status	Indicates whether information is being sent to or from the InterSystems IRIS system, as follows: `Free` — no information is being sent and the connection is ready to process the next request. `In Use` — information is being transmitted through the connection. `Private` — the connection is state-aware (preserve mode 1) and not free for general use. `Server` — the connection is being used by the InterSystems IRIS server.
Idle time/Timeout	Indicates the amount of time that the connection has been idle against the timeout applied to that connection. The timeout is the 'No Activity Timeout' for connections in the state-less pool and the 'Application timeout' for connections marked as 'Private' (state-aware).
Activity	Number of transactions (hits) the connection has processed.

Item	Function
Interrupt	For a connection with status 'In Use', an Interrupt button will attempt to interrupt the corresponding InterSystems IRIS process and return it to the state where it is ready to accept the next web request.
Close	If available, allows you to forcefully close down the connection by selecting it. See Closing Connections Manually.

12.1.2 InterSystems IRIS Servers Table

The second status table (InterSystems IRIS Servers) displays information on InterSystems IRIS servers.

Item	Function
Server Number	The number that the Web Gateway assigns to the server.
Server Name	Name of the InterSystems IRIS system connected to.
Mirror Member	For mirror-aware configurations, the name of the mirror member.
Mirror Status	For mirror-aware configurations, the name of the mirror configuration along with the mirror status of the server. The member type, Failover or Async, will be shown and the Primary will be labeled as 'Primary'.
Total Connections	Number of connections to the InterSystems IRIS system.
Connections In-Use	Number of connections that are currently in use (actively serving a Web request).
Private Connections	Number of connections that are currently in use as state-aware sessions (preserve mode 1).
Total Activity	Number of transactions (hits) the InterSystems IRIS system has processed.
Queued Requests	Number of Web requests that are held in a queue waiting for a free connection to the InterSystems IRIS system. Queued requests are an indication that the InterSystems IRIS license should be increased in order to maintain good performance.
Close	Close all connections on this InterSystems IRIS server. See Closing Connections Manually.

12.1.3 Application Paths Table

The third status table displays information for application paths.

Item	Function
Path Number	The number that the Web Gateway assigns to the application path.
Path	The application path.
Server Number	The number that the Web Gateway assigns to the InterSystems IRIS server.
Server Name	The name of the InterSystems IRIS system connected to.
Activity	The number of requests processed by this server for this path since the last Gateway.

Item	Function
Status	The status for this server, one of `Disabled`, `Enabled` or `Offline`. Also the current master (or primary) server in the set is indicated in this column.
Action	If a server is marked as `Offline`, this column contains a button allowing Administrators to mark it `Online/Enabled` again.

12.1.4 Web Gateway Cache Table

The fourth status table lists the forms held in the Web Gateway response cache.

Item	Function
Cached Forms	Name (including path) of cached form.
Cached Data (Bytes)	Amount of cached form data held in the Gateway (in Bytes).
Cache Blocks In Use	Total number of cache memory blocks in use.
Cache File	Name of physical file if permanent storage (on the web server host) is used to cache the file.
Cache Form Activity	Total number of times this form has been requested from the cache.
Clear	Clear this form from the cache. See Clearing the Cache.

12.1.5 Closing Connections Manually

If your InterSystems IRIS system shuts down while a CSP connection is still active, CSP continues to try to connect to the system until one of the following occurs:

- It successfully reconnects to the system.
- CSP is shut down.
- The connection is manually closed.

If your InterSystems IRIS system is scheduled for extensive downtime, you may want to close the connections. You can close sessions manually using the **Close** button on the **System Status** page.

Note that you can close the connections while the InterSystems IRIS system is down.

12.1.6 Clearing the Cache

Under certain circumstances, such as during the development process for web applications, it may be necessary to clear the Web Gateway cache. To do this:

1. From the Management Portal, navigate to **System Administration** > **Configuration** > **Web Gateway Management** and select **System Status**.
2. On the **System Status** page, there are a number of tables. To clear the cache, in the **Cached Forms** table, select the button in the **Clear** column (the right-most column) and the **Total** row (the bottom row). If the **System Status** page does not display a **Cached Forms** table, then there is no currently cached content. This may be because the cache has been cleared recently and nothing has been cached since then.

This action clears all cached content for the Web Gateway and removes the **Cached Forms** table from the page until there is new cached content.

12.2 Testing Server Connections

The **Test Server Connection** option is useful to test Web Gateway connectivity to your InterSystems IRIS systems. Note that you must be a system manager to use this feature.

To test CSP connectivity:

1. From the Web Gateway management pages main menu, select **Test Server Connection**.
2. Select the desired InterSystems IRIS system from the displayed list.
3. Select **Connect**.

Depending on your selection and the state of the server connection, you receive one of the following results:

Result	Meaning
CSP Test Form	The Web Gateway is working correctly and is able to connect to InterSystems IRIS. The form shows the basic parameters returned by the target InterSystems IRIS server (version and process ID).
Server Availability Error	This error occurs any time that InterSystems IRIS is unreachable. If there are no additional error messages, check to ensure your InterSystems IRIS system is running. Also, check the Web Gateway Event Log for specific connectivity error messages.

In all cases where an error condition is returned, check the Web Gateway Event Log for additional and more specific error information. Consider raising the log level to capture even more diagnostic information where necessary.

12.3 Viewing the Event Log

Use the **View Event Log** option from the Web Gateway management pages main menu to read the contents of the Web Gateway Event Log. The Event Log Level parameter determines what information is logged.

The Web Gateway stores the event log in a log file, named CSP.log by default. For convenience, the Web Gateway's **About Web Gateway** management page indicates the location of this file. When the active log file reaches its capacity as specified by **Event Log Rotation Size**, it is copied to filename.old, where *filename* is the full original filename. If **Event Log Rotation Size** is blank (the default), the file grows until manually cleared. To save all logs in files named with date and time, select **Retain All Log Files** on the **Default Parameters** page. Each log entry is marked with a header record which captures the date, time and additional information with respect to the context in which the log entry was made.

Log entries follow the same machine-readable format which an InterSystems IRIS instance uses for its structured logging feature. This means you can use the same third-party tools to analyze the Web Gateway Event Log which you use to analyze the logs for your InterSystems IRIS instances. For added efficiency, the Web Gateway Event Log uses several of the same field names which other structured logs use: `when`, `level`, `event`, `pid`, and `text`. On the **View Event Log** page, the `text` and `details` fields are presented without their field names. However, the log entries available in the CSP.log file fully subscribe to the name-value pair (NVP) format.

What follows is an example of a Web Gateway Event Log entry as it appears in CSP.log. Each entry in CSP.log occupies one line. However, for readability this example is divided up onto multiple lines (marked by the \ character).

```
local-time="Thu Jul 21 11:39:20 2022" \
  wg-build="RT 2202.1825 (win32/apapi:srv=2.4.52/apr=1.7.0/apu=1.6.1/mpm=WinNT)" \
  wg-log-level=0 when="2022-07-21 15:39:20.831" level=WARNING event=WebGateway.SessionOpen \
  pid=17216 thread-id=2072 text="Warning" \
  details="A Connection between the Web Gateway and InterSystems IRIS has been found to be \
  closed (possibly as a result of an intermediary, such as a firewall, timing-out the TCP session)"
```

Select **Clear Log** to clear all current entries from the Event Log.

The Log can be displayed in either ascending date/time order (the default) or descending date/time order. Select the link at the top right-hand corner of the form to reverse the display order. This link acts as a toggle between the two modes.

Finally, most browsers are unable to render more than about 1MB of log data in a single form. Therefore, as the volume of log data returned approaches 1MB, the Web Gateway terminates the display and prompts for the next page of data. See the **More** link at the bottom left-hand corner of the form. Additionally, a **Top** link is provided at the bottom right-hand corner of the form to allow you to quickly go back to the first form in the series.

12.4 Using the HTTP Trace Facility

The HTTP trace facility is accessed via the **View HTTP Trace** option.

The trace window consists of two main frames. The left-hand frame contains a list of HTTP requests processed by the Web Gateway by time and a unique request ID (assigned by the Web Gateway). As each request is selected, the request and response data is shown in the right-hand frame. Links allow easy navigation between the request and response message.

Note: Note that the HTTP request headers reported by the Web Gateway are reconstituted because the hosting web server always assumes responsibility for parsing the request headers. The Web Gateway reassembles the complete header from the CGI environment variables supplied by the web server. However, if a request is passed directly through the NSD component (that is, effectively bypassing the web server), then the request header recorded is byte-for-byte the same as it was when dispatched from the client.

13

CGI Environment Variables Passed by the Web Gateway

CGI environment variables are derived both from the client's HTTP request headers and from the environment in which the web server is operating. The Web Gateway transmits the common environment variables to InterSystems IRIS® with each and every request. If extra environment variables are required by the application, they must be explicitly requested in the Web Gateway configuration (via the **Extra CGI Environment Variables** setting in the **Application Access** section of the configuration). In the InterSystems IRIS Management Portal, navigate to **System Administration** > **Configuration** > **Web Gateway Management** and select **Application Access**.

The list of environment variables transmitted is shown in the table below together with a brief description of each. Further documentation can be obtained from standard web text books.

Environment Variable	Value
AUTH_PASSWORD	Value entered in the client's authentication dialog. This variable is available only if Basic authentication is used.
AUTH_TYPE	Contains the authentication method that the server uses to validate users when they attempt to access a protected script.
CONTENT_TYPE	For requests which have attached information, such as HTTP POST and PUT, this is the content type of the data.
GATEWAY_INTERFACE	Revision of the CGI specification to which this server complies. Format: CGI/revision
HTTP_ACCEPT	Value of the Accept request header that contains a list of accepted formats (MIME types). For example: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-excel. The values of the fields for the HTTP_ACCEPT variable are concatenated, and separated by a comma (,).
HTTP_ACCEPT_CHARSET	Comma-delimited list of the character encodings that the client accepts.
HTTP_ACCEPT_LANGUAGE	Contains a string describing the language to use for displaying content (such as en-us).
HTTP_AUTHORIZATION	Contains the Base-64 encoded username, password, scheme and realm sent by the client.
HTTP_COOKIE	Holds the contents of the client's cookie(s).

Environment Variable	Value
HTTP_REFERER	Holds a string that contains the URL of the page that referred the request to the current page using an HTML <A> tag. Note that the URL is the one that the user typed into the browser address bar, which may not include the name of a default document. If the page is redirected, HTTP_REFERER is empty.
HTTP_SOAPACTION	SOAPAction HTTP request header field can be used to indicate the intent of the SOAP HTTP request. The value is a URI identifying the intent. SOAP places no restrictions on the format or specificity of the URI or that it is resolvable. An HTTP client MUST use this header field when issuing a SOAP HTTP Request.
HTTP_USER_AGENT	Browser the client is using to send the request. General format: software/version library/version.
HTTPS	Set to either `On` or `Off` (using word, not numerical value). Set to `on` if the script is being called through a secure server (that is, using SSL/TLS).
PATH_TRANSLATED	Translated version of PATH_INFO, in which any virtual-to-physical mapping is applied to the path.
REMOTE_ADDR	IP address of the remote host making the request.
REMOTE_HOST	Hostname making the request. If the server does not have this information, it should set REMOTE_ADDR and leave this parameter unset.
REMOTE_IDENT	If the HTTP server supports RFC 931 identification, then this variable is set to the remote username retrieved from the server.
REMOTE_USER	Name of the user as it is derived from the authorization header sent by the client
REQUEST_METHOD	Method with which the request was made. For HTTP, this is `GET`, `HEAD`, `POST`, and so on.
SERVER_NAME	The server's hostname, DNS alias, or IP address as it would appear in self-referencing URLs.
SERVER_PORT	Port number to which the request was sent. For example: 80
SERVER_PORT_SECURE	Set to either 0 or 1. If the request is being handled on the web server's secure port, then it is set to 1. Otherwise, it is set to 0.
SERVER_PROTOCOL	Name and revision of the information protocol that the request came in with. Format: protocol/revision
SERVER_SOFTWARE	Name and version of the web server software responding to the request. Format: name/version.

14

HTTP Response Headers Returned by the Web Gateway

Web applications in InterSystems IRIS® (including REST-based applications) usually assume the responsibility for formulating a full HTTP response header. For performance reasons, the Web Gateway by default streams the response headers, together with the following content, directly to the client via the web server. This mode of operation is known as the *non-parsed header* (NPH) approach. The Web Gateway does not grant the hosting web server any control over the response headers by passing them back through the dedicated API functions provided by the server. It is assumed that it is the client that needs to read and interpret the response header directives rather than the web server.

However, this assumption breaks down in cases where it necessary for the web server to interpret the response headers in order to invoke further web server-based functionality implied in the header directives generated by the CSP engine) For example, by invoking output filters to further process the response (compression and encryption utilities etc.). Such output filters are usually found not to work for CSP content returned according to the nonparsed header mode of operation.

A facility exists to instruct the Web Gateway to explicitly pass the response headers through the hosting web server instead of streaming them directly to the client.

To use this facility, set the following CSP header directive: `CSP-nph: false`

This directive must be set in the **OnPreHTTP()** method. For example:

```
<script language=objectscript method=OnPreHTTP arguments=""
returntype=%Boolean>
Do %response.SetHeader("CSP-nph", "false")
Quit 1 </script>
```

When set to `false`, (the default setting for the Web Gateway is `true`), the `CSP-nph` directive ensures that the hosting web server is properly notified as to the nature of the response through the response headers returned from the CSP engine. As a result, any further processing can be performed as necessary. This is parsed header mode.

When the Web Gateway is operating in parsed header mode, the hosting web server interprets the response headers and perhaps add header directives of its own. At the very least it adds a Server header to the response. For example:

```
Server: Apache/2.0.48 (Win32)
```

OR:

```
Server: Microsoft-IIS/5.1
```

Note that this facility only applies to the use of Web Gateway implementations that work directly to web server APIs. In other words: everything other than CGI.

If the Web Gateway CGI modules are used and this facility is required then you must configure the web server to use the non-NPH versions of the CSP CGI modules. For example, use `CSPcgi` instead of `nph-CSPcgi`. The `nph-` prefix used

in the name of a CGI module is the standard way of informing the web server that it is not required to read and interpret the response headers returned by the module: in other words operate in non parsed header mode.

The essential difference between the parsed and non-parsed versions of these modules lies in the way the HTTP response status line is formulated. This is the first line in the header block.

For parsed headers, format the HTTP status line as follows:

`Status:` <*status_code*>

Example:

```
Status: 200 OK
```

For nonparsed headers, format the HTTP status line as follows:

`HTTP/1.1`<*status_code*>

Example:

```
HTTP/1.1 200 OK
```

The CGI modules supplied with the Web Gateway automatically handle these differences internally. The CSP engine always return a standard HTTP header block (2).

See also the Non-parsed Headers parameter in Adding an Application Path.

15

Compressing the Response to Requests for CSP Forms (GZIP/ZLIB)

Compressing the response generated by the CSP engine before dispatching it to the client is advantageous because it can dramatically reduce the network bandwidth required to transport the response to the client. From the client's perspective the performance of the application is improved. This is particularly true for clients accessing the application through mobile devices over slower telecommunications networks. There is, of course, a cost in terms of the web server host's CPU time that's required to actually compress the data but this is a small price to pay for the advantages.

The advantage of serving compressed response data is particularly marked for CSP pages for which large volumes of response data are generated.

There are two methods for implementing GZIP in a web server environment.

- Using the Web Gateway's own interface to the GZIP library described here.
- Using a GZIP output filter as an add-on to the hosting web server.

Most web servers offer add-on facilities for compressing data. Windows/IIS offers a `gzip` filter (implemented as an ISAPI filter). The Apache Group offer a compression filter implemented as an add-on module (mod_deflate.c – which, rather confusingly, implements `gzip` compression not `deflate`). There is also a third-party module for Apache called mod_gzip.c. There are a number of third-party GZIP products available as add-ons for most web servers.

The advantages of implementing a compression solution directly in the Web Gateway are as follows:

- Ease of setup and configuration.
- Greater flexibility in controlling which CSP files are to be compressed.
- Compression tends to work better if the data is submitted to the compressor functions in large buffers. The Web Gateway receives the response content from InterSystems IRIS in fairly large chunks; therefore the performance of the compression and the degree of compression achieved are good.

It has been discovered that if Chunked Transfer Encoding is enabled at the Web Gateway level and if the Apache mod_deflate output filter is enabled for the same resources, then browsers are occasionally unable to display the response content.

The Web Gateway makes use of the freely available GZIP (or `zlib`) library for implementing data compression. The compression algorithm used is described in RFCs (Request for Comments) 1950 to 1952.

15.1 The GZIP/ZLIB Library

The GZIP/ZLIB library was developed by Jean-loup Gailly and Mark Adler (Copyright (C) 1995-2009). A pre-built version of this library is provided with InterSystems IRIS distributions on Windows. On UNIX systems, the Web Gateway uses the OS-supplied build of ZLIB.

The Web Gateway dynamically links to the ZLIB library when response compression is requested for the first time. Thereafter the ZLIB library remains loaded until the Web Gateway is closed down.

If the Web Gateway is able to load the ZLIB library on demand and identify all the required functions, an initialization message is written to the Event Log in the following format (with *x.x.xx* standing in for the version of the library on your system):

Web Gateway Initialization
The ZLIB library is loaded - Version *x.x.xx*.
(This library is used for the optional GZIP compression facility)

If the Web Gateway cannot find or link to the ZLIB library, it operates as before and pages are returned without being compressed. A statement of failure is written to the Event Log.

15.2 Using the GZIP/ZLIB Library

The Web Gateway implements two modes of operation (1 and 2) for compressing the response data using the ZLIB library:

1. In this mode, the Web Gateway streams all data received from InterSystems IRIS into the compressor. When all the data has been processed, the compressor streams the compressed data back to the Web Gateway at which point it is forwarded on to the client.

   This mode offers the best possible compression at the expense of slightly higher latency. Of course, the latency is more pronounced for larger forms.

2. In this mode, the Web Gateway streams all data received from InterSystems IRIS into the compressor. On each and every call, the compressor makes as much compressed data as it can available to the Web Gateway at which point it is forwarded on to the client.

   This mode offers the lowest possible latency at the expense of slightly reduced level of compression. Of course, the reduction in the degree of compression achieved is more pronounced for larger forms. Generally speaking, mode 2 is more appropriate for web applications where it is usually not possible to know, in advance, how much data a response contains.

If (and only if) the Web Gateway is able to successfully compress the data stream returned from InterSystems IRIS, it modifies the HTTP response headers to include the appropriate Content-Encoding directive. For example:

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=ISO-8859-1
Set-Cookie: CSPSESSIONID=000000000002119qMwh3003228403243; path=/csp/samples/;
Cache-Control: no-cache
Connection: Close
Date: <date and time>
Expires: <date and ttime>
Pragma: no-cache
Content-Encoding: gzip
```

Before attempting to compress response data, the Web Gateway always checks the value of the `Accept-Encoding` HTTP request header (the HTTP_ACCEPT_ENCODING CGI environment variable). The Web Gateway only compresses a response if the client has indicated that it is capable of dealing with compressed content.

For example:

```
Accept-Encoding: gzip, deflate
```

There are several methods for specifying that a CSP response should be compressed. These are discussed in the following sections.

15.3 Specifying Compression for Individual Pages

Within web applications, the `%response` object contains a property called `GzipOutput`. If this property is set to true (or the mode required) the Web Gateway attempts to compress the response.

```
<script language=objectscript method=OnPreHTTP arguments=""
         returntype=%Boolean>
         Set %response.GzipOutput = 2
         Quit 1
</script>
```

Compression can also be specified on a per-page basis by adding the **CSP-gzip** directive to the HTTP response headers. This must, of course, be done in the `OnPreHTTP` method. For example:

```
<script language=objectscript method=OnPreHTTP arguments=""
        returntype=%Boolean>
        Do %response.SetHeader("CSP-gzip", "2")
        Quit 1
</script>
```

The `CSP-gzip` header directive should be set to the compression mode required (1 or 2).

15.4 Specifying Compression for All Pages within an Application Path

Compression can be specified on a per-application path basis. This, incidentally, is the most common method for indicating that compression should be used when using a web server output filter (such as `mod_deflate`).

Use the following configuration parameters in the Web Gateway Application Access section:

Item	Function
GZIP Compression	If Enabled, all CSP output for that path is compressed. Default is Enabled.
GZIP Minimum File Size	Controls the minimum response size in bytes for which compression is activated. If left empty, then all responses for which GZIP is enabled are compressed.
GZIP Exclude File Types	List of file types to be excluded from GZIP compression. Files can be listed by MIME type (such as image/jpeg) or by common extension (such as jpeg). By default, these common (natively compressed) image files are excluded: `GZIP Exclude File Types: jpeg gif ico png gz zip mp3 mp4 tiff`. Separate additional types or extensions with a space.

15.5 Monitoring

Log level V3 instructs the Web Gateway to record the degree of compression achieved for all responses that were successfully compressed. The size of the compressed data and the original uncompressed data stream is recorded.

For example:

```
GZIP Compression for /csp/samples/inspector.csp
GZIP Mode=1; Uncompressed Content Size=19042; Compressed Content Size=2499 (13 percent)
```

16

Implementing HTTP Authentication for Web Applications

The Apache modules (mod_csp*.so/dll and CSPa*[Sys].so/dll) to allow HTTP authentication to be controlled by InterSystems IRIS®.

HTTP authentication of web requests is normally carried out between the web server and client (browser). Because of this, it is not usually possible to implement HTTP authentication in custom request handlers hosted by the web server – such as CGI programs and web server API-based request handlers. Of course, such extensions can issue a `401 Authorization Required` response header and, in response to this, the browser displays the HTTP login dialog. However, in the subsequent request, the web server intercepts the user's login details and attempts to authenticate the user using its own built-in functionality. The username and password are not, at least in the first instance, passed along to the request handling extension until the web server has authenticated the user on its own terms.

This scheme presents a problem for users of third-party development technologies (such as CSP) who wish to perform HTTP authentication locally (and programmatically) within their technology of choice.

The feature described here overcomes these technical difficulties and allows users to perform HTTP authentication in the InterSystems IRIS environment for Apache-hosted web applications. Users of Apache can choose between the three approaches described in the following sections.

16.1 Standard HTTP authentication in Apache (mod_auth)

This method is the standard mechanism provided by Apache (through the `mod_auth` module) and does not involve the Web Gateway. It is mentioned here for the sake of completeness.

As an example, the basic parameters required for protecting the CSP samples using Apache-based authentication are shown in the following configuration block (httpd.conf):

```
<Location "/csp/samples/">
    AuthType Basic
    AuthName "CSP samples"
    AuthUserFile conf/csp.pwd
    require valid-user
</Location>
```

Where:

AuthType is the type of authorization required (usually `Basic`).

AuthName is the realm.

AuthUserFile is the file (relative to the web server root) holding usernames and their associated passwords (in encrypted form). This file is created and maintained by the Apache `htpasswd` utility.

The *require* parameter lists the users who may access the protected resource (the CSP samples in this case). The *valid-user* argument indicates that the user must be defined in the username/password file (as declared in AuthUserFile).

Apache provides for users to be grouped together in user 'groups' – see the *AuthGroupFile* directive for further details:

`https://httpd.apache.org/docs/2.4/mod/mod_authz_groupfile.html#authgroupfile`

16.2 Authenticating in CSP at the Same Time as the Request is Processed.

This is the preferred (and best performing) method for implementing HTTP authentication in web applications.

The basic parameters required for protecting the CSP samples using CSP-based authentication are shown in the following Apache configuration block (httpd.conf):

```
<Location "/csp/samples/">
    AuthType Basic
    AuthName "CSP samples"
    require valid-user
    AuthCSPEnable On
    AuthBasicAuthoritative Off
</Location>
```

The parameters *AuthType*, *AuthName* and *require* are the standard Apache parameters used for triggering authentication.

The additional *AuthCSPEnable* parameter instructs the CSP module to bypass the authentication checks that would otherwise be performed by Apache (in mod_auth) and pass the user's name and password, along with the original web request, to InterSystems IRIS for authentication. The web application must check the user using the following CGI environment variables:

- AUTH_TYPE: This is Basic.
- REMOTE_USER: The user's name.
- AUTH_PASSWORD: The user's password (as plain text).

If the user can be successfully authenticated based on the values held in these parameters then the application should continue and process the request (i.e. return the requested CSP resource). If not, it should return a HTTP `401 Authorization Required` response which, at the very least, should be something like:

```
HTTP/1.1 401 Authorization Required
WWW-Authenticate: Basic realm="CSP samples"
Content-Type: text/html
Connection: close
<html>
<head><title>401 Authorization Required</title>
</head><body> <h1>Authorization Required</h1>
<p> The server could not verify that you are authorized
to access the application. Check your username and password.
</p>
<hr>
</body>
</html>
```

On receiving this message the browser redisplays the login dialog unless the user has used-up all his/her login attempts (usually 3) in which case the message following the header is displayed instead.

Users can implement this method of authentication by modifying the login page. If a request comes in and the user does not have the necessary privileges to run the application then the login page is called, the processing for which can extract the authentication information from the request (such as AUTH_TYPE, REMOTE_USER and AUTH_PASSWORD). If these parameters are correct, the login script can then redirect control to the application page that was originally requested. It should not be necessary to repeat the authentication procedure for all public pages provided the InterSystems security control layer is deployed.

16.3 Authenticating in CSP before the Request is Processed.

This is an alternative method for implementing HTTP authentication in InterSystems IRIS. It is intended primarily for cases where performing authentication at request-processing time in the web application would be awkward or time consuming.

In this method, the user is authenticated by calling a dedicated authentication class. The Web Gateway performs this check before dispatching the original request to InterSystems IRIS. When the user's details have been successfully checked by the authentication class, the web application need not perform any further any further checking.

Of course, this method bears the overhead of processing two requests (to InterSystems IRIS) per web request: one for authentication and one for actually dealing with the request for the CSP resource.

The basic parameters required for implementing this method of authentication are shown in the following Apache configuration block (httpd.conf):

```
<Location "/csp/samples/">
    AuthType Basic
    AuthName "CSP samples"
    require valid-user
    AuthCSPEnable On
    AuthCSPClass /csp/samples/%CSP.HTTPAuthentication.cls
    AuthBasicAuthoritative Off
</Location>
```

The parameters *AuthType*, *AuthName*, *require* and *AuthCSPEnable* are the same as for method (2).

The additional *AuthCSPClass* parameter defines a class that performs user authentication. The class must extend %CSP.Page and, using the appropriate CGI environment variables, should check the user's login details and return either a `200 OK` response header if the operation is successful or a `401 Authorization Required` response header if not.

A simple authentication class in which user login details are checked against records held in the %Users file is shown below:

```
Class %CSP.HTTPAuthentication Extends %CSP.Page
{
        ClassMethod OnPreHTTP() As %Boolean
        {
                Set %response.ContentType = "text/html"
                Set %session.Preserve = 0
                Quit 1
        }
        ClassMethod OnPage() As %Status
        {
                Set crlf=$Char(13,10)
                Set type=%request.GetCgiEnv("AUTH_TYPE", "")
                Set user=%request.GetCgiEnv("REMOTE_USER", "")
                Set pwd=%request.GetCgiEnv("AUTH_PASSWORD", "")
                Set httpauth=%request.GetCgiEnv("HTTP_AUTHORIZATION", "")
                If httpauth'="" {
      Set type=$Piece(httpauth," ",1)
      Set user=$system.Encryption.Base64Decode($Piece(httpauth," ",2))
      Set pwd=$Piece(user,":",2)
      Set user=$Piece(user,":",1)
                }
                Set auth=0
                If $ZConvert(type,"L")'="basic" Set auth=1
                If auth=0,user'="",$Get(^%Users(user))=pwd Set auth=1
```

```
                If auth=1 {
                        Write "HTTP/1.1 200 OK"_crlf
                        Write "Content-Type: text/html"_crlf
                        Write "Content-Length: 0"_crlf
                        Write "Connection: close"_crlf_crlf
                }
                Else {
                        Write "HTTP/1.1 401 Authorization Required"_crlf
                        Write "WWW-Authenticate: Basic realm=""CSP samples"""_crlf
                        Write "Content-Type: text/html"_crlf
                        Write "Content-Length: 0"_crlf
                        Write "Connection: close"_crlf_crlf
                }
                Quit $$$OK
        }
        ClassMethod OnHTTPHeader(ByRef OutputBody As %Boolean) As %Status
        {
                Quit $$$OK
        }
}
```

For methods (1) and (3) a custom error page can be specified for login failure by using the Apache `ErrorDocument` directive. For example:

```
ErrorDocument /error/my_authentication_error.html
```

Of course, for method (2) the text of the error message is controlled by the web application.

17

Load Balancing, Failover, and Mirrored Configurations

17.1 Load Balancing and Failover Between Multiple Web Servers

In most environments, multiple web servers are used to balance load and provide high availability at the web server layer. A load balancer is typically required to direct user connections to participating web servers. For best performance and resilience, it is recommended that a hardware-based solution is used. A Load Balancing system such as Cisco ACE 4710 or the F5 BigIP LTM appliance is placed in front of a set of web servers. In this configuration, if there are also multiple InterSystems IRIS server instances, such as in a distributed cache cluster, each web server (and by implication, Web Gateway instance) should be configured to connect to a specific InterSystems IRIS® server instance.

Software based load-balancing and failover systems, though not as robust as hardware based solutions, are much less costly to deploy. Examples of software based solutions include HAProxy and the Apache Group's mod_proxy_balancer. For more information, see the HAProxy site www.haproxy.org

Important: Persistent ("sticky") sessions should always be enabled for web applications. It is essential that each user session "sticks" to the same back-end InterSystems IRIS server for the lifetime of the session – unless, of course, a failover event occurs.

Although the above approach is the primary recommendation, the Web Gateway provides a basic (software-based) system for implementing load balancing and failover between multiple InterSystems IRIS servers (that is, the CSP servers for multiple InterSystems IRIS instances). This facility is described in the following section.

17.2 Load Balancing and Failover Between Multiple InterSystems IRIS Server Instances

In configurations with multiple (equivalent) InterSystems IRIS server instances, such as in a distributed cache cluster, the Web Gateway provides a basic (software-based) facility for implementing load balancing and failover between those InterSystems IRIS instances for web applications. An external solution like those described previously is the primary recommendation, however.

The failover mechanism provided by the Web Gateway is not necessary to implement failover between multiple InterSystems IRIS database servers in a typical High Availability configuration, such as failover clustering or InterSystems IRIS mirroring. Those technologies provide Virtual IP based failover and the Web Gateway can be configured to connect to that IP address.

The remainder of this section describes the load balancing and failover capabilities provided by the Web Gateway.

Web Gateway load balancing and failover is configured in the **Application Access** section of the Web Gateway management pages. See Configuring Application Access.

Navigate to **System Administration** > **Configuration** > **Web Gateway Management** and select **Application Access**. A list of InterSystems IRIS servers may be defined for an application (path). Use the options listed under the **Use Alternative Servers For** parameter to select the purpose for which they are to be used. The following options are available:

- **Fail-Over**
- **Load-Balancing and Fail-Over**

The default course of action is to use the first InterSystems IRIS server defined in the list. Following this default server is the list of alternative InterSystems IRIS servers, each designated as **Server #** where # is the server number.

The configuration screen initially shows only three empty server slots, but additional slots appear that enable you to define any number of alternative servers. Each server can be marked as **Enabled** or **Disabled**. The default setting is **Enabled**.

Load balancing is implemented in a round robin fashion. Each new user session is connected to the next available alternative server. Once a user session is established on a server, the Web Gateway maintains the session on that server unless it becomes unavailable, in which case the session is failed-over to the next available server in the list. State-aware sessions (preserve mode = 1) cannot be failed-over under any circumstances and, consequently, the session is closed if the hosting server becomes unavailable.

If a CSP server does not respond with the span of time specified by the **Server Response Timeout**, the Web Gateway marks the server as offline and does not use it for load balancing. However, if Web Gateway Registry functions are enabled (they are enabled by default), then the Web Gateway periodically attempts to reconnect to CSP servers which are offline. If the connection to the CSP server succeeds, the Web Gateway marks it online and uses it for load balancing again, automatically.

17.3 Mirrored Configurations

With mirrored InterSystems IRIS configurations, a database is duplicated (or *mirrored*) between participating *mirror members*. An InterSystems IRIS *mirror set* configuration represents the set of participating mirror members for an installation. For a complete description of InterSystems IRIS mirroring, see Mirroring in the *High Availability Guide*.

If Mirror Virtual IP (or an equivalent technology) is used to provide network redirection to the primary member, then configure the Web Gateway to connect to that address. No further action is required. The Virtual IP address is always mapped to the mirror primary.

For configurations where the Mirror Virtual IP cannot be used (or does not operate in certain disaster scenarios), it is possible to configure the Web Gateway to be *mirror-aware*. When the Web Gateway is mirror-aware, it assumes responsibility for determining which member is primary. To make a Web Gateway configuration mirror-aware, in the Web Gateway's **Server Access** section, select **Configuration is Mirror Aware** and provide the address of one of the mirror members.

Note: There are situations where it is not appropriate for a Web Gateway configuration to be mirror-aware. For example, a Web Gateway configuration supporting the Management Portal should never be configured to be mirror-aware as the portal must always connect to a specific InterSystems IRIS server regardless of its mirror status.

If a mirror-aware Web Gateway configuration connects to an InterSystems IRIS server that is not a mirror member then the connection fails and the affected client receives a `Server Availability` error.

The Web Gateway obtains – from the Member that it first connects to – a list of failover members and disaster recovery (DR) members. The Web Gateway persists this list in its local configuration file (CSPRT.ini). If the Web Gateway subsequently cannot connect to the member defined in its configuration then it uses the list previously recorded locally to enable it to identify and connect to alternative members.

The Web Gateway cycles through the members list until it finds the primary. If it cannot find the primary, the Web Gateway defaults to the server defined in the Gateway configuration.

- The Web Gateway repeatedly cycles through the list until it finds a member defined as primary.
- To avoid the negative performance impact of a tight looping structure, the Web Gateway pauses after each cycle for a number of seconds equal to the number of tries.
- For a given HTTP request, the Web Gateway spends no more time attempting to find the primary than that defined in the **Server Response Timeout** parameter.
- When searching for the primary, the Web Gateway always connects to failover members first. It only tries async members if it cannot find the primary amongst the failover members. An async member only becomes primary if you manually designate it as primary.

Mirror members appear in the Web Gateway System Status form when the first connection is made. Mirror members are shown named as the current configuration name (as defined under the Web Gateway's **Server Access** section) with the mirror set name, mirror, and mirror member name shown as a tooltip.

The columns, **Mirror Name** and **Mirror Status** appear in the 'InterSystems IRIS Servers' table. The name of the mirror set and mirror member are shown in the **Mirror Name** column. The current member status is shown in the **Mirror Status** column: the **Member Type** (`Failover` or `Async`) is shown and the primary member is labelled as `Primary`.

18

Process Affinity and State-Aware Mode (Preserve Mode 1)

The architecture of the web is *stateless*. In order to get the best out of web architecture in terms of performance, maintainability and scalability, web applications should embrace the stateless paradigm.

By default, web applications operate in a stateless environment with respect to the hosting InterSystems IRIS® server. The Web Gateway maintains a pool of connections to InterSystems IRIS and distributes the workload amongst them and increases, within configured limits, (or decreases) the size of the connection pool. Each connection is associated with a single InterSystems IRIS process (as identified by the *$Job* variable).

For a normal web application operating in stateless mode, consider the choice of backend InterSystems IRIS process used to serve each request for a client session to be random. The Web Gateway chooses whichever connection/process happens to be free.

However, in the interests of efficiency, the Web Gateway does implement a form of InterSystems IRIS *process affinity*. In other words, it attempts, where possible, to route a request for a session to the same InterSystems IRIS process that was used to serve the previous request for that session.

In addition to a measure of process affinity based on session ID, the Web Gateway also attempts to implement process affinity based on namespace. The Web Gateway keeps track of the namespace to which each connection is pointing and delivers, where possible, requests to a connection that is already pointing at the namespace required to process the request. This helps in avoiding the overhead incurred in moving resources between different namespaces on receiving each web request.

In terms of precedence, session affinity always overrides all other considerations in the selection of a connection. If an incoming request cannot be assigned to the same connection previously used to serve the client session, namespace affinity is used instead to influence the final choice.

CSP includes a mode whereby the Web Gateway routes all requests for a session to a reserved (or private) InterSystems IRIS connection/process. This mode of operation provides a *state-aware* environment with respect to the relationship between web sessions and their corresponding InterSystems IRIS processes.

State-aware mode is implemented as CSP Preserve Mode 1

The original motivation for the provision of a state-aware mode of operation (preserve mode 1) was to make it relatively easy to migrate legacy application code from a fixed client-server environment (e.g. terminal applications) to the web. Support for transactions that spanned several HTTP requests was also a consideration in its introduction. However, the limitations outlined in the following paragraphs should be borne in mind when creating state-aware applications.

State-aware applications do not scale as well as their stateless counterparts and it is therefore recommended that new applications (and modifications to existing ones) be designed to be stateless as far as is practically possible. It is recommended that state-aware mode, if used at all, should be applied sparingly in predominantly stateless applications.

Writing complete applications to operate in state-aware mode is not recommended. Apart from the scalability issues that arise as a consequence of the need to reserve an InterSystems IRIS process for each and every session, state-aware applications are unable to take full advantage of modern load balancing and failover solutions because of the very specific requirements for routing requests. Also, state-aware applications are not as fault-tolerant as their stateless counterparts. For example, the recycling of a web server worker process can happen transparently beneath a stateless application but results in all associated state-aware sessions closing. Of course, you can avoid the latter restriction by using the Web Gateway's NSD component to separate the management of the Web Gateway process pool from the hosting web server.

Creating a successful state-aware application (or state-aware sections within a predominantly stateless application) requires a certain amount of discipline.

Since all requests for a session must be processed by the same InterSystems IRIS process, a queue must be maintained to serialize access to the private InterSystems IRIS process for cases where the client simultaneously dispatches several requests. The original HTTP v1.1 standard mandated that a client should simultaneously open no more than 2 connections to each server (RFC2616). However, this limit is configurable and, indeed, the latest generation of web browsers support, by default, up to 8 connections to each server. Needless to say, an increase in the maximum number of connections to each server can have a profound effect on state-aware web applications: an application can expect up to 8 requests to be fired concurrently and subsequently held in the queue responsible for controlling access to the single private InterSystems IRIS process.

Another potential pitfall in state-aware mode is the effect of the Server Response Timeout operating between the Web Gateway and InterSystems IRIS. When the Web Gateway does not receive a response within the prescribed time limit imposed by the response timeout it has no option but to close the connection with the consequential loss of the state-aware session.

Finally, the effect of client interrupts can cause problems with applications operating in state-aware mode. When a client interrupts a request at (and beyond) the point at which InterSystems IRIS is generating a response, the Web Gateway attempts to absorb the (now unwanted) response payload in order to retain the connection. If it is unable to do this in a timely fashion it, again, has no option but to interrupt whatever the InterSystems IRIS process is doing by closing the connection and the session is lost. Bear in mind that while the Web Gateway is attempting to absorb the payload for an interrupted request, further requests for the same session may be arriving and placed in the queue.

In summary, follow the following design goals when creating state-aware applications.

- As far as possible avoid (or use sparingly) client constructs that generate many simultaneous requests (for example: HTML Frameset documents).
- Ensure that responses are generated quickly. This reduces the scope for issues related to timeout and/or client interrupt events. It also relieves pressure on the session queue. If a task in InterSystems IRIS potentially requires an extended time to complete, then consider performing it in another process so that the primary private process can quickly return a response to the Web Gateway (and client).

18.1 Launching State-Aware Mode

Mark a session as *state-aware* by setting the preserve mode as follows:

```
Set %session.Preserve = 1
```

It is recommended that a session be marked as state-aware in the form's OnPreHTTP method:

```
<script language=objectscript method=OnPreHTTP arguments="" returntype=%Boolean>
Set %session.Preserve = 1
Quit 1
</script>
```

Issuing the instruction here means that the CSP engine can mark the session cookie (or token) as state-aware before formulating and dispatching the HTTP response headers to the Web Gateway.

Sessions can be marked as state-aware after the **OnPreHTTP** method has fired but in this case the session cookie/token has already been formulated. The CSP engine passes the `preserve=1` instruction to the Web Gateway in the response footers (dispatched after the response payload) and the Web Gateway marks the connection as `private` and caches the instruction against the session ID so that it can recognize the unmodified session token as state-aware when subsequent requests arrive.

If the session is marked as state-aware in the **OnPreHTTP** method, the Web Gateway has no need to cache the transition against the session since the information is carried in the session cookie/token which effectively resides on the client.

18.2 Maintaining State-Aware Mode and Responding to Errors

Once a session is marked as state-aware and the Web Gateway has acknowledged the state-transition and marked the connection as *private*, the session transparently operates in state-aware mode until one of the following events occurs:

- The application transitions back to a stateless mode of operation.
- The application programmatically ends the session or the session times out.
- The private connection closes prematurely as a result of some error condition.

If the private connection hosting a state-aware application is prematurely closed (perhaps as a result of an error condition), the Web Gateway routes the request to a free stateless connection in the pool and InterSystems IRIS error number 5974 is returned:

```
CSP error occurred
Error: The persistent session is no longer available because the server process does not exist
ErrorNo: 5974
CSP Page: /csp/samples/loop.csp
Namespace: %SYS
Class: <Unknown>
```

At this point, the request is operating in stateless mode and it is the application's responsibility to respond to this error: for example, by directing the user back to the login form for the application.

When operating in state-aware mode, the value of %session.NewSession should be checked in every page. Alternatively, the application should check the validity of user specific authentication data stored in %session.Data when the user was first authorized to access the application. These checks are important for security reasons and to ensure that the user session is still securely locked-in to a state-aware mode of operation. An error condition is not automatically raised under these circumstances because it is possible that the session had already (and legitimately) transitioned out of state-aware mode. For example, consider the situation where an incoming session token is still marked as state-aware but the application had already transitioned to stateless mode – this situation arising as a result of a session token being embedded in a form (as CSPCHD) that was served before the transition was made.

Finally bear in mind that when a session is terminated (for example, after it has timed out) the CSP engine deletes all operational data associated with the session, after which point any further incoming requests for that session are treated as though they are for a new session.

The embedded security mechanisms provided by InterSystems IRIS for web applications offer protection against the eventualities outlined above. Users are automatically directed to the login form in all cases where a loss of continuity within a state-aware application occurs (with respect to InterSystems IRIS process).

18.3 Terminating State-Aware Mode

An application can revert back to a *stateless* mode of operation by setting the preserve mode as follows:

```
Set %session.Preserve = 0
```

It is recommended that this code be executed in the form's OnPreHTTP method:

```
<script language=objectscript method=OnPreHTTP arguments="" returntype=%Boolean>
    Set %session.Preserve = 0
    Quit 1
</script>
```

Issuing the instruction here means that the CSP engine can mark the session cookie (or token) as stateless before formulating and dispatching the HTTP response headers to the Web Gateway.

A session can be immediately terminated as follows:

```
Set %session.EndSession = 1
```

When you set this property, the session terminates immediately after serving the current request.

You can set a session to timeout as follows:

```
Set %session.AppTimeout = 900
```

The session times out and terminates after the prescribed number of seconds of inactivity. The default is 900 seconds (15 minutes).

19

Web Gateway Registry in InterSystems IRIS

The InterSystems Web Gateway Registry registers each connected Web Gateway installation with InterSystems IRIS® and provides the infrastructure to allow InterSystems IRIS code to interact with those installations (to clear caches and so on). Such programmatically controlled interactions may include reading and modifying the Web Gateway's runtime configuration and collecting system status and log information. The relevant classes are as follows:

```
%CSP.Mgr.GatewayRegistry (The Gateway Registry)
%CSP.Mgr.GatewayMgr  (A Connected Gateway)
```

The following code lists all connected (i.e. active) Web Gateway installations and writes the web server IP address, port and Web Gateway build number to the console window.

```
Set reqistry = $system.CSP.GetGatewayRegistry()
Set gateways = reqistry.GetGatewayMgrs()
For no=1:1:gateways.Count() {
     Set gateway = gateways.GetAt(no)
     Write !,no, " : "
     Write gateway.IPAddress,":",gateway.Port," ",gateway.Version
}
```

When InterSystems IRIS is first started this list is empty. As Administrator and User activity increases, expect at least one entry to appear for the Web Gateway which serves your instance's applications.

You can find further documentation associated with the classes listed above. Some code examples follow to illustrate common tasks.

List Default Parameters

```
Kill defaults
Do gateway.GetDefaultParams(.defaults)
ZWrite defaults
```

Update Default Parameter(s)

```
Kill newpars
Set newpars("Server_Response_Timeout")=30
Do gateway.SetDefaultParams(.newpars)
```

List Servers

```
Set status = gateway.GetServers(.servers)
For no=1:1:$ListLength(servers) {
     Set server = $List(servers,no)
     Write !,no, " : ",server
}
```

List Server Parameters

```
Kill serverpars
Do gateway.GetServerParams("LOCAL",.serverpars)
ZWrite serverpars
```

Update Server Parameter(s)

```
Kill newpars
Set newpars("Maximum_Server_Connections")=250
Do gateway.SetServerParams("LOCAL",.newpars)
```

List Application Paths

```
Set status = gateway.GetApplicationPaths(.paths)
For no=1:1:$ListLength(paths) {
     Set path = $List(paths,no)
     Write !,no, " : ",path
}
```

List Application Parameters

```
Kill pathpars
Do gateway.GetApplicationParams("/csp",.pathpars)
ZWrite pathpars
```

Update Application Parameter(s)

```
Kill newpars
Set newpars("GZIP_Compression")="Enabled"
```

Clear Gateway cache

```
Do gateway.ClearCache("*")
```

19.1 Forcing the Web Gateway to Reload Its Configuration

There are occasions when the Web Gateway's configuration is modified by external agents (i.e. agents other than the Web Gateway's own Systems Management Suite).

There are two methods for interactively instructing the Web Gateway to reload its configuration, and in a way that doesn't require a complete restart.

19.1.1 Using the InterSystems IRIS Web Gateway Registry

The following Registry Method is provided:

```
Set status = %CSP.Mgr.GatewayMgr.ActivateCSPIni()
```

When successfully called, the Web Gateway reads its configuration file and activates all changes made.

19.1.2 Using Scripts External to InterSystems IRIS

Scripts should add the following line (case-sensitive) to the SYSTEM section of the modified Web Gateway configuration file:

```
[SYSTEM]
RELOAD=1
```

The Web Gateway caretaker daemon checks the *RELOAD* flag approximately every minute and, if correctly set, reloads and reactivates its configuration and removes the flag from the file. The following message is written to the Event Log after a successful reload operation:

```
Gateway Management
Gateway Configuration Reloaded and Reactivated
```

Web Gateway Configuration File (CSP.ini) Parameter Reference

The InterSystems® Web Gateway maintains its configuration information in its CSP.ini file, which is located in the same directory as the Web Gateway binaries.

In most cases, InterSystems recommends using the Web Gateway management pages or Web Gateway Registry methods to modify the Web Gateway configuration, rather than editing the CSP.ini file directly. However, this is impractical for some use cases. For example, to initialize pre-configured `webgateway` containers automatically as part of a scalable Kubernetes cluster, it is necessary to leverage the CSP.ini merge feature to populate each Web Gateway's CSP.ini file upon deployment. See Using the InterSystems Web Gateway Container for details regarding this use case.

After you make changes to CSP.ini parameters, you can reload the Web Gateway configuration procedurally using the Web Gateway Registry **%CSP.Mgr.GatewayMgr.ActivateCSPIni()** method or by adding the `RELOAD` to the `[SYSTEM]` section of the CSP.ini file.

The parameters in the CSP.ini file are organized into functional sections of several types. This reference documents the contents of the CSP.ini file for each section.

[SYSTEM]

Describes the default parameters for the Web Gateway. The CSP.ini file contains only one `[SYSTEM]` section.

Available Parameters

RELOAD

A flag that instructs the Web Gateway caretaker daemon to reload and reactivate the Web Gateway configuration. The caretaker daemon then removes the flag. To procedurally trigger a reload using this flag, specify `RELOAD=1`.

DPAPI

(For Web Gateways installed on a Windows machine only.) A flag that determines whether or not password fields within the CSP.ini file are automatically encrypted using Microsoft's Data Protection API (DPAPI). Password encryption is enabled by default and InterSystems recommends that users leave it enabled. Where no alternative solution exists, you can disable DPAPI password encryption by specifying `DPAPI=Disable`.

Instance_Host_Name

The network host name for this Web Gateway (`server_name:port`), if different from the default. This parameter is transmitted to InterSystems IRIS along with the request data as system variable *CSPIHN*.

MAX_CONNECTIONS

The maximum number of connections that this Web Gateway can establish to InterSystems IRIS application server processes. The default value is `1024`.

MAX_CACHE_SIZE

The maximum amount of shared memory allocated for caching application response data, defined in kilobyes (`128K`) or megabytes (`16M`). The default value is `256K`.

Web_Server_ID_Cookie

Whether the Web Server ID Cookie (*CSPWSERVERID*) is `Enabled` or `Disabled` for load balancers to implement passive cookie affinity. The default value is `Enabled`.

SM_Forms

Whether access to the Web Gateway management pages is `Enabled` or `Disabled`. The default value is `Enabled`.

Username

The username required to access the Web Gateway management pages, if specified.

Password

The password required to access the Web Gateway management pages, if specified.

SM_Timeout

When you have configured authentication for the Web Gateway management pages, the allowable idle time before the expiration of an authenticated session, in seconds. The default value is `28800`.

Custom_SM_Login_Form

Full file path or relative URL path to a custom login form that controls access to the Web Gateway management pages

System_Manager

A list of IP addresses corresponding to machines permitted to access the Web Gateway management pages. You can use the wildcard character (`*`) can be used to specify a range of addresses—for example, `190.8.7.*` or `*.*.*.*`. By default, access is only granted to the host machine. See Enabling Access from Additional Client Addresses for further guidance in configuring this parameter. By default, only the local machine is permitted.

Accept_X_Forwarded_For

When specified, whether to allow access to the Web Gateway management pages in response to requests forwarded by a proxy server from an origin IP address specified by the System_Manager parameter. Allowed values are `Enabled` (to accept) and `Disabled` (to deny).

System_Manager_UNPW_Override

When specified, determines whether the Web Gateway management pages require a username and password for machines specified by the System_Manager parameter. Allowed values are `1` (override enabled) or `0` (override disabled).

Server_Response_Timeout

The maximum amount of time allowed for a target InterSystems IRIS application server to respond to a request from the web server, in seconds. The minimum allowable value is `5`. The default value is `60`.

Queued_Request_Timeout

The maximum amount of time allowed for a request to remain in a queue waiting for an appropriate InterSystems IRIS server, in seconds. The minimum allowed value is `5`. The default value is

No_Activity_Timeout

For stateless connections, the maximum amount of time before the Web Gateway closes an idle connection to an InterSystems IRIS application server, in seconds. The default value is `86400`.

Timeout_All_Connections

When specified, determines whether to terminate any connection which remains idle for the length of time specified by the No_Activity_Timeout parameter, including connections in the minimal connection pool.

Env_Parameters

The Event Log Level.

Event_Log_File

When specified, determines an alternative filesystem location and file name for storing the Web Gateway Event Log. (If not specified, the Web Gateway saves the log as CSP.log within the same directory as the Web Gateway binary files).

Event_Log_Rotation_Size

When specified, determines the maximum size the Event Log file can reach before Event Log rotation takes place. This quantity can be defined in bytes (`128000`, with no suffix), kilobytes (`256K`), or megabytes (`8M`). The minimum allowed value is `100K`. This parameter is not specified by default, which means you must clear the Event Log file manually.

Maximum_Logged_Request_Size

Maximum volume of data to log for any given HTTP request. The minimum allowed value is `40K`. The default value is `256K`.

Retain_All_Log_Files

When specified, determines whether to save all old log files which Event Log rotation generates, instead of only the most recent. Allowed values are `Enabled` (to retain all) or `Disabled` (to retain only one).

SA_Exclude_File_Types

When specified, preserves asynchronous processing for static files of the specified types for state-aware applications (as well as stateless applications). This parameter accepts a space-separated list of file type extensions.

Document_Root

When specified, provides the full physical path to the web server's web document root directory. This parameter is only required for web applications which render content using the Microsoft ASP engine.

ASP_Directory

When specified, provides the full physical path to a directory where the Web Gateway can temporarily store Microsoft ASP content. This parameter is only required for web applications which render content using the Microsoft ASP engine.

WS_Service_Status

For Web Gateway implementations which use the Network Service Daemon (NSD), determines whether the internal HTTP server for the NSD is enabled to respond to raw HTTP requests. Allowed values are `Enabled` or `Disabled`. The default value is `Enabled`.

For security reasons, InterSystems recommends disabling this facility.

NSD_Document_Root

In rare cases where the NSD serves as a stand-alone server, provides the full physical path to the web document root directory for the NSD.

Server_Error

The path and filename for the custom page the Web Gateway should display when it encounters an internal error.

Server_Busy

The path and filename for the custom page the Web Gateway should display when all available connections are in use.

Server_Unavailable

The path and filename for the custom page the Web Gateway should display when the InterSystems IRIS application server (or application) has been deliberately disabled from within the configuration.

Server_Timeout

The path and filename for the custom page the Web Gateway should display when a request times out.

Connection_Closed

The path and filename for the custom page the Web Gateway should display when a user logs out of a state-aware session.

Default_Server

Specifies the server to which the Web Gateway directs requests when the application access profile for an application path specifies no destination server. By default, this parameter specifies the automatically-generated LOCAL server access profile.

[<server>]

Describes the server access profile with the name <*server*>. The CSP.ini file contains one `[<server>]` section for each server access profile defined within the Web Gateway configuration.

Details

Each InterSystems IRIS application server to which the Web Gateway connects corresponds to a server access profile. The CSP.ini file maintains each server access profile as a section of the CSP.ini file. Each server access profile section begins with the header line `[<server>]`, where <*server*> represents the name of the server access profile. For example, the section corresponding to a server access profile named `irisserver1` would begin with the line `[irisserver1]`.

In addition to the parameters described in this page, you can also specify values for the following parameters, which would override any defaults specified in the [SYSTEM] section:

Available Parameters

Ip_Address

The DNS host name or IP address of the InterSystems IRIS application server.

TCP_Port

The TCP port number at which the superserver for this InterSystems IRIS instance is listening for incoming Web Gateway connections.

Mirror_Aware

If specified, identifies the InterSystems IRIS application server as a mirror primary, accessing mirrored databases. Specify a value of `1` to enable.

Minimum_Server_Connections

The minimum number of process-affinitive connections that the Web Gateway should make to this InterSystems IRIS application server before beginning to share the connections among clients. The default value is `3`.

Maximum_Server_Connections

The maximum number of connections that the Web Gateway is allowed to make to this InterSystems IRIS application server. By default this is unspecified, and inherits the value of *MAX_CONNECTIONS* for the Web Gateway.

Maximum_Session_Connections

The maximum number of connections to this InterSystems IRIS application server which can be used concurrently by an individual session. The default value is `3`.

Connection_Security_Level

A numeric value indicating how you have chosen to secure the connection between the Web Gateway and this InterSystems IRIS application server. Allowed values are:

- `0` — Password
- `1` — Kerberos
- `2` — Kerberos with Packet Integrity
- `3` — Kerberos with Encryption

- 4 — SSL/TLS

Username

The username the Web Gateway must use to authenticate its connection to the InterSystems IRIS server.

Password

The password which the Web Gateway must use to authenticate its connection to the InterSystems IRIS application server.

Alternatively, on UNIX®/Linux/macOS systems, this parameter can specify an operating system command within braces ({}). For example: `Password={sh /tmp/PWretrieve.sh}`. The Web Gateway executes the command when the command is saved as part of a server access profile within the Web Gateway management pages or when the `RELOAD=1` flag is found in the CSP.ini file's `[SYSTEM]` section. The output of the command is then stored as the password for the application server within memory.

The value of this parameter is stored as a hash value within the CSP.ini file.

Product

A numeric value indicating what InterSystems product the application server is associated with (InterSystems IRIS). Allowed values are:

- 0 — InterSystems Caché®
- 1 — InterSystems Ensemble®
- 2 — InterSystems IRIS, InterSystems IRIS for Health, or HealthShare® products

Service_Principal_Name

The service principal name which identifies this InterSystems IRIS server within your implementation of Kerberos-based authentication for Web Gateway connections.

Keytable

The location of the keytab file, if you are using Kerberos-based authentication.

SSLCC_Protocol_Min

A numeric value indicating the minimum SSL/TLS protocol version the Web Gateway and the InterSystems IRIS application server can use to secure their connection. Allowed values are:

- 4 — TLSv1.0
- 8 — TLSv1.1
- 16 — TLSv1.2
- 32 — TLSv1.3 (where supported)

When TLSv1.3 is supported, the default value is 16. Otherwise, the default value is 8.

SSLCC_Protocol_Max

A numeric value indicating the maximum SSL/TLS protocol version the Web Gateway and the InterSystems IRIS application server can use to secure their connection. Allowed values are:

- 4 — TLSv1.0

- 8 — TLSv1.1
- 16 — TLSv1.2
- 32 — TLSv1.3 (where supported)

When TLSv1.3 is supported, the default value is 32. Otherwise, the default value is 16.

SSLCC_Verify_Peer

If specified, requires peer certificate verification for the InterSystems IRIS application server. Specify a value of 1 to enable.

SSLCC_Cipher_Suites

Specifies the accepted cipher suites when the connection is secured with TLSv1.2 or below. The default value is ALL:!aNULL:!eNULL:!EXP:!SSLv2.

SSLCC_Cipher_Suites_1_3

Specifies the accepted cipher suites when the connection is secured with TLSv1.3. The default value is TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256.

SSLCC_Certificate_File

The full path to the SSL/TLS certificate file for the Web Gateway. Supported file formats for certificate files are the same as those supported for InterSystems IRIS TLS Configurations.

SSLCC_Certificate_Key_File

The full path to the private key associated with the Web Gateway's SSL/TLS certificate. Supported file formats for certificate files are the same as those supported for InterSystems IRIS TLS Configurations.

SSLCC_Key_Type

A numeric value indicating the cryptographic algorithm to which the key corresponds. Allowed values are:

- 1 — DSA
- 2 — RSA

SSLCC_Private_Key_Password

If specified, the password required to access the Web Gateway's private key file.

Alternatively, on UNIX®/Linux/macOS systems, this parameter can specify an operating system command within braces. For example: SSLCC_Private_Key_Password={sh /tmp/tlsPWretrieve.sh}. The Web Gateway executes the command when it is saved as part of a server access profile within the Web Gateway management pages or when the RELOAD=1 flag is present in the CSP.ini file's [SYSTEM] section. The output of the command is then stored as the private key password within memory.

This password is stored as a hash value within the CSP.ini file.

SSLCC_CA_Certificate_File

The full path to the certificate for Certificate Authority (CA) for the Web Gateway's certificate. Supported file formats for certificate files are the same as those supported for InterSystems IRIS TLS Configurations.

[SYSTEM_INDEX]

Describes the status (`Enabled` or `Disabled`) for each server access profile defined within the Web Gateway configuration. The CSP.ini file contains only one `[SYSTEM_INDEX]` section.

Details

The `[SYSTEM_INDEX]` section of the CSP.ini file specifies the InterSystems IRIS application servers to which the Web Gateway is actively routing requests. Each line in this section identifies an application server by the name of its server access profile and specifies whether routing to that server is `Enabled` or `Disabled`.

For example, the CSP.ini file for a Web Gateway configured with two active server access profiles named `irisserver1` and `irisserver2` would include the following lines:

```
[SYSTEM_INDEX]
irisserver1=Enabled
irisserver2=Enabled
```

[APP_PATH:<appPath>]

Describes the application access profile for the application path <*appPath*>. The CSP.ini file contains one `[APP_PATH:<appPath>]` section for each application access profile defined within the Web Gateway configuration.

Details

An application access profile defines the way that the Web Gateway routes requests for an application path. The CSP.ini file maintains each application access profile as a section of the CSP.ini file. Each application access profile section begins with the header line `[APP_PATH:<appPath>]`, where <*appPath*> represents the application path. For example, the section corresponding to a server access profile named `irisserver1` would begin with the line `[irisserver1]`.

Available Parameters

Default_Server

The name of the server access profile to which the Web Gateway should attempt to route requests for this application path by default, before any alternative servers. You should also define this server as an Alternative_Server, usually *Alternative_Server_0*.

Alternative_Servers

A numeric value indicating the way the Web Gateway should route requests to alternative InterSystems IRIS application servers, when alternative servers are defined. Allowed values are:

- `FailOver` — enables failover from one application server onto the next available alternative, in the order specified
- `LoadBalancing` — enables load-balancing among application servers in a round-robin fashion, along with failover
- `Disabled` — disables failover and load balancing capabilities

You must define all the application servers you want to use for an application path using *Alternative_Server_*<*n*> parameters.

Alternative_Server_0, Alternative_Server_1,... Alternative_Server_<*n*>

Strings which specify server access profiles within the Web Gateway's routing behavior for the application access profile, and whether those InterSystems IRIS application servers are enabled or disabled within the application access profile. When failover is enabled, the Web Gateway attempts to route requests for the application to the application server identified by the *Alternative_Server_0* parameter, then the *Alternative_Server_1* parameter, then *Alternative_Server_2*, and so on. The Web Gateway can continue this failover behavior for an arbitrary number (<*n*>) of alternative servers—in other words, up to the *Alternative_Server_*<*n*> parameter.

Each *Alternative_Server_*<*n*> parameter begins with a bit specifying the status of the application server within the application access profile (`1` for enabled, `0` for disabled) and ends with the name of the server access profile for the desired InterSystems IRIS application server. When the Web Gateway updates the value of this parameter, it separates the status and the server access profile name using an arbitrary number of ~ characters, for legibility.

For example, if you want to specify that the third active InterSystems IRIS application server in your application's failover sequence should be the server which corresponds to a server access profile named `irisserv1`, then you would add the following line to the application access profile block:

```
Alternative_Server_2=1~~~~~~irisserv1
```

KeepAlive

Specifies how the Web Gateway should implement HTTP `Keep-Alive` connection behavior for this application. Allowed values are:

- `No Action` — the Web Gateway allows the HTTP response headers for each request to determine the `Keep-Alive` status of a connection. This is the default value
- `Enabled`
- `Disabled`

Non_Parsed_Headers

Specifies whether the Web Gateway streams HTTP response headers directly to the client (`Enabled`) or submits them to the hosting web server for parsing and filtering (`Disabled`). The default value is `Enabled`.

Response_Size_Notification

Specifies the method the Web Gateway uses to notify clients about the amount of data contained in an HTTP response for this application. Allowed values are:

- `Chunked Transfer Encoding and Content Length` — the default value.
- `Chunked Transfer Encoding`
- `Content Length`

Response_Size_Notification_Always

Specifies whether the Web Gateway sends response size notifications for all HTTP responses which are associated with this application. Allowed values are `Enabled` or `Disabled`. The default value is `Disabled`.

GZIP_Compression

Specifies whether the Web Gateway compresses all pages returned for this application. Allowed values are `Enabled` or `Disabled`. The default value is `Disabled`.

GZIP_Minimum_File_Size

Specifies the minimum response size (in bytes) for which the Web Gateway invokes GZIP compression. The default value is `500`.

GZIP_Exclude_File_Types

The file types which the Web Gateway excludes from GZIP compression for this application, specified as a space-separated list of file type extensions. The default value includes natively compressed files: `jpeg gif ico png gz zip mp3 mp4 tiff`.

Extra_CGI_Env_Variables

Additional CGI environment variables which the Web Gateway should provide to the InterSystems IRIS application server with each request for this application. For the default set of variables which the Web Gateway provides, see CGI Environment Variables Passed by the Web Gateway.

Proc_Class

The name of a custom request handler which the Web Gateway should use to process requests for this application.

[APP_PATH_INDEX]

Describes the status (`Enabled` or `Disabled`) for each application access profile defined within the Web Gateway configuration. The CSP.ini file contains only one `[APP_PATH_INDEX]` section.

Details

The `[APP_PATH_INDEX]` section of the CSP.ini file specifies the application paths for which the Web Gateway is actively routing requests. Each line in this section identifies an application path for which you have configured an application access profile and specifies whether routing for that application path is `Enabled` or `Disabled`.

For example, the CSP.ini file for a Web Gateway configured to serve the application paths `/` (root), `/csp`, `/irisserver1`, `/irisserver2`, and `/apps/criticalapp` would include the following lines:

```
[APP_PATH_INDEX]
/=Enabled
/csp=Enabled
/irisserver1
/irisserver2
/apps/criticalapp
```

A

Using the NSD (Windows)

This page describes how to use the Network Service Daemon (NSD) with the InterSystems Web Gateway on Microsoft Windows. This is not the typical installation but is appropriate in some cases.

A.1 When to Use the NSD

There are three situations in which you might choose to use the NSD to separate the Web Gateway from the web server so that you can manage the Web Gateway independently of the web server:

- If your web server distributes its load over multiple server processes, an instance of the Web Gateway is then attached to each web server process.
- If you have a very large web server installation for which CSP is only a small part; for example, a web server that serves php, static content, .NET, and .ASP applications, as well as web applications.
- If you are using the Nginx web server.

A.2 NSD Module Install Locations

If you use the NSD Module in Microsoft Windows, you install the following two utilities:

- CSPnsd.exe
- CSPnsdSv.exe

On an IIS installation, these are installed in this location:

C:\Inetpub\CSPGateway\nsd

On an Apache installation, these are installed in this location:

C:\Program Files\Apache Group\Apache\WebGateway\nsd

Run the NSD from within its home directory, C:\Inetpub\CSPGateway\nsd. The configuration file and the log file are written in this directory for NSD-based connectivity options.

A.3 Operating the NSD

Use the following procedure to start the NSD.

1. Change to the NSD home directory, such as:

   C:\Inetpub\CSPGateway\nsd

2. Start the NSD with:

   ```
   CSPnsd
   ```

   The NSD starts as a Windows service (`CSPnsdSv.exe`). Once registered as a service, you can manage the NSD entirely through the Windows Service Manager.

3. Close down the NSD, by issuing the following command:

   ```
   CSPnsd -stop
   ```

Alternatively, you can enter:

```
CSPnsd
```

This shows the status of the NSD's Windows Service and allows you to perform one of the following actions:

- Stop the NSD service if it is running.
- Continue the NSD service if it is paused.
- Remove the NSD service from the services database.

Alternatively, you can use the Windows Service Manager to manage the NSD. The NSD can be identified in the Service Manager by the description:

```
Cache Server Pages – Network Service Daemon
```

All errors are reported in the Web Gateway Event Log.

Other Startup Options

1. Display help information.

   ```
   CSPnsd -h
   ```

2. Run the NSD interactively in a command window as opposed to as a Windows service. You must use this mode of operation if you are running multiple instances of the NSD.

   ```
   CSPnsd -v
   ```

A.3.1 Starting NSD on Alternative TCP Port

By default, the NSD listens for incoming requests on TCP port 7038. You can override this by starting the service as follows:

`CSPnsd –v` [*port_no*]

Or:

`CSPnsd –v -p`[*port_no*]

- where *port_no* is the TCP port number of your choice.

On startup, the NSD creates the CSPnsd.ini file, which typically contains the following lines:

```
[SYSTEM]
Ip_Address=127.0.0.1
TCP_Port=7038
```

In this context, the clients are the Web Gateway modules contained within, or dynamically linked to, the web server and/or the CSP CGI modules invoked by the server. It is, therefore, essential that this file is not deleted or moved. It is also important that the web server processes can read this file. Set the privileges accordingly, bearing in mind the Windows user under which your web server is operating. The NSD clients attempt to find this file in a location contained within the Windows PATH variable (for example: C:\Windows). Consequently, the CSPnsd.ini file must be moved to this location before starting the web server.

It is inappropriate to store the NSD port number in the CSPnsd.ini file for the scenario in which multiple instances of the NSD are running. For Apache servers, there is a much better mechanism for communicating the TCP port number of the NSD to its clients. Specifically, set the following environment variables in the Apache configuration to indicate the address and port of the target NSD installation. The values specified in these environment variables take precedence over any values found in the CSPnsd.ini file:

- CSP_NSD_NAME — This is the IP address of the NSD. Only use this parameter if the NSD is operating on a remote computer.
- CSP_NSD_PORT — This is the TCP port of the NSD.

A.3.1.1 Example 1: Two Apache Virtual Hosts

Distribute the load for two Apache virtual hosts (say, 123.123.1.1 and 123.123.1.2) between two independent NSD installations (listening on TCP port 7038 and 7039).

Add the following directives to the Apache configuration (httpd.conf):

```
<VirtualHost 123.123.1.1>
    ServerName 123.123.1.1
    SetEnv CSP_NSD_PORT 7038
</VirtualHost>
<VirtualHost 123.123.1.2>
    ServerName 123.123.1.2
    SetEnv CSP_NSD_PORT 7039
</VirtualHost>
```

A.3.1.2 Example 2: Two Web Applications

Distribute the load for two web applications (say, /csp1 and /csp2) between two independent NSD installations (listening on TCP port 7038 and 7039).

1. Add the following directives to the Apache configuration (httpd.conf):

   ```
   <Location /csp1>
       SetEnv CSP_NSD_PORT 7038
   </Location>
   <Location /csp2>
       SetEnv CSP_NSD_PORT 7039
   </Location>
   ```

2. Restart Apache after making changes to its configuration.

In cases where multiple instances of the NSD are running, it is recommended that the separate instances be installed in separate directories, each maintaining its own copies of the configuration file and the log file. The Web Gateway management pages for each instance can easily be accessed by using the NSD internal HTTP server. For example:

```
http://localhost:7038/csp/bin/Systems/Module.cxw
```

```
http://localhost:7039/csp/bin/Systems/Module.cxw
```

B

Using the NSD (UNIX®/Linux/macOS)

This page describes how to use the Network Service Daemon (NSD) or use with the Web Gateway on UNIX®, Linux, or macOS. This is not the typical installation but is appropriate in some cases.

B.1 When to Use the NSD

There are three situations in which you might choose to use the NSD to separate the Web Gateway from the web server so that you can manage the Web Gateway independently of the web server:

- If your web server distributes its load over multiple server processes, an instance of the Web Gateway is then attached to each web server process.
- If you have a very large web server installation for which CSP is only a small part; for example, a web server that serves php, static content, .NET, and .ASP applications, as well as web applications.
- If you are using the Nginx web server.

B.2 NSD Module Install Locations

The NSD Module, if required, is CSPnsd.

The default location for this module is:

```
/opt/webgateway/bin
```

The NSD should be run from within its home directory (above). The configuration file and log file are written in this directory for NSD-based connectivity options.

B.3 Operating the NSD

To run the NSD:

1. Change to the following directory:

   /opt/webgateway/bin

2. Enter the following command to start the NSD:

   ```
   ./CSPnsd
   ```

Before retiring to the background, the NSD displays a banner indicating its running configuration. It shows the TCP port number dedicated to this service, which is, by default, port number 7038.

You can suppress all startup messages for this command using the `-s` qualifier. For example, to start the NSD from a script invoked at system boot, use:

```
/opt/webgateway/bin/CSPnsd –s
```

Other common startup options:

- Display help information.

  ```
  ./CSPnsd -h
  ```

- Pause the operation of the NSD. This command sends a stop signal (SIGSTOP) to the NSD process.

  ```
  ./CSPnsd -pause
  ```

- Continue the operation of the NSD (after a pause). This command sends a continue signal (SIGCONT) to the NSD process.

  ```
  ./CSPnsd -cont
  ```

- Give permission to others to run the NSD. Administrators of the NSD (CSPnsd) component can give permission to a group or others to start/stop the NSD using `CSPnsd –m=`*s* where *s* is a startup option.

  s can be one of

  – `u` for the current user (default)

  – `g` for the current group

  – `o` for others

  – `a` for everyone (`m=ugo`)

  Example: `CPSnsd -m=ug` gives permissions to the group (the Administrator group) to run the NSD. This command gives the CPSnsd.pid permissions of: `-rw-rw---`

  When the command to stop the CSPnsd is issued, it tries to signal the CSPnsd parent process to shut down as before. If this is not possible because the service was started by a different user, a flag is written to the CSPnsd.ini file and the service gracefully closes itself down when it acknowledges this flag. This process takes up to 20 seconds to complete.

To close down the NSD, enter:

```
./CSPnsd -stop
```

Alternatively:

```
kill –TERM `cat /opt/webgateway/bin/CSPnsd.pid`
```

These commands close down the NSD in an orderly manner – it gracefully terminates all open connections to InterSystems IRIS and releases all its system resources before terminating. Do not use the **kill –9** command to terminate the NSD.

All errors are reported in the Web Gateway Event Log.

B.3.1 Starting the NSD on Alternative TCP Port

By default, the NSD listens for incoming requests on TCP port 7038. You can override this by starting the service as follows, where *port_no* is the TCP port number of your choice.

`./CSPnsd [port_no]`

Or:

`./CSPnsd -p=[port_no]`

On startup, the NSD creates the following file:

/opt/webgateway/bin/CSPnsd.ini

Typically, this file contains the following lines:

```
[SYSTEM]
Ip_Address=127.0.0.1
TCP_Port=7038
```

In this context, the clients are the Web Gateway module contained within, or dynamically linked to, the web server and/or the Web Gateway CGI modules invoked by the server. It is, therefore, essential that this file is not deleted or moved. It is also important that the web server processes can read this file. Set the privileges accordingly, bearing in mind the UNIX® username under which your web server is operating. The NSD clients attempt to find this file in the following locations:

/opt/webgateway/bin

/etc

If the NSD is operating in a different directory, you have to move the CSPnsd.ini file to one of the locations listed.

It is inappropriate to store the NSD port number in the CSPnsd.ini file in situations in which multiple instances of the NSD are running. For Apache servers, there is a much better mechanism for communicating the TCP port number of the NSD to its clients. Specifically, set the following environment variables in the Apache configuration to indicate the address and port of the target NSD installation.

- `CSP_NSD_NAME` — This is the IP address of the NSD. Only use this parameter if the NSD is operating on a remote computer.
- `CSP_NSD_PORT` — This is the TCP port of the NSD.

The values specified in these environment variables take precedence over any values found in the CSPnsd.ini file.

B.3.1.1 Example 1: Two Apache Virtual Hosts

To distribute the load for two Apache virtual hosts (123.123.1.1 and 123.123.1.2) between two independent NSD installations (listening on TCP port 7038 and 7039), add the following directives to the Apache configuration (httpd.conf):

```
<VirtualHost 123.123.1.1>
    ServerName 123.123.1.1
    SetEnv CSP_NSD_PORT 7038
</VirtualHost>
<VirtualHost 123.123.1.2>
    ServerName 123.123.1.2
    SetEnv CSP_NSD_PORT 7039
</VirtualHost>
```

B.3.1.2 Example 2: Two Web Applications

To distribute the load for two web applications (/csp1 and /csp2) between two independent NSD installations (listening on TCP port 7038 and 7039), add the following directives to the Apache configuration (httpd.conf):

```
<Location /csp1>
    SetEnv CSP_NSD_PORT 7038
</Location>
<Location /csp2>
    SetEnv CSP_NSD_PORT 7039
</Location>
```

Restart Apache after making changes to its configuration.

In cases where multiple instances of the NSD are running, it is recommended that the separate instances be installed in separate directories, each maintaining its own copies of the configuration file and log file. The Web Gateway management pages for each instance can easily be accessed by using the NSD's internal HTTP server. For example:

```
http://localhost:7038/csp/bin/Systems/Module.cxw
```

```
http://localhost:7039/csp/bin/Systems/Module.cxw
```

B.3.1.3 Spreading the Load over Multiple NSD Processes

By default, the NSD operates in a two-process mode of operation (one parent and one child worker).

However, there are limits to the number of threads that a single UNIX® process can start. If the concurrent load of the web application is resulting in requests queuing for available threads, consider raising the number of processes used by the NSD.

```
./CSPnsd –c=[no_processes]
```

- where no_processes is the number of child (or worker) processes to start.

It should be noted that there are even advantages in setting the number of child processes to one.

```
./CSPnsd –c=1
```

Under these circumstances, the NSD actually starts two processes: a parent and one child worker process. The presence of the parent processes when using the '–c' directive improves the resilience of the NSD because if a fault develops in one of the worker processes the parent can replace the process. For the single, multi-threaded architecture, the NSD cannot always recover from serious internal error conditions.

State-aware connectivity (preserve mode 1) should not be used in cases where the number of worker processes exceeds one.

B.3.1.4 Granting Administrator Rights to the NSD

Administrators of the NSD (CSPnsd) component can have some control over the user (or group) permitted to start/stop this service.

In the default scenario, the CSPnsd master process ID (PID) file (CSPnsd) is created such that only the user who started the service can subsequently close it down.

Administrators can now choose, for example, to allow all users belonging to the current UNIX® group to manage the service. This is the group to which the administrating user belongs.

```
NSD start-up option: [-m=s]
    Define the user(s) permitted to manage this service
        where 's' is:
            'u' for the current user (the default),
            'g' for the current group,
            'o' for others,
            'a' for everyone (m=ugo),
```

Example:

```
./CSPnsd -m=ug
```

This allows the current user and all others in the current user's group to manage the NSD.

When the command to stop the NSD is issued, it first tries to signal the CSPnsd parent process to shut down as before. If this is not possible due to the service having been started by a different user, a flag is written to the CSPnsd.ini file and the service gracefully closes itself down when it acknowledges this flag. This process takes up to 20 seconds to complete.

C

Alternative Options for Apache (UNIX®/Linux/macOS)

This page describes additional possible Apache configurations for use with the InterSystems Web Gateway on UNIX®, Linux, and macOS (apart from a locked-down Apache, discussed separately). To get started with all these configurations, read the first section. Then follow the directions in the section that applies to your configuration.

C.1 Install Locations (All Atypical Options)

This section describes directory locations for Web Gateway files and CSP static files.

1. The NSD module is:

   ```
   CSPnsd
   ```

   The default location of this module is:

   ```
   /opt/webgateway/bin
   ```

   The NSD should be run from within its home directory /opt/webgateway/bin. The CSP.ini and the **CSP.log** are written in this directory.

   In order to avoid disrupting existing Gateway installations on upgrading InterSystems IRIS®, the installation places the following modules in the common location /opt/webgateway/bin. This location is not related to a particular InterSystems IRIS instance.

2. CGI and other dynamically-linked modules:

   - CSPcgi (Runtime module)
   - nph-CSPcgi (Copy of CSPcgi)
   - CSPcgiSys (Systems-Management module)
   - nph-CSPcgiSys (Copy of CSPcgiSys)
   - mod_csp24.so (Apache Version 2.4.x — Apache module as a DSO, if supplied)

In order to avoid disrupting existing Gateway installations on upgrading InterSystems IRIS, the installation procedures place these modules in the following common location. This location is not related to a particular InterSystems IRIS instance.

```
/usr/cspgateway/bin
```

The original location (*install-dir*/csp/bin) is used to hold the Web Gateway components required for serving the Management Portal for the specific instance of InterSystems IRIS.

The modules with Sys appended access the Web Gateway management pages. The runtime modules (that is, those without Sys) have no access to the Web Gateway management pages.

3. The default location for the HyperEvents components:

   - CSPBroker.js
   - CSPxmlhttp.js

   and miscellaneous static resources (such as image files) are required by the CSP Samples and the Management Portal is:

   install-dir\csp\broker

C.1.1 Requirements for using Apache API Modules (Recommended Option and Alternative Option 1)

Before following instructions for either the recommended option or atypical option 1 (Alternative Option 1: Apache API Module with NSD (mod_csp24.so)), check that your build of Apache includes the built-in module for managing shared objects (mod_so). To perform this check, run the following command which lists the modules currently available within Apache:

```
httpd -l
```

The shared object module (mod_so) should appear in the list of modules displayed. The following shows a typical module listing (with mod_so included):

```
Compiled in modules:
  core.c
  mod_access.c
  mod_auth.c
  mod_include.c
  mod_log_config.c
  mod_env.c
  mod_setenvif.c
  prefork.c
  http_core.c
  mod_mime.c
  mod_status.c
  mod_autoindex.c
  mod_asis.c
  mod_cgi.c
  mod_negotiation.c
  mod_dir.c
  mod_imap.c
  mod_actions.c
  mod_userdir.c
  mod_alias.c
  mod_so.c
```

If mod_so is not included in the list for your Apache installation, see your Apache documentation and follow the procedure for rebuilding Apache to include this module.

C.2 Alternative Option 1: Apache API Module with NSD (mod_csp24.so)

If the CSP module is supplied with your distribution as a pre-built shared object (mod_csp24.so), then start at Runtime Configuration. To build the shared object from the supplied source file mod_csp.c choose Method 1 or Method 2 below. Method 1 is preferred.

Be sure to read the following instructions regarding the creation of shared objects in conjunction with the specific documentation contained within your Apache distribution. Note that the instructions given here assume that the root directory for the Apache installation is /usr/apache. In practice, this directory name usually has the Apache version number appended to it.

C.2.1 Method 1: Building the CSP Module as Shared Object with apxs (APache eXtenSion) Tool

The following command builds and installs the shared library, mod_csp24.so, in the Apache /modules directory using the Apache extension tool, **apxs**. It also adds a directive to load the module to the Apache configuration file /conf/httpd.conf.

```
apxs –c –o mod_csp24.so mod_csp.c
```

Copy the shared object produced (mod_csp24.so) into the following directory: /opt/webgateway/bin.

C.2.2 Method 2: Building the CSP Module as Shared Object Manually

Perform the following steps to manually build the CSP module as a shared object:

1. Install the module source file mod_csp.c in the following directory: /usr/apache/src/modules/extra
2. Return to the /usr/apache/src directory and edit the Configuration file. Near the end of this file, locate the following line:

   ```
   # AddModule modules/example/mod_example.o
   ```

   After this line, add the following line:

   ```
   ShareModule modules/extra/mod_csp24.so
   ```

3. Configure the build process using the following command:

   ```
   ./Configure
   ```

4. Build the shared object using the following command:

   ```
   make
   ```

   To produce shared object mod_csp24.so in /usr/apache/src/modules/extra

Note: For further information about the `apxs` tool, see the Apache documentation at https://httpd.apache.org/docs/2.4/programs/apxs.html.

C.2.3 Runtime Configuration

Edit the Apache configuration file httpd.conf. For the standard Apache distribution, this file is in:

```
/usr/apache/conf
```

For Red Hat Linux, the runtime version of httpd.conf is in:

```
/etc/httpd/conf
```

Assuming that you wish to invoke the CSP engine for requested files that contain a .csp, .cls, or .zen extension, add the following section to the end of httpd.conf.

```
LoadModule csp_module /opt/webgateway/bin/mod_csp24.so
CSPFileTypes csp cls zen cxw
Alias /csp/ /opt/webgateway/csp/
<Directory "/opt/webgateway/csp">
        AllowOverride None
        Options MultiViews FollowSymLinks ExecCGI
        Require all granted
        <FilesMatch "\.(log|ini|pid|exe)$">
                Require all denied
        </FilesMatch>
        <Files CSPnsd>
                Require all denied
        </Files>
</Directory>
ScriptAlias /csp-bin/ "/opt/webgateway/bin/"
ScriptAliasMatch /csp/bin/Systems/Module.cxw "/opt/webgateway/bin/nph-CSPcgiSys"
ScriptAliasMatch /csp/bin/RunTime/Module.cxw "/opt/webgateway/bin/nph-CSPcgi"
<Directory "/opt/webgateway/bin/">
        AllowOverride None
        Options None
        Require all granted
</Directory>
```

Restart Apache after making changes to httpd.conf.

C.2.3.1 Controlling Connection Pooling

The size of the connection pool can be controlled by the following Apache configuration parameter (specified in http.conf):

```
CSPMaxPooledNSDConnections <no>
```

In the absence of this parameter, a default value of 32 is used internally – which is effectively:

```
CSPMaxPooledNSDConnections 32
```

To switch-off connection pooling, set this parameter to zero:

```
CSPMaxPooledNSDConnections 0
```

If, for any reason, it becomes necessary to use the legacy (asymmetric) mode of operation (whereby the Web Gateway notifies the end of response transmission by closing the connection on its side), set this parameter to minus 1:

```
CSPMaxPooledNSDConnections -1
```

C.2.3.2 Operating and Managing the Web Gateway with Apache API and NSD

This connectivity option depends on the Web Gateway's network service daemon (NSD).

1. Start the CSP NSD as described in Operating the NSD.
2. Restart Apache after making changes to its configuration (httpd.conf).

The order in which Apache and the NSD are started is unimportant.

3. To access the Web Gateway management pages, enter the following URL in your browser.

   ```
   http://<hostname>:<port_no>/csp-bin/nph-CSPcgiSys
   ```

   If you see an `Unauthorized User` error message, see Enabling Access from Additional Client Addresses.

C.3 Alternative Option 2: CGI Modules with NSD (nph-CSPcgi)

The web server should be configured such that it recognizes InterSystems file types and passes them to the Web Gateway for processing.

The web server configuration file (httpd.conf) is found in the following directory:

```
/usr/apache/conf
```

For Red Hat Linux, the runtime version of httpd.conf is found in:

```
/etc/httpd/conf
```

Add the following section to the end of httpd.conf:

```
<LocationMatch "/*\.([Cc][Ss][Pp]|[Cc][Ll][Ss]|[Zz][En][Nn])$">
    AllowOverride None
    Options FollowSymLinks ExecCGI
    Require all granted
</LocationMatch>
ScriptAliasMatch /csp/bin/Systems/Module.cxw "/opt/webgateway/bin/nph-CSPcgiSys"
ScriptAliasMatch /csp/bin/RunTime/Module.cxw "/opt/webgateway/bin/nph-CSPcgi"
ScriptAliasMatch /*\.([Cc][Ss][Pp]|[Cc][Ll][Ss])$ "/opt/webgateway/bin/nph-CSPcgi"
Alias /csp/ instance-installation-directory
<Directory "instance-installation-directory">
    AllowOverride None
    Options MultiViews FollowSymLinks ExecCGI
   Require all granted
    <FilesMatch "\.(log|ini|pid|exe)$">
       Require all denied
    </FilesMatch>
    <Files CSPnsd>
       Require all denied
    </Files>
</Directory>
ScriptAlias /csp-bin/ "/opt/webgateway/bin/"
<Directory "/opt/webgateway/bin/">
    AllowOverride None
    Options None
    Require all granted
</Directory>
```

The above configuration block relies on the Regular Expressions (`regex`) processor being available to the Apache environment. Sometimes this is not the case and CSP files are consequently not served (`File not found` errors are returned). To remedy this situation, you can associate the (virtual) root location of your web applications with the CGI module instead of making the association through the CSP file extensions. For example, your web applications are contained in /csp. To associate the CSP CGI module with files under /csp, replace the following configuration block:

```
<LocationMatch "/*\.([Cc][Ss][Pp]|[Cc][Ll][Ss]|[Zz][En][Nn])$">
    AllowOverride None
    Options FollowSymLinks ExecCGI
    Require all granted
</LocationMatch>
ScriptAliasMatch /*\.([Cc][Ss][Pp]|[Cc][Ll][Ss])$ "/opt/webgateway/bin/nph-CSPcgi"
```

with:

```
<Location "/csp">
    AllowOverride None
    Options FollowSymLinks ExecCGI
   Require all granted
</Location>
ScriptAlias /csp "/opt/webgateway/bin/nph-CSPcgi"
```

These directives work for URLs of the following form, using the *<baseURL>* for your instance:

```
http://<baseURL>/csp/*.csp
```

Duplicate the configuration block for other root Locations. For example, repeat the process for /myapps for URLs of the form:

```
http://<baseURL>/myapps/*.csp
```

Another approach to avoiding the `regex` issue is to use an `Action` directive in conjunction with a CSP MIME type. However, it should be noted that `Action` is essentially a content filtering technique and, as such, requires that your CSP files are physically present on the web server host even if the InterSystems IRIS server is installed on a separate computer. If you wish to use this approach, first add a new MIME type to the end of the Apache mime.types file and associate it with file types representing CSP content. The mime.types file are found in the same directory as the httpd.conf file.

```
text/csp                csp cls
```

Now, add the `Action` directive to the end of the CGI configuration block in httpd.conf such that it reads:

```
Alias /csp/ /opt/webgateway/csp/
<Directory "/opt/webgateway/csp">
    AllowOverride None
    Options MultiViews FollowSymLinks ExecCGI
    Require all granted
<Files CSPnsd>
       Require all denied
    </Files>
<Files CSP.ini>
       Require all denied
    </Files>
<Files CSP.log>
       Require all denied
    </Files>
<Files CSPnsd.ini>
       Require all denied
    </Files>
<Files CSPnsd.pid>
       Require all denied
    </Files>
</Directory>
ScriptAlias /csp-bin/ "/opt/webgateway/bin/"
<Directory "/opt/webgateway/bin/">
    AllowOverride None
    Options None
    Require all granted
</Directory>
Action text/csp “/csp-bin/nph-CSPcgi”
```

Restart Apache after making changes to httpd.conf.

Finally, note that because CGI is an open standard, The CSP CGI modules work with any web server.

C.3.1 Operating and Managing the Web Gateway with CGI and NSD

This connectivity option depends on the Web Gateway’s network service daemon (NSD).

1. Start the CSP NSD as described in Operating the NSD
2. Restart Apache after making changes to its configuration (httpd.conf).

The order in which Apache and the NSD are started is unimportant.

3. To access the Web Gateway management pages, enter one of the following URLs in your browser.

```
http://<hostname>:<port_no>/csp/bin/Systems/Module.cxw
http://<hostname>:<port_no>/csp-bin/nph-CSPcgiSys
```

If you see an `Unauthorized User` error message, see Enabling Access from Additional Client Addresses.

C.4 Alternative Option 3: Built-in Apache API Module with NSD (mod_csp.c)

Before embarking on setting up this more complicated option you should bear in mind that, for most modern UNIX® systems, the performance advantage in static linking over linking the module at runtime as a shared object (option 1) is minimal (if anything at all).

Be sure to read these instructions in conjunction with the specific documentation contained within your Apache distribution.

C.4.1 Build Apache to Include CSP Module Source Code

Refer to the Apache documentation for this step.

http://httpd.apache.org/

C.4.2 Check the Apache Binary Produced

Run the following command to check that the CSP module has been successfully included in the Apache core (this command lists all modules currently built-into Apache):

```
./httpd -l
```

For example:

```
Compiled in modules:
        core.c
        mod_access.c
        mod_auth.c
        mod_include.c
        mod_log_config.c
        mod_env.c
        mod_setenvif.c
        prefork.c
        http_core.c
        mod_mime.c
        mod_status.c
        mod_autoindex.c
        mod_asis.c
        mod_cgi.c
        mod_negotiation.c
        mod_dir.c
        mod_imap.c
        mod_actions.c
        mod_userdir.c
        mod_alias.c
        mod_csp.c
```

C.4.3 Runtime Configuration

Edit the Apache configuration file httpd.conf. For the standard Apache distribution this file is in:

```
/usr/apache/conf
```

For Red Hat Linux, the runtime version of httpd.conf is in:

```
/etc/httpd/conf
```

Assuming that you wish to invoke the CSP engine for requested files that contain a .csp, .cls, or .zen extension, add the following section to the end of httpd.conf:

```
CSPFileTypes csp cls zen cxw
ScriptAliasMatch /csp/bin/Systems/Module.cxw "/opt/webgateway/bin/nph-CSPcgiSys"
ScriptAliasMatch /csp/bin/RunTime/Module.cxw "/opt/webgateway/bin/nph-CSPcgi"
Alias /csp/ /opt/webgateway/csp/
<Directory "/opt/webgateway/csp">
        AllowOverride None
        Options MultiViews FollowSymLinks ExecCGI
        Require all granted
        <FilesMatch "\.(log|ini|pid|exe)$">
                Require all denied
        </FilesMatch>
        <Files CSPnsd>
                Require all denied
        </Files>
</Directory>
ScriptAlias /csp-bin/ "/opt/webgateway/bin/"
<Directory "/opt/webgateway/bin/">
        AllowOverride None
        Options None
        Require all granted
</Directory>
```

Note that all requests to Apache are serviced by a set of modules invoked in a predefined sequence. The CSP module is one of the first modules invoked, provided its definition was added near the end of the Configuration file as suggested.

Restart Apache after making changes to httpd.conf.

C.4.4 Operating and Managing the Web Gateway with Apache API and NSD

This connectivity option depends on the Web Gateway's network service daemon (NSD).

1. Start the CSP NSD as described in Operating the NSD.
2. Restart Apache after making changes to its configuration (httpd.conf).

   The order in which Apache and the NSD are started is unimportant.
3. To access the Web Gateway management pages, point your browser at one of the following locations.

   ```
   http://<hostname>:<port_no>/csp/bin/Systems/Module.cxw
   http://<hostname>:<port_no>/csp-bin/nph-CSPcgiSys
   ```

   If you see an `Unauthorized User` error message, see Enabling Access from Additional Client Addresses.

D

Add the Web Gateway to a Locked-Down Apache Installation (UNIX®/Linux/macOS)

In some cases—most notably, Security Enhanced Linux (SELinux) on RHEL systems—the Apache web server is locked-down; it cannot access files outside of the Apache file system without additional configuration. Without this additional configuration, attempted Web Gateway connections result in an `HTTP 403 Forbidden` error.

If the installer automatically configured a web server connection for your InterSystems IRIS instance or your standalone Web Gateway, then the installer has already performed the additional configuration steps necessary to enable Web Gateway connections on SELinux systems. This page provides the instructions necessary to perform this additional configuration manually.

Two solutions are available:

- Modify the security context for the Web Gateway's home directory so that Apache can access files held in this location. This is often the most straightforward solution.
- Move the Web Gateway's home directory to a location under the Apache root file system (which is pre-configured to be accessible to Apache in the SELinux setup).

D.1 Modify the Security Context for the Web Gateway Files

First, modifying the SELinux security context for the Web Gateway's home directory involves the following steps.

We use, as an example, a Web Gateway home directory of /opt/webgateway/bin, the InterSystems IRIS Superserver listening on port 1972 and InterSystems IRIS installed in /usr/iris/.

The *chcon* command sets file context and takes effect immediately.

```
sudo chcon -R -t httpd_sys_content_t /usr/iris/csp
sudo chcon -R -t httpd_sys_rw_content_t /opt/webgateway/conf/CSP.ini
sudo chcon -R -t httpd_sys_rw_content_t /opt/webgateway/conf/CSPRT.ini
sudo chcon -R -t httpd_sys_rw_content_t /opt/webgateway/logs/CSP.log
sudo chcon -t httpd_modules_t /opt/webgateway/bin/CSPa2.so
sudo chcon -t httpd_modules_t /opt/webgateway/bin/CSPa2Sys.so
sudo chcon -t httpd_modules_t /opt/webgateway/bin/CSPa22.so
sudo chcon -t httpd_modules_t /opt/webgateway/bin/CSPa22Sys.so
sudo chcon -t httpd_modules_t /opt/webgateway/bin/CSPa24.so
sudo chcon -t httpd_modules_t /opt/webgateway/bin/CSPa24Sys.so
```

However, changes made by the chcon command are lost after the next relabeling; therefore it is necessary use the semanage fcontext facility in addition to chcon. The following commands allow the Web Gateway to run when SELinux is enforced:

```
semanage fcontext -a -t lib_t "/opt/webgateway/bin/(.*\.so)?" 2> /dev/null
semanage fcontext -a -t httpd_sys_rw_content_t "/opt/webgateway/bin/temp(/.*)?" 2> /dev/null
semanage fcontext -a -t httpd_sys_rw_content_t "/opt/webgateway/bin/CSP(.*\.ini)?" 2> /dev/null
semanage fcontext -a -t httpd_sys_rw_content_t "/opt/webgateway/bin/CSP.log" 2> /dev/null
restorecon -vr /opt/webgateway/bin > /dev/null
```

Then use the commands shown below. Note that it is extremely important to properly set the context of the Superserver port (as shown in the last line); otherwise, the Web Gateway will not be able to access it, resulting in "Server unavailable" errors.

```
sudo /usr/sbin/semanage fcontext -a -t httpd_sys_content_t
    "/usr/iris/csp(/.)?"
sudo /usr/sbin/semanage fcontext -a -t httpd_sys_rw_content_t
    "/opt/webgateway/conf/CSP.ini"
sudo /usr/sbin/semanage fcontext -a -t httpd_sys_rw_content_t
    "/opt/webgateway/conf/CSPRT.ini"
sudo /usr/sbin/semanage fcontext -a -t httpd_sys_rw_content_t
    "/opt/webgateway/logs/CSP.log"
sudo /usr/sbin/semanage fcontext -a -t httpd_modules_t
    "/opt/webgateway/bin/CSPa2.so"
sudo /usr/sbin/semanage fcontext -a -t httpd_modules_t
    "/opt/webgateway/bin/CSPa2Sys.so"
sudo /usr/sbin/semanage fcontext -a -t httpd_modules_t
    "/opt/webgateway/bin/CSPa22.so"
sudo /usr/sbin/semanage fcontext -a -t httpd_modules_t
   "/opt/webgateway/bin/CSPa22Sys.so"
sudo /usr/sbin/semanage fcontext -a -t httpd_modules_t
    "/opt/webgateway/bin/CSPa24.so"
sudo /usr/sbin/semanage fcontext -a -t httpd_modules_t
    "/opt/webgateway/bin/CSPa24Sys.so"
sudo /usr/sbin/semanage port -a -t http_port_t -p tcp 51773
```

These are the basic steps for granting the Web Gateway (operating in the context of the hosting Apache server) access to files in its home directory.

D.2 Move the Web Gateway Directory

An alternative approach (and the one that should be used if the method suggested above is not acceptable) is to configure the Web Gateway to work within the pre-configured directories provided by Apache. The following commands assume that Apache is installed in /usr/apache.

- CGI modules should be copied to: /usr/apache/cgi-bin/

  ```
  cp /usr/iris/csp/bin/*cgi* /usr/apache/cgi-bin/
  ```

- API modules should be copied to: /usr/apache/modules/

  ```
  cp /usr/iris/csp/bin/*.so /usr/apache/modules/
  ```

- Static files should be copied to locations under: /usr/apache/htdocs/

  ```
  cp /usr/iris/csp/samples/* /usr/apache/htdocs/csp/samples/
  cp /usr/iris/csp/broker/* /usr/apache/htdocs/csp/broker/
  cp /usr/iris/csp/sys/* /usr/apache/htdocs/csp/sys/
  ```

  Also, copy any sub-directories held under the above locations.

Having moved the Web Gateway installation, the appropriate changes to the paths specified in the Apache configuration must be made.

The sections below provide examples Apache configuration directives to configure the Web Gateway for each deployment option. Note that these examples configure the Apache server to invoke the Web Gateway globally, but only for InterSystems file types; your configuration directives should invoke the Web Gateway to serve requests according to your organization's needs.

D.2.1 Recommended Option: Apache API Modules (CSPa24.so)

```
LoadModule cspsys_module_sa /usr/apache/modules/CSPap24.so
CSPSYSModulePath /usr/apache/modules/
CSPFileTypes csp cls zen cxw
```

D.2.2 Alternative Option 1: Apache API Module with NSD (mod_csp.so)

```
LoadModule csp_module /usr/apache/modules/mod_csp.so
CSPFileTypes csp cls zen cxw
ScriptAliasMatch /csp/bin/Systems/Module.cxw "/usr/apache/cgi-bin/nph-CSPcgiSys"
ScriptAliasMatch /csp/bin/RunTime/Module.cxw "/usr/apache/cgi-bin/nph-CSPcgi"
```

D.2.3 Alternative Option 2: CGI Modules with NSD (nph-CSPcgi)

```
<LocationMatch "/*\.([Cc][Ss][Pp]|[Cc][Ll][Ss]|[Zz][En][Nn])$">
AllowOverride None
Options FollowSymLinks ExecCGI
Require all granted
</LocationMatch>
ScriptAliasMatch /csp/bin/Systems/Module.cxw "/usr/apache/cgi-bin/nph-CSPcgiSys"
ScriptAliasMatch /csp/bin/RunTime/Module.cxw "/usr/apache/cgi-bin/nph-CSPcgi"
ScriptAliasMatch /*\.([Cc][Ss][Pp]|[Cc][Ll][Ss])$ "/usr/apache/cgi-bin/nph-CSPcgi"
```

D.2.4 Alternative Option 3: Built-in Apache API Module with NSD (mod_csp.c)

```
CSPFileTypes csp cls zen cxw
ScriptAliasMatch /csp/bin/Systems/Module.cxw "/apache/cgi-bin/nph-CSPcgiSys"
ScriptAliasMatch /csp/bin/RunTime/Module.cxw "/usr/apache/cgi-bin/nph-CSPcgi"
```

E

Alternative Options for IIS 7 or Later (Windows)

This page contains instructions for configuring atypical options for configuring Microsoft IIS for use with the InterSystems Web Gateway. For these options:

1. Perform the following steps, as described in the recommended procedure:
   a. Set permissions for the Web Gateway components
   b. Configure the web application paths for your applications
   c. Enable URLs which contain .
2. Install ISAPI and GCI services, as described in Installing the ISAPI and CGI Services.
3. Use the instructions in *one* of the following section:
   - Alternative Option 1: Using the ISAPI Modules (CSPms*.dll)
   - Alternative Option 2: Using a Native Module with the NSD (CSPcms.dll)
   - Alternative Option 3: Using an ISAPI Module with the NSD (CSPcms.dll)
   - Alternative Option 4: Using the CGI Modules with the NSD (nph-CSPcgi*.exe)

E.1 Installing the ISAPI and CGI Services

IIS 7 does not, by default, run **ISAPI extension**, **ISAPI filters**, or **CGI modules**. For all the atypical options for IIS 7, you must install these services.

Note that, with the **ISAPI extensions** service installed, all versions of the Web Gateway work with IIS 7.

Install these legacy services through the Windows Control Panel.

1. Open the Windows Control Panel.
2. Select **Programs and Features** and select **Turn Windows Features on or off**.
3. Navigate to **Internet Information Services** and expand **World Wide Web Services** and **Application Development Features**.

   Select **ISAPI Extensions**. Also select **ISAPI Filters** and **CGI**, if these additional services are required. Select **OK**.

4. In the Windows **Control Panel**, open **Administrative Tools** and **Internet Information Services (IIS) Manager**.
5. In the left panel, highlight **[MACHINE_NAME] ([machine_name]\[user_name])**
6. In the middle panel, double-click the **Modules** icon.
7. In the right panel, select **Add Native Module**.
8. In the left panel, expand the top level, expand **Web Sites** and expand **Default Web Site**

```
[MACHINE_NAME] ([machine_name]\[user_name])
              Web Sites
                     Default Web Site
```

9. In the middle panel, double-click **Handler Mappings**.
10. In the middle panel, highlight the **ISAPI-dll** handler.
11. In the right panel, select **Edit Handler Permissions**.
12. Select **Execute** and select **OK**. This allows ISAPI extensions to be invoked through direct calls to the name of the ISAPI DLL.

E.2 Alternative Option 1: Using the ISAPI Modules (CSPms*.dll)

Use this option if your Web Gateway DLLs are unable to support the Native Module interface (the Recommended Option). This is the default (and best performing) solution that was supplied for earlier versions of IIS.

IIS 7 does not, by default, run **ISAPI extensions**, **ISAPI filters** or **CGI modules**. This option requires the **ISAPI extensions** service.

Follow the instructions in Installing the ISAPI and CGI Services (If Required) for installing and configuring the ISAPI extensions service.

The web server should be configured such that it recognizes InterSystems file types and passes them to the Web Gateway for processing.

E.2.1 Enabling the ISAPI Extensions

DLLs: CSPms.dll and CSPmsSys.dll

Before these extensions can be used they must be registered with IIS as being “Allowed” applications. This is done in the **Internet Information Services (IIS) Manager** control panel.

1. Open the **Internet Information Services (IIS) Manager** window.
2. In the left panel, highlight **[MACHINE_NAME] ([machine_name]\[user_name])**.
3. In the middle panel, double-click **ISAPI and CGI Restrictions**.
4. In the right panel, select **Add**.
5. In the **Add ISAPI or CGI Restriction** dialog, enter the following details:

   ISAPI or CGI Path: C:\Inetpub\CSPGateway\CSPms.dll

   Description: WebGatewayRunTime

Allow extension path to execute: Select

Select **OK**

E.2.2 Mapping InterSystems IRIS File Extensions

Choose *one* of the following configuration methods:

- Serve all content (including static content) from InterSystems IRIS. Map * to the Web Gateway. See Mapping Additional File Types.
- Serve static content from the web server. Map *only* the InterSystems file types to the Web Gateway.

If you are serving static files from the web server, map the InterSystems file types to the Web Gateway ISAPI extensions as follows:

Extension	Binary
*.csp	C:\Inetpub\CSPGateway\CSPms.dll
*.cls	C:\Inetpub\CSPGateway\CSPms.dll
*.zen	C:\Inetpub\CSPGateway\CSPms.dll
*.cxw	C:\Inetpub\CSPGateway\CSPms.dll

1. Open the **Internet Information Services (IIS) Manager** window.
2. In the left panel expand the top level to reveal the **Web Sites** section, then the **Default Web Site** section. Highlight the **Default Web Site** section:

   ```
   [MACHINE_NAME] ([machine_name]\[user_name])
              Web Sites
                      Default Web Site
   ```

   Note: This activates CSP for the whole web site. To restrict the use of CSP to specific virtual sub-directories (such as /csp/) focus control on the appropriate subdirectory (under **Default Web Site**) before creating the mappings. Repeat the process for each virtual subdirectory from which CSP content is to be served.

3. In the middle panel, double-click the **Handler Mappings** icon.
4. In the right panel, select **Add Script Map**.
5. In the **Add Script Map** dialog, enter:

   Request Path: *.csp

   Executable: C:\Inetpub\CSPGateway\CSPms.dll

   Name: WebGateway_csp

6. Select **Request Restrictions**.

   Clear: **Invoke handler only if request is mapped to**

   Select **OK** to return to **Add Script Map** dialog.

   Select **OK**.

7. At this point you may be prompted as follows:

   “Would you like to enable this ISAPI extension? If yes, we add your extension as an “Allowed” entry in the ISAPI and CGI Restrictions list. If the extension already exists we allow it.”

Select **Yes**.

You can later find the list of allowed applications as follows:

In the left panel, highlight:

[MACHINE_NAME] ([machine_name]\[user_name])

In the middle panel, double-click **ISAPI and CGI Restrictions**.

If the Web Gateway ISAPI components are not included in the list of allowed applications, add them.

You can add text of your own choice in the **Description** field. For example:

```
WebGatewayManagement for CSPmsSys.dll
```

```
WebGatewayRunTime for CSPms.dll
```

8. Repeat the above process: Use the **Add Script Map** dialog to enter the following two mappings:

   Request Path: *.cls

   Executable: C:\Inetpub\CSPGateway\CSPms.dll

   Name: WebGateway_cls

   Request Path: *.zen

   Executable: C:\Inetpub\CSPGateway\CSPms.dll

   Name: WebGateway_zen

   Request Path: *.cxw

   Executable: C:\Inetpub\CSPGateway\CSPms.dll

   Name: WebGatewayManagement

E.2.3 Operating and Managing the Web Gateway

To access the Web Gateway's systems management suite, point your browser at one of the following locations:

http://<hostname>:<port>/csp/bin/Systems/Module.cxw

http://<hostname>:<port>/csp/bin/CSPmsSys.dll

If you see an unauthorized user error message, see Enabling Access from Additional Client Addresses and Web Gateway and Security.

E.3 Alternative Option 2: Using a Native Module with the NSD (CSPcms.dll)

IIS 7 does not, by default, run **ISAPI extensions**, **ISAPI filters**, or **CGI modules**. This option requires the **CGI modules** service for running the Web Gateway Management module (nph-CSPcgiSys.exe).

Follow the instructions in Installing the ISAPI and CGI Services (If Required.

Configure the web server so that it recognizes InterSystems file types and passes them to the Web Gateway for processing.

E.3.1 Registering the Runtime Native Module

DLL: CSPcms.dll

Before this module can be used, it must be registered with IIS. This is done in the **Internet Information Services (IIS) Manager** control panel.

1. Open the **Internet Information Services (IIS) Manager** window.
2. In the left panel, highlight:

   [MACHINE_NAME] ([machine_name]\[user_name])
3. In the middle panel, double-click the **Modules** icon.
4. In the right panel, select **Add Native Module**.
5. Select **Register** and enter the following details in the **Register Native Module** dialog:

   Name: CSPcms

   Path: C:\Inetpub\CSPGateway\CSPcms.dll

   Select **OK**.
6. In the left panel, expand the top level to reveal the **Web Sites** section, then the **Default Web Site** section. Highlight the **Default Web Site** section:

```
[MACHINE_NAME] ([machine_name]\[user_name])
            Web Sites
                    Default Web Site
```

7. In the right panel, select **Add Native Module**.
8. In the **Add Native Module** dialog select **CSPcms** then select **OK**.

E.3.2 Enabling the CGI module for Web Gateway Management

Executable: nph-CSPcgiSys.exe

Before this module can be used, it must be registered with IIS as being an Allowed application. This is done in the **Internet Information Services (IIS) Manager** control panel.

1. Open the **Internet Information Services (IIS) Manager**.
2. In the left panel, highlight:

   [MACHINE_NAME] ([machine_name]\[user_name])
3. In the middle panel, double-click the **ISAPI and CGI Restrictions** icon.
4. In the right panel, select **Add**.
5. In the **Add ISAPI or CGI Restriction** dialog, enter:

   ISAPI or CGI Path: C:\Inetpub\CSPGateway\nph-CSPcgiSys.exe

   Description: WebGatewayManagement

   Allow extension path to execute: Select

   Select **OK**.

E.3.3 Mapping InterSystems IRIS File Extensions

Note: Do not use **Add Wildcard Script Mapping** utility for this file extension mapping process; it gives an error. Instead, use the utility called **Add Module Mapping for *.**

Choose *one* of the following configuration methods:

- Serve all content (including static content) from InterSystems IRIS. Map * to the Web Gateway. If you are configuring the web application so that the InterSystems IRIS server serves all static files, then see Mapping Additional File Types.
- Serve static content from the web server.

  Map *only* InterSystems file types to the Web Gateway.

If you are serving static files from the web server, map the InterSystems file types to the Web Gateway modules as follows:

Extension	Native Module	Binary
*.csp	CSPms	C:\Inetpub\CSPGateway\CSPms.dll
*.cls	CSPms	C:\Inetpub\CSPGateway\CSPms.dll
*.zen	CSPms	C:\Inetpub\CSPGateway\CSPms.dll
*.cxw		C:\Inetpub\CSPGateway\nph-CSPcgiSys.exe

1. Open the **Internet Information Services (IIS) Manager** window.
2. In the left panel, expand the top level to reveal the **Web Sites** section, then the **Default Web Site** section. Highlight the **Default Web Site** section:

   ```
   [MACHINE_NAME] ([machine_name]\[user_name])
               Web Sites
                       Default Web Site
   ```

   Note: This activates CSP for the whole web site. To restrict the use of CSP to specific virtual sub-directories (such as /csp/) focus control on the appropriate subdirectory (under **Default Web Site**) before creating the mappings. Repeat the process for each virtual subdirectory from which CSP content is to be served.

3. In the middle panel, double-click the **Handler Mappings** icon.
4. In the right panel, select **Add Module Mapping**.
5. In the **Add Module Mappings** dialog, enter:

   Request Path: *.csp

   Module: Select CSPcms

   Name: WebGateway_csp

6. Select **Request Restrictions**.

   Clear: **Invoke handler only if request is mapped to**

   Select **OK** to return to the **Add Module Mappings** dialog.

   Select **OK**.

7. Repeat the above process to add the following Module Mappings:

   Request Path: *.cls

   Module: Select **CSPcms**

Name: WebGateway_cls

and

Request Path: *.zen

Module: Select **CSPcms**

Name: WebGateway_zen

8. In the left panel, highlight the **Default Web Site** section:

```
[MACHINE_NAME] ([machine_name]\[user_name])
            Web Sites
                    Default Web Site
```

9. In the middle panel, double-click the **Handler Mappings** icon.
10. In the right panel, select **Add Script Map**.
11. In the **Add Script Map** dialog, enter:

    Request Path: *.cxw

    Executable: C:\Inetpub\CSPGateway\nph-CSPcgiSys.exe

    Name: WebGatewayManagement

12. Select **Request Restrictions**.

    Clear: **Invoke handler only if request is mapped to**

    Select **OK** to return to the **Add Script Map** dialog.

    Select **OK**.

13. You may be prompted as follows: “Would you like to enable this ISAPI extension? If yes, we add your extension as an “Allowed” entry in the ISAPI and CGI Restrictions list. If the extension already exists we allow it.”

    Select **Yes**.

    You can later find the list of allowed applications as follows:

    In the left panel, highlight:

    [MACHINE_NAME] ([machine_name]\[user_name])

    In the center panel, double-click the **ISAPI and CGI Restrictions** icon.

    If the Web Gateway Management CGI module is not included in the list of allowed applications, add it.

    You can add text of your own choice in the **Description** field. For example:

    `WebGatewayManagement for nph-CSPcgiSys.exe`

E.3.4 Operating and Managing the Web Gateway

This connectivity option depends on the Web Gateway’s network service daemon (NSD).

Start the CSP NSD as described in the section, *Starting the NSD*.

To access the Web Gateway’s Systems Management suite, point your browser at one of the following locations:

http://<hostname>:<port>/csp/bin/Systems/Module.cxw

http://<hostname>:<port>/csp-bin/nph-CSPcgiSys

If you see an unauthorized user error message, see Enabling Access from Additional Client Addresses and Web Gateway and Security.

E.4 Alternative Option 3: Using an ISAPI Module with the NSD (CSPcms.dll)

Use this option if your Web Gateway DLLs are unable to support the Native Module interface (Alternative Option 2).

IIS 7 does not, by default, run **ISAPI extensions**, **ISAPI filters** or **CGI modules**. This option requires both the **ISAPI extensions** and the **CGI modules** service.

Follow the instructions in Installing the ISAPI and CGI Services.

The web server should be configured such that it recognizes InterSystems file types and passes them to the Web Gateway for processing.

E.4.1 Enabling the Runtime ISAPI Extension

DLLs: CSPcms.dll

Before this extension can be used, it must be registered with IIS as being "Allowed" applications. This is done in the **Internet Information Services (IIS) Manager** control panel.

1. Open the **Internet Information Services (IIS) Manager** window.
2. In the left panel, highlight: **[MACHINE_NAME] ([machine_name]\[user_name])**
3. In the middle panel, double-click the **ISAPI and CGI Restrictions** icon.
4. In the right panel, select **Add**.
5. In the **Add ISAPI or CGI Restriction** dialog, enter:

   ISAPI or CGI Path: C:\Inetpub\CSPGateway\CSPcms.dll

   Description: WebGatewayRunTime

   Allow extension path to execute: Select

   Select **OK**

E.4.2 Enabling the CGI module for Web Gateway Management

Executable: nph-CSPcgiSys.exe

Before this module can be used, it must be registered with IIS as being an Allowed application. This is done in the **Internet Information Services (IIS) Manager** control panel.

1. Open the **Internet Information Services (IIS) Manager** window.
2. In the left panel, highlight: **[MACHINE_NAME] ([machine_name]\[user_name])**
3. In the middle panel, double-click the **ISAPI and CGI Restrictions** icon.
4. In the right panel, select **Add**.
5. In the **Add ISAPI or CGI Restriction** dialog, enter:

ISAPI or CGI Path: C:\Inetpub\CSPGateway\nph-CSPcgiSys.exe

Description: WebGatewayManagement

Allow extension path to execute: Select

Select **OK**.

E.4.3 Mapping InterSystems IRIS File Extensions

Choose *one* of the following configuration methods:

- Serve all content (including static content) from InterSystems IRIS. Map * to the Web Gateway. If you are configuring the web application in InterSystems IRIS so that the InterSystems IRIS server serves all static files, see Mapping Additional File Types.
- Serve static content from the web server.

  Map *only* InterSystems file types to the Web Gateway.

If you are serving static files from the web server, map the InterSystems file types to the Web Gateway Modules as follows:

Extension	Binary
*.csp	C:\Inetpub\CSPGateway\CSPcms.dll
*.cls	C:\Inetpub\CSPGateway\CSPcms.dll
*.zen	C:\Inetpub\CSPGateway\CSPcms.dll
*.cxw	C:\Inetpub\CSPGateway\nph-CSPcgiSys.exe

1. Open the **Internet Information Services (IIS) Manager** window.
2. In the left panel, expand the top level and expand **Web Sites**. Highlight **Default Web Site**.

   ```
   [MACHINE_NAME] ([machine_name]\[user_name])
              Web Sites
                      Default Web Site
   ```

   Note: This activates CSP for the whole web site. To restrict the use of CSP to specific virtual sub-directories (such as /csp/) focus control on the appropriate subdirectory (under **Default Web Site**) before creating the mappings. Repeat the process for each virtual subdirectory from which CSP content is to be served.

3. In the middle panel, double-click **Handler Mappings**.
4. In the right panel, select **Add Script Map**.
5. In the **Add Script Map** dialog, enter:

   Request Path: *.csp

   Executable: C:\Inetpub\CSPGateway\CSPcms.dll

   Name: WebGateway_csp
6. Select **Request Restrictions**.

   Clear: **Invoke handler only if request is mapped to**

   Select **OK** to return to the ‘Add Script Map’ dialog.

   Select **OK**.

7. At this point you may be prompted as follows:

   "Would you like to enable this ISAPI extension? If yes, we add your extension as an "Allowed" entry in the ISAPI and CGI Restrictions list. If the extension already exists we allow it."

   Select **Yes**.

   You can later find the list of allowed applications as follows:

   In the left panel, highlight:

   [MACHINE_NAME] ([machine_name]\[user_name])

   In the middle panel, double-click the **ISAPI and CGI Restrictions** icon.

   If the Web Gateway ISAPI module is not included in the list of allowed applications, add it.

   You can add text of your own choice in the **Description** field. For example:

   ```
   WebGatewayRunTime for CSPcms.dll
   ```

   ```
   WebGatewayManagement for nph-CSPcgiSys.exe
   ```

8. Repeat the above process: Use the **Add Script Map** dialog to enter the following two mappings:

   Request Path: *.cls

   Executable: C:\Inetpub\CSPGateway\CSPcms.dll

   Name: WebGateway_cls

   Request Path: *.zen

   Executable: C:\Inetpub\CSPGateway\CSPcms.dll

   Name: WebGateway_zen

   Request Path: *.cxw

   Executable: C:\Inetpub\CSPGateway\nph-CSPcgiSys.exe

   Name: WebGatewayManagement

E.4.4 Operating and Managing the Web Gateway

This connectivity option depends on the Web Gateway's network service daemon (NSD).

1. Start the CSP NSD as described in the section dedicated to this service.

To access the Web Gateway's Systems Management suite, point your browser at one of the following locations:

http://<hostname>:<port>/csp/bin/Systems/Module.cxw

http://<hostname>:<port>/csp-bin/nph-CSPcgiSys

If you see an unauthorized user error message, see Enabling Access from Additional Client Addresses and Web Gateway and Security.

E.5 Alternative Option 4: Using the CGI Modules with the NSD (nph-CSPcgi*.exe)

In most cases, the all-inclusive Native Module-based solution (the Recommended Option) is the option of choice, and is the implementation that gives the best performance. The CGI/NSD hybrid is useful for cases where it is necessary, for operational reasons, to manage the Web Gateway independently of the hosting web server. For example, if multiple instances of the web server are to share the same Web Gateway installation. In option 1 each instance of the core web server process binds to its own instance of the Web Gateway.

Another factor in choosing this approach might be that the in-house requirements of your web master (or ISP) dictate that all web server extensions are implemented using the CGI protocol.

IIS 7 does not, by default, run **ISAPI extensions**, **ISAPI filters** or **CGI modules**. This option requires the **CGI modules** service.

Follow the instructions in for installing the CGI service, Installing the ISAPI and CGI Services.

Configure the web server so that it recognizes InterSystems file types and passes them to the Web Gateway for processing.

E.5.1 Enabling the CGI Modules

Executables: nph-CSPcgi.exe and nph-CSPmsSys.exe

Before these modules can be used they must be registered with IIS as being “Allowed” applications. This is done in the **Internet Information Services (IIS) Manager** control panel.

1. Open the **Internet Information Services (IIS) Manager** window.
2. In the left panel, highlight:

   [MACHINE_NAME] ([machine_name]\[user_name])
3. In the middle panel, double-click the **ISAPI and CGI Restrictions** icon.
4. In the right panel, select **Add**.
5. In the **Add ISAPI or CGI Restriction** dialog, enter:

   ISAPI or CGI Path: C:\Inetpub\CSPGateway\nph-CSPcgi.exe

   Description: WebGatewayRunTime

   Allow extension path to execute: Select

   Select **OK**.
6. Repeat the above steps for nph-CSPcgiSys.exe, entering the following details in the **Restrictions** dialog:

   ISAPI or CGI Path: C:\Inetpub\CSPGateway\nph-CSPcgiSys.exe

   Description: WebGatewayManagement

   Allow extension path to execute: Select

E.5.2 Mapping InterSystems IRIS File Extensions

Choose *one* of the following configuration methods:

- Serve all content (including static content) from InterSystems IRIS. Map * to the Web Gateway. If you are configuring the web application in InterSystems IRIS so that the InterSystems IRIS server serves all static files, see Mapping Additional File Types.
- Serve static content from the web server.

  Map *only* InterSystems file types to the Web Gateway.

If you are serving static files from the web server, map the InterSystems file types to the Web Gateway CGI modules as follows:

Extension	Binary
*.csp	C:\Inetpub\CSPGateway\nph-CSPcgi.exe
*.cls	C:\Inetpub\CSPGateway\nph-CSPcgi.exe
*.zen	C:\Inetpub\CSPGateway\nph-CSPcgi.exe
*.cxw	C:\Inetpub\CSPGateway\nph-CSPcgiSys.exe

1. Open the **Internet Information Services (IIS) Manager** window.
2. In the left panel expand the top level to reveal the **Web Sites** section, then the **Default Web Site** section. Highlight the **Default Web Site** section:

   ```
   [MACHINE_NAME] ([machine_name]\[user_name])
              Web Sites
                     Default Web Site
   ```

   Note: This activates CSP for the whole web site. To restrict the use of CSP to specific virtual sub-directories (such as /csp/) focus control on the appropriate subdirectory (under **Default Web Site**) before creating the mappings. Repeat the process for each virtual subdirectory from which CSP content is to be served.

3. In the middle panel, double-click the **Handler Mappings** icon.
4. In the right panel, select **Add Script Map**.
5. In the **Add Script Map** dialog, enter:

   Request Path: *.csp

   Executable: C:\Inetpub\CSPGateway\nph-CSPcgi.exe

   Name:WebGateway_csp
6. Select **Request Restrictions**.

   Clear: **Invoke handler only if request is mapped to**

   Select **OK** to return to the **Add Script Map** dialog.

   Select **OK**.
7. At this point you may be prompted as follows: “Would you like to enable this ISAPI extension? If yes, we add your extension as an “Allowed” entry in the ISAPI and CGI Restrictions list. If the extension already exists we allow it.”

   Select **Yes**.
8. You can later find the list of allowed applications as follows:

   In the left panel, highlight:

   [MACHINE_NAME] ([machine_name]\[user_name])

   In the middle panel, double-click the **ISAPI and CGI Restrictions** icon.

If the Web Gateway CGI components are not included in the list of allowed applications, add them.

You can add text of your own choice in the **Description** field. For example:

```
WebGatewayManagement for nph-CSPcgiSys.exe
WebGatewayRunTime for nph-CSPcgi.exe
```

9. Repeat the above process: Use the **Add Script Map** dialog to enter the following two mappings:

   Request Path: *.cls

   Executable: C:\Inetpub\CSPGateway\nph-CSPcgi.exe

   Name: WebGateway_cls

   Request Path: *.zen

   Executable: C:\Inetpub\CSPGateway\nph-CSPcgi.exe

   Name: WebGateway_zen

   Request Path: *.cxw

   Executable: C:\Inetpub\CSPGateway\nph-CSPcgiSys.exe

   Name: WebGatewayManagement

E.5.3 Operating and Managing the Web Gateway

This connectivity option depends on the Web Gateway's network service daemon (NSD).

1. Start the CSP NSD as described in the section dedicated to this service.

To access the Web Gateway's Systems Management suite, point your browser at one of the following locations:

http://<hostname>:<port>/csp/bin/Systems/Module.cxw

http://<hostname>:<port>/csp-bin/nph-CSPcgiSys

If you see an unauthorized user error message, see Enabling Access from Additional Client Addresses and Web Gateway and Security.