



# Corticon Web Console



# Copyright

---

© 2022 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.

These materials and all Progress<sup>®</sup> software products are copyrighted and all rights are reserved by Progress Software Corporation. The information in these materials is subject to change without notice, and Progress Software Corporation assumes no responsibility for any errors that may appear therein. The references in these materials to specific platforms supported are subject to change.

#1 Load Balancer in Price/Performance, 360 Central, 360 Vision, Chef, Chef (and design), Chef Habitat, Chef Infra, Code Can (and design), Compliance at Velocity, Corticon, Corticon.js, DataDirect (and design), DataDirect Cloud, DataDirect Connect, DataDirect Connect64, DataDirect XML Converters, DataDirect XQuery, DataRPM, Defrag This, Deliver More Than Expected, DevReach (and design), Driving Network Visibility, Flowmon, Inspec, Ipswitch, iMacros, K (stylized), Kemp, Kemp (and design), Kendo UI, Kinvey, LoadMaster, MessageWay, MOVEit, NativeChat, OpenEdge, Powered by Chef, Powered by Progress, Progress, Progress Software Developers Network, SequeLink, Sitefinity (and Design), Sitefinity, Sitefinity (and design), Sitefinity Insight, SpeedScript, Stylized Design (Arrow/3D Box logo), Stylized Design (C Chef logo), Stylized Design of Samurai, TeamPulse, Telerik, Telerik (and design), Test Studio, WebSpeed, WhatsConfigured, WhatsConnected, WhatsUp, and WS\_FTP are registered trademarks of Progress Software Corporation or one of its affiliates or subsidiaries in the U.S. and/or other countries.

Analytics360, AppServer, BusinessEdge, Chef Automate, Chef Compliance, Chef Desktop, Chef Workstation, Corticon Rules, Data Access, DataDirect Autonomous REST Connector, DataDirect Spy, DevCraft, Fiddler, Fiddler Classic, Fiddler Everywhere, Fiddler Jam, FiddlerCap, FiddlerCore, FiddlerScript, Hybrid Data Pipeline, iMail, InstaRelinker, JustAssembly, JustDecompile, JustMock, KendoReact, OpenAccess, PASOE, Pro2, ProDataSet, Progress Results, Progress Software, ProVision, PSE Pro, Push Jobs, SafeSpaceVR, Sitefinity Cloud, Sitefinity CMS, Sitefinity Digital Experience Cloud, Sitefinity Feather, Sitefinity Thunder, SmartBrowser, SmartComponent, SmartDataBrowser, SmartDataObjects, SmartDataView, SmartDialog, SmartFolder, SmartFrame, SmartObjects, SmartPanel, SmartQuery, SmartViewer, SmartWindow, Supermarket, SupportLink, Unite UX, and WebClient are trademarks or service marks of Progress Software Corporation and/or its subsidiaries or affiliates in the U.S. and other countries. Java is a registered trademark of Oracle and/or its affiliates. Apache and Kafka are either registered trademarks or trademarks of the Apache Software Foundation in the United States and/or other countries. Any other marks contained herein may be trademarks of their respective owners.

Please refer to the NOTICE.txt or Release Notes – Third-Party Acknowledgements file applicable to a particular Progress product/hosted service offering release for any related required third-party acknowledgements.

**Last updated with new content:** Corticon 6.3

**Updated:** 2025/05/14



# Table of Contents

<b>About the Corticon Web Console.....</b>	<b>7</b>
<b>User Guide.....</b>	<b>13</b>
Components in a Corticon deployment.....	15
Server groups and Servers.....	16
Add Server groups and Servers.....	16
Edit Server groups and Servers.....	20
Explore Server features.....	22
Decision Services and Applications .....	26
How to use Applications .....	28
Add or Edit a Decision Service.....	28
Undeploy a Decision Service on a Server.....	34
Decision Service General Information.....	34
Decision Service Details.....	34
Application Details.....	36
Test Execution.....	36
Simplified JSON in requests.....	37
WSDL.....	39
Monitored Attributes.....	39
Batch Configurations.....	41
Add Batch Configurations.....	41
Edit Batch Configurations.....	45
Run Batch Configurations.....	46
How to view the Activity Log.....	47
<b>Administrator Guide.....</b>	<b>49</b>
User management.....	50
How to use LDAP for Web Console authentication.....	52
Configure the Activity Log.....	53
Configure auto logout.....	53
Reset the administrator password.....	54



---

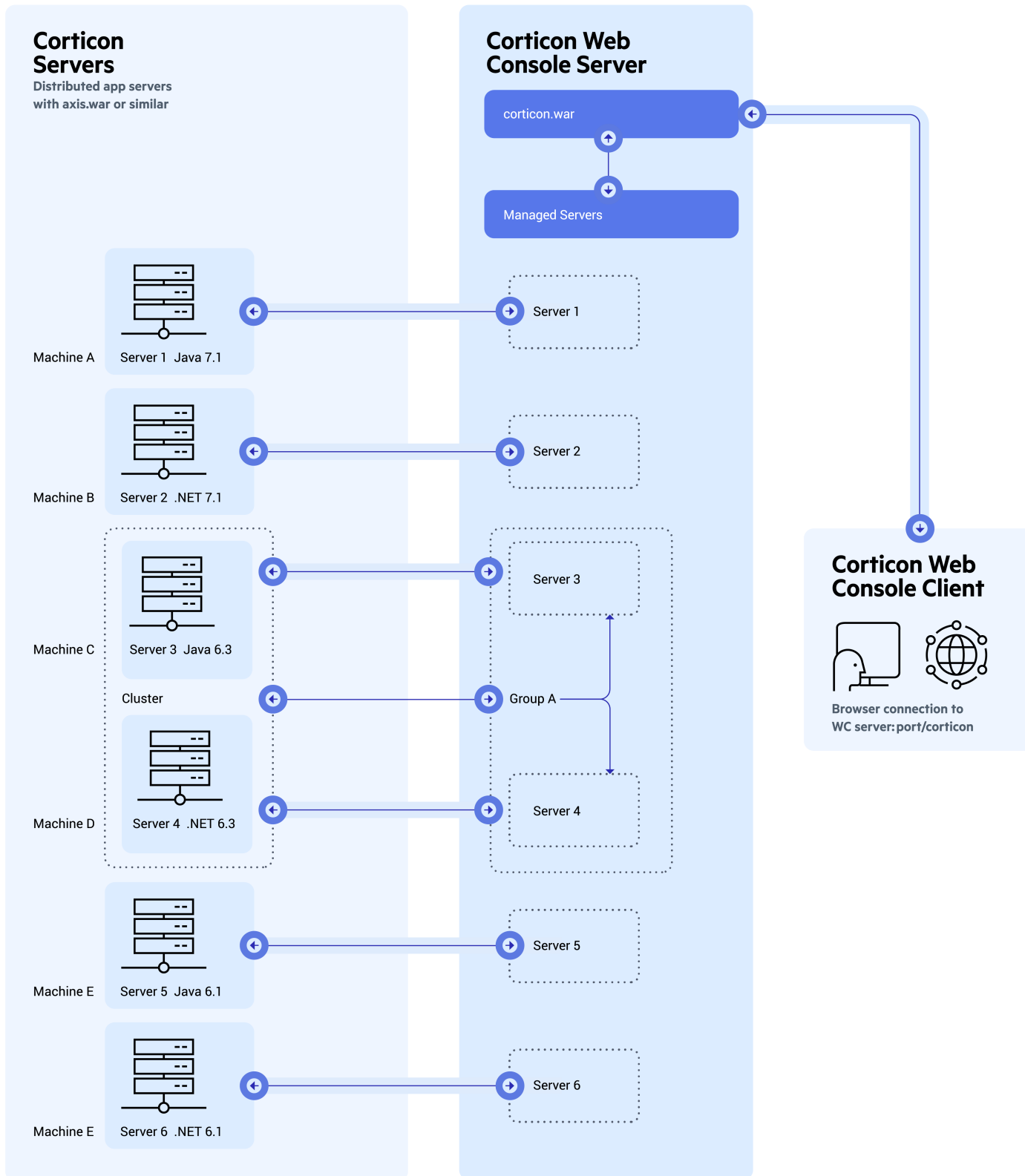
**1**

# About the Corticon Web Console

---

The Unified One Web Console

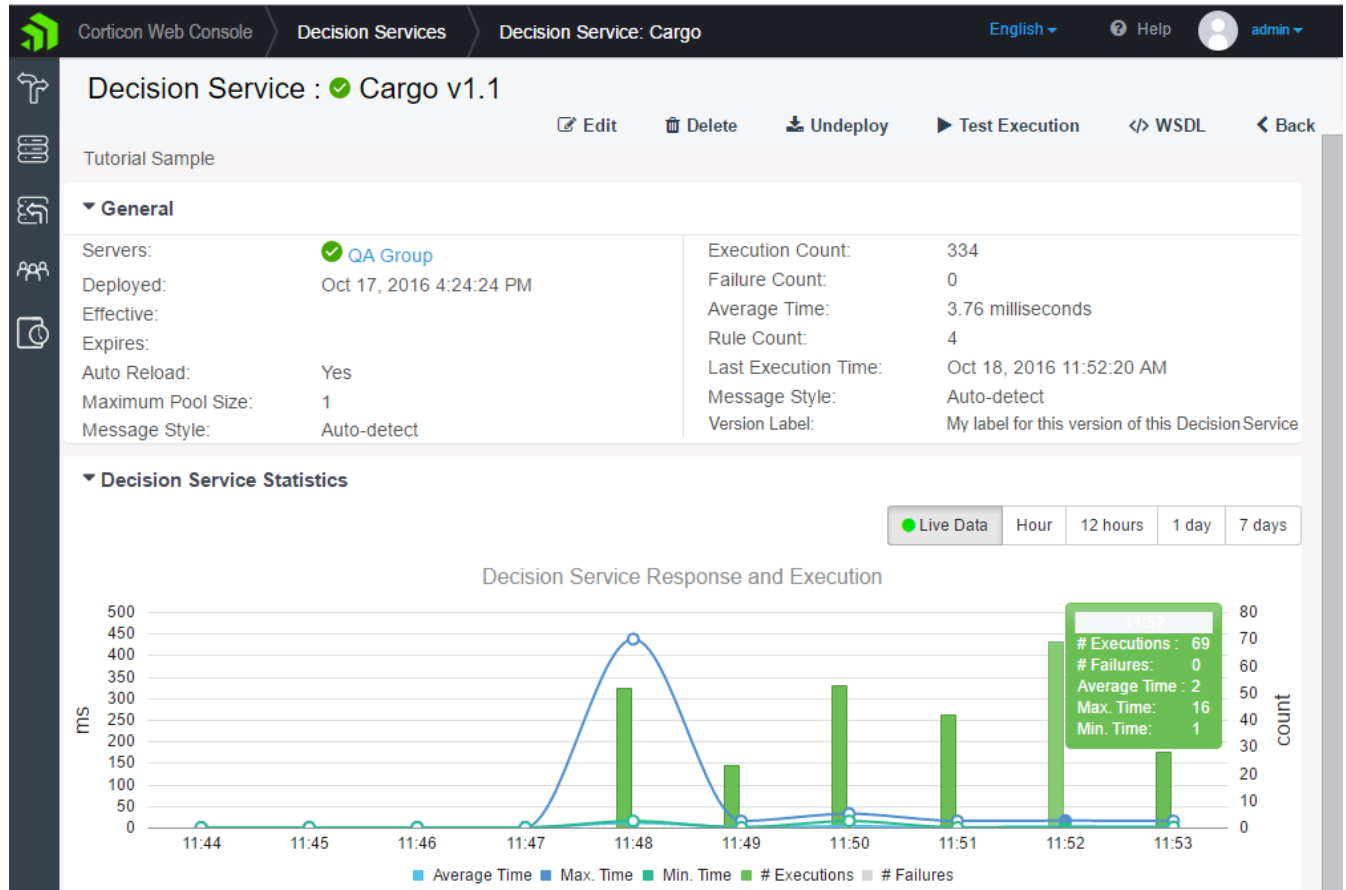
The Corticon Web Console is a distinct installation option that creates a management server accessed from a browser to manage distributed application servers hosting Corticon deployments of various versions, as illustrated:



Starting with **Corticon 6.3.4**, the Web Console has been updated to match the version available in **Corticon 7.1**. This update provides users with the latest enhancements, security updates, and capabilities. The Web Console Update is only available for 6.3 starting with Corticon 6.3.4. If you are using an earlier version of Corticon 6.3, upgrade to 6.3.4 to take advantage of the latest Web Console features and security improvements.

## A look at what the Web Console can do

Here is Web Console view of a Decision Service with a graph of the responses and executions over a span of a several minutes:



Corticon's Web Console provides a central point for administering and monitoring your Java and .NET Corticon Decision Services. Through the console you can easily deploy individual Decision Services to one or more Corticon Servers. You can also group related Decision Services into an Application to deploy and manage them as one. Once deployed, you can easily monitor the performance of the Decision Services and Corticon Servers and view both individual and aggregated metrics. Actions on Decision Services associated with a Server Group are automatically applied to each server member of the group that is running. For example, if you have a Decision Service managed by an Application which is deployed to a Server Group and add another server to the group, the Decision Service will be automatically deployed to the new server. This helps you scale up or scale down the servers in a deployment to meet demand.

The Web Console is a web application that can be installed in the same application server as the Corticon Server for single-server environments or installed separately for multiple-server environments. The choice is yours, depending on the nature of your Corticon deployment. The Web Console maintains configuration information and historical metrics in a local data store. The historical metrics let you see changes in the performance of your Decision Services and Corticon Servers over time.

Corticon's Windows **Start** menu provides shortcut to **Start Corticon Server**. When the Web Console is installed standalone, this starts just the Web Console. When the Web Console is installed together with the Corticon Server, this shortcut starts both of them. Then, the **Corticon Web Console** shortcut launches your default browser to connect to the local Web Console.

---

This guide describes user activities in the Web Console interface, followed by an administrator's section that touches on architectural features and management functions.



## User Guide

---

A server administrator uses a web browser to connect to a running Web Console Server. You will see how the Web Console interface works, how you manage Corticon Servers in a distributed architecture, and how you manage and monitor the Decision Services that run on those servers.

---

**Note: If you are getting started with the Corticon Web Console** - Requires access to a running network-accessible Corticon Server 6.3 that has the Web Console component. See the *Corticon Installation Guide* for installation information.

---

**Note: If you upgraded the Corticon Server** - An updated Corticon Server with the Web Console component will likely have left residual display and link data in its cache. Clear the browser cache to ensure that the Web Console starts cleanly.

---

### To connect to a running Web Console Server:

- On any device, in a supported browser, enter the hostname where Web Console is running followed by the port value (typically 8850) and then `/corticon`. For example:

```
http://localhost:8850/corticon
```

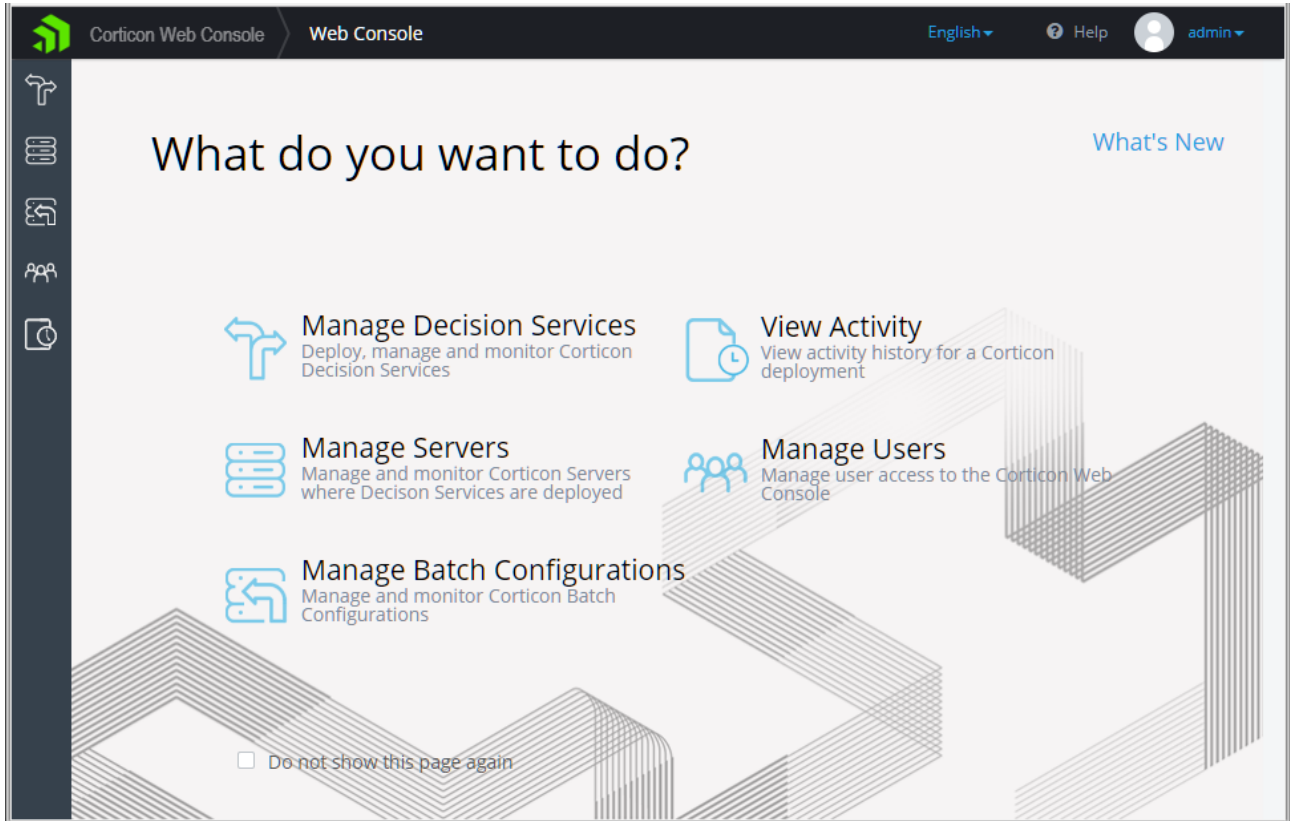
- When you are on the machine that hosts the Web Console installation, simply choose **Start > Progress > Corticon 6.3 Web Console**

### Logging in to the Web Console

Enter your user credentials in the Web Console login page. When you start using the Web Console, the one pre-defined user is the administrative user, `admin`, with the default password `admin`. If you are the administrator, you should change the default password soon after you log in. Only the `admin` user can add new users. All users have rights to deploy and manage Decision Services. If your role is as a user, obtain your user credentials from your Web Console administrator.

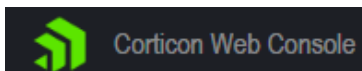
## What Do You Want To Do?

When you log in for the first time, you see a welcome page that acquaints you with the Web Console's functions. You also can access **What's New** in Corticon.



Click any action button on the welcome page, the title bar, or the function pane to close the welcome page and open the chosen page.

You can re-open the welcome page by clicking in the upper left corner of the page:



## Navigation

The general navigation elements of Web Console pages are:

- **Title bar:**
  - The navigation path to the current page in the Web Console.
  - **English:** The default language is shown. Choose your preferred available language from its drop-down list to view text in that language as well as localized formatting of dates and numbers.
  - **Help:**
    - **Help Contents:** Opens a new tab linked to the Web Console help for this release.
    - **About:** Version information about the connected Web Console Server.
    - **Community:** Opens a new tab linked to the Progress Corticon community site.
  - **admin** (the User Name that enabled log in)
    - **Profile:** Lets the user change their password, full name, and email address.

- **Logout:** Closes the session and logs the user off the Web Console Server.
- **Function bar** on the left provides access to the functional areas described on the page:



---

**Note: Automatic logout** - A user gets logged out of their Web Console session when they are inactive for a period of time specified by the Web Console administrator. A warning message is issued several seconds before the Web Console logs out with the opportunity to click **OK** to reset the inactive timeout period.

---

For details, see the following topics:

- [Components in a Corticon deployment](#)
- [Server groups and Servers](#)
- [Decision Services and Applications](#)
- [Batch Configurations](#)
- [How to view the Activity Log](#)

## Components in a Corticon deployment

The components that you work with in the Web Console are:

- **Decision Services** - The Corticon Decision Services added to the Web Console. A Decision Service is a set of Corticon rules and supporting assets packaged for deployment.
- **Applications** - Collections of one or more Decision Services to be managed as set. For example, a set of Decision Services in support of a business process that you want to deploy or monitor as a whole.
- **Servers** - Individual instances of Corticon Java or .NET Servers that have been registered with the Web Console. Once registered, the servers are available for deployment of Decision Services.
- **Server Groups** - Groups of one more Servers. Server Groups are useful when you want to deploy Decision Services to a set of Servers. For example, a set of Servers behind a load balancer, or in a regional location.
- **Users** - Defined users who can use the Web Console to administer a Corticon deployment.
- **Activity Logs** - Record of user actions in the Web Console and other asynchronous events such as a server going offline.

## Sorting and filtering components

In the pages that list Servers, Decision Services, and Users, you can readily adjust the column sizes and display as well as sort and filter which lines qualify for display by clicking on a column header, as illustrated:

Decision Service Name ▲ ▾	Version ▲ ▾	Expires ▾	CDD ▾	Server
✓ AllocateTrade			✓	✓ localhost
✓ Candidates			✓	✓ localhost
✓ Cargo			✓	✓ localhost
✓ ProcessOrder			✓	✓ localhost

Sort Ascending
Sort Descending
Columns ▶
Filter ▶

Show items with value that:
Starts with ▾
Ca
Filter
Clear

## Server groups and Servers

The Web Console allows you to manage and monitor Corticon Servers. The servers can be managed individually or in groups. Server Groups are useful when you want to deploy Decision Services to a set of Corticon Servers. A common use case is a set of Corticon Servers running behind a load balancer where each Server needs to have the same set of Decision Services deployed. Additionally, you can view aggregate metrics for the performance of the servers in a group.

### Servers that automatically register with the Web Console

In an elastic deployment environment, new server instances might not be recognized by the Web Console. These new server instances must be added manually to a Server Group in the Web Console. Server instances might spin up and down based on load. When a new server instance can register itself during startup with the Web Console, the Web Console can automatically report the server's metrics.

See *"Server registration with Web Console"* in the *Corticon Server Guide* for details on the procedures and configuration.

Also see the various configuration patterns in the following video:

## Add Server groups and Servers

As Corticon Servers are the deployment platform that runs Corticon Decision Services, your Web Console requires that you have one or more Corticon Servers under management so that you can deploy Decision Services and Applications. You can create **Server Groups** to enable common distribution of Decision Services to all servers in the group, and immediate provisioning of new servers added to the group.

---

**Note:** When you first start the Web Console in a new installation, no servers are under management unless you installed both Corticon Server for Java and Corticon Web Console. In that case, the Corticon Server is, by default, brought under management in the Web Console as the server `localhost`.

---

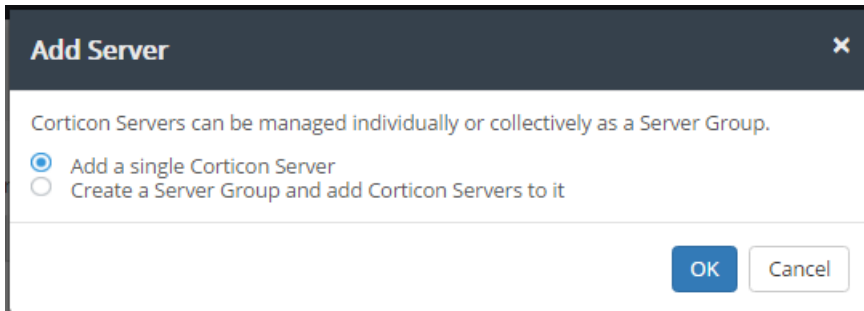
**To add servers and server groups:**

1. Connect to the Web Console server where you want to add servers and server groups.

2. Click the **Servers** button: 

3. Click **+ Add Server**: 

4. In the **Add Server** dialog box, choose whether to add a single server or a server group:



5. Click **OK**.

6. There are a few ways to add servers. The following entries are common **Server** information for each of them:

- **Protocol:** Default is HTTP. You can choose HTTPS, if this server has enabled it.
- **Hostname:** Enter the DNS-resolvable name or static IP address (avoid *localhost* and *127.0.0.1*)
- **Port:** 8850 is the default HTTP port, 8851 for HTTPS, 80 typically on IIS
- **Context URL:** The default is *axis*
- **Server Requires Authentication:** When authentication has been enabled on a server, choose this option, and then supply the user name and password for the Web Console to use to establish a connection to the server.

---

**Note:** In addition, the default context URL, **axis**, can be replaced with a preferred context URL, such as **CorticonProduction**. This functionality -- renaming a default *axis.war* file to a preferred *.war* name -- enables multiple server deployments to use the same host port and supporting resources.

---

## Adding a single Corticon Server

If you choose **Add a single Corticon Server**, the following dialog box opens:

### Add Server

Add to a Server Group

**Name**

**Description**

**Server**

Server Requires Authentication

Enter the name you want to describe this server, and a description. Then enter then the server information. Click **Save** when your entries are complete.

## Adding a Corticon Server to a new Server Group

If you choose **Create a Server Group and add Servers to it**, the following dialog box opens:

### New Server Group

**Name**

**Description**

The one or more Java Servers and .NET Servers where the solution's stakeholders will evaluate its current functionality.

**Servers** + Add

	IP Address	Port	Context URL
<input type="checkbox"/>	UAT.ourCompany.com	8850	axis

Enter the group name and a description, then click **+ Add** to open the following dialog box:

**Add a single Corticon Server**
×

▾

Server Requires Authentication

Enter then the server information. Click **Add** when your entries are complete.

If you want to add more servers at this time, click **+ Add** and follow the steps.

When your new server group is complete, click **Save**.

## Adding a Corticon Server to an existing group

When you choose **Add a single Corticon Server**, the **Add Server** dialog box provides a way to add the server to an existing group:

Click **Select Server Group** to choose a group, and then enter the server information. When your new server and its group assignment are complete, click **Save**.

**Note:** Adding a server -- individually or within a group -- as `localhost` might seem practical during evaluation and testing, but when you access Web Console from a remote machine that has a server installation that you want to add, you might find that references to `localhost` are distracting as it is not *this* localhost. It is a good practice to always use DNS-resolvable hostnames or static IP addresses.

## Edit Server groups and Servers

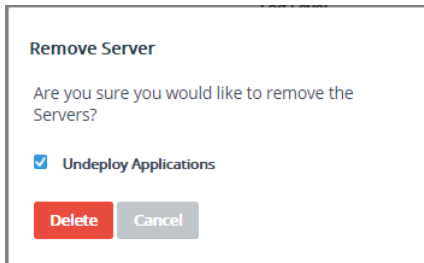
After adding a Server group or a server, you can change its configuration.

### To edit a Server Group:

Select **Edit** on the server group's **Details** page to open its edit dialog box:

	IP Address	Port	Context URL	Log Level	Monitoring
<input type="checkbox"/>	nbbbedgsaintma5	8850	axis	INFO	Enabled
<input type="checkbox"/>	nbbbedgsaintma5	80	axis	INFO	Enabled

1. Edit the name and description as appropriate
2. Click **+ Add** to add more servers.
3. Select a server to access its edit and delete functions. **Edit** lets you change the server information. **Delete** removes the server from the group and the Web Console. You are asked to decide whether to undeploy any Decision Services before deletion, and then confirm the deletion action:

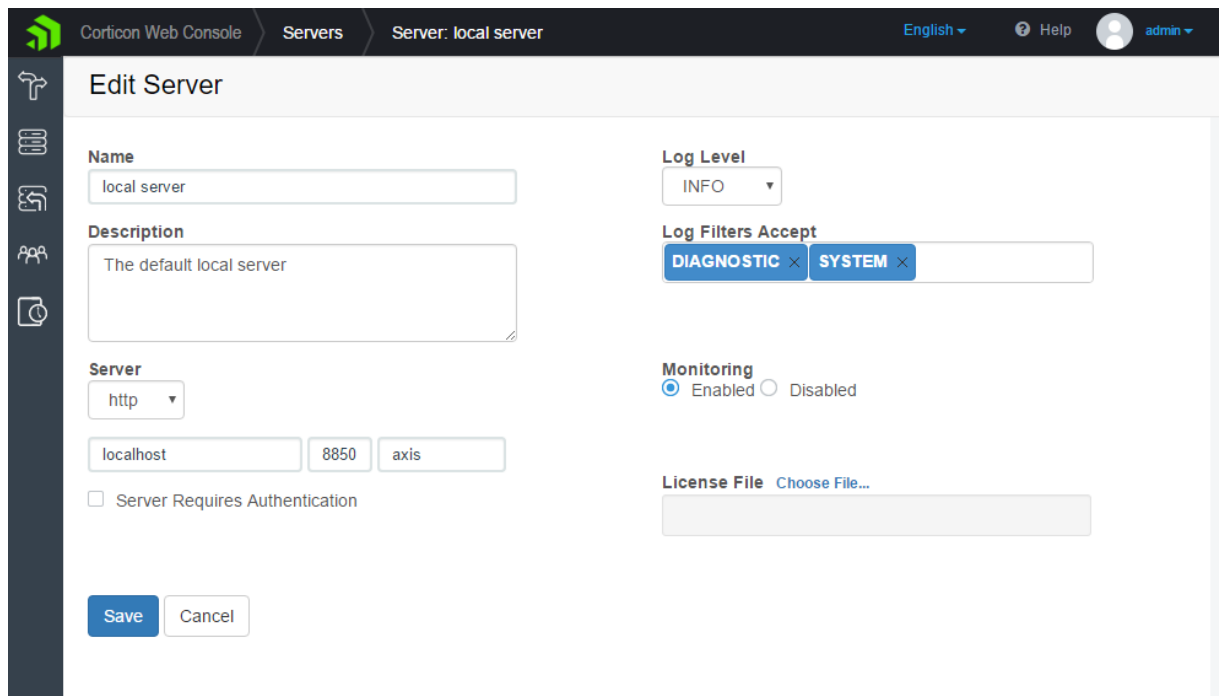


Note that deleting a server from the Web Console does not stop or delete the actual running server instance; it just removes the registration of the server with the Web Console. The server continues to run and could be added back to the Web Console.

4. You can change other server properties that will apply to all servers in the group as illustrated on the right side of the dialog box: **Log Level**, **Log Filters Accept**, **Monitoring**, and **License File**.

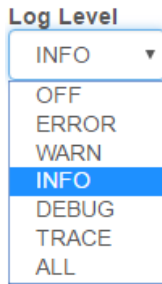
#### To edit a Server:

Select **Edit** on the server's **Details** page to open its edit dialog box:




Edit:

- **Name**
- **Description**
- **Server** hostname/IP address, port, and context URL
- **Log level** - The log level on the selected server. The default level is INFO. When you change the level and save the edits, it is immediately applied to that server without stopping and restarting the server. The logs promptly reflect the changed level of detail.



- **Monitoring** - Determines whether the statistics from this server are gathered by the Web Console and stored for later analysis.
- **License File** - Copies the selected `CcLicense.jar` (or its preferred name) from the machine where the browser is connected to the Web Console (or a network-accessible location) to the `CcServerSandbox` on the machine hosting this server.

## Explore Server features

When you click the **Servers** button:  in the left panel, the servers and server groups are listed. In this example, there is one Server Group and one Server:

The screenshot shows the "Servers" page in the Corticon Web Console. The page title is "Servers" and it includes a "+ Add Server" button. Below the title, there is a section titled "Servers" with the subtitle "Manage and monitor Servers where Decision Services are deployed." A table lists the servers and server groups:

Name	Type	Description	Status	Executions	Average Time (ms)
local server	Server	The default local server	✓	0	0.00
QA Group	Server Group	Test servers	✓	334	3.76

At the bottom of the table, there is a pagination control showing "1" and "1 - 2 of 2 items".

Clicking on a server or server group **Name** selects it, and then opens its **Details** page to display the deployment and operational information about it.

The screenshot shows the 'Server Details' page for 'localhost:8850'. The page includes a navigation bar with 'Corticon Web Console', 'Servers', and 'Server: localhost:8850'. Below the navigation bar, there are tabs for 'Execution Metrics', 'Server Statistics', 'Properties', and 'License'. The 'Execution Metrics' tab is active, showing a table with the following data:

Execution Metrics	
Execution Count:	0
Failure Count:	0
Average Time:	0 milliseconds

For the selected Server, you can choose **Edit**, **Delete**, **View Log**, or **Download Logs**.

## Server Execution Metrics

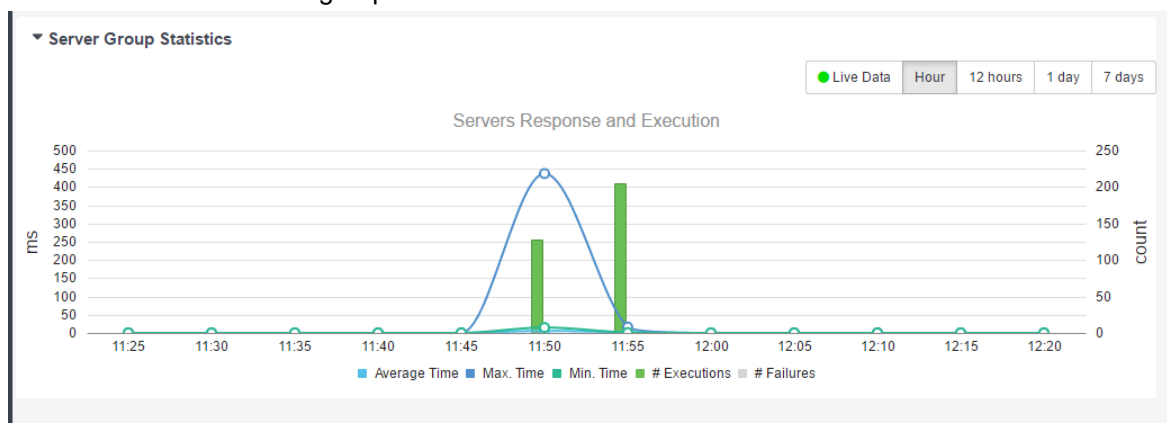
Execution metrics provide counts and performance data of all Decision Services running on the selected server, or aggregated across a server group.

The screenshot shows the 'Execution Metrics' section with the following data:

Execution Metrics	
Execution Count:	334
Failure Count:	0
Average Time:	3.76 milliseconds

## Server Statistics

You can look at metrics and statistics at several levels from for all Decision Services running on server or aggregated for all Decision Services and Servers in a server group. The following view shows the categories of information for a server group:



## Properties

A Server's **Properties** lists important settings and platform environment data of the server, from its point-of-view:

▼ Properties			
Maintenance Service Enabled	Yes	Java Version	1.8.0_131
Maintenance Service Interval	30000	Java Vendor	Oracle Corporation
Monitoring Service Enabled	Yes	Operating System	Windows 10
Maximum Memory	1003 MB	Operating System Version	10.0
Total Memory	503 MB	Architecture	amd64
Free Memory	273 MB	Autoload Directory	C:/_60x/work_dir/Server/cdd
Number Of Cores	8	Sandbox Directory	C:/_60x/work_dir/Server/SER/CcServerSandbox

Properties are specific to a Corticon Server on the machine where it is installed and running. They are accessed for an individual server, or a member of a server group.

## License

License information shows the location of the Corticon license that a specific server is using, as well as essential information about that license:

▼ License	
License Path	C:/_60x/work_dir/Server/pas/server/webapps/axis/WEB-INF/lib/CcLicense.jar/CcLicense.lic
Licensed To	Evaluation
License Deactivation Date	Jun 1, 2019
License Database Access	Yes

The license file that enabled the server to run is typically updated only when a new license has been provided that changes the expiration and enabled features for that server.

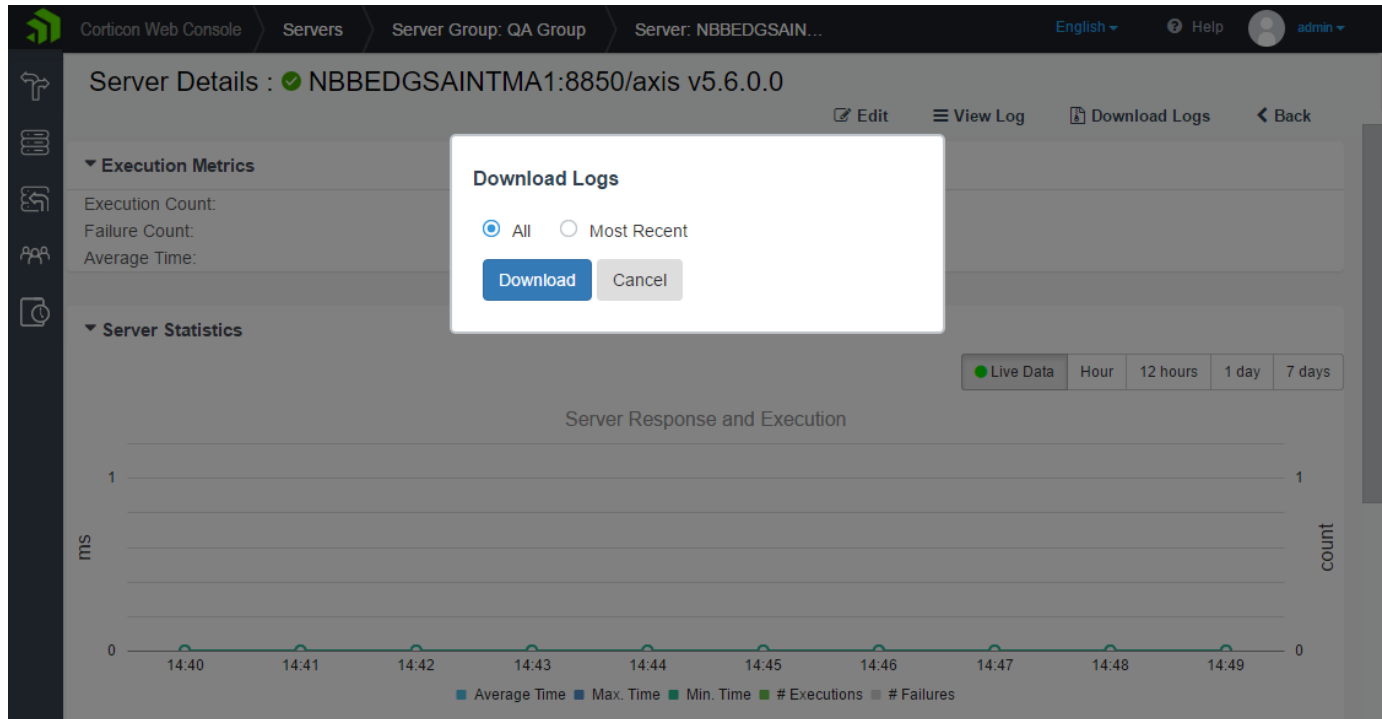
## View log

Servers lets you access the tail of the current `CcServer.log` file that the server is using:

Timestamp	Level	Logger	Marker	Message
Oct 18, 2016 2:48:30 PM	INFO	Cc	DIAGNOSTIC	id=1476816510115,shp=510.625,shp=263.09999084472656,sex=0,slq=0,sec=278,sfc=6,saex=3.539568345323741,sawt=0.5071942446043165
Oct 18, 2016 2:48:30 PM	INFO	Cc	DIAGNOSTIC	id=1476816510115,ds=ProcessOrder.1.10,ec=0,aex=0.0,awt=0.0,fc=0
Oct 18, 2016 2:48:30 PM	INFO	Cc	DIAGNOSTIC	id=1476816510115,ds=Candidates.1.14,ec=0,aex=0.0,awt=0.0,fc=0
Oct 18, 2016 2:48:30 PM	INFO	Cc	DIAGNOSTIC	id=1476816510115,ds=AllocateTrade.1.14,ec=0,aex=0.0,awt=0.0,fc=0
Oct 18, 2016 2:48:30 PM	INFO	Cc	DIAGNOSTIC	id=1476816510115,ds=Cargo.1.1,ec=278,aex=3.539568345323741,awt=0.5071942446043165,fc=6
Oct 18, 2016 2:48:30 PM	INFO	Cc	DIAGNOSTIC	id=1476816510115,ds=Cargo.0.16,ec=0,aex=0.0,awt=0.0,fc=0
Oct 18, 2016 2:48:00 PM	INFO	Cc	DIAGNOSTIC	id=1476816480099,shp=510.6875,shp=317.77515411376953,sex=0,slq=0,sec=278,sfc=6,saex=3.539568345323741,sawt=0.5071942446043165
Oct 18, 2016 2:48:00 PM	INFO	Cc	DIAGNOSTIC	id=1476816480099,ds=ProcessOrder.1.10,ec=0,aex=0.0,awt=0.0,fc=0
Oct 18, 2016 2:48:00 PM	INFO	Cc	DIAGNOSTIC	id=1476816480099,ds=Candidates.1.14,ec=0,aex=0.0,awt=0.0,fc=0
Oct 18, 2016 2:48:00 PM	INFO	Cc	DIAGNOSTIC	id=1476816480099,ds=AllocateTrade.1.14,ec=0,aex=0.0,awt=0.0,fc=0
Oct 18, 2016 2:48:00 PM	INFO	Cc	DIAGNOSTIC	id=1476816480099,ds=Cargo.1.1,ec=278,aex=3.539568345323741,awt=0.5071942446043165,fc=6

## Download log

Corticon Web Console enables you to download and view Server log files. This is especially useful when you need to locally examine a remote Server's log files to identify the source of a problem. To download a Server's log files, click **Download Logs** in the Server page. This opens a dialog box where you can choose to download **All** log files or only the **Most Recent**. If you choose **All**, all log files that have been retained since installation will be downloaded. If you choose **Most Recent**, you will get all log files that have been modified by the Server in the last 24 hours. Select the appropriate option and click **Download**. This downloads a ZIP file named `CcServerLog.zip` that contains the Server log files.



To know more about troubleshooting a problem by examining Server logs, see *Troubleshooting Corticon Server in the Server Guide*.

## Decision Services and Applications

### Types of Decision Services

Many Decision Services might be deployed on a Corticon Server. There are two types of Decision Services from the point of view of the Web Console, based on how they were deployed:

- **Managed Decision Services** are those deployed through the Web Console. For managed Decision Services the Web Console has the EDS file, and can perform more management activities such as deploying it to additional Corticon Servers. Managed Decision Services can be:
  - Added directly through the Web Console's **Add Decision Service** feature.
  - Added directly from Corticon Studio using the Studio's **Package and Deploy** feature. Studio prompts for the Application where the Decision Service will be added, and the Server or Server Group where it will be deployed.
- **Discovered Decision Services** are those deployed not through the Web Console but through another means. The management operations the Web Console can perform on discovered Decision Services is limited so as not to conflict with how they were deployed. Discovered Decision Services could be:
  - Decision Services packaged and deployed directly from Corticon Studio or any of the deployment tools.
  - These are Decision Services deployed through Corticon Deployment Descriptors (CDDs) -- text-based files that specify a Decision Service to be deployed and its deployment properties. CDD files are automatically loaded by the Corticon Server.

In most deployments, you will likely use either managed or unmanaged Decision Services. The approach you take for deployment and management depends on your needs.

## How Decision Service types are displayed

When you deploy Corticon rules through CDD files, your unmanaged Decision Services are *discovered*, as shown:

▼ Discovered Decision Services						
Decision Services not deployed using the Web Console, can be viewed here but not fully managed.						
Decision Service Name ▲	Version	CDD	Server	Executions	Average Time (ms)	
✓ AllocateTrade	1.14	✓	✓ localhost:8850/axis	0	0.00	
✓ AllocateTrade	1.14	✓	✓ NBBEDGSAINTMA5:8850/axis	0	0.00	
✓ AllocateTrade	1.14	✓	✓ NBBEDGSAINTMA1:8850/axis	0	0.00	
✓ Candidates	1.14	✓	✓ NBBEDGSAINTMA5:8850/axis	0	0.00	
✓ Candidates	1.14	✓	✓ localhost:8850/axis	0	0.00	
✓ Candidates	1.14	✓	✓ NBBEDGSAINTMA1:8850/axis	0	0.00	
✓ Cargo	0.16	✓	✓ NBBEDGSAINTMA1:8850/axis	0	0.00	
✓ Cargo	0.16	✓	✓ NBBEDGSAINTMA5:8850/axis	0	0.00	
✓ Cargo	0.16	✓	✓ localhost:8850/axis	0	0.00	
✓ ProcessOrder	1.10	✓	✓ NBBEDGSAINTMA5:8850/axis	0	0.00	

1 - 10 of 12 items

When you deploy Decision Services through the Web Console, they are shown as *managed*:

▼ Managed Decision Services				
Decision Services deployed using the Web Console, can be fully managed here.				
Decision Service Name ▲	Version	Servers	Executions	Average Time (ms)
✓ Cargo	1.1	✓ QA Group	0	0.00

1 - 1 of 1 items

When you use applications to group your Decision Services, each *managed* Application lists its Decision Services:

▼ Managed Decision Services																			
Decision Services deployed using the Web Console, can be fully managed here.																			
Application Name ▲	Servers	Executions	Average Time (ms)																
✓ Insurance	✓ local server	0	0.00																
<table border="1"> <thead> <tr> <th>Decision Service Name ▲</th> <th>Version</th> <th>Executions</th> <th>Average Time (ms)</th> <th></th> </tr> </thead> <tbody> <tr> <td>✓ Generate Policy</td> <td>0.41</td> <td>0</td> <td>0.00</td> <td></td> </tr> <tr> <td>✓ Price Policy</td> <td>0.41</td> <td>0</td> <td>0.00</td> <td></td> </tr> </tbody> </table>					Decision Service Name ▲	Version	Executions	Average Time (ms)		✓ Generate Policy	0.41	0	0.00		✓ Price Policy	0.41	0	0.00	
Decision Service Name ▲	Version	Executions	Average Time (ms)																
✓ Generate Policy	0.41	0	0.00																
✓ Price Policy	0.41	0	0.00																

1 - 1 of 1 items

## Opening the Decision Services and Applications page

1. Connect to the Web Console server where you manage Decision Services.

2. Click the **Decision Services** button:



The Decision Services page shows all the types of Decision Services on the managed servers, as illustrated:

The screenshot displays the 'Decision Services' page in the Corticon Web Console. The interface includes a top navigation bar with 'Corticon Web Console', 'Decision Services', and user information. A sidebar on the left contains navigation icons. The main content area is divided into two sections:

- Managed Decision Services:** This section is for services deployed via the Web Console. It contains two tables:
  - Insurance:** Deployed on 'local server'. It lists two decision services: 'Generate Policy' (version 0.41) and 'Price Policy' (version 0.41), both with 0 executions and 0.00 ms average time.
  - Cargo:** Deployed on 'QA Group'. It lists one decision service: 'AllocateTrade' (version 1.1) with 0 executions and 0.00 ms average time.
- Discovered Decision Services:** This section shows services not deployed via the Web Console. It lists three 'AllocateTrade' services (version 1.14) on different servers: 'localhost:8850/axis', 'NBBEDGSAINTMA5:8850/axis', and 'NBBEDGSAINTMA1:8850/axis'. Each has 0 executions and 0.00 ms average time.

## How to use Applications

An Application is a group of Decision Services that you can deploy to a Server or Server Group. When you deploy an Application to a Server Group, all Decision Services in the Application are deployed to each of the Corticon Servers in the Server Group. Further, if a new server is added to the Server Group, the Web Console automatically deploys the Application to it. An Application is therefore, a unit of deployment. It enables you to manage a set of related Decision Services more easily.

In order to add a Decision Service to an Application, you need to have a Decision Service file (.eds) that was packaged from a Ruleflow. There are several toolsets that perform this task, as discussed in the section *"How to package and deploy Decision Services" in the Deployment Guide*.

A feature of Corticon Studio, as described in *"Deploy to Corticon Web Console" in the Deployment Guide*, enables you to select Ruleflows in a project to deploy as Decision Services that are sent to a new or existing Application assigned to a server or server group managed in a Web Console. As a result, the Decision Services are immediately deployed (or redeployed) to the server or all active servers in the Server Group.

## Add or Edit a Decision Service

The following procedures show to bring a Decision Service under management either as an independent Decision Service, or as a member of an Application.

---

**Note:** As the general steps are common to both adding and editing a Decision Service, this topic focuses on the tasks when adding a Decision Service, and then shows how to access a Decision Service to edit it.

---

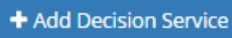
**To add a Decision Service:**

1. Connect to the Web Console server where you want to add Decision Services.

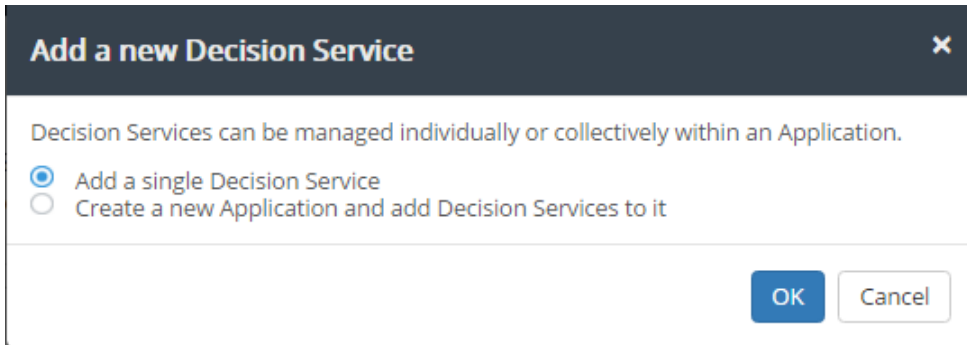
2. Click the **Decision Services** button:



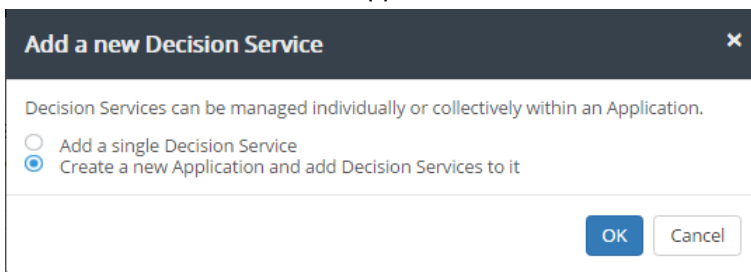
3. Click **+ Add Decision Service**:



4. The **Add a new Decision Service** dialog box opens:



5. You can choose to create an Application for the Decision Service you are adding:



a. If you choose that option and click **OK**, the **New Application** dialog opens:

Corticon Web Console Decision Services New Application English Help admin

## New Application

**Name**  
Insurance

**Description**  
Insurance Sample

**Servers**  
QA Group

**Rule Messages Restrictions**

- Restrict Info Rule Messages
- Restrict Warning Rule Messages
- Restrict Violation Rule Messages

**Decision Services** + Add

	Name	File Name	Database Access Mode
<input type="checkbox"/>	GeneratePolicy	GeneratePolicy.eds	None
<input type="checkbox"/>	PricePolicy	PricePolicy.eds	None

Save Save & Deploy Cancel

- b. Enter a Name and Description.
  - c. Choose the server or server group where the Application's Decision Services are to be deployed.
  - d. Set options that will apply to all Decision Services in the Application.
  - e. Click **+ Add** for each Decision Service you want to add to the Application.
6. On either path, the **Add Decision Service** dialog box opens at the **Decision Service** tab:

### Add Decision Service

Decision Service Database Advanced Monitored Attributes

When adding a Decision Service you must specify a name, select a server and provide the EDS file of the Decision Service. Other properties are optional. To add the Decision Service to an existing Application select "Add to an Existing Application"

**Name**  
Decision Service Name

**Description**

**EDS File** Choose File...

**Servers**  
Select Servers

Add to an Existing Application

Save Save & Deploy Cancel

- a. Enter a name. Note that this will be its name when deployed, not the name of the EDS file you choose.
- b. Add a description.
- c. Click **Choose file** to locate an EDS file.
- d. Choose a server or server group
- e. If you started this process as a single Decision Service, you can choose to add it to an existing application from the list that will be offered.

7. Click the **Database** tab to access its options:

The screenshot shows a dialog box titled "Add Decision Service" with a close button (X) in the top right corner. The dialog has four tabs: "Decision Service", "Database", "Advanced", and "Monitored Attributes". The "Database" tab is currently selected. Below the tabs, there is a text instruction: "If this Decision Service connects to a database you must provide a Datasource Configuration file with connection parameters." Below this instruction is a label "Datasource Configuration File" followed by a "Choose File..." button. Underneath is a large, empty text input field. Another text instruction follows: "If this Decision Service uses Corticon EDC to connect to a database you must select a EDC Access Mode of Read Only or Read/Update." Below this is a section titled "EDC Access Mode" with three radio button options: "None" (which is selected), "Read Only", and "Read/Update". At the bottom of the dialog, there are three buttons: "Save", "Save & Deploy", and "Cancel".

- a. **Datasource Configuration File:** Specify the XML file that contains the data source access properties. To learn how to generate this file from Corticon Studio, see the topic: *"Export the Datasource Configuration file" in the Data Integration Guide.*
- b. You can change the **EDC Access Mode** option to either **Read Only** or **Read/Update** to extend the dialog tab to display additional configuration settings:

**Add Decision Service** ✕

Decision Service Database **Advanced** Monitored Attributes

If this Decision Service connects to a database you must provide a Datasource Configuration file with connection parameters.

Datasource Configuration File [Choose File...](#)

If this Decision Service uses Corticon EDC to connect to a database you must select a EDC Access Mode of Read Only or Read/Update.

**EDC Access Mode**

None  Read Only  Read/Update

**EDC Entities Returned Mode**

All Entities  Incoming and New Entities

EDC Caching

Save Save & Deploy Cancel

- c. In the **EDC Access Mode**, choose the appropriate access option. This setting controls how a Decision Service will access connected databases. Select **Read Only** or **Read/Update** to then expose additional settings that you need to configure:
- **EDC Entities Returned Mode:** Choosing **All Entities** returns all records from the database when the Decision Service executes. Choosing **Incoming and New Entities** returns entities that were in the request message and only those entity records that are added or modified in the database when the Decision Service executes. Select the appropriate option.
  - **EDC Caching:** Database caching enables Corticon to store often-used data in a cache. This improves the performance of the Decision Service since it can read and write data in the cache faster than if this data was in the database. If you choose **Enabled**, database caching will be enabled for the Decision Service. To learn more about database caching, see the topic: *"How to work with database caches" in the Data Integration Guide.*

---

**Important: Turning caching on or off** - If you want to enable or disable caching on a deployed Decision Service, the mechanisms of caching require that you undeploy and delete the Decision Service, and then add and deploy the Decision Service again with the cache enablement setting you want.

---

8. Click the **Advanced** tab to access its options:

Consult the *Server Guide* for more information about these settings.

- a. In the **Maximum Pool Size** field, specify how many execution threads for this Decision Service will be added to the execution queue. If you leave this field blank, the Web Console will set a default value of **1**.
  - b. In the **XML message style** section, choose whether request messages for this Decision Service should contain a **Flat** or **Hierarchical** payload structure. **Auto Detect** accepts either style.
9. Click the **Monitored Attributes** tab to access its options:

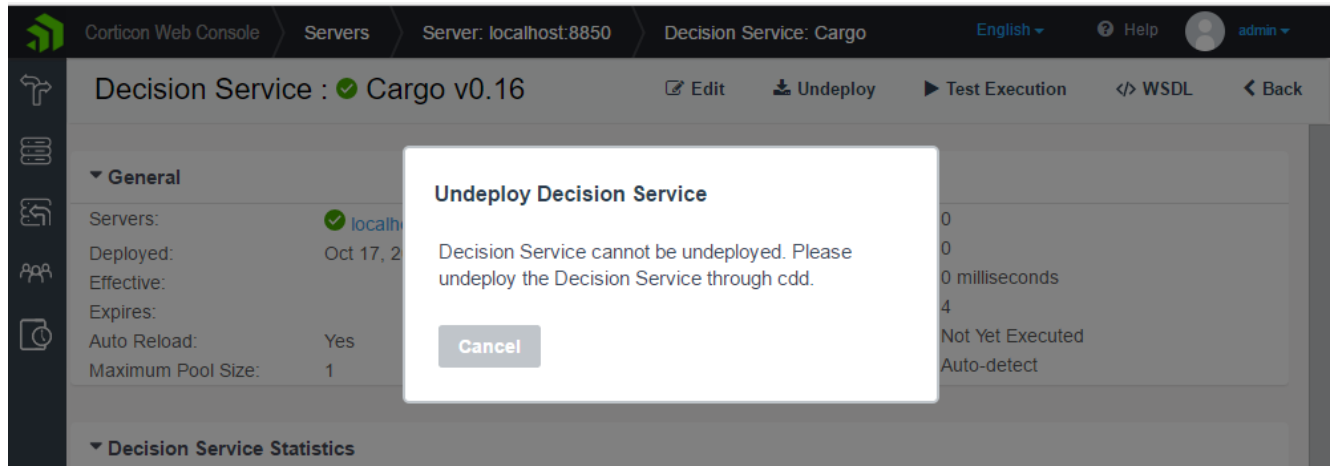
See how to set up use Monitored Attributes in the topic [Monitored Attributes](#) on page 39

10. Click **Save** to store the Decision Service but not deploy it. Click **Save & Deploy** to store the Decision Service and also deploy it. Click **Cancel** to close without making changes.

## Undeploy a Decision Service on a Server

You can undeploy Decision Services by selecting the Decision Service and clicking **Undeploy**. Performing this operation on a managed Decision Service takes you to the Application details page, which has options to remove individual Decision Services or undeploy the Application altogether.

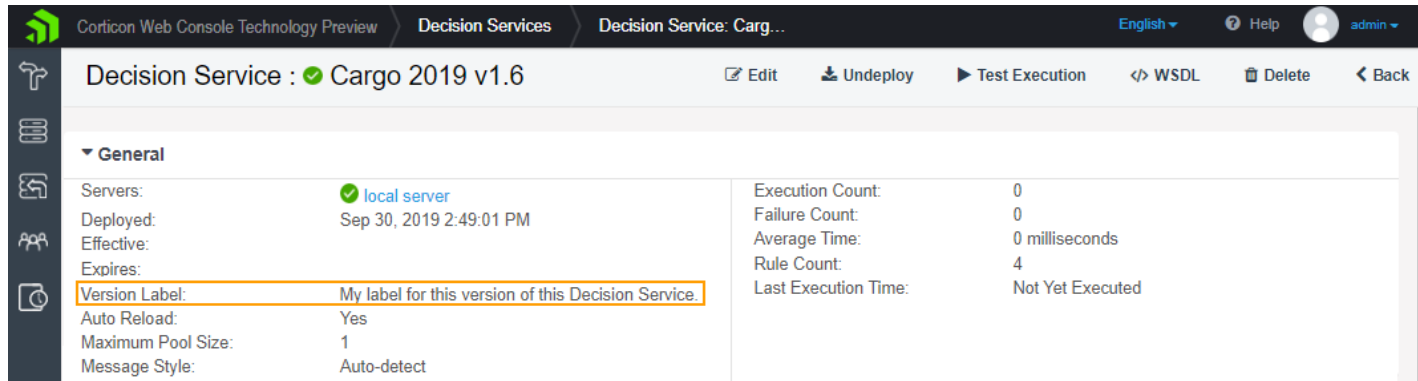
**Note:** In the Web Console, you cannot undeploy a Decision Service that was deployed using a CDD file.



## Decision Service General Information

General metrics are a simple table of the count of all request executions of a Decision Service on the selected server, the count of failures, and the average execution time. The average time is average execution time for execution of all the Decision Services on this server.

The Version Label is the text that was added to the Ruleflow that generated the versioned Decision Service.



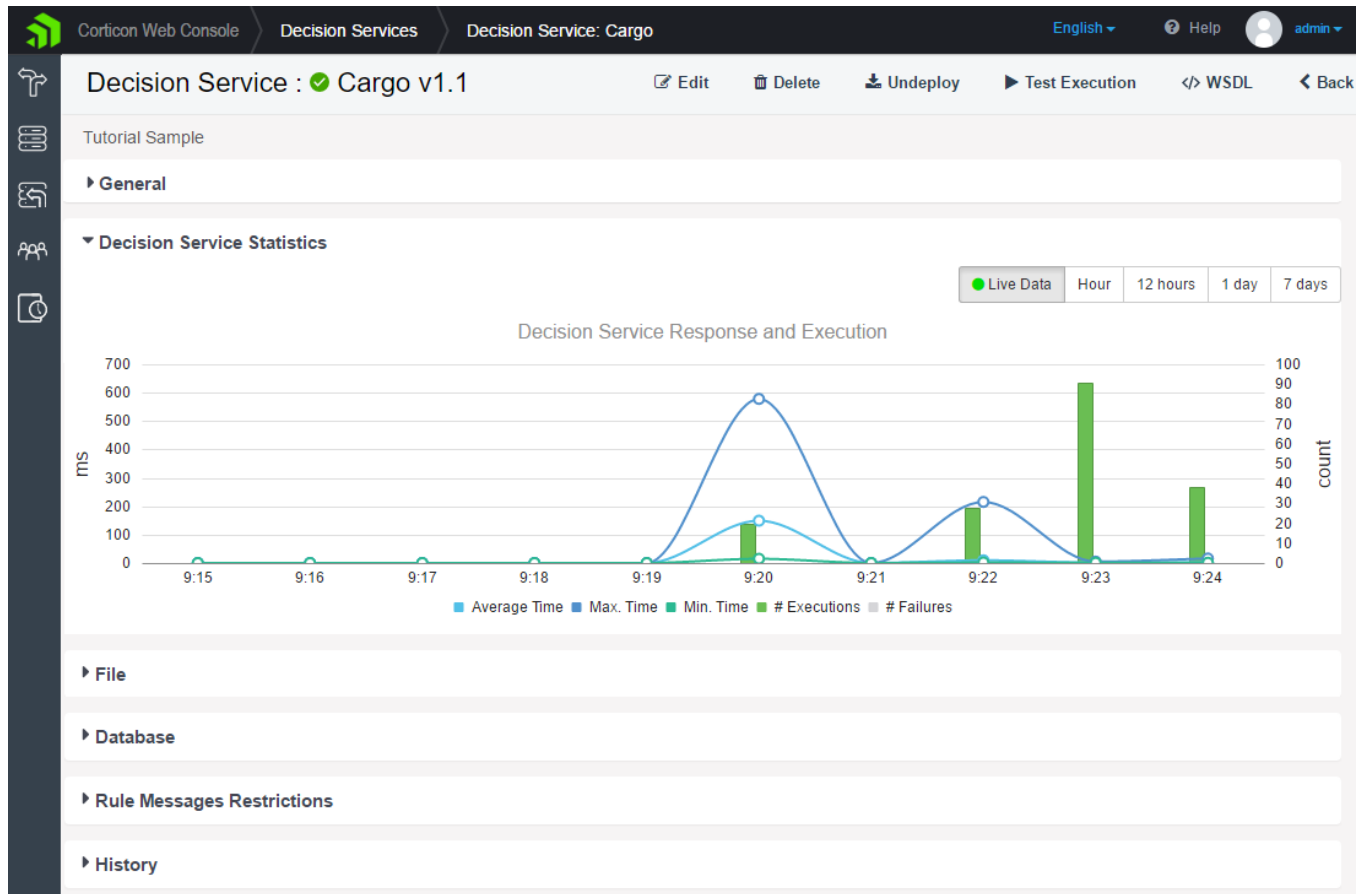
**Note:** These metrics are reset when a server restarts.

## Decision Service Details

Click on a Decision Service to display its operational and performance data.

If the Decision Service is deployed to a Server Group, the operational and performance data is an aggregate of that Decision Service from all servers in the server group.

The actions available let you **Edit**, **Delete**, **Undeploy**, **Test Execution**, and display **WSDL**.



You can collapse and expand sections of the page to manage the display, as illustrated:

The screenshot displays the Corticon Web Console interface for a Decision Service. The breadcrumb navigation shows 'Corticon Web Console' > 'Decision Services' > 'Decision Service: Freig...'. The user is logged in as 'admin'. The main content area is titled 'Decision Service : ✔ Freight1 v1.1' and includes several sections:

- General:** Servers: My Five (status: ✔); Deployed: Nov 1, 2016 6:55:29 PM; Effective: ; Expires: ; Auto Reload: Yes; Maximum Pool Size: 1. Execution Count: 0; Failure Count: 0; Average Time: 0 milliseconds; Rule Count: 4; Last Execution Time: Not Yet Executed; Message Style: Auto-detect.
- Decision Service Statistics:** (Collapsible section)
- Monitored Attributes:** (Collapsible section)
- File:** Local File: Cargo56.eds; EDS File Timestamp: Oct 31, 2016 1:56:42 PM.
- Database:** Database Access Mode: None; Database Access Properties File: ; Database Entities Returned Mode: ALL; Database Caching: No; Use Execution Recording Service: No.
- Rule Messages Restrictions:** Restrict Info Rule Messages: System Default; Restrict Violation Rule Messages: System Default; Restrict Warning Rule Messages: System Default.
- History:** Created On: Oct 31, 2016 9:56:41 PM; Created By: admin.

## Application Details

The general metrics shown for an application are a rollup of the metrics of the Decision Services in the Application. For example, the average execution time shown on an application is the average execution time of all its Decision Services.

## Test Execution

The **Test Execution** option lets you test your Decision Service by making a REST or SOAP request to it. When you select the Test Execution, you choose a server where the Decision Service is deployed, whether to make a REST or SOAP request, and then locate a JSON or XML file for the payload of the request.

---

**Note:** While the Decision Service name is essential for Corticon requests, this panel ignores the `decisionServiceName` parameter in the request as it is focused on the current Decision Service.

---

To execute a test against a selected deployment of the current Decision Service :

1. Click **Server** to select a server that has the Decision Service deployed.
2. In the **Choose Request File** area, click **Choose File**, then locate and open an XML or JSON request appropriate for the Decision Service. The **Request** area shows the request text.
3. Choose its **Request Type**.
4. Click **Execute**.

The request executes, and then adds the **Response** text, as shown:

The screenshot displays the Corticon Web Console interface for a test execution. The breadcrumb navigation shows 'Corticon Web Console' > 'Decision Services' > 'Application: Cargo' > 'Test Execution: Cargo'. The page title is 'Test Execute: Cargo'. On the right, there are 'Execute' and 'Back' buttons. The main content area is divided into several sections:

- Server:** A dropdown menu showing 'NBBEDGSAINTMA1:8850/axis'.
- Choose Request File:** A 'Choose File...' button and a text input field containing 'Sample.xml'.
- Request Type:** Radio buttons for 'REST/JSON' and 'SOAP/XML', with 'SOAP/XML' selected.
- Request:** A text area containing an XML request:
 

```
<?xml version="1.0" encoding="UTF-8"?>
<CorticonRequest xmlns="urn:Corticon" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
decisionServiceName="InsertDecisionServiceName">
  <WorkDocuments>
    <Cargo id="Cargo_id_1">
      <weight>1000</weight>
      <volume>10</volume>
      <container xsi:nil="true" />
    </Cargo>
    <Cargo id="Cargo_id_2">
      <weight>1000</weight>
      <volume>40</volume>
      <container xsi:nil="true" />
    </Cargo>
    <Cargo id="Cargo_id_3">
      <weight>30000</weight>
      <volume>20</volume>
    </Cargo>
  </WorkDocuments>
</CorticonRequest>
```
- Response:** A text area containing an XML response:
 

```
<?xml version="1.0" encoding="UTF-8"?><CorticonResponse xmlns="urn:Corticon" decisionServiceName="Cargo"
decisionServiceTargetVersion="1.1">
  <WorkDocuments>
    <Cargo id="Cargo_id_1">
      <container>standard</container>
      <volume>10</volume>
      <weight>1000</weight>
    </Cargo>
    <Cargo id="Cargo_id_2">
      <container>oversize</container>
      <volume>40</volume>
      <weight>1000</weight>
    </Cargo>
    <Cargo id="Cargo_id_3">
      <container>heavyweight</container>
      <volume>20</volume>
      <weight>30000</weight>
    </Cargo>
  </WorkDocuments>
</CorticonResponse>
```

## Simplified JSON in requests

Some users find that their JSON requests have metadata only at the root, expecting that the decision service can infer the metadata for subordinate levels. That tactic is now supported, although the output provides the metadata at all levels. The following example shows the `coupons` Ruletest in the advanced tutorial. It executes as expected with either request, and produces identical responses (both with metadata).

## How to: Use Simplified JSON in requests

1. Copy the request you want to simplify to an editor.
2. Add the decision service name as name at the top.
3. Delete (or leave out) the metadata for related entities.
4. Retain the metadata for the primary entity.

For example:

```

1 {
2   "__metadataRoot": {"#locale": ""},
3   "Objects": [{
4     "preferredCard": {
5       "cumulativeCashBack": 9.24,
6       "__metadata": {
7         "#type": "PreferredAccount",
8         "#id": "PreferredAccount_id_1"
9       },
10      "cardNumber": "12"
11    },
12    "ShoppingCart": {
13      "totalAmount": null,
14      "Item": [
15        {
16          "price": 25,
17          "name": "Miller Beer",
18          "__metadata": {
19            "#type": "Item",
20            "#id": "Item_id_1"
21          },
22          "department": null,
23          "barCode": "39-291-1234"
24        },
25        {
26          "price": 1.99,
27          "name": "RedBull",
28          "__metadata": {
29            "#type": "Item",
30            "#id": "Item_id_2"
31          },
32          "department": null,
33          "barCode": "39-285-98765"
34        }
35      ],
36      "useCashBack": true,
37      "savings": null,
38      "__metadata": {
39        "#type": "ShoppingCart",
40        "#id": "ShoppingCart_id_1"
41      }
42    },
43    "__metadata": {
44      "#type": "Customer",
45      "#id": "Customer_id_1"
46    },
47    "Name": ""
48  ]
49 }

```

```

1 {
2   "__metadataRoot": {"#locale": ""},
3   "name": "MyAdvancedTutorial",
4   "Objects": [{
5     "preferredCard": {
6       "cumulativeCashBack": 9.24,
7       "cardNumber": "12"
8     },
9     "ShoppingCart": {
10      "totalAmount": null,
11      "Item": [
12        {
13          "price": 25,
14          "name": "Miller Beer",
15          "department": null,
16          "barCode": "39-291-1234"
17        },
18        {
19          "price": 1.99,
20          "name": "RedBull",
21          "department": null,
22          "barCode": "39-285-98765"
23        }
24      ],
25      "useCashBack": true,
26      "savings": null
27    },
28    "__metadata": {
29      "#type": "Customer",
30      "#id": "Customer_id_1"
31    },
32    "Name": ""
33  ]
34 }

```

### Note

- In the example, Name at the bottom is an attribute of the root entity.
- Mind your commas, braces, and brackets when you are paring down a known-good request.

## WSDL

The **WSDL** option displays the current Decision Service's WSDL, and also provides a link to WSDL data in an editor:

The screenshot shows the Corticon Web Console interface. A modal window titled "WSDL" is open, displaying the WSDL for the "Cargo" Decision Service. The WSDL content is as follows:

```
<?xml version="1.0" encoding="UTF-8"?><definitions xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="urn:Corticon">
  <types>
    <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespace="urn:Corticon" elementFormDefault="qualified">
      <xsd:element name="CorticonRequest" type="tns:CorticonRequest"/>
      <xsd:element name="CorticonResponse" type="tns:CorticonResponse"/>
      <xsd:complexType name="CorticonRequest">
        <xsd:sequence>
          <xsd:element name="ExecutionProperties" maxOccurs="1" type="tns:ExecutionProperties" minOccurs="0"/>
          <xsd:element name="WorkDocuments" type="tns:WorkDocuments"/>
        </xsd:sequence>
        <xsd:attribute name="decisionServiceName" use="required" type="xsd:string" fixed="Cargo.eds"/>
        <xsd:attribute name="decisionServiceTargetVersion" use="optional" type="xsd:nonNegativeInteger"/>
        <xsd:attribute name="decisionServiceEffectiveTimestamp" use="optional" type="xsd:dateTime"/>
      </xsd:complexType>
      <xsd:complexType name="CorticonResponse">
        <xsd:sequence>
          <xsd:element name="ExecutionProperties" maxOccurs="1" type="tns:ExecutionProperties" minOccurs="0"/>
          <xsd:element name="WorkDocuments" type="tns:WorkDocuments"/>
          <xsd:element name="Messages" type="tns:Messages"/>
        </xsd:sequence>
        <xsd:attribute name="decisionServiceName" use="required" type="xsd:string" fixed="Cargo.eds"/>
        <xsd:attribute name="decisionServiceTargetVersion" use="optional" type="xsd:nonNegativeInteger"/>
        <xsd:attribute name="decisionServiceEffectiveTimestamp" use="optional" type="xsd:dateTime"/>
      </xsd:complexType>
      <xsd:complexType name="ExecutionProperties">
        <xsd:sequence>
          <xsd:element name="ExecutionProperty" maxOccurs="unbounded" type="tns:ExecutionProperty" minOccurs="0"/>
        </xsd:sequence>
      </xsd:complexType>
      <xsd:complexType name="ExecutionProperty">
        <xsd:sequence>
          <xsd:attribute name="executionPropertyName" use="required" type="xsd:string"/>
          <xsd:attribute name="executionPropertyValue" use="required" type="xsd:string"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:schema>
  </types>
  <message name="CorticonRequest" type="tns:CorticonRequest"/>
  <message name="CorticonResponse" type="tns:CorticonResponse"/>
  <portType name="CorticonRequestResponse" >
    <input message="tns:CorticonRequest" type="tns:CorticonRequest"/>
    <output message="tns:CorticonResponse" type="tns:CorticonResponse"/>
  </portType>
  <binding name="CorticonRequestResponse" type="tns:CorticonRequestResponse">
  </binding>
  <service name="CorticonRequestResponse" >
    <port name="CorticonRequestResponse" type="tns:CorticonRequestResponse"/>
  </service>
</definitions>
```

Below the WSDL content, the "Cargo v1.1 WSDL URL:" is displayed as <http://NBBEDGSAINTMA5:8850/axis/dswsdl/Cargo/1/1>. A "Close" button is visible in the bottom right corner of the modal.

The background interface shows the "File" section with "Local File: Cargo.eds" and "EDS File Timestamp: Oct 17, 2016 4:24:24 PM". The "Database" section shows settings for "Database Access Mode: None", "Database Access Properties File:", "Database Entities Returned Mode: ALL", "Database Caching: No", and "Use Execution Recording Service: No".

## Monitored Attributes

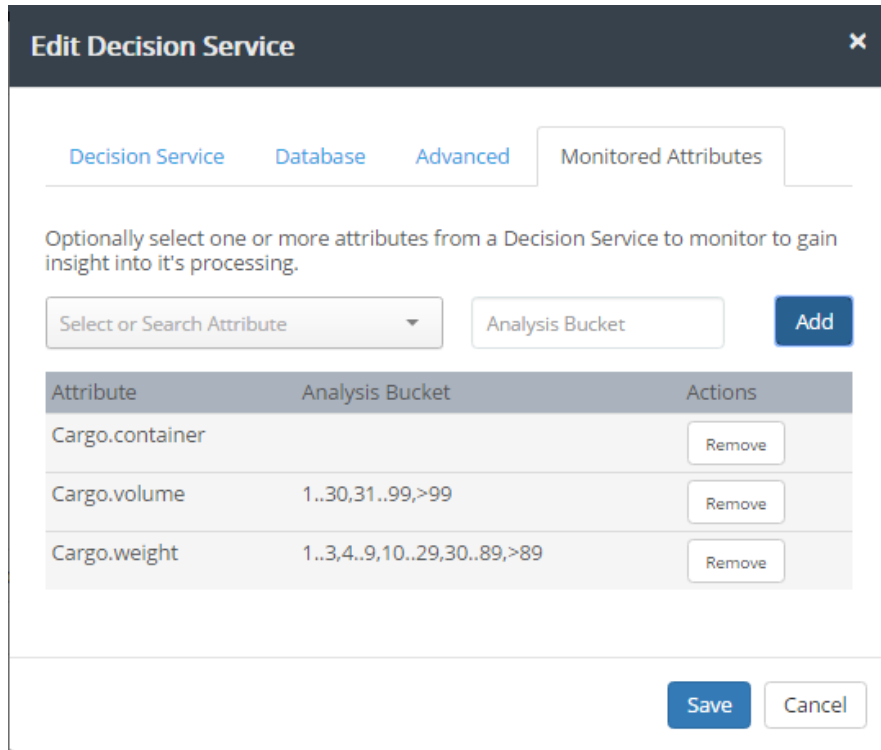
The Web Console lets you monitor the value distribution of one or more attributes in a deployed Decision Service. By choosing attributes to monitor, you can view the statistical breakdown of attribute values over the course of many Decision Service executions.

For example, the Ruleflow created in the [Tutorial: Basic Rule Modeling in Corticon Studio](#) *Basic Rule Modeling tutorial* reads integer values for `Cargo.volume` and `Cargo.weight` in the request, and assigns a text value to the attribute `Cargo.container`. To monitor these attributes, select the name in the **Monitored Attribute** dialog, enter comma-separated values or value ranges in the **Analysis Buckets** entry area, and then click **Add**.

When you set *bucket* ranges of values, you can analyze categories of data. Bucketing is useful when a wide range of numeric or date data is possible. For this example, the three buckets for `Cargo.volume` are 1 to 30 kilos, 31 to 99 kilos, and greater than 99.

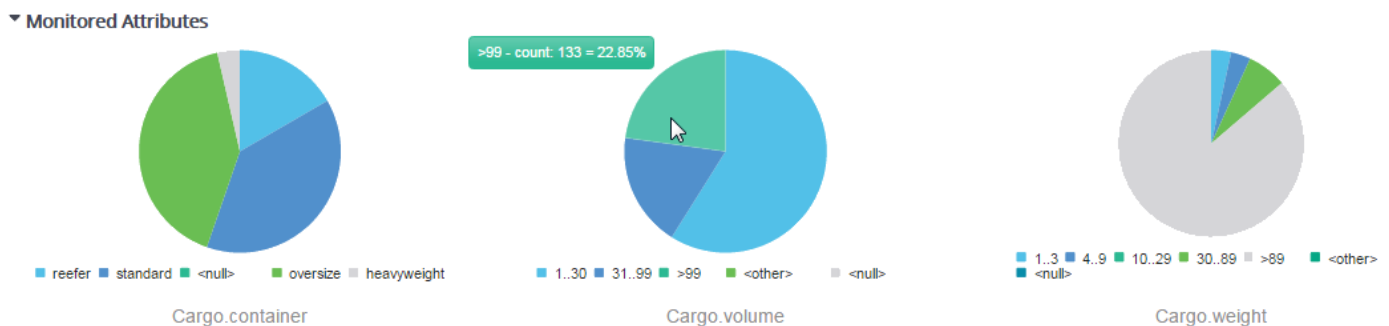
Entering no values can be useful for string values, especially when there is a small set of values defined in a Custom Data Type (such as `Cargo.container` in this example), or there is small set of known values, such as risk ratings `high`, `medium`, `low`.

The monitored attributes in this example are listed as shown:



Click **Save** to enable your selections.

In this example, the integer values are examined across narrower ranges than the rules, perhaps as a study to see whether new container categories should be considered. The results of attribute monitoring are visualized as follows:



# Batch Configurations

Corticon Web Console lets you connect to remote Web Console servers that in turn connect to managed Corticon Servers where deployed Decision Services are defined that integrate with data sources. When these Decision Services use defined SQL batch queries linked through the Datasource, you can define batch configurations and run batch jobs.

As a result, you can ensure that high-volume rules-based processing occurs on a specified schedule.

## Add Batch Configurations

To add batch configurations:

1. Connect to the Web Console server where you maintain batch configurations.

2. Click the **Batch configurations** button:



3. Click **+ New Batch Configuration**:

+ New Batch Configuration

4. The **New Batch Configuration** dialog box opens:

where:

- **Name** - Unique text that you want to use to refer to this configuration
- **Description** - Optional supporting text for the configuration

- **Decision Service** - List of managed, deployed Decision Services that have at least one component that has batch queries in its connected database.
- **Datasource** - The name of the Datasource connection that the Decision Service uses, as assigned in the Vocabulary.

For example, in an export configuration file named `myConfig.xml` where the first few lines are...

```
...
<decisionService>
  <datasources>
    <database useForQueryService="true" name="Patient Data">
      <connection-url>jdbc:progress:sqlserver://localhost:1433;
        databaseName=PatientRecords</connection-url>
```

...the **Datasource** value is `Patient Data`.

- **Query** - The name of the batch query stored in one of the Decision Service's connected databases

5. Click to access the **Advanced Properties** tab:

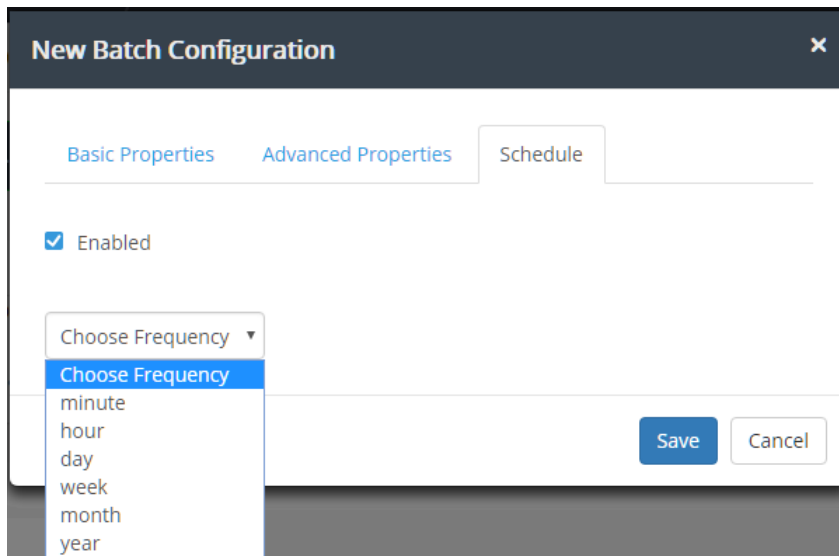
where:

- **Number of ID's per Fetch** - Number of Ids that will be retrieved by each Datasource Fetch. Default value is 1000.
- **Entities per Payload** - Number of entities that will be added to each payload sent to the Corticon Server execute method. Default value is 1.
- **Number of Processing Threads** - The number of execution threads the Corticon Server will spawn when executing the batch. The Default value is the number of cores on the Corticon Server's machine.
- **Log Path** - The folder that will store the logs produced for this batch configuration on the server that runs the batch process. Default location is `[CORTICON_WORK_DIR]\logs\ .`

The log file name is set as `DecisionServiceName(Version)_Threads_Timestamp.log`. For example, `PatientUpdate(1.2)_4_1515014748084.log`

- **Logging enabled checkbox** - To the right of the **Log Path** entry, the checkbox lets you decide whether to do logging for this batch configuration.

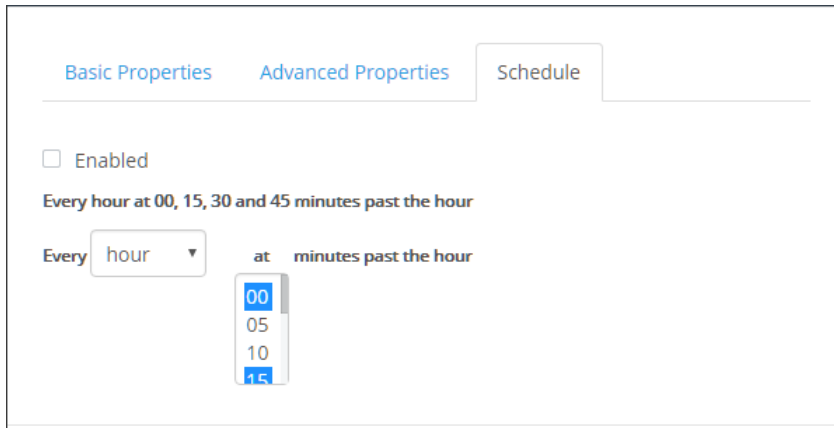
6. Click to access the **Schedule** tab:

The screenshot shows a dialog box titled "New Batch Configuration" with a close button (X) in the top right corner. It has three tabs: "Basic Properties", "Advanced Properties", and "Schedule". The "Schedule" tab is selected. Below the tabs, there is a checkbox labeled "Enabled" which is checked. Below that is a dropdown menu labeled "Choose Frequency" with a downward arrow. The dropdown menu is open, showing the following options: "Choose Frequency", "minute", "hour", "day", "week", "month", and "year". At the bottom right of the dialog box, there are two buttons: "Save" and "Cancel".

where:

- **Enabled** - Chooses to repeat the batch process with the frequency you specify.
- *Choose Frequency:*
  - **minute** - Once every minute.
  - **hour** - At specified minute past every hour.
  - **day** - At the specified time of every day.
  - **week** - At specified week day at the specified time of that day.
  - **month** - At specified day every month at the specified time of that day.
  - **year** - At specified day and month every year at the specified time of that day.

**Note:** On most of the frequency options, you can use Control+click to choose multiple values, as illustrated:




The screenshot shows a web interface with three tabs: 'Basic Properties', 'Advanced Properties', and 'Schedule'. The 'Schedule' tab is active. It contains a checkbox labeled 'Enabled' which is unchecked. Below it, the text reads 'Every hour at 00, 15, 30 and 45 minutes past the hour'. Underneath, there is a configuration area with 'Every' followed by a dropdown menu set to 'hour', the word 'at', and 'minutes past the hour'. A small calendar-style popup is open over the 'at' field, showing a list of minutes: 00, 05, 10, and 15. The '15' option is highlighted with a blue background, indicating it is selected. The '00' option is also highlighted with a blue background, suggesting a multi-select state where both 00 and 15 are chosen.

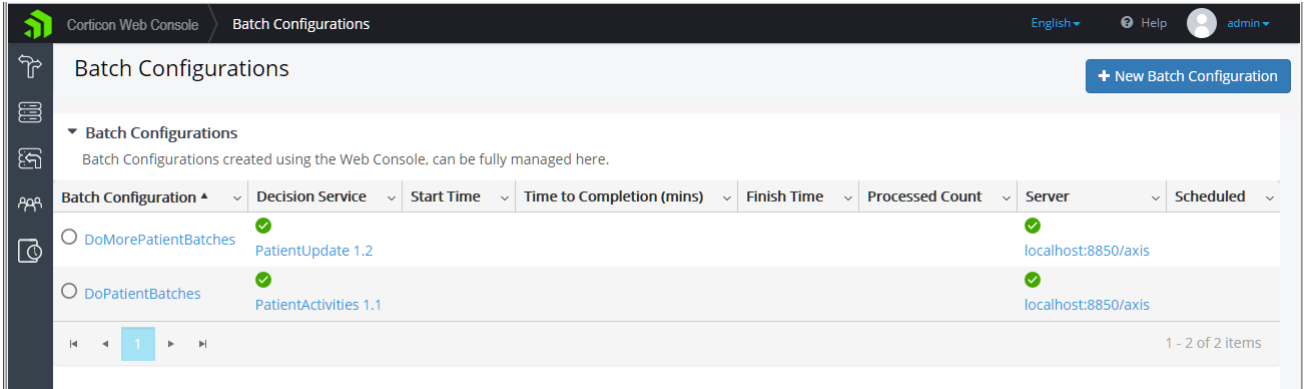
7. Click **Save**.

## Edit Batch Configurations

To maintain batch configurations:

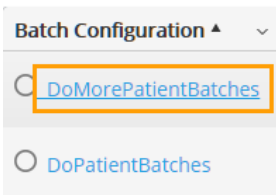
1. Connect to the Web Console server where you maintain batch configurations.

2. Click the **Batch configurations** button:  The Batch Configuration page opens and displays the current batch configurations, as illustrated:



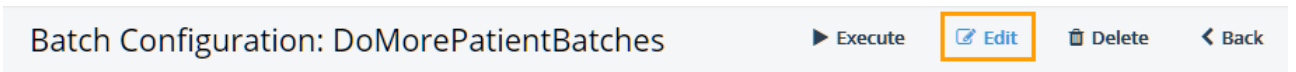
Batch Configuration	Decision Service	Start Time	Time to Completion (mins)	Finish Time	Processed Count	Server	Scheduled
<input type="radio"/> DoMorePatientBatches	PatientUpdate 1.2					localhost:8850/axis	
<input type="radio"/> DoPatientBatches	PatientActivities 1.1					localhost:8850/axis	

3. Click the Batch Configuration name you want to edit, as illustrated:



Batch Configuration
<input type="radio"/> DoMorePatientBatches
<input type="radio"/> DoPatientBatches

4. On the Details page, click **Edit**, as shown:



Batch Configuration: DoMorePatientBatches	Execute	Edit	Delete	Back


5. The **Edit Batch Configuration** dialog box opens.

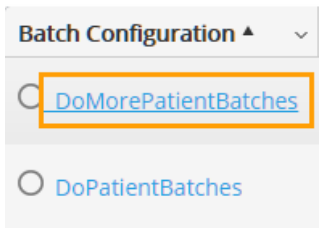
6. Follow the steps for the dialog box as described in [Add Batch Configurations](#) on page 41

7. Click **Save**.

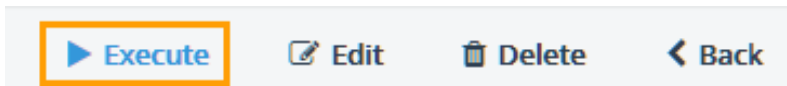
## Run Batch Configurations

To run a batch configuration:

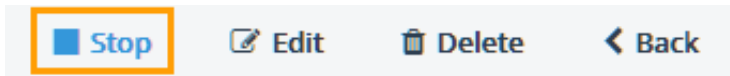
1. Connect to the Web Console server where you maintain batch configurations.
2. Click the **Batch Configurations** button:  The Batch Configuration page opens and displays the current batch configurations:
3. Click the Batch Configuration name you want to edit, as illustrated:



4. On the Details page, click **Execute**, as shown:



**Note:** When execution is running, you can terminate it by clicking **Stop**, as shown:



The job statistics show the time and counts of the most current run, as shown:

### ▼ Statistics

Start Time:	Jan 3, 2018 1:12:22 PM	Processed Count:	4
Finish Time:	Jan 3, 2018 1:12:23 PM	Retrieved Count:	4
Running Time:	0 mins		
Time to Completion:	0 mins		

The logs are produced on the server that ran the deployed Decision Service at the location you specified or the default location [CORTICON\_WORK\_DIR]\logs. The filename for each run is `DecisionServiceName(Version)_Threads_Timestamp.log`

**Note:** For an example of batch runs, see "Getting Started with Batch" in the *Data Integration Guide*.

# How to view the Activity Log

Corticon Web Console maintains a log of its activities. The log includes:

- User actions such as deploying or undeploying Decision Services and creating or modifying Applications and Servers.
- System events such as deployment failures and lost connections to Servers.

**To view the activity log:**

1. Connect to the Web Console server where you want to view the Activity Log.

2. Click the **Activity Log** button:




The Activity Log page opens and displays the log in a three-column table:

The screenshot shows the Corticon Web Console Activity Log page. The page header includes the Corticon logo, 'Corticon Web Console', 'Activity Log', 'English', 'Help', and 'admin'. The main content area has a 'Configuration' button and filter options for User, Component, Action, and Status, all set to 'All'. There is also a 'Filter' button and a 'Page Size' dropdown set to '10'. The log entries are displayed in a table with the following data:

Time	User Name	Message
Oct 18, 2016 2:20:48 PM		✔ Connection to "NBBEDGSAINTMA5:8850/axis" established
Oct 18, 2016 2:20:48 PM		✔ Connection to "NBBEDGSAINTMA1:8850/axis" established
Oct 18, 2016 2:20:47 PM		✔ Connection to "localhost:8850/axis" established
Oct 18, 2016 2:05:06 PM	admin	✔ Application "nbbedmciel2" successfully removed
Oct 18, 2016 2:04:37 PM	admin	✔ Server "nbbedmciel2" successfully removed
Oct 18, 2016 2:04:36 PM	admin	✔ Decision Service "nbbedmciel2" successfully undeployed from "nbbedmciel2:8850/axis"
Oct 18, 2016 2:04:32 PM	admin	✔ Server "nbbedmciel2_2" successfully removed
Oct 18, 2016 2:04:21 PM		✘ Connection to "nbbedmciel2:8087/axis" failed
Oct 18, 2016 2:04:09 PM	admin	✔ Server nbbedmciel2_2 successfully added
Oct 18, 2016 2:04:09 PM	admin	✘ Connection to "nbbedmciel2:8087/axis" failed

At the bottom of the table, there is a pagination bar with buttons for 'First', 'Previous', '21', '22', '23', '24', '25', '26', '27', '28', '29' (selected), '30', 'Next', and 'Last'. Below the pagination bar, it says 'Page: 29 / 41'.

Some log messages, such as those relating to failed deployment of Decision Services, have additional information about the problem that is not displayed in the table. To view this information, hover over a *Failed* log message, and then click on the information button  at the end of that line. An alert opens with additional information on the issue.

You can filter the table to view a subset of the log messages. To do this, select the filters you want from the drop-down lists, and then click **Filter**. For example, to view all failed Decision Services deployments by a user, select the username from the **User** drop-down, select **Decision Service** in **Component**, select **Deploy** in **Action** and finally, select **Failed** in the **Status** drop-down. You can also add a date range to the filter to narrow the information to only log messages recorded between specified dates.

The screenshot shows the 'Activity Log' page in the Corticon Web Console. At the top, there are navigation elements like 'Corticon Web Console', 'Activity Log', 'English', 'Help', and a user profile 'admin'. Below this, there are filter controls for 'User', 'Component', 'Action', and 'Status'. The 'Status' dropdown is currently set to 'Failed' and is open, showing options for 'All', 'Successful', and 'Failed'. There is also a 'Filter' button and a 'Page Size' dropdown set to '10'. Below the filters is a table with columns 'Time', 'User Name', and 'Message'. The table contains 11 rows of log entries, all of which are failed. The messages include connection failures and execution failures of Decision Service 'Cargo'. At the bottom of the table, there are pagination controls: 'First', 'Previous', '1', '2', 'Next', 'Last'. Below the pagination, it says 'Page: 1 / 2'. There is also a 'Configuration' button in the top right corner.

Time	User Name	Message
Oct 18, 2016 5:12:48 PM		Connection to "NBBEDGSAINTMA5:8850/axis" failed
Oct 18, 2016 5:09:33 PM		Connection to "NBBEDGSAINTMA1:8850/axis" failed
Oct 18, 2016 2:04:21 PM		Connection to "nbbedmciel2:8087/axis" failed
Oct 18, 2016 2:04:09 PM	admin	Connection to "nbbedmciel2:8087/axis" failed
Oct 18, 2016 2:04:06 PM	admin	Execute of Decision Service "Cargo" on "NBBEDGSAINTMA1:8850/axis" Failed
Oct 18, 2016 2:04:06 PM	admin	Execute of Decision Service "Cargo" on "NBBEDGSAINTMA1:8850/axis" Failed
Oct 18, 2016 2:04:06 PM	admin	Execute of Decision Service "Cargo" on "NBBEDGSAINTMA1:8850/axis" Failed
Oct 18, 2016 2:04:06 PM	admin	Execute of Decision Service "Cargo" on "NBBEDGSAINTMA1:8850/axis" Failed
Oct 18, 2016 2:04:06 PM	admin	Execute of Decision Service "Cargo" on "NBBEDGSAINTMA5:8850/axis" Failed
Oct 18, 2016 2:04:06 PM	admin	Execute of Decision Service "Cargo" on "NBBEDGSAINTMA1:8850/axis" Failed

The Web Console maintains this log for a configurable period of time. This setting is visible only to Web Console Administrators. To know more about configuring the Activity Log, see the topic [Configure the Activity Log](#) on page 53.

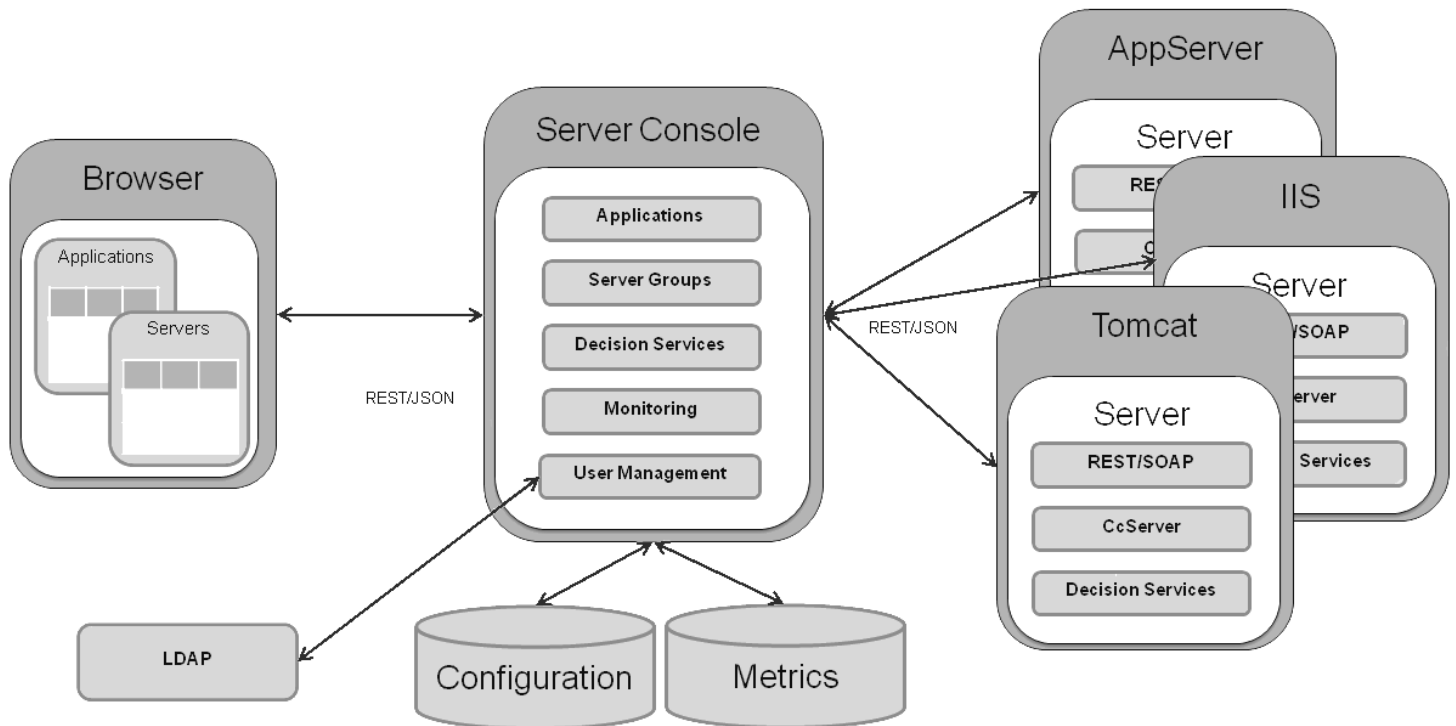
---

# Administrator Guide

---

**Architecture Overview** - The Web Console is a separate web application (`corticon.war`) from the Corticon Server (`axis.war`), deployable to either the same or separate application server as the Corticon Server.

When managing a group of Corticon Servers the recommended practice is to deploy the Web Console to a separate application server as depicted in this diagram:



Key aspects of this diagram:

- There is a single application server hosting the Web Console and three application servers hosting Corticon Servers. The Web Console is agnostic to the application server hosting a Corticon Server, this includes a mix of Java and Corticon Server for .NETs.
- REST/JSON is used for communication between the browser and the Web Console and between the Web Console and the Corticon Server.
- The Web Console stores all configurations locally. This includes definition of server groups, applications, and Decision Services (including the EDS file).
- The Web Console stores historical metrics locally. A retention policy will be supported for determining how long to keep historical metrics.

### Installation

The Corticon Java and Corticon Server for .NET installers are in one installer, together with the new Web Console. See the *Corticon Installation Guide* for more information. See the Progress Software web page [Corticon Supported Platforms Matrix](#) to review the currently supported browsers, platforms and application servers.

For details, see the following topics:

- [User management](#)
- [Configure the Activity Log](#)
- [Configure auto logout](#)
- [Reset the administrator password](#)

## User management

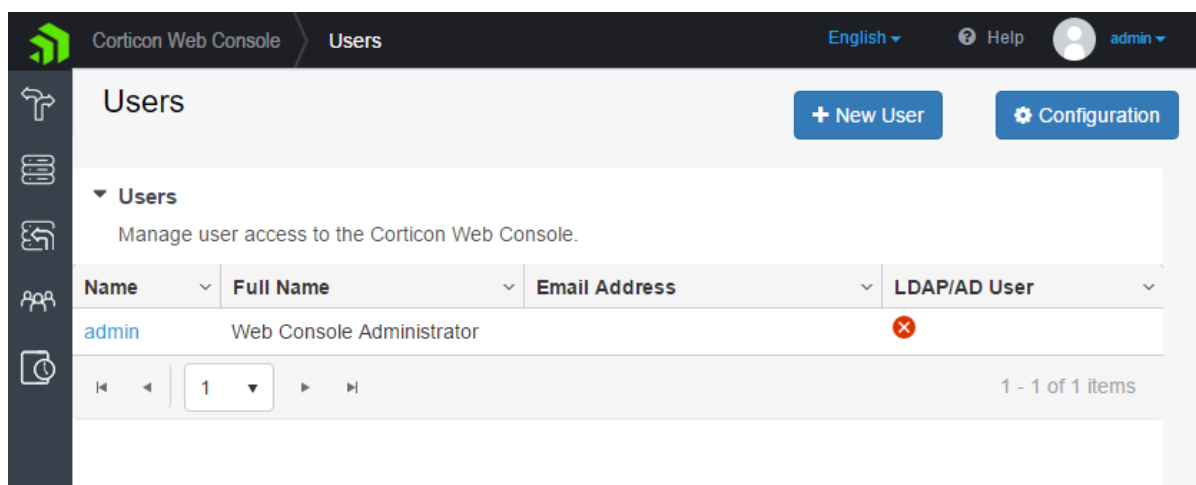
The Web Console provides secure access. The administrator (User Name `admin`) is a preset user that cannot be deleted. You can change the administrator's password -- that's a task you should do as soon as you get started with the Web Console and take the administrator's role.

The administrator is the only user that can access user management to create, edit, and delete other users. Note that the case matters in the user name and password.

### To display users:

1. Connect to the Web Console server as `admin` where you want to manage users.

2. Click the **Users** button:  The Users page opens:



Corticon Web Console > Users English Help admin

## Users

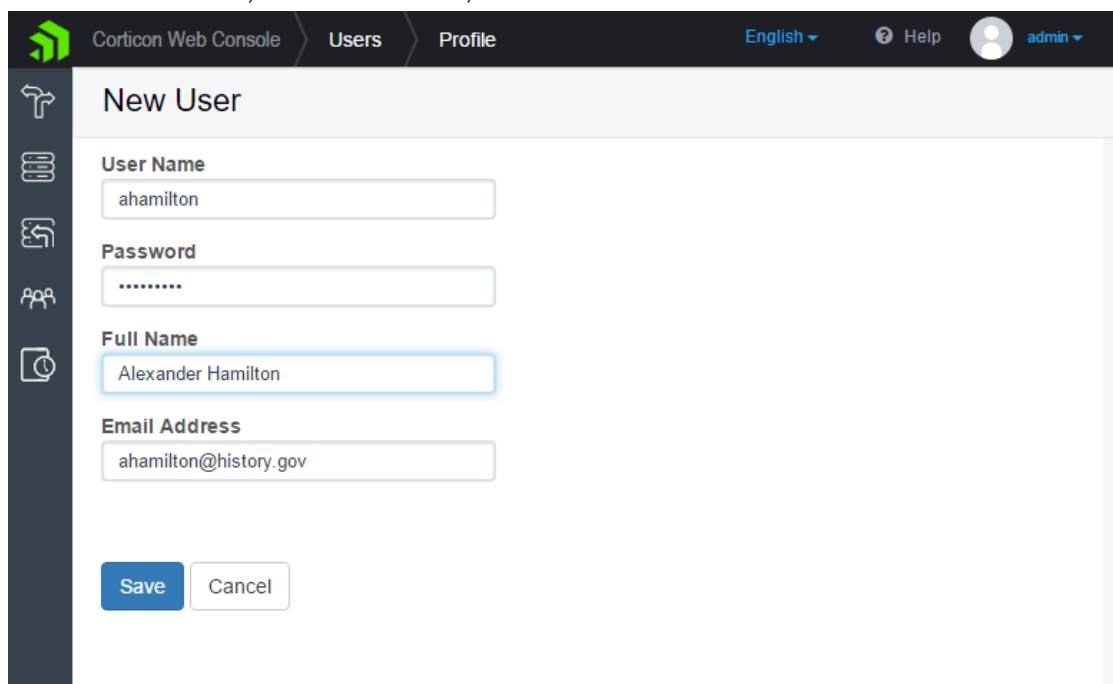
+ New User Configuration

▼ Users  
Manage user access to the Corticon Web Console.

Name	Full Name	Email Address	LDAP/AD User
admin	Web Console Administrator		✘

1 - 1 of 1 items

To create new users, click **+ New User**, and then enter the user information and click **Save**:



Corticon Web Console > Users > Profile English Help admin

## New User

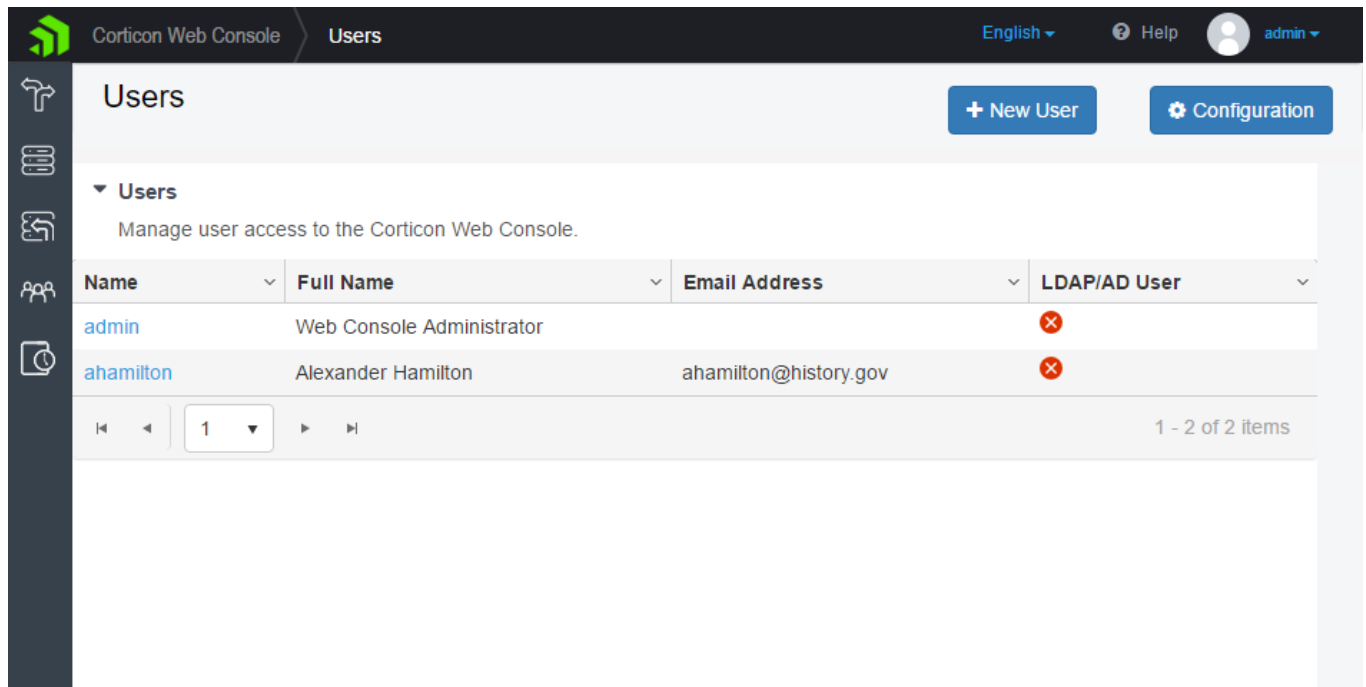
User Name  
ahamilton

Password  
.....

Full Name  
Alexander Hamilton

Email Address  
ahamilton@history.gov

Save Cancel



## How to use LDAP for Web Console authentication

You can also set up LDAP authentication, if business needs require your users to be authenticated through an LDAP server. After LDAP authentication is set up, LDAP users who log in to the Web Console are added to the **Users** page. LDAP users are differentiated from other users by the **LDAP/AD** annotation. Note that while LDAP users can be deleted from the **Users** page, their details cannot be modified in the Web Console.

To configure LDAP authentication, edit the file `CorticonServerConsoleConfig.groovy` located in `[CORTICON_WORK_DIR]\etc`.

Uncomment all property lines in this file and enter values for the first four properties. Here is an example:

```
/*Configure the manager distinguished name based on your ldap configuration*/
grails.plugin.springsecurity.ldap.context.managerDn = 'uid=admin,ou=system'

/*Ldap password*/
grails.plugin.springsecurity.ldap.context.managerPassword = 'secret'

/*Ldap server url */
grails.plugin.springsecurity.ldap.context.server = 'ldap://localhost:10389'

/*Ldap search base*/
grails.plugin.springsecurity.ldap.search.base = 'o=mojo,ou=groups'
```

To map Web Console Admin and User roles to LDAP user groups, specify the user group names in the `ldap{ }` section at the bottom as shown. Use commas to define multiple user groups for each role.

```
corticon
{
  ldap {
    admin = "Admin,DBAdmin"
    users = "User,Analyst,Developer"
    authenticateUserswithGroups = "false" //This flag is used to
    authenticate the Admin and User groups listed above. If the flag is
    false, and the group is not listed above, the user will be defaulted
    to user role.
  }
}
```

After setting these properties, save the file and restart Corticon Server. LDAP users can log then in to Web Console using their LDAP user credentials. Once an LDAP user logs in, they are added to the **USERS** page in Web Console.

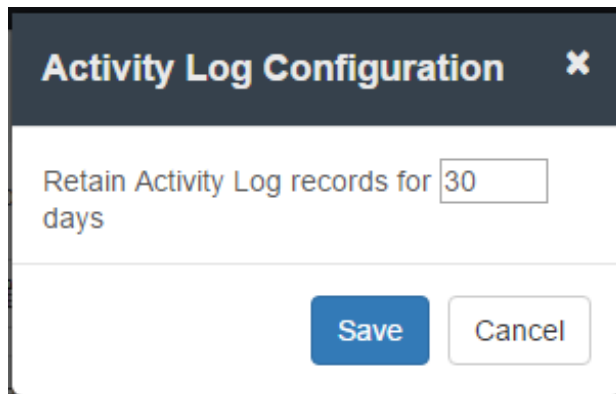
**Note:** Setting up LDAP authentication adds LDAP users to the Web Console user base. You can add other users in the **USERS** page and have them access Web Console using their Web Console user credentials.

## Configure the Activity Log

Corticon Web Console maintains a log of its activities. The log includes:

- User actions such as deploying or undeploying Decision Services and creating or modifying Applications and Servers.
- System events such as deployment failures and lost connections to Servers.

A Web Console Administrator can view the Activity Log as well as configure the duration for which Corticon Web Console maintains log records. To view the Activity Log, click **ACTIVITY LOG** on the left pane. To configure the duration for which Web Console keeps log records, click **Configuration** on the Activity Log page and set the number of days for which Web Console maintains log records.



To know more about viewing and filtering the Activity Log, see the topic [How to view the Activity Log](#) on page 47.

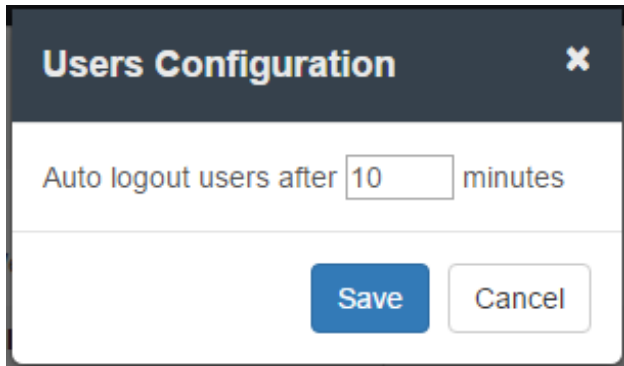
---

**Note:** The Web Console Activity Log is different from a Server log, which logs user actions, system events, and other information for a specific instance of Corticon Server based on configurable log levels.

---

## Configure auto logout

As part of user management, you can define a period of inactivity (in minutes) after which a user is automatically logged out of the Web Console. To configure this setting, click **Configuration** on the **Users** page. In the **User Configuration** dialog box, enter the duration of inactivity, as shown below:



## Reset the administrator password

If the login password of the Web Console administrative user (`admin`) is lost, Corticon provides a way to reset the password to the default (also `admin`).

To reset the administrator's password:

1. Stop the Corticon Server that is running the Web Console.
2. Select **Start > Progress > Corticon 6.x Command Prompt**.
3. Enter `set JAVA_OPTS=-DCORTICON_RESET_ADMIN_PASSWORD=true`.
4. Enter `Server\tomcat\bin\startup.bat`.

Corticon Server starts and resets the administrator's password.

After completing these steps, you can connect the Web Console and log in with the default administrator credentials, user `admin`, password `admin`. It is good idea to immediately replace the default password with your preferred administrator password.

This procedure applies to the application server that is installed by Corticon Server, Apache Tomcat. You can perform similar steps for other supported application servers and platforms. Consult your application server documentation for how to pass the JVM system property `CORTICON_RESET_ADMIN_PASSWORD` to the server.

---

**Note:** Do not set this property in startup scripts as it will reset the password on each startup. This should be only done only when the password needs to be reset. Subsequent launches of Corticon Server and the Web Console should use the normal startup procedures.

---