



## **Tutorial: Deploying a Progress Corticon Decision Service as a Web Service for Java**



# Copyright

---

© 2019 Progress Software Corporation and/or one of its subsidiaries or affiliates. All rights reserved.

These materials and all Progress® 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.

Corticon, DataDirect (and design), DataDirect Cloud, DataDirect Connect, DataDirect Connect64, DataDirect XML Converters, DataDirect XQuery, DataRPM, Deliver More Than Expected, Icenium, Kendo UI, Kinvey, NativeScript, OpenEdge, Powered by Progress, Progress, Progress Software Developers Network, Rollbase, SequeLink, Sitefinity (and Design), Sitefinity, SpeedScript, Stylus Studio, TeamPulse, Telerik, Telerik (and Design), Test Studio, and WebSpeed 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, DataDirect Autonomous REST Connector, DataDirect Spy, SupportLink, DevCraft, Fiddler, JustAssembly, JustDecompile, JustMock, Kinvey Chat, NativeChat, NativeScript Sidekick, OpenAccess, ProDataSet, Progress Results, Progress Software, ProVision, PSE Pro, SmartBrowser, SmartComponent, SmartDataBrowser, SmartDataObjects, SmartDataView, SmartDialog, SmartFolder, SmartFrame, SmartObjects, SmartPanel, SmartQuery, SmartViewer, SmartWindow, 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. Any other marks contained herein may be trademarks of their respective owners.

**Updated: 2019/07/31**



# Table of Contents

**Tutorial - Deploying a Progress Corticon Decision Service as a Web Service for Java.....7**

**Setting up the tutorial.....9**

**Preparing the Vocabulary.....13**

**Deploying the Decision Service .....17**

**Testing the Decision Service.....27**



## **Tutorial - Deploying a Progress Corticon Decision Service as a Web Service for Java**

---

Once the rules in a Ruleflow are modeled, analyzed and tested, they are ready to be deployed to Progress Corticon Server as a Decision Service.

There are a number of ways you can deploy a Decision Service. You can deploy Decision Service in-process or as a Web Service for either Java or .NET. In this tutorial, you will learn how to deploy a Decision Service as a Web Service for Java.

Corticon Server is a set of core Java classes that comprises all the functionality required to host and manage Decision Services. These Java classes are deployed on an application server. For this tutorial, we will use the default application server, Progress Application Server.

To deploy a Decision Service as a Web Service, you need to perform the following tasks:

1. Prepare Vocabulary for deployment
2. Deploy the Ruleflow as a Decision Service to Corticon Server
3. Test if the Decision Service can be accessed as a Web Service

In this tutorial, you will learn how to perform these tasks.

This tutorial is designed for hands-on use. We recommend that you follow along in Corticon Studio and Corticon Server, using the instructions and illustrations that are provided.





## Setting up the tutorial

---

Before you work on deploying a Decision Service, you need to set-up your environment for this tutorial.

You must:

1. Install and start Corticon Studio
2. Install and start Corticon Server
3. Import a Sample Project into Studio

### Step 1: Install and start Corticon Studio

First, let's install the development environment, the Corticon Studio.

1. Download and run the Studio installer.
2. On the **Introduction** page of the installation wizard, click **Next**.
3. On the **License Agreement** page, select **I accept the terms of the License Agreement** and click **Next**.
4. On the pages that follow, retain the default settings and click **Next**.
5. Click **Install**. When the installation is finished, click **Done**.
6. On the **Start** menu, select **Progress > Corticon 6.0 Studio**.

### Step 2: Install and start Corticon Server

Now, let's install the deployment runtime environment, the Corticon Server.

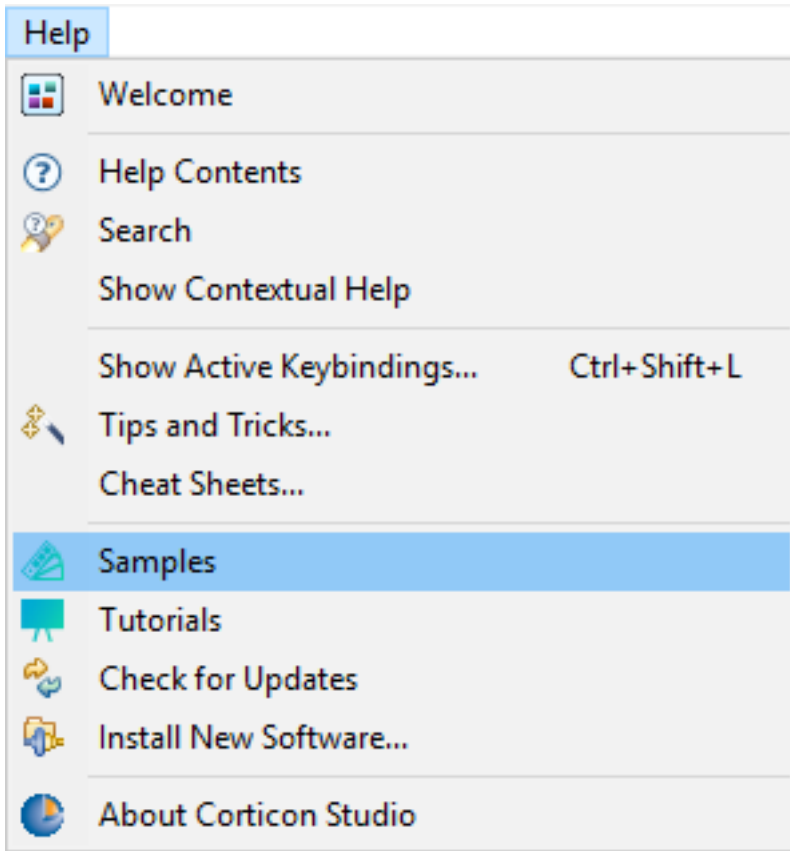
1. Download and run the Server installer.
2. On the **Introduction** page of the installation wizard, click **Next**.
3. On the **License Agreement** page, select **I accept the terms of the License Agreement** and click **Next**.

4. On the **Corticon Server Install and Work Directory** page, retain the default settings and click **Next**.
5. On the **Choose Server Components** page, select **Corticon Server for Java** and **Corticon Web Console** and click **Next**.
6. On the pages that follow, retain the default settings and click **Next**.
7. Click **Install**. When the installation is finished, click **Done**.
8. On the **Start** menu, select **Progress > Start Corticon 6.0 Server**.

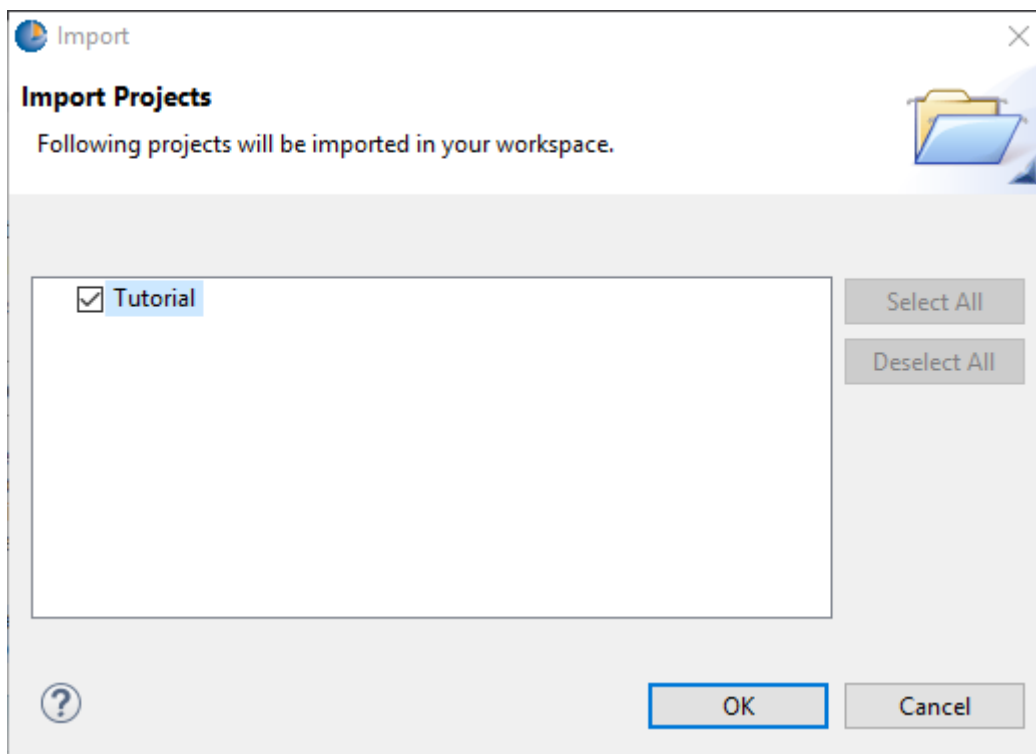
### Step 3 : Importing a Sample Project into Studio

The Corticon Studio installation includes several sample rule projects. We will use one for this tutorial. To import the sample:

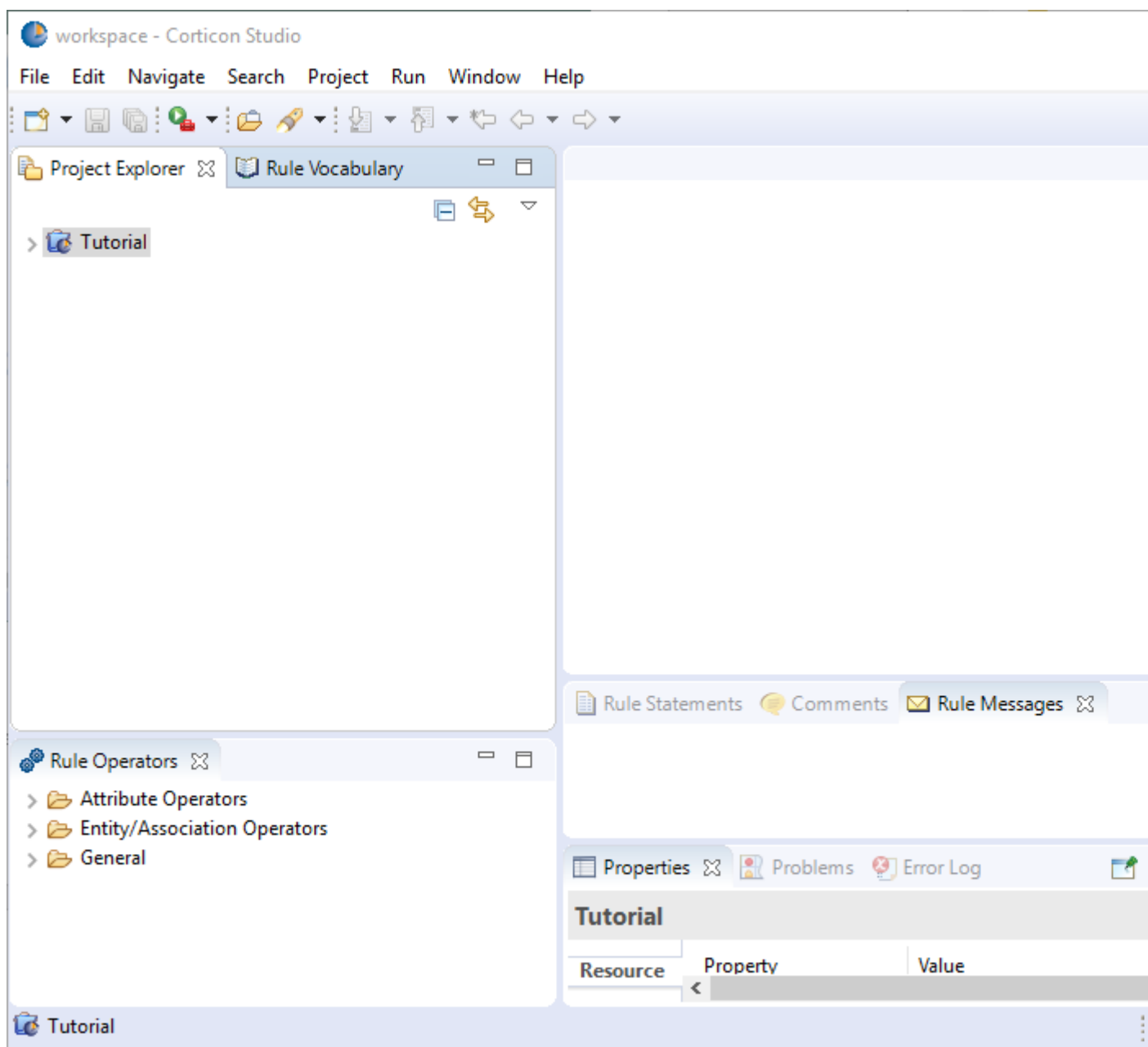
1. In Studio, choose **Help > Samples**.



2. On the Samples page, select **Tutorial** and click **Open**.
3. In the **Import Projects** window, choose **Tutorial**, and then click **OK**.



4. The sample rule project opens in Corticon Studio.



---

## Preparing the Vocabulary

---

When a Web Service client sends a request message to a Decision Service that is exposed as a Web Service, the message contains data enclosed within XML or JSON tags.

For Corticon to correctly process the request message, you must ensure that XML or JSON tag names map to entity and attribute names in the Corticon rule Vocabulary.

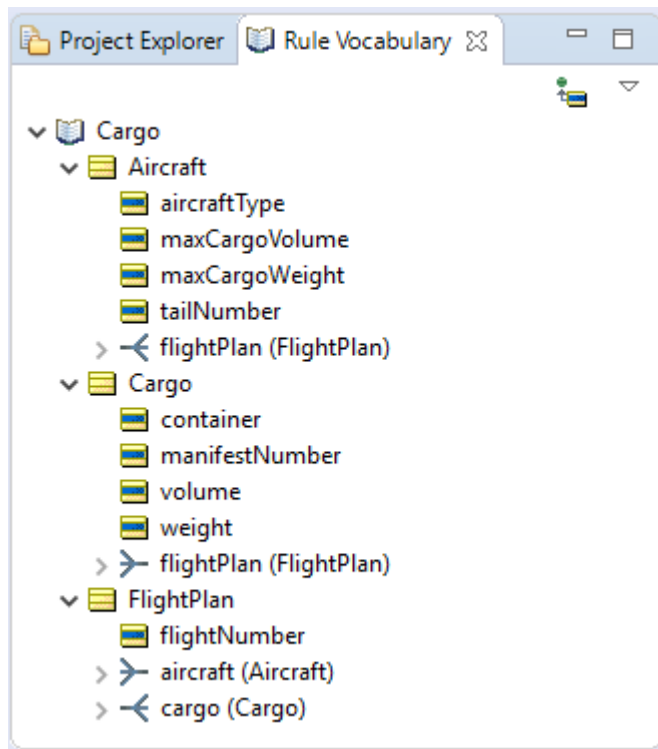
By default, Corticon Studio specifies the entity or attribute name as the XML Element Name. If XML or JSON tag names are different from the entity or attribute names, you need to prepare the Vocabulary by manually mapping them.

For this tutorial, our request messages uses JSON.

The JSON request messages looks like this:

```
{
  "__metadataRoot": {},
  "name": "Shipment",
  "Objects": [
    {
      "volume": 10,
      "container": null,
      "itemweight": 1000,
      "__metadata": {
        "#type": "Cargo",
        "#id": "Cargo_id_1"
      }
    }
  ]
}
```

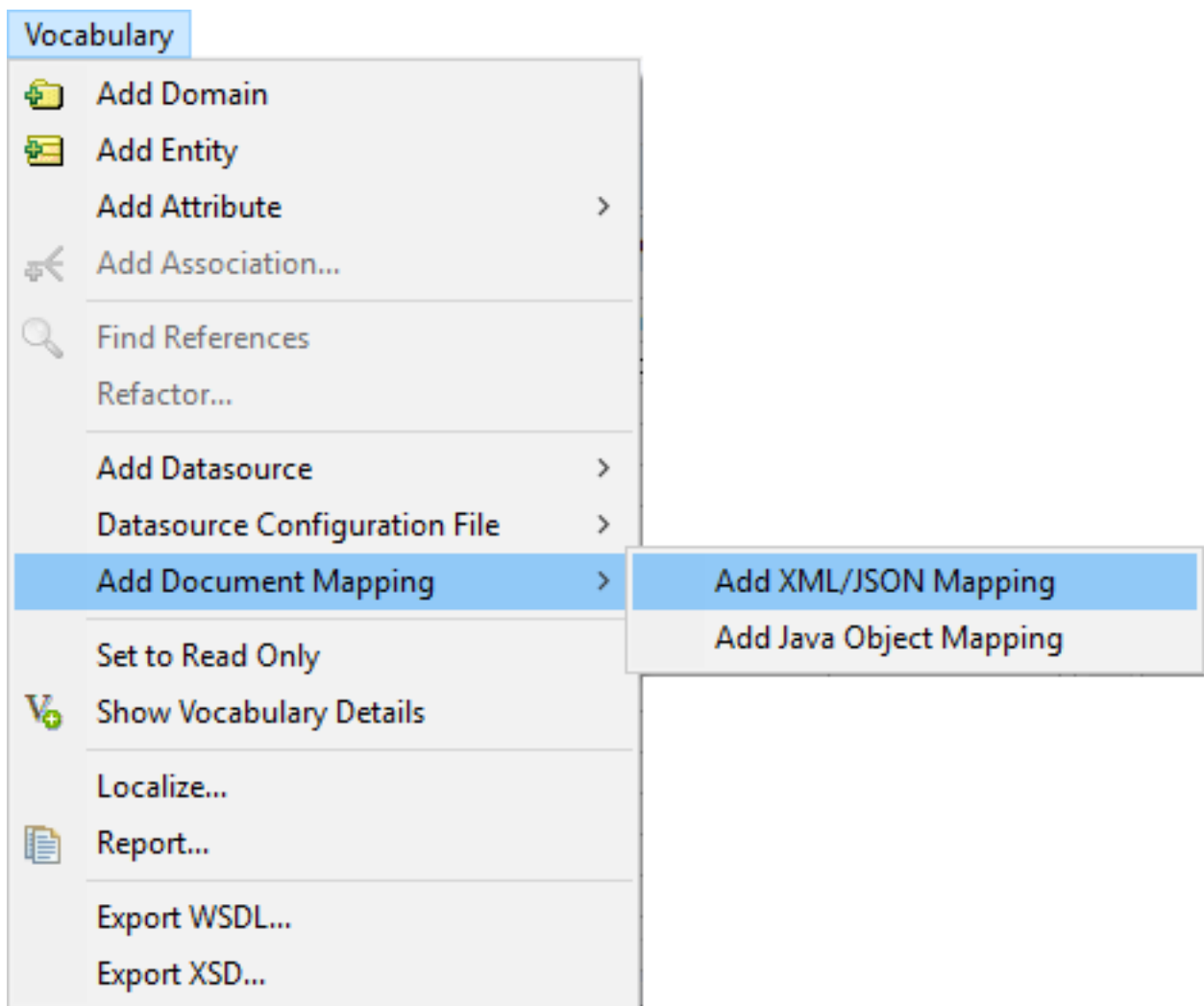
Our project Vocabulary looks like this:



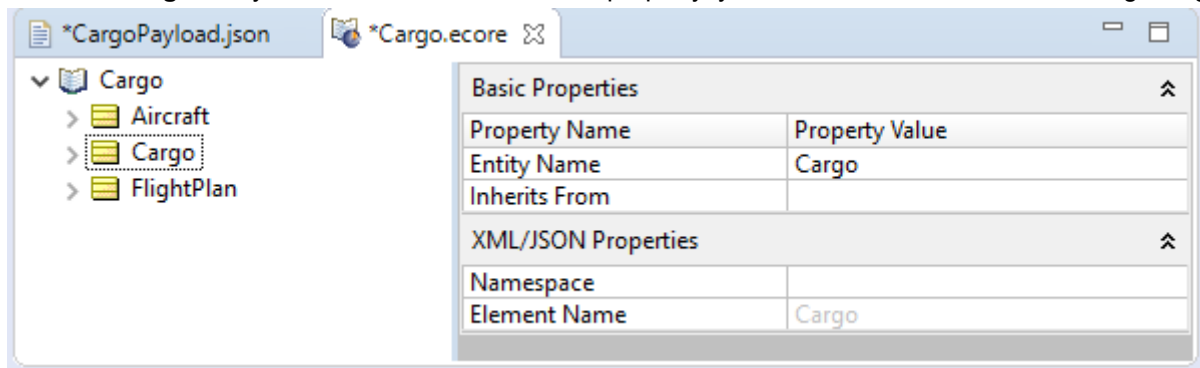
Notice that the request message contains two tag names 'Shipment' and 'itemWeight' that do not appear in our Vocabulary. They correspond to the Cargo entity and the Cargo.weight attribute.

You need to prepare the Vocabulary by manually mapping 'Shipment' to Cargo and **itemWeight** to **Cargo.weight**:

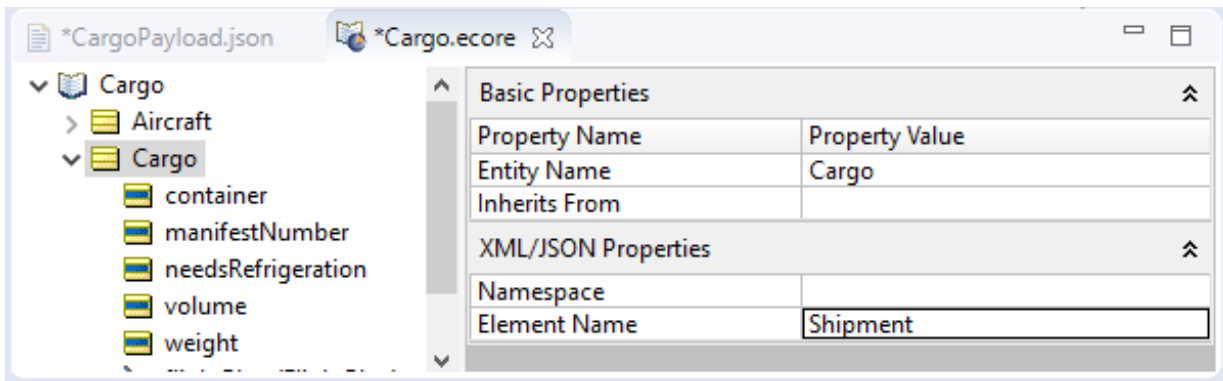
1. Open the Vocabulary file **Cargo.ecore** in the **Tutorial-Done** folder.
2. On the Vocabulary menu, choose **Add Document Mapping > Add XML/JSON Mapping**.



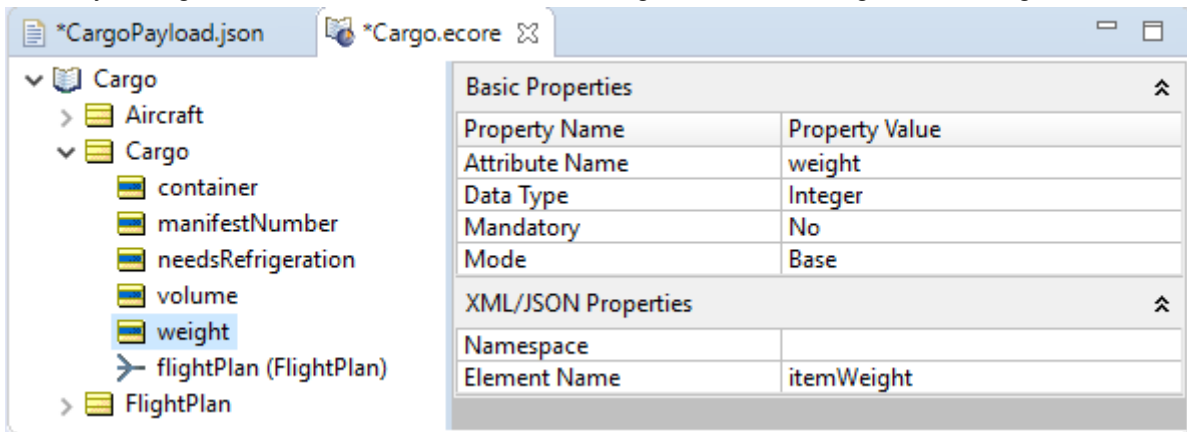
3. Click the **Cargo** entity. In the **XML Element Name** property, you can see the default name 'Cargo' in grey.



4. Change the default name by double-clicking the property value and then entering Shipment.



5. Similarly, change the **XML Element Name** for the weight attribute in Cargo to itemWeight.



6. Save the file.

You have now prepared the Vocabulary.



---

## Deploying the Decision Service

---

Now that you have prepared the Vocabulary, you can deploy rules created in Rulesheets, and prepared in a Ruleflow that uses the Vocabulary. You will package and deploy a Decision Service to Corticon Server.

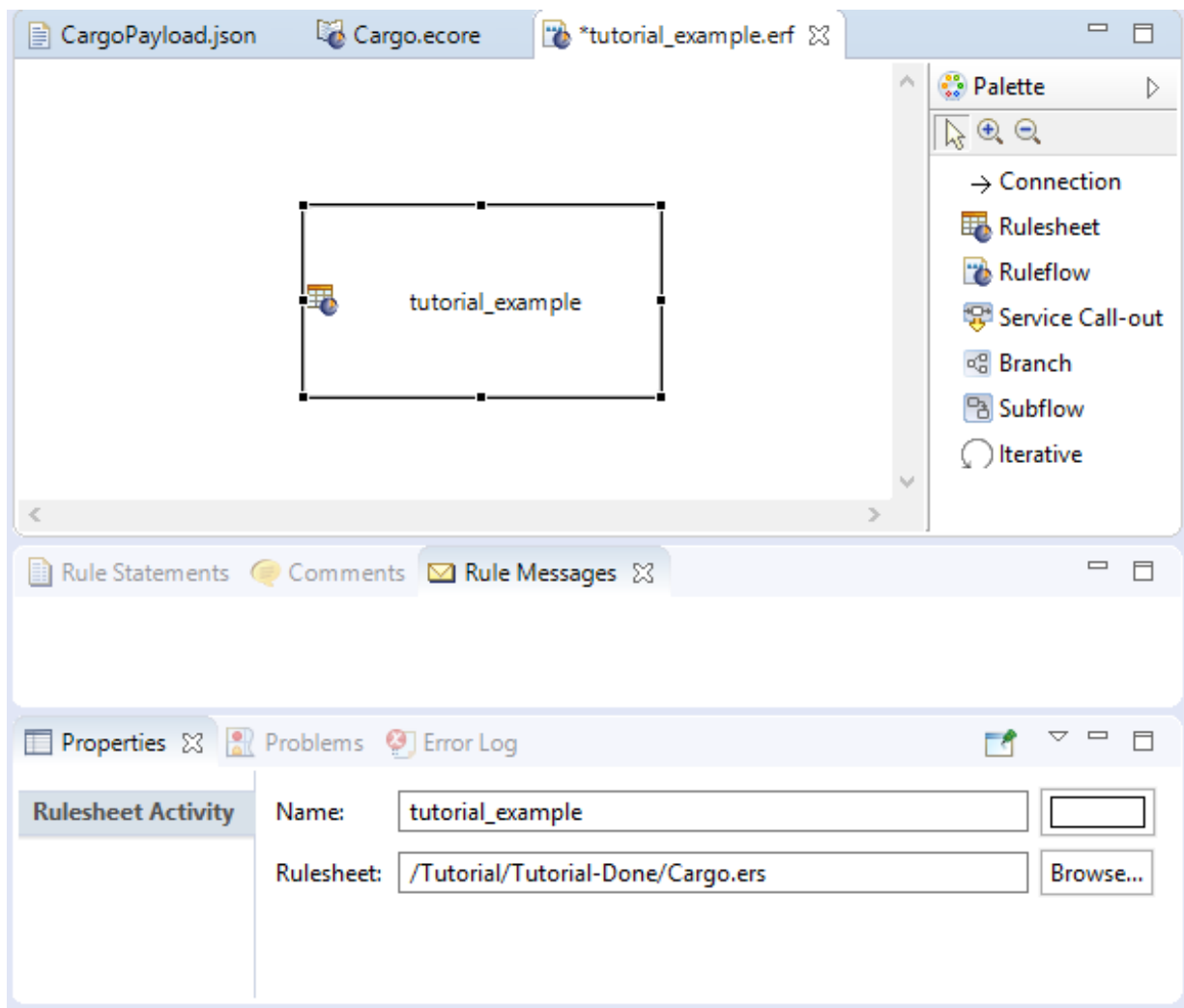
There are different ways to deploy a Ruleflow to Corticon Server. In this tutorial, you will package the Ruleflow in a pre-compiled EDS file and then deploy it to Corticon Server using the Web Console.

To learn about other methods of deployment, refer to the Deployment topics in the Corticon Information Hub.

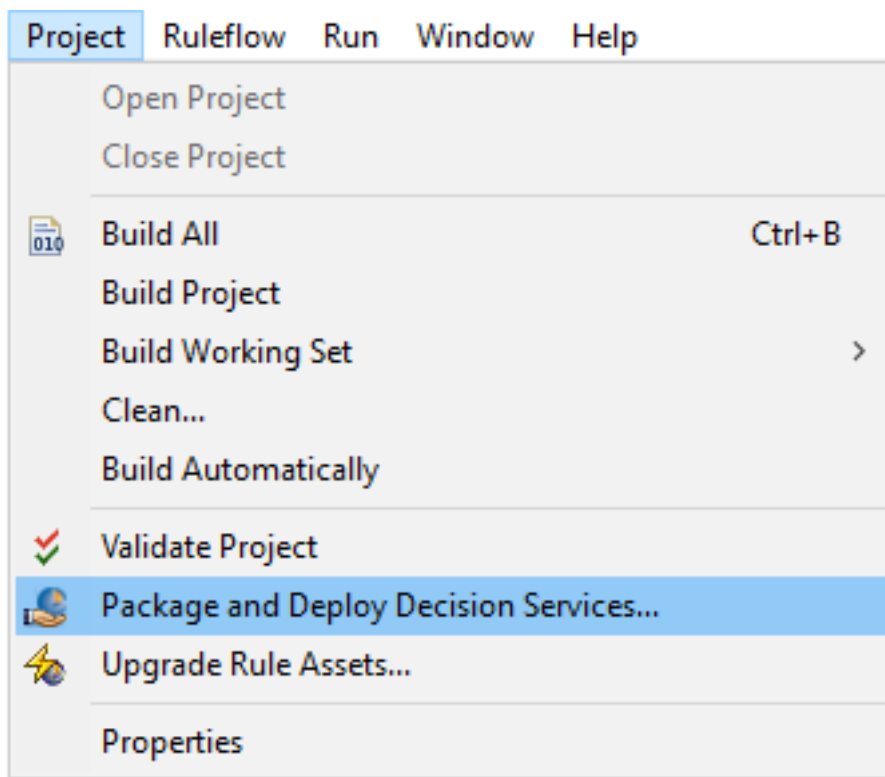
### Packaging the Ruleflow into an Decision Service file

Let's start by packaging the Ruleflow in Corticon Studio into a Decision Service, an EDS file.

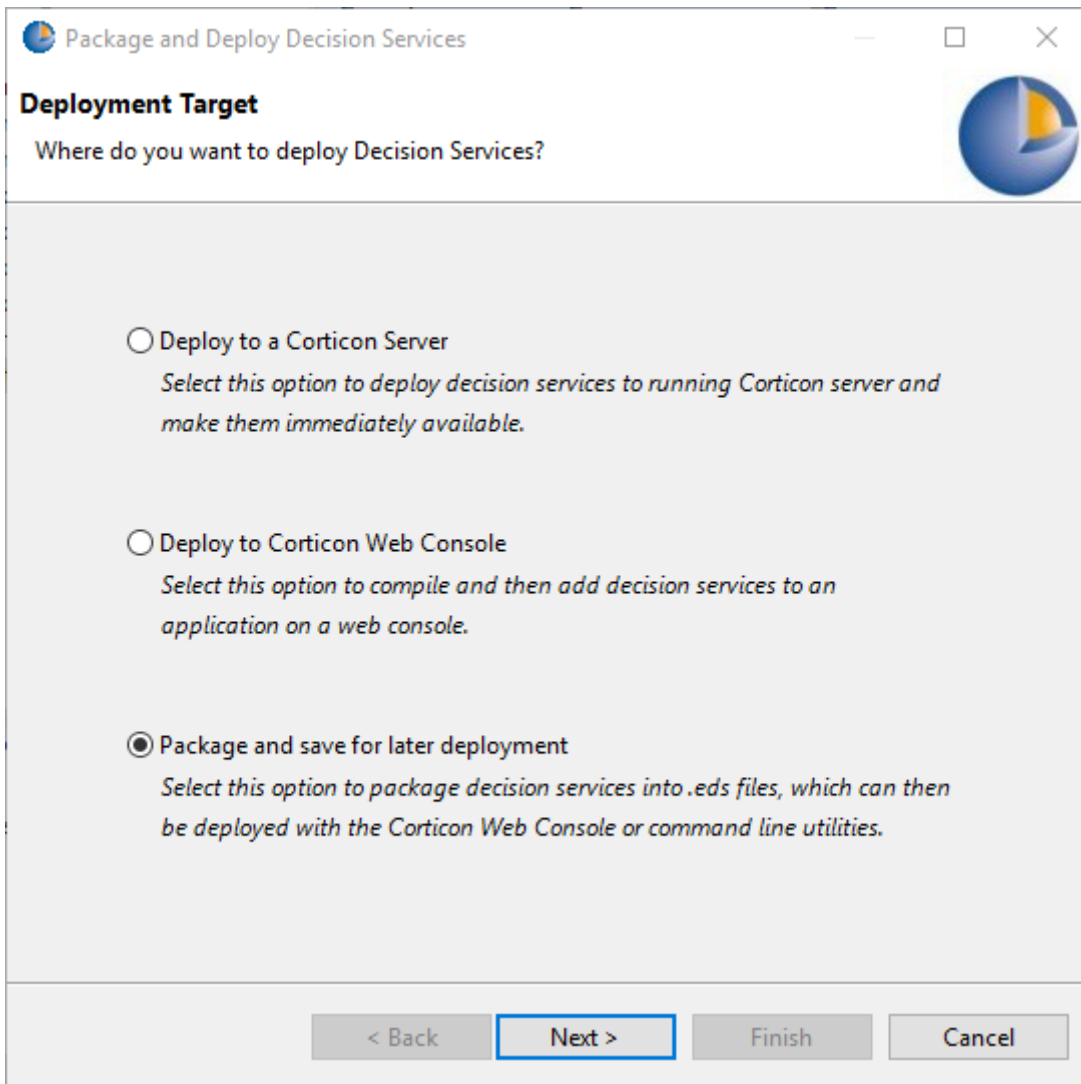
1. Open the **tutorial\_example.erf** file, and then click on the **tutorial\_example** box.



2. The Ruleflow has just one Rulesheet—Cargo.ers – which is renamed to tutorial\_example in the Ruleflow. When you create an EDS file, related Ruleflow, Rulesheet, and the Vocabulary are compiled into a package that can be deployed to run rules on Corticon Server.
3. To package the Ruleflow, select **Project > Package and Deploy Decision Services**.



4. In the **Deployment Target** window, choose **Package** and save for later deployment and click **Next**.



5. In the **Package Ruleflows** screen:
  - a. Select the **tutorial\_example** Ruleflow.
  - b. Keep the default directory or specify a different location to save the EDS file by clicking the **Browse** button.
  - c. Click **Next**.

Package and Deploy Decision Services

### Package Ruleflows

Select the ruleflows to be packaged for later deployment

Select All Deselect All


	Decision Service Name	Version	Database Mode	Ruleflow
<input checked="" type="checkbox"/>	tutorial_example	0.16	None	/Tutorial/Tutorial-Done/tutorial_example.erf

To directory: C:\\_60x\workspace Browse

< Back Next > Finish Cancel

A message indicates that the Ruleflow was successfully compiled into an EDS file and saved in the location you specified.

Compilation Success

 Ruleflows have been successfully compiled into .eds files in directory C:\\_60x\workspace.

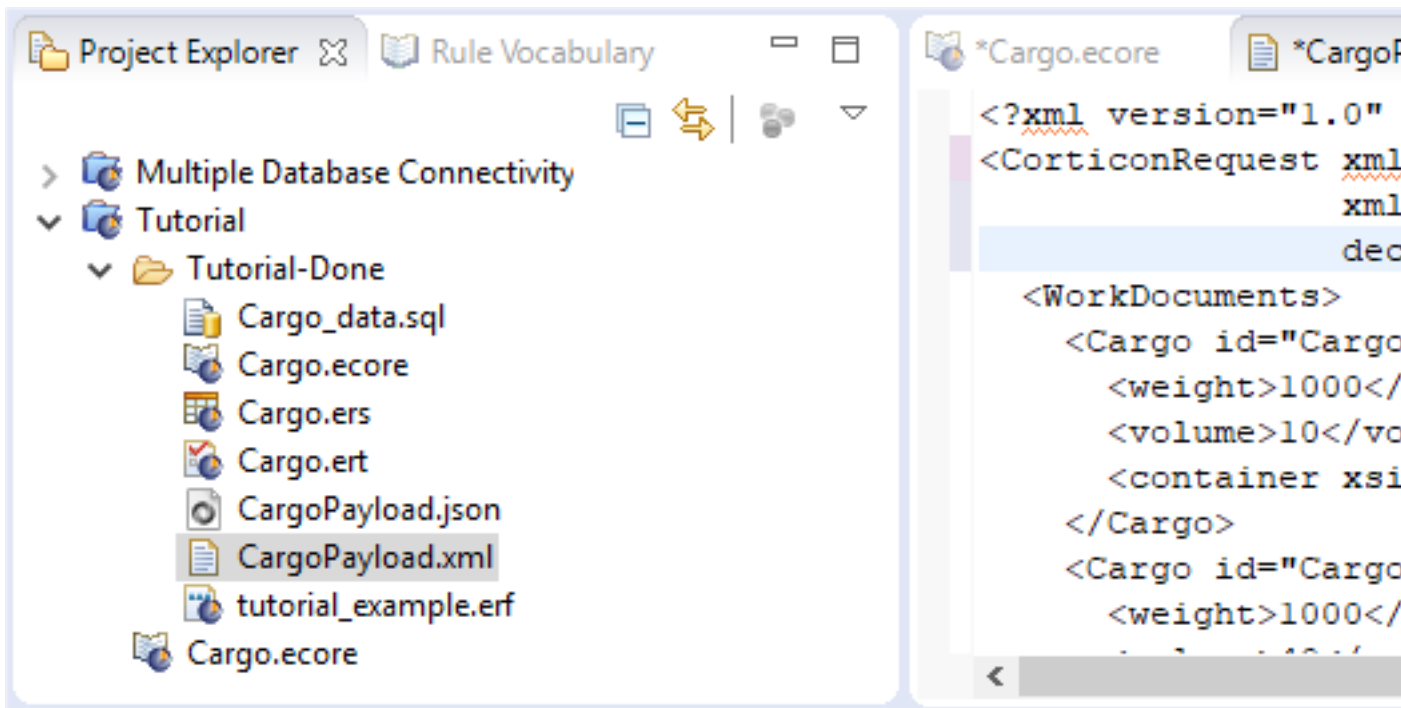
OK

We'll deploy the Decision Service to Corticon Server using the Web Console, but first let's create a request that we can use to test the deployed Decision Service.

## Using an XML Test Request in Studio

If you prefer to use an XML request instead of JSON, the sample XML request requires you to declare the Decision Service name.

1. In the Studio, click on **CargoPayload.xml** to open it in the editor.
2. Change the **decisionServiceName** to the name we will give the Decision Service when we deploy it, **Cargo Shipment**.

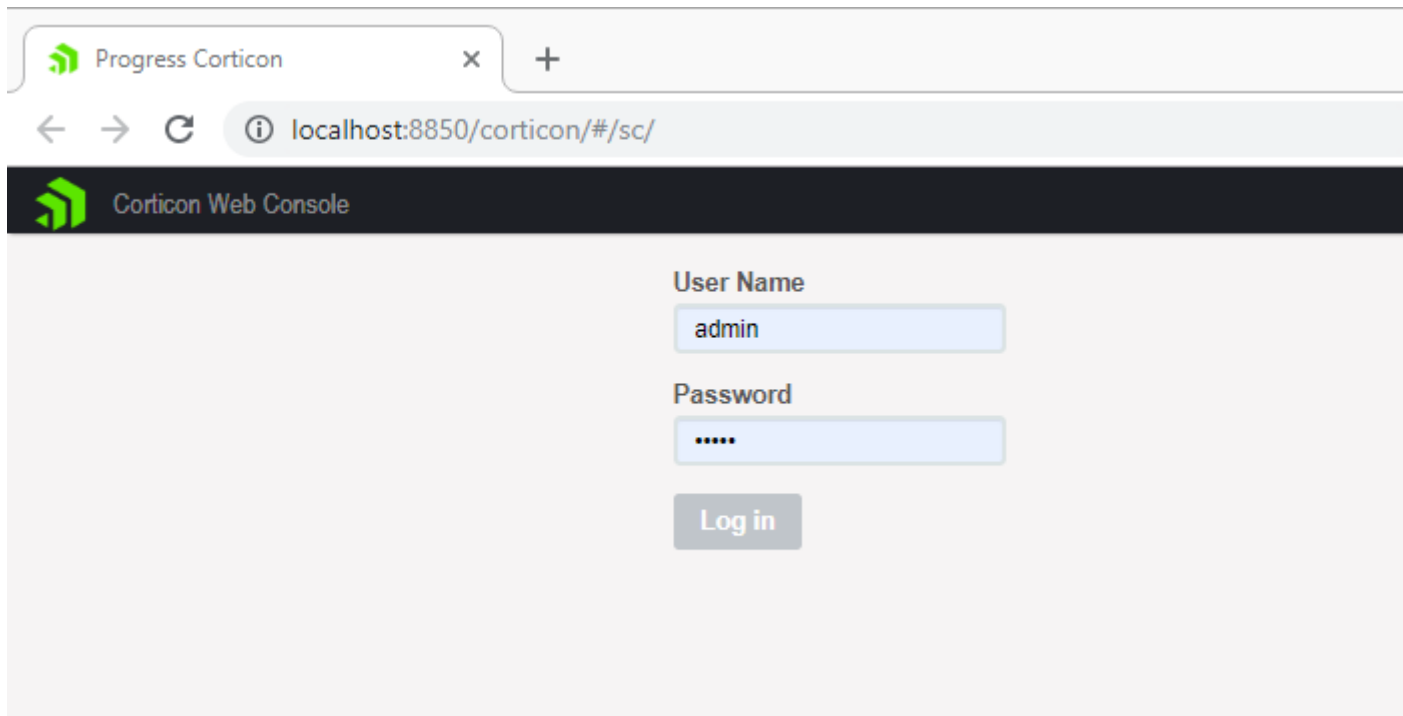


3. Save the file as **Cargo Shipment Test.xml**.
4. Save the file where you can access it from the browser where you run the Web Console.

### Deploying the Decision Service using the Web Console

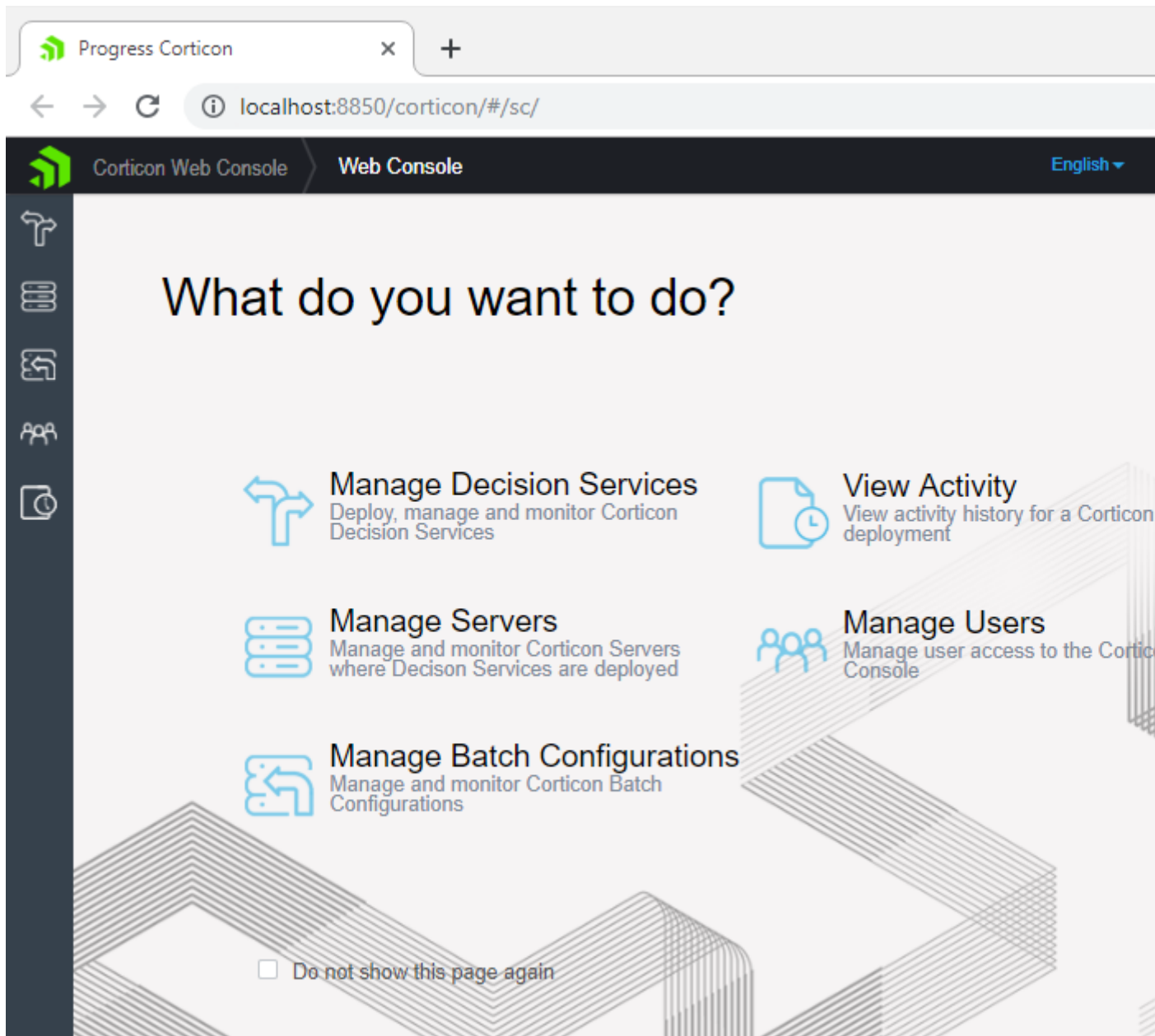
The Web Console is the tool for administering and monitoring your Decision Services. Before you launch the Web Console, ensure that Corticon Server is started.

1. On the Start menu, choose **Progress > Start Corticon 6.0 Server**. Wait a minute or so for the server startup to complete.
2. Launch the Web Console by selecting Start for Corticon 6.0 Web Console.
3. The Web Console login page opens. Enter **admin/admin** in the User Name/Password fields and click **Log in**.

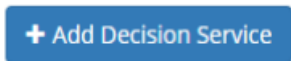


The Web Console Home page opens.

4. Click on Manage Decision Services.



5. On the Decision Services page, click **+ Add Decision Service**.



6. In the options panel, choose Add a new Decision Service.



## Add a new Decision Service

Decision Services can be managed individually or collectively within an Application

- ☒ Add a single Decision Service
- ☐ Create a new Application and add Decision Services to it

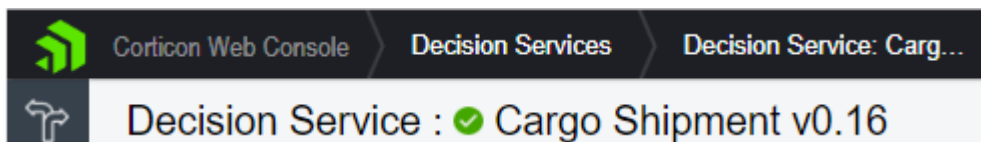
OK

The Add Decision Service dialog box

opens:

7. In the Add Decision Service dialog box:
  - a. In the Name field, enter Cargo Shipment as the name for the Decision Service.
  - b. In the EDS File section, click Choose File. In the window that opens, navigate to the location where you saved the EDS file and click Open.
  - c. On the Select Servers pulldown, choose local server.
  - d. Click Save & Deploy to deploy the Decision Service to the Corticon Server.

When the Decision Service is successfully deployed, you see a green checkmark next to the Decision Service name:



You have now successfully deployed the Ruleflow as a Decision Service on Corticon Server.

## Testing the Decision Service

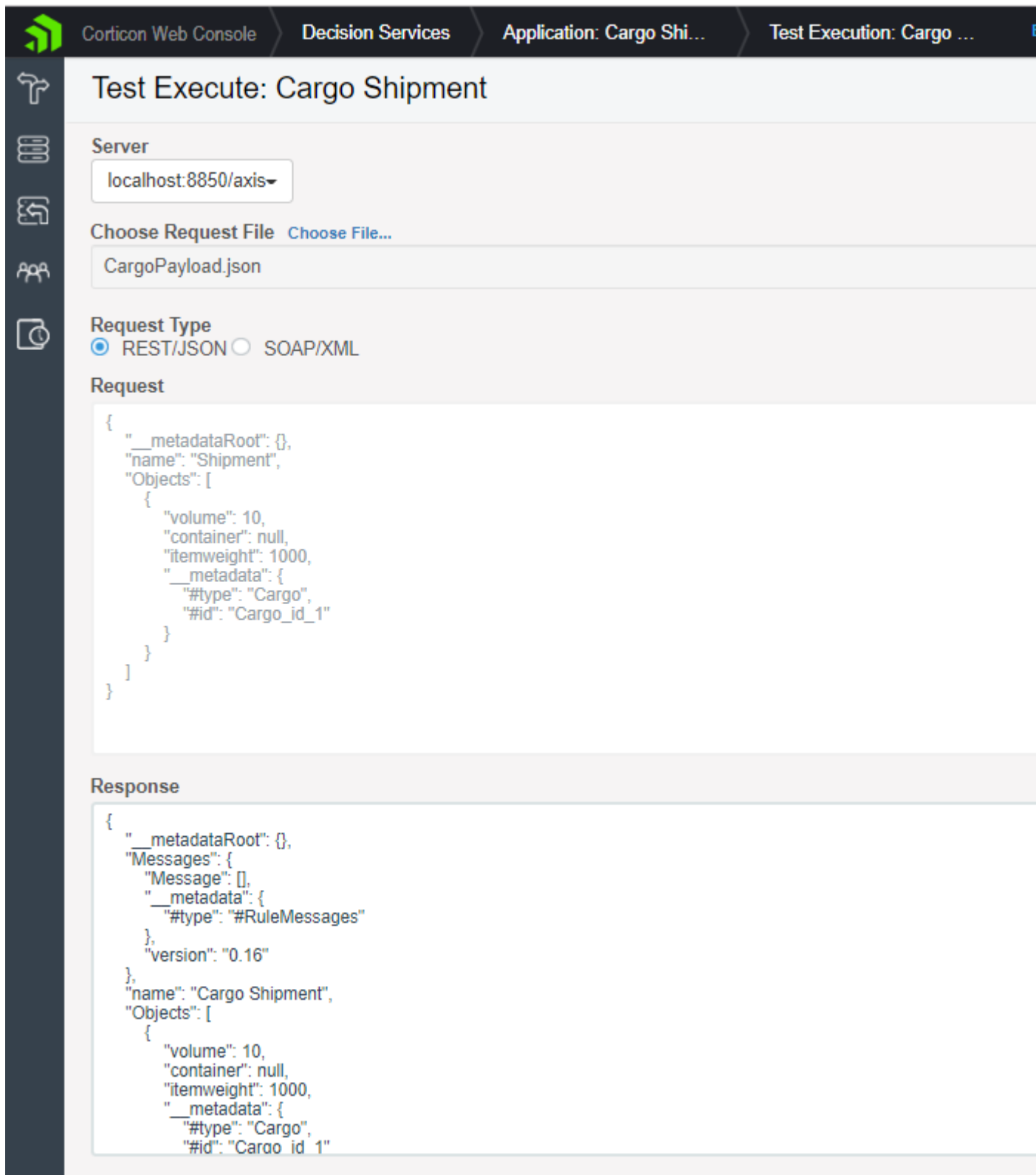
After you deploy a Decision Service, you can test the Decision Service to verify that it can be accessed as a Web Service, and to make sure that the Vocabulary has been properly prepared. The Corticon Web Console lets you send an XML or JSON request to invoke the deployed Decision Service.

1. On the Decision Service page for Cargo Shipment, click Test Execution.

The screenshot shows the Corticon Web Console interface. The breadcrumb navigation at the top reads 'Corticon Web Console > Decision Services > Decision Service: Carg...'. The main heading is 'Decision Service : ✓ Cargo Shipment v0.16'. To the right of the heading are three buttons: 'Edit', 'Undeploy', and 'Test Execution'. Below the heading, a 'General' section is expanded, displaying the following information:

Servers:	✓ local server	Execution Count:
Deployed:	Jul 18, 2019 8:55:36 AM	Failure Count:
Effective:		Average Time:
Expires:		Rule Count:
Auto Reload:	Yes	Last Execution Time:
Maximum Pool Size:	1	
Message Style:	Auto-detect	

2. On the Test Execution page, click Choose File. Let's try the JSON request first.
3. Locate the JSON request file in the project. The Request Section is now populated.
4. Click Execute.



**Corticon Web Console** > **Decision Services** > **Application: Cargo Shi...** > **Test Execution: Cargo ...**

## Test Execute: Cargo Shipment

**Server**  
localhost:8850/axis▼

**Choose Request File** [Choose File...](#)  
CargoPayload.json

**Request Type**  
☒ REST/JSON ☐ SOAP/XML

**Request**


```
{
  "__metadataRoot": {},
  "name": "Shipment",
  "Objects": [
    {
      "volume": 10,
      "container": null,
      "itemweight": 1000,
      "__metadata": {
        "#type": "Cargo",
        "#id": "Cargo_id_1"
      }
    }
  ]
}
```






**Response**

```
{
  "__metadataRoot": {},
  "Messages": {
    "Message": [],
    "__metadata": {
      "#type": "#RuleMessages"
    }
  },
  "version": "0.16",
  "name": "Cargo Shipment",
  "Objects": [
    {
      "volume": 10,
      "container": null,
      "itemweight": 1000,
      "__metadata": {
        "#type": "Cargo",
        "#id": "Cargo_id_1"
      }
    }
  ]
}
```

You can see the Response field is now populated with a JSON-formatted response message.

5. Now click Choose File again to select the XML file you edited. Then click Execute.

Corticon Web ConsoleDecision ServicesApplication: Cargo Shi...Test Execution: Cargo ...



## Test Execute: Cargo Shipment

**Server**  
localhost:8850/axis▼

**Choose Request File** [Choose File...](#)  
Cargo Shipment Test.xml

**Request Type**  
☐ REST/JSON ☒ SOAP/XML

**Request**

```
<?xml version="1.0" encoding="UTF-8"?>
<CorticonRequest xmlns="urn:Corticon" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
decisionServiceName="Cargo Shipment">
  <WorkDocuments>
    <Shipment id="Cargo_id_1">
      <itemWeight>1000</itemWeight>
      <volume>10</volume>
      <container xsi:nil="true" />
    </Shipment>
    <Shipment id="Cargo_id_2">
      <itemWeight>1000</itemWeight>
      <volume>40</volume>
      <container xsi:nil="true" />
    </Shipment>
    <Shipment id="Cargo_id_3">
      <itemWeight>30000</itemWeight>
      <volume>20</volume>
      <container xsi:nil="true" />
    </Shipment>
  </WorkDocuments>
</CorticonRequest>
```

**Response**

```
<?xml version="1.0" encoding="UTF-8"?><CorticonResponse xmlns="urn:Corticon" decisionServiceName="Cargo
Shipment" decisionServiceTargetVersion="0.16">
  <WorkDocuments>
    <Shipment id="Cargo_id_1">
      <container>standard</container>
      <itemWeight>1000</itemWeight>
      <volume>10</volume>
    </Shipment>
    <Shipment id="Cargo_id_2">
      <container>oversize</container>
      <itemWeight>1000</itemWeight>
      <volume>40</volume>
    </Shipment>
    <Shipment id="Cargo_id_3">
      <container>heavyweight</container>
      <itemWeight>30000</itemWeight>
      <volume>20</volume>
    </Shipment>
  </WorkDocuments>
</CorticonResponse>
```

Notice that the request and the response messages contain the terms Shipment and itemWeight. Corticon translated the Shipment and itemWeight tags in the request message to Cargo and Cargo.weight in the Vocabulary, and, after processing, translated the results back to Shipment and itemWeight.

You have now successfully tested the deployed Decision Service.

Congratulations! You have completed this tutorial.

You have prepared the Vocabulary for deployment, deployed a Ruleflow as a Decision Service to Corticon Server, and confirmed that the Decision Service can be accessed as a Web Service.