Progress®

Corticon Tutorial

Deploying a Decision Service as a Web Service for Java

Progress® Corticon®

Copyright

Last updated with new content: Corticon 6.3.1

Updated: 2022/09/23

Table of Contents

1

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. This tutorial uses the installed Tomcat 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.

2

Set 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 Studio**.

Step 2: Install and start Corticon Server

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

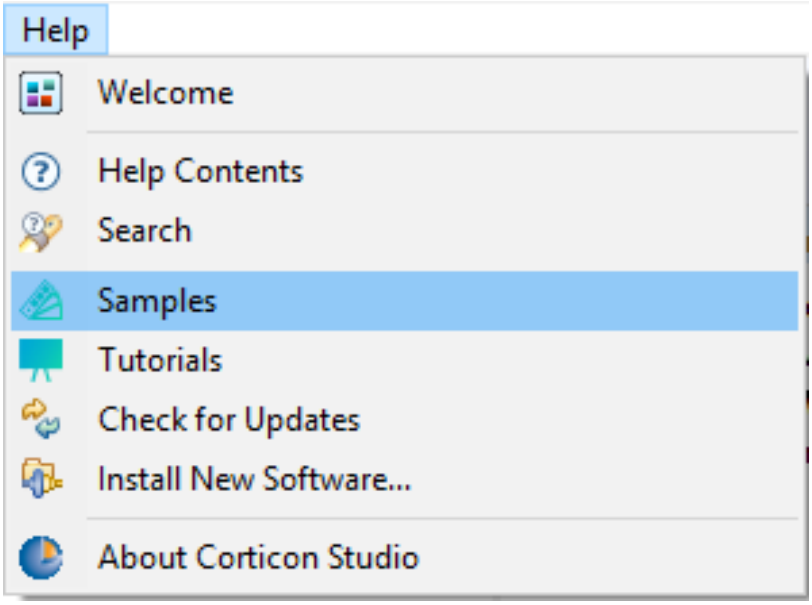
1. Download and run the Server installer.
2. On the **Introduction** page of the installation wizard, click **Next**.
3. On the **License Agreement** page, select **I accept the terms of the License Agreement** and click **Next**.

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 Server**.

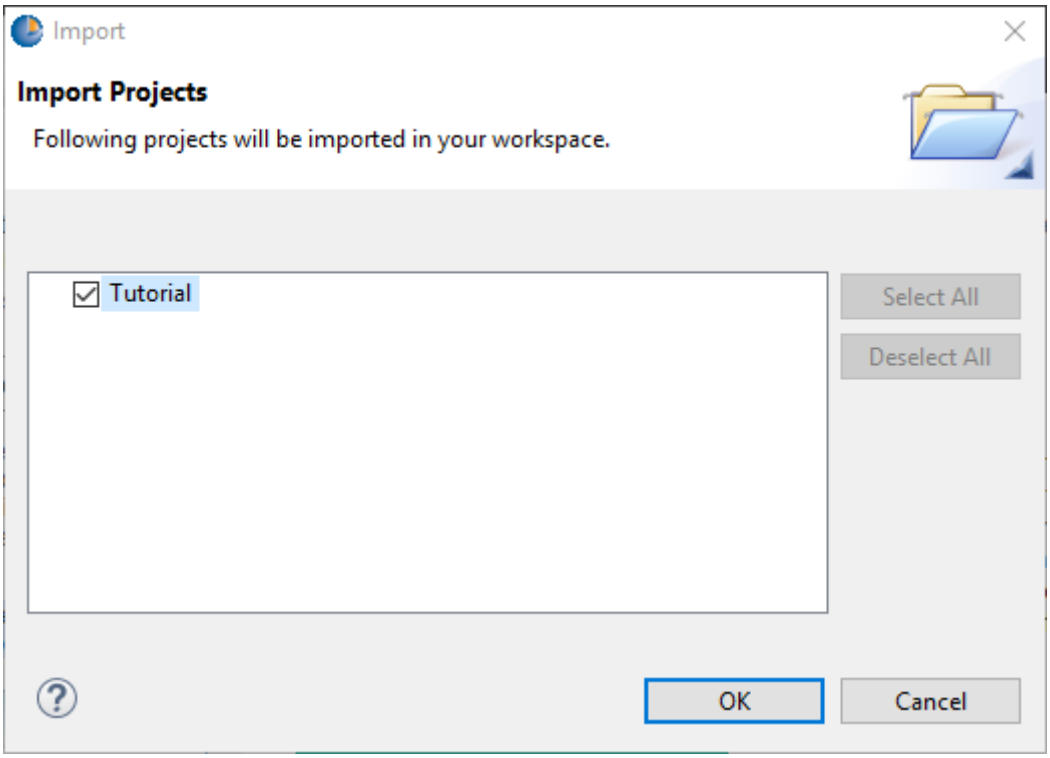
Step 3 : Importing a Sample Project into Studio

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

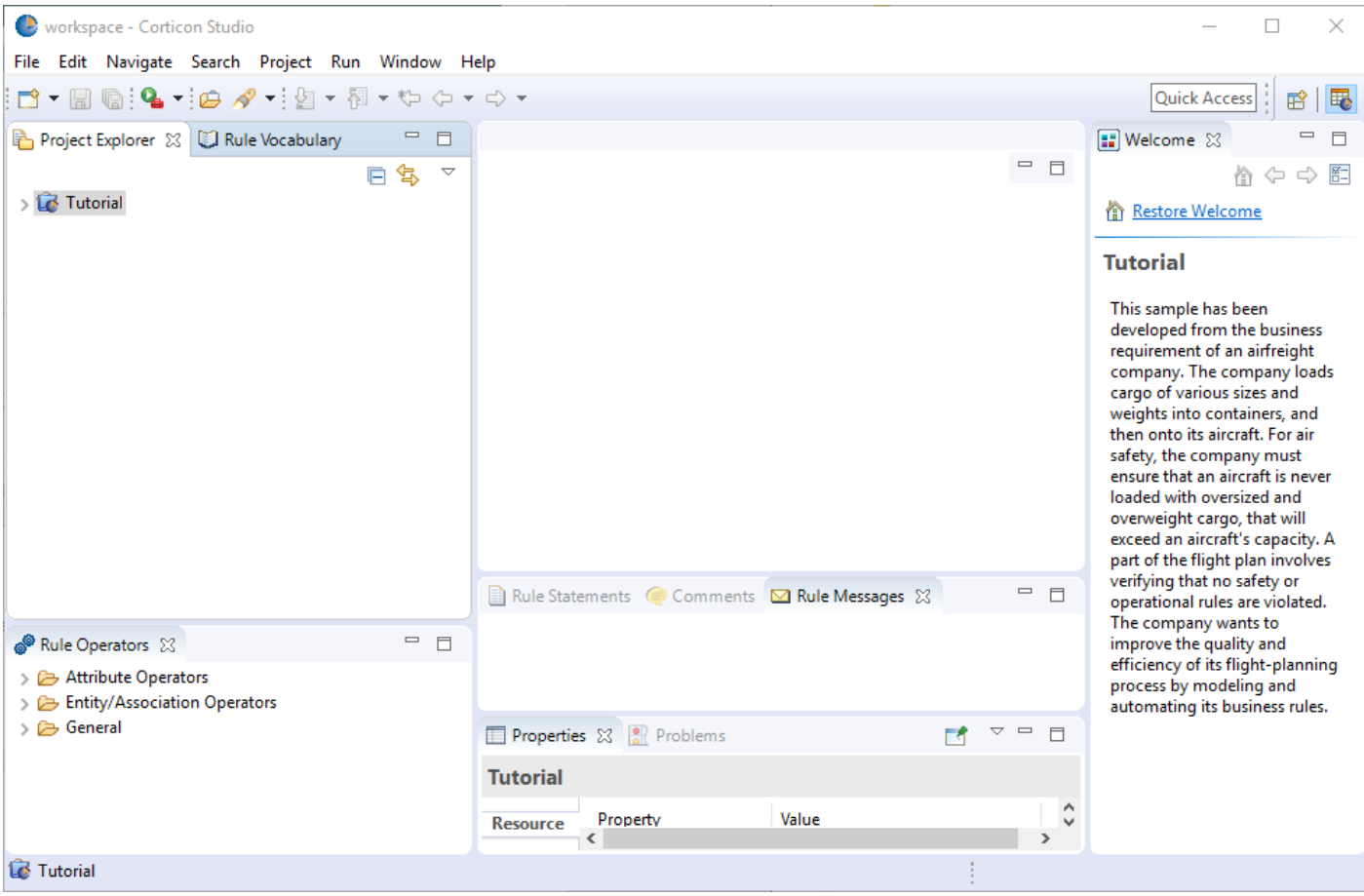
1. In Studio, choose **Help > Samples**.



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



4. The sample rule project opens in Corticon Studio.



3

Prepare requests

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 this tutorial, our request messages use JSON and XML.

The JSON request message looks like this:

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

Copy this code, and then save it as **ShipmentPayload.json**.

The XML request message looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<CorticonRequest xmlns="urn:Corticon"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" decisionServiceName="Cargo">
  <WorkDocuments>
    <Shipment id="Shipment_id_1">
      <itemWeight>1000</itemWeight>
      <volume>10</volume>
      <container xsi:nil="true" />
    </Shipment>
   </WorkDocuments>
</CorticonRequest>
```

Copy this code, and then save it as **ShipmentPayload.xml**.

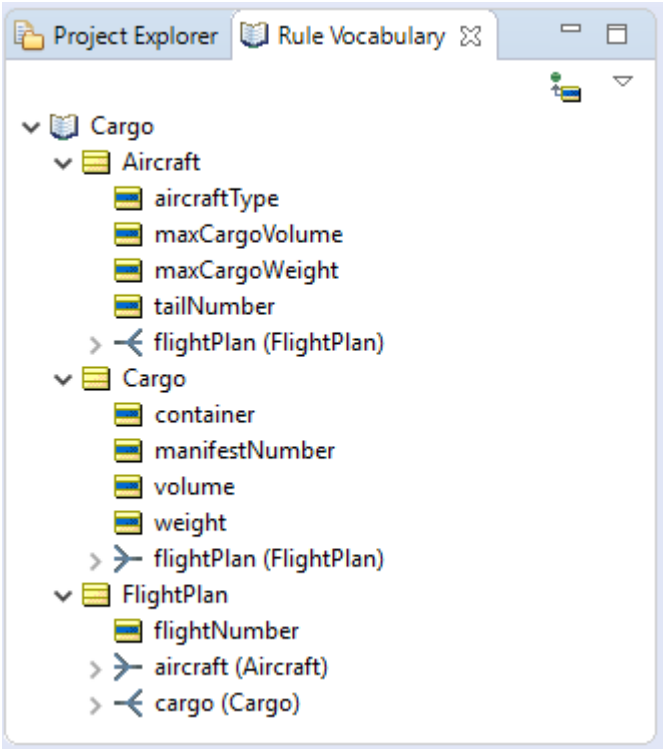
4

Prepare the Vocabulary and mappings

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.

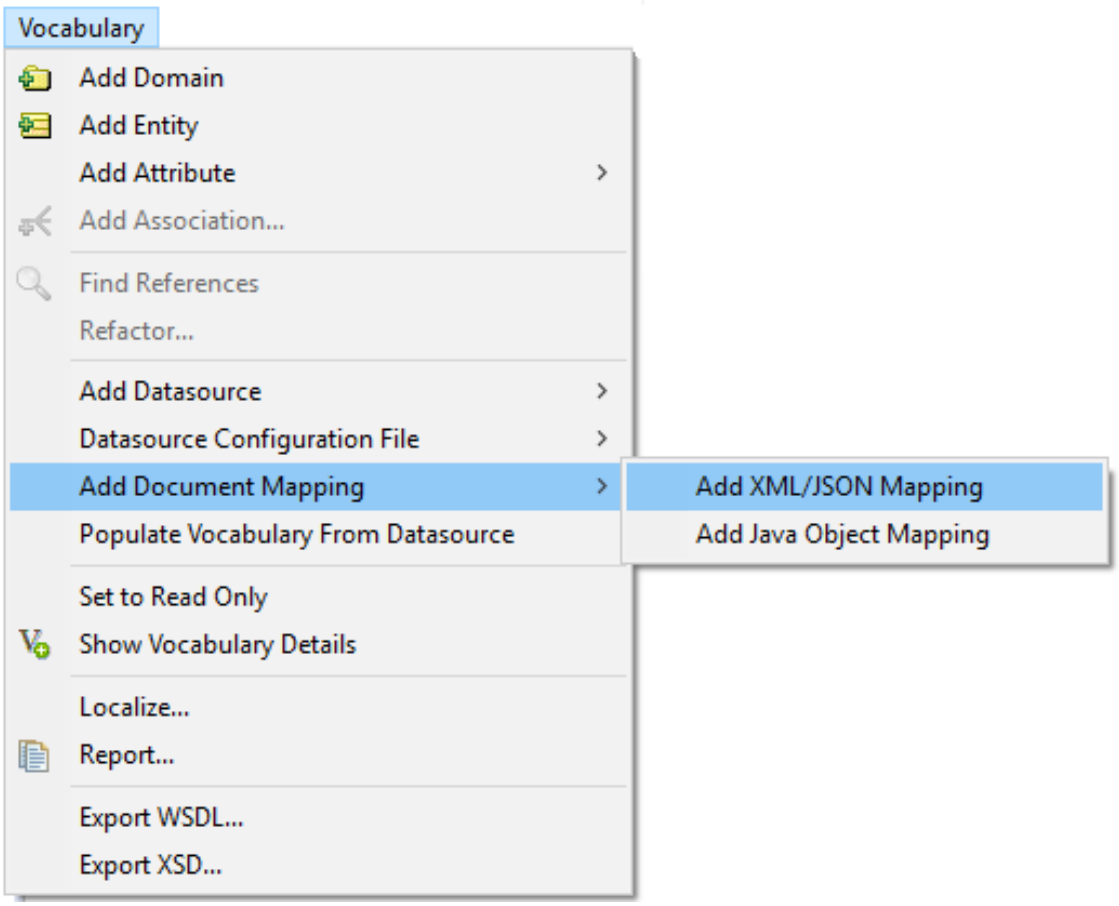
Our project Vocabulary looks like this:



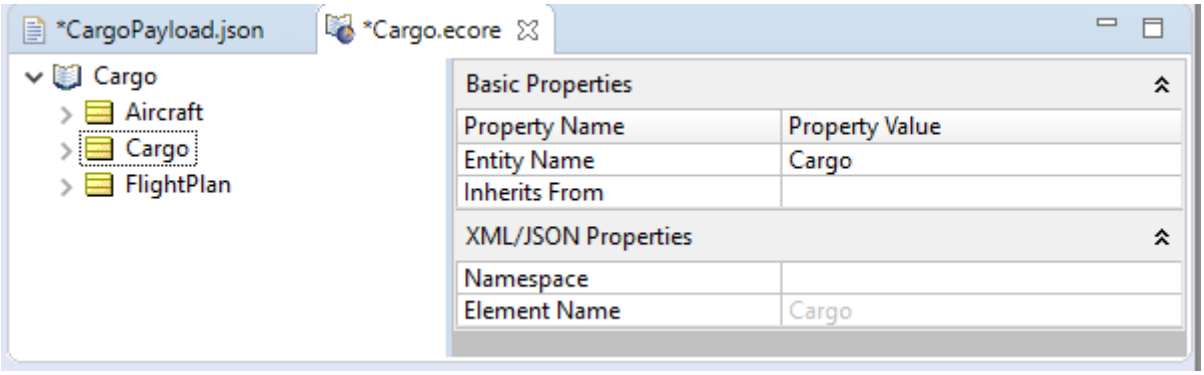
Notice that the request message contains two tag names ‘Shipment’ and ‘itemWeight’ that do not appear in our Vocabulary. They correspond to the Cargo entity and the Cargo.weight attribute.

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

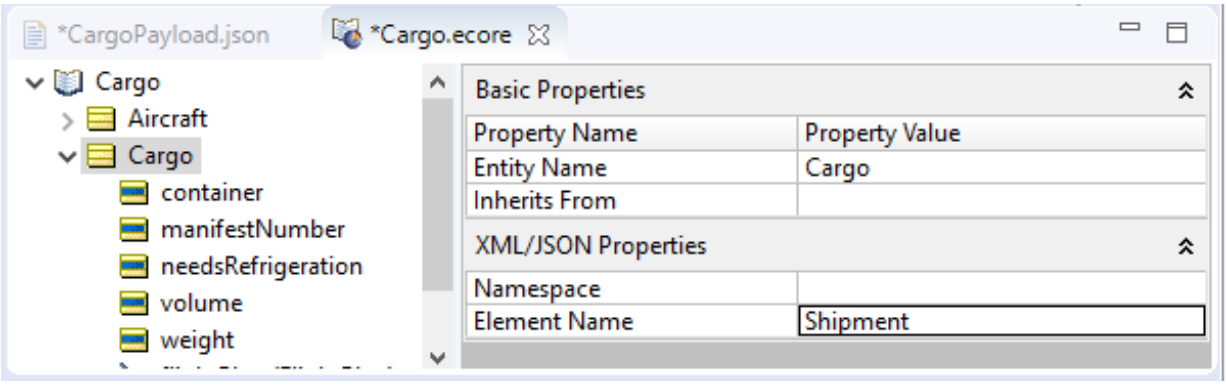
1. Open the Vocabulary file **Cargo.ecore** in the **Tutorial-Done** folder.
2. On the Vocabulary menu, choose **Add Document Mapping > Add XML/JSON Mapping**.



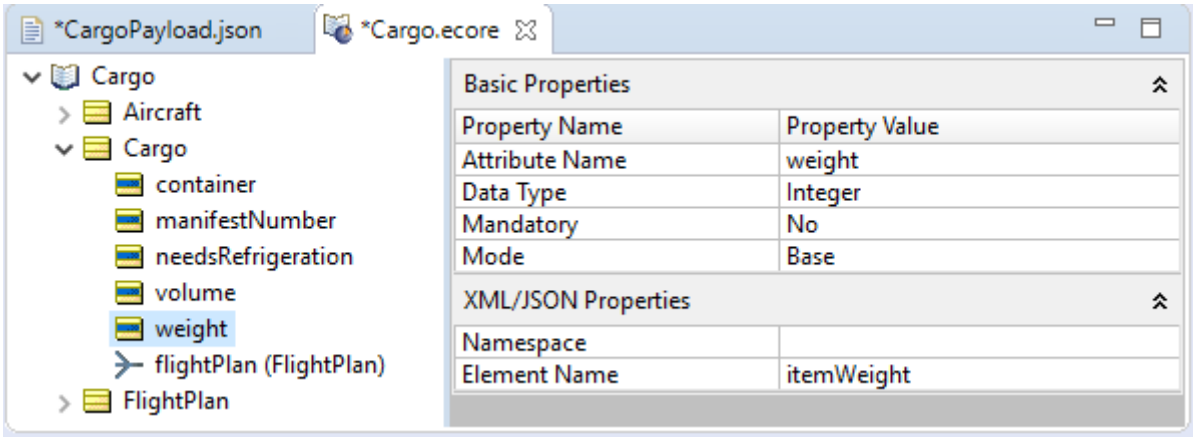
3. Click the **Cargo** entity. In the **XML Element Name** property, you can see the default name ‘Cargo’ in grey.



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.

5

Deploy 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.

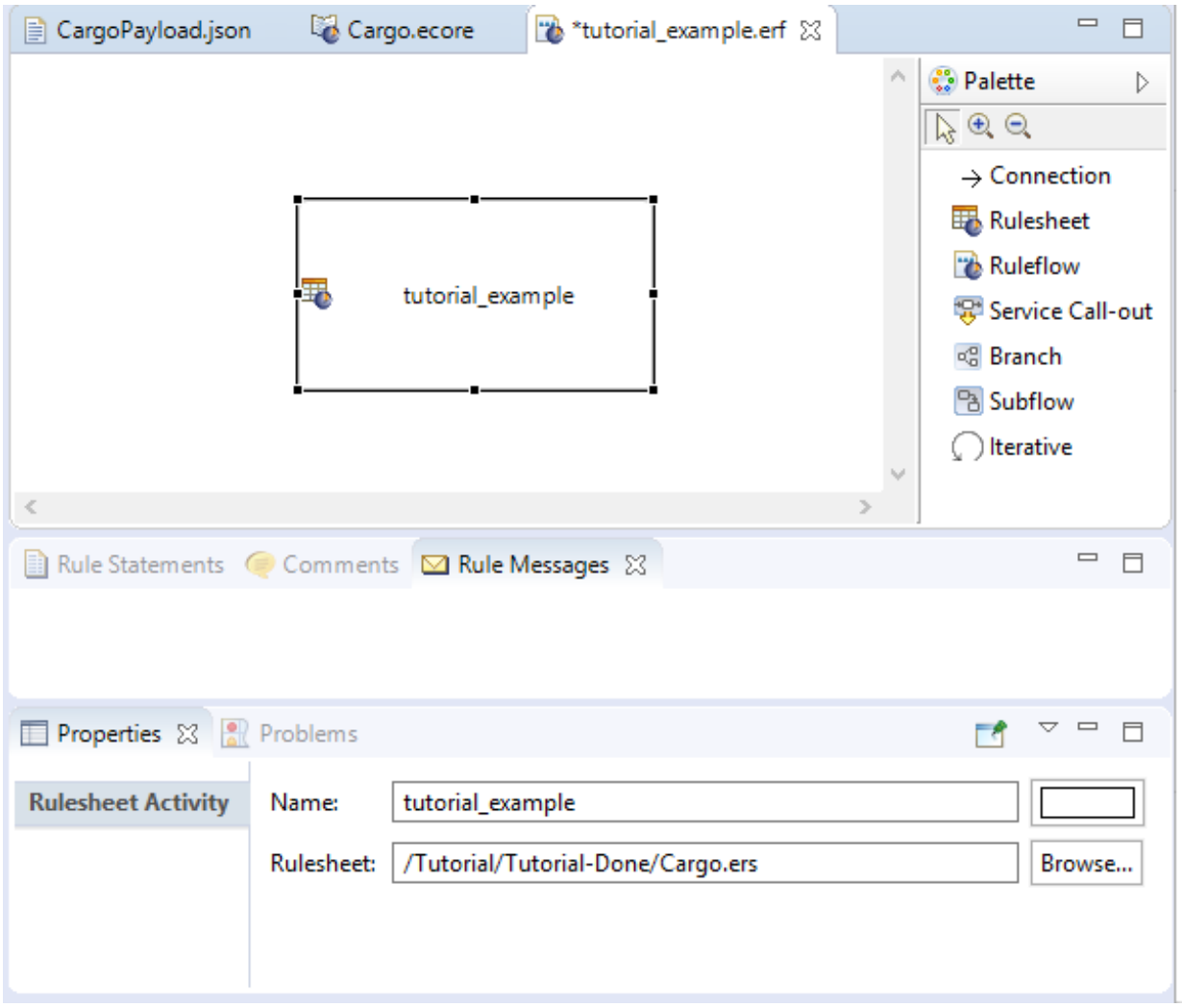
There are different ways to deploy a Ruleflow to Corticon Server. In this tutorial, you will package the Ruleflow in a pre-compiled EDS file and then deploy it to Corticon Server using the Web Console.

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

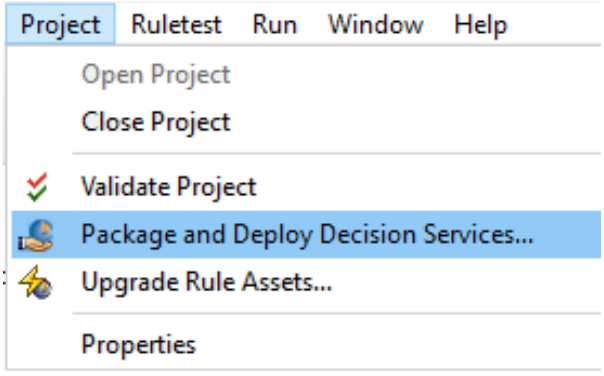
Packaging the Ruleflow into an Decision Service file

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

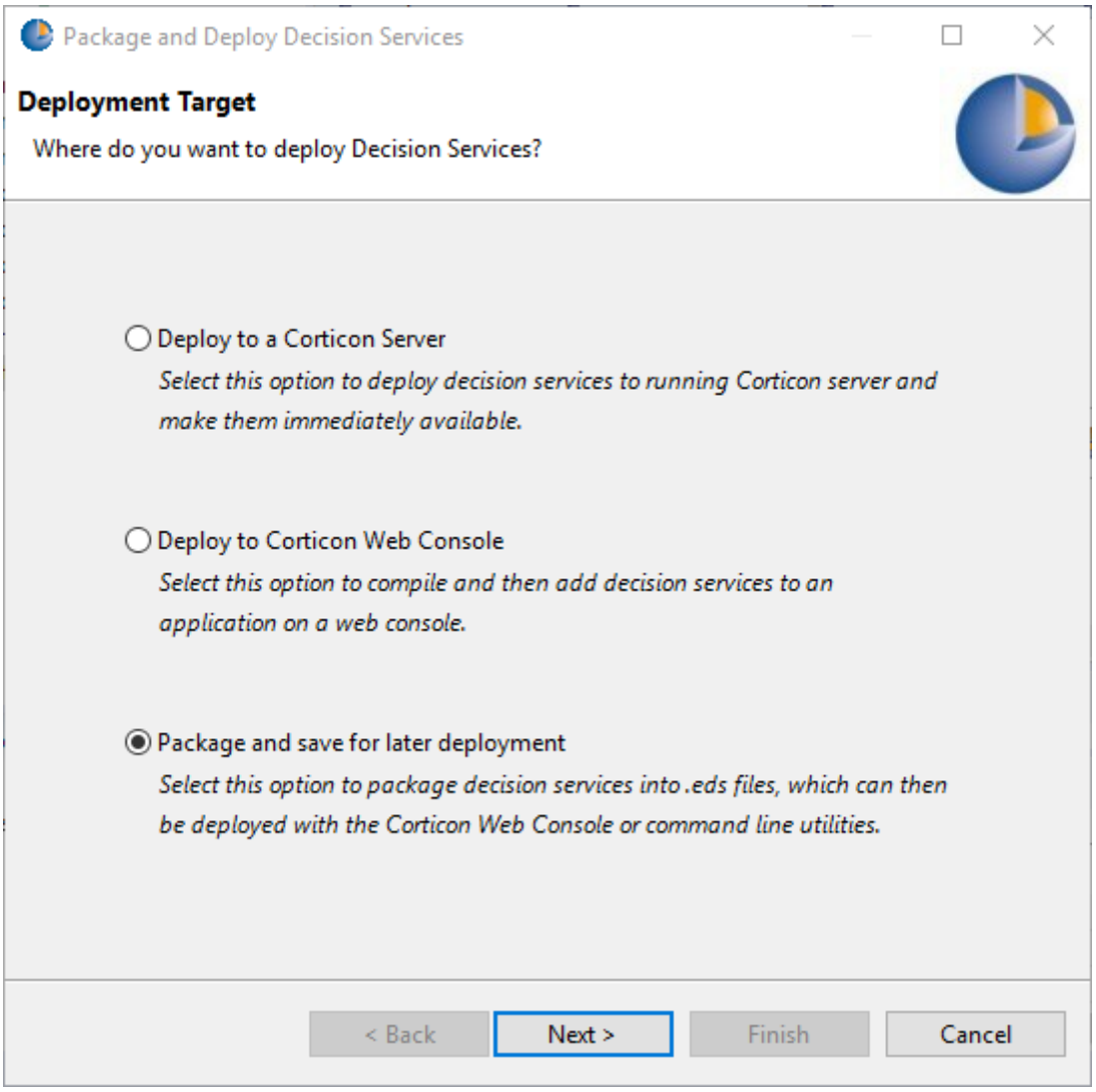
1. Open the **tutorial_example.erf** file, and then click on the **tutorial_example** box.



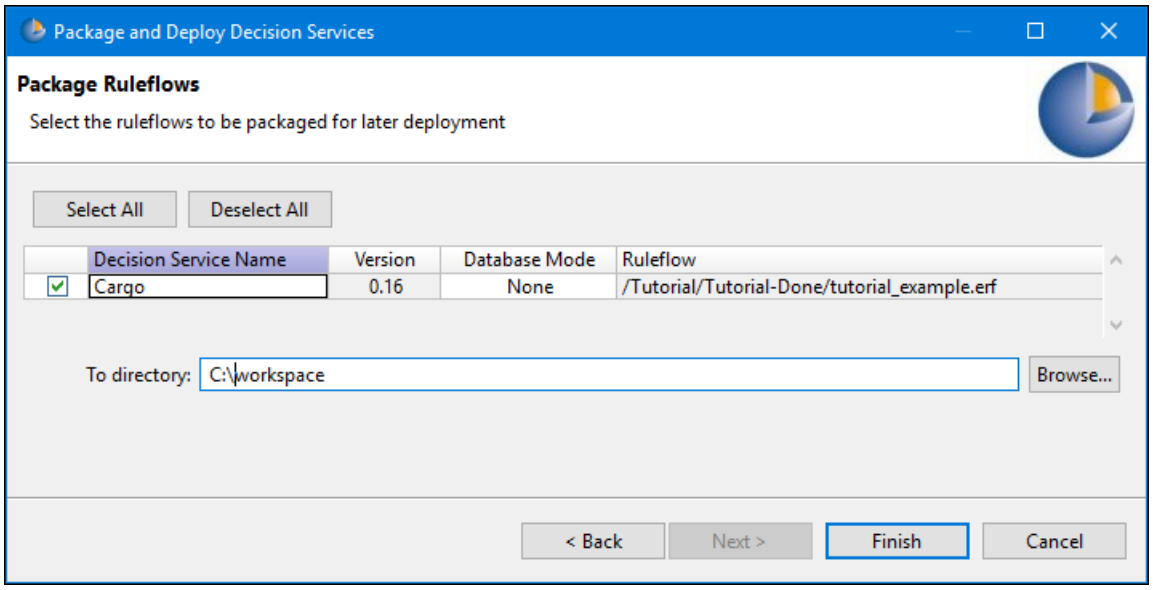
2. The Ruleflow has just one Rulesheet—Cargo.ers—which is renamed to tutorial_example in the Ruleflow. When you create an EDS file, related Ruleflow, Rulesheet, and the Vocabulary are compiled into a package that can be deployed to run rules on Corticon Server.
3. To package the Ruleflow, select **Project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



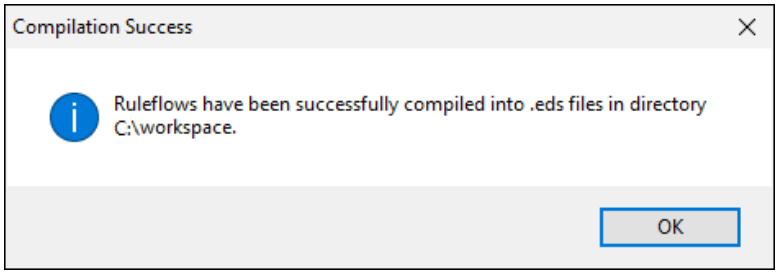
4. In the **Deployment Target** window, choose **Package** and save for later deployment and click **Next**.



5. In the **Package Ruleflows** screen:
   a. Select the **tutorial_example** Ruleflow.
   b. Click twice on the Decision Service Name, and then type in our preferred name, `Cargo`.
   c. Click **Finish**.



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

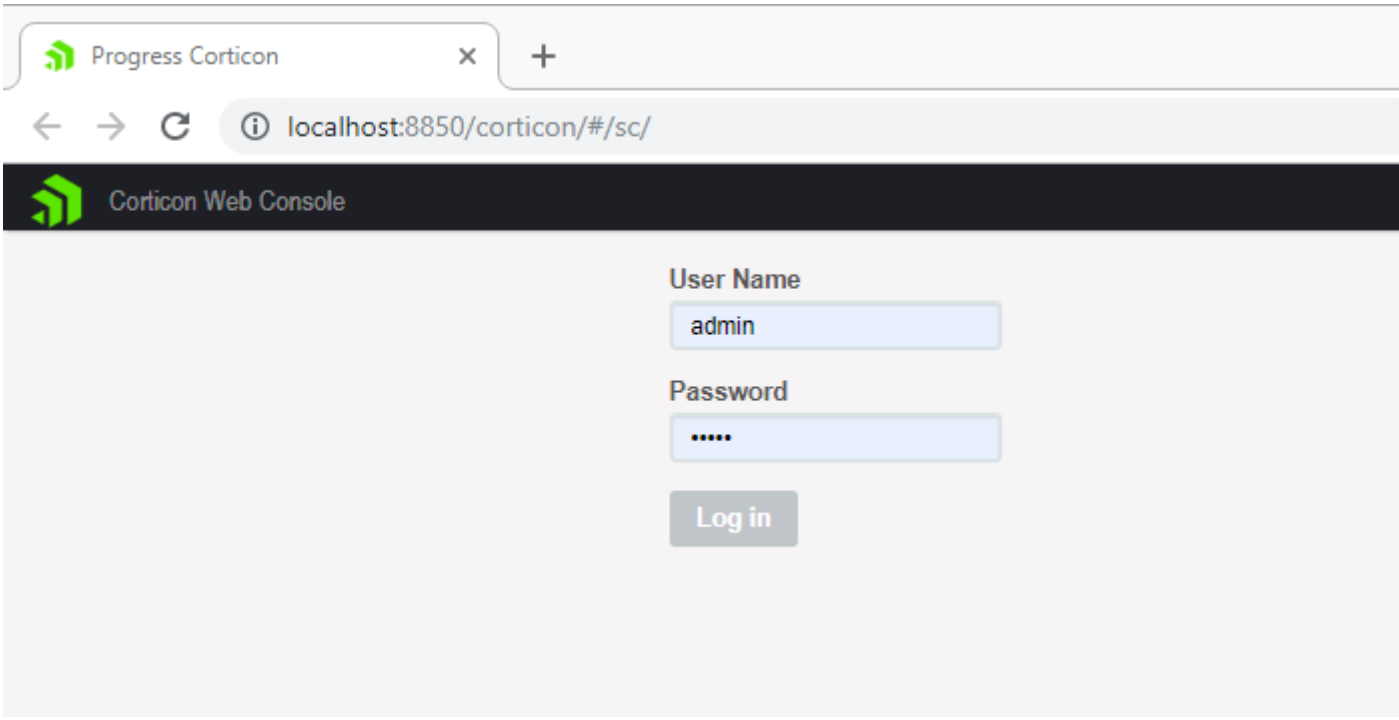


We'll deploy the Decision Service to Corticon Server using the Web Console, and then test it there with our sample requests.

Deploying the Decision Service using the Web Console

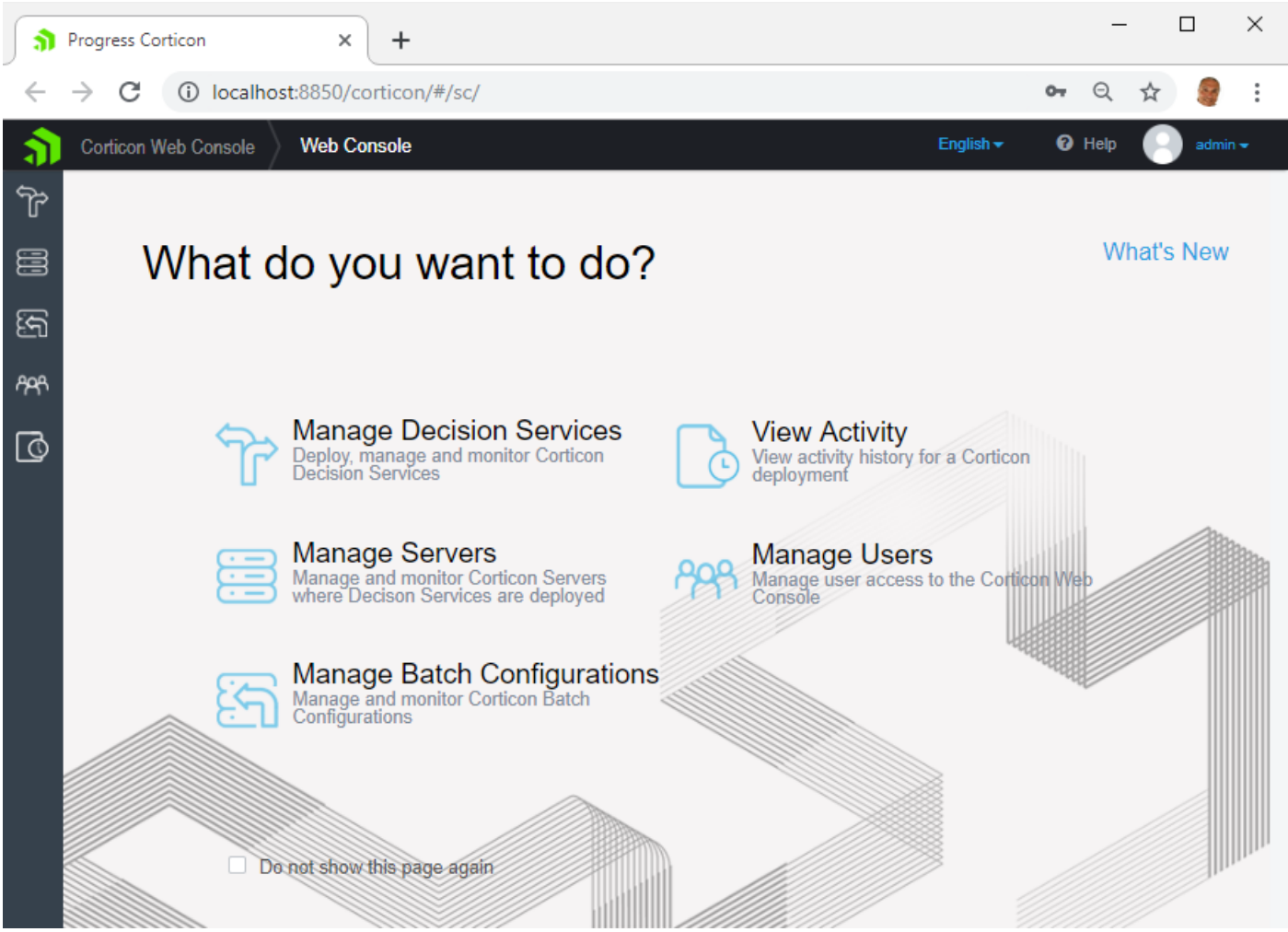
The Web Console is the tool for administering and monitoring your Decision Services. Before you launch the Web Console, ensure that Corticon Server is started.

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

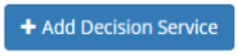


The Web Console Home page opens.

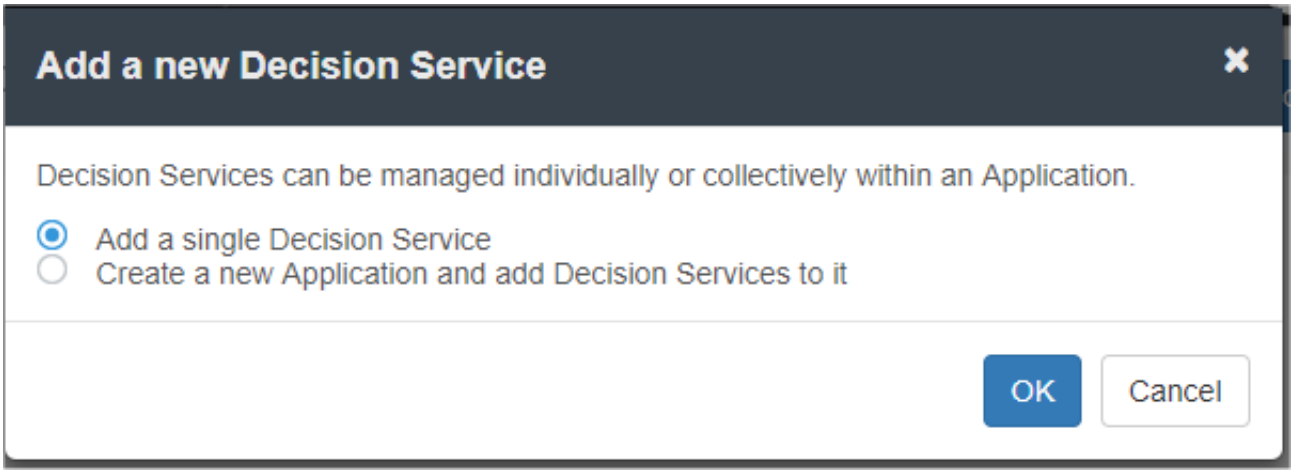
4. Click on Manage Decision Services.



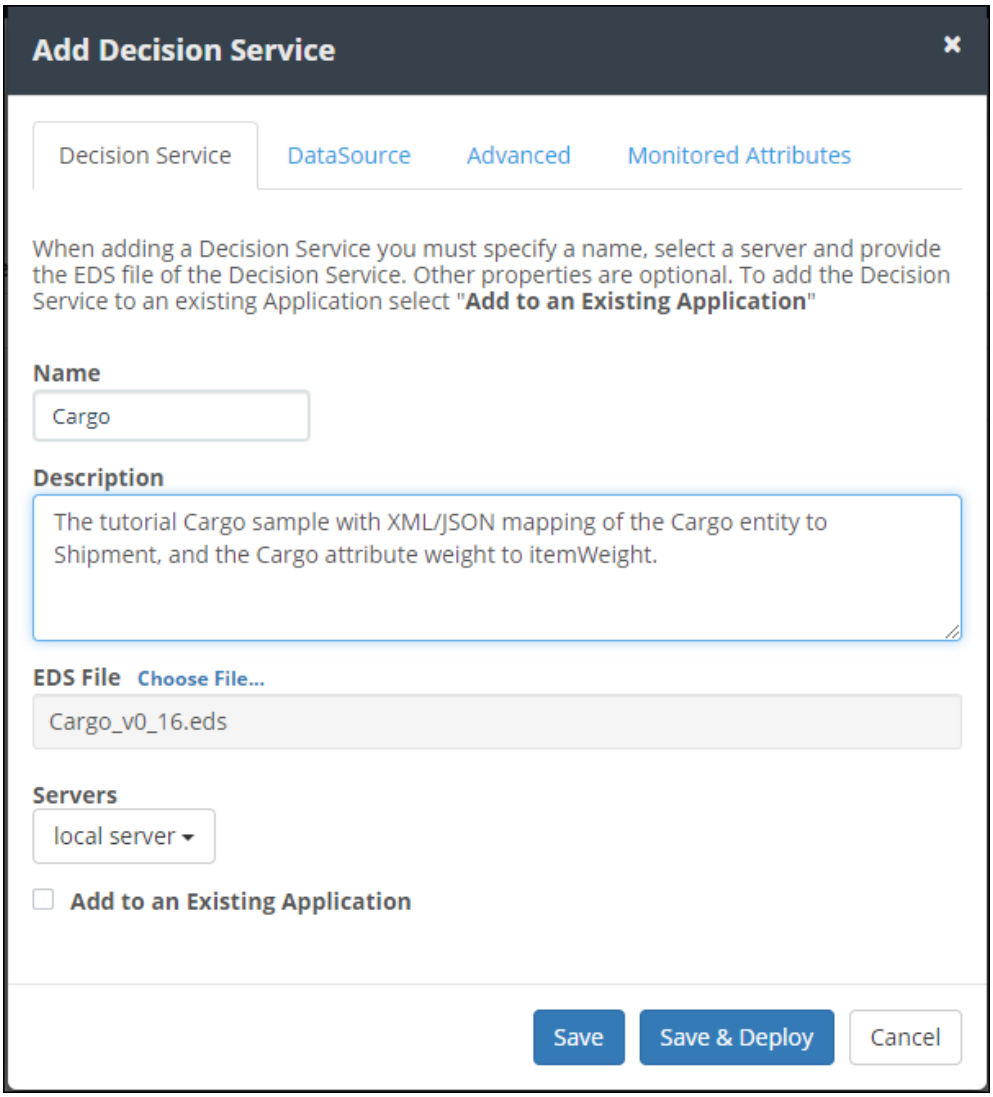
5. On the Decision Services page, click **+ Add Decision Service**.



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

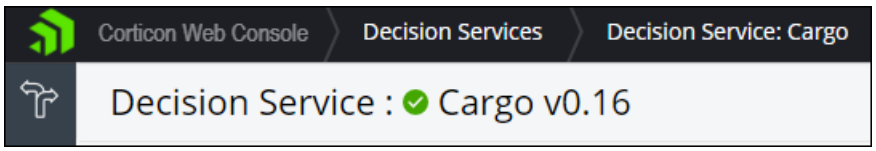


The Add Decision Service dialog box opens:



7. In the Add Decision Service dialog box:
   a. In the Name field, enter Cargo 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:



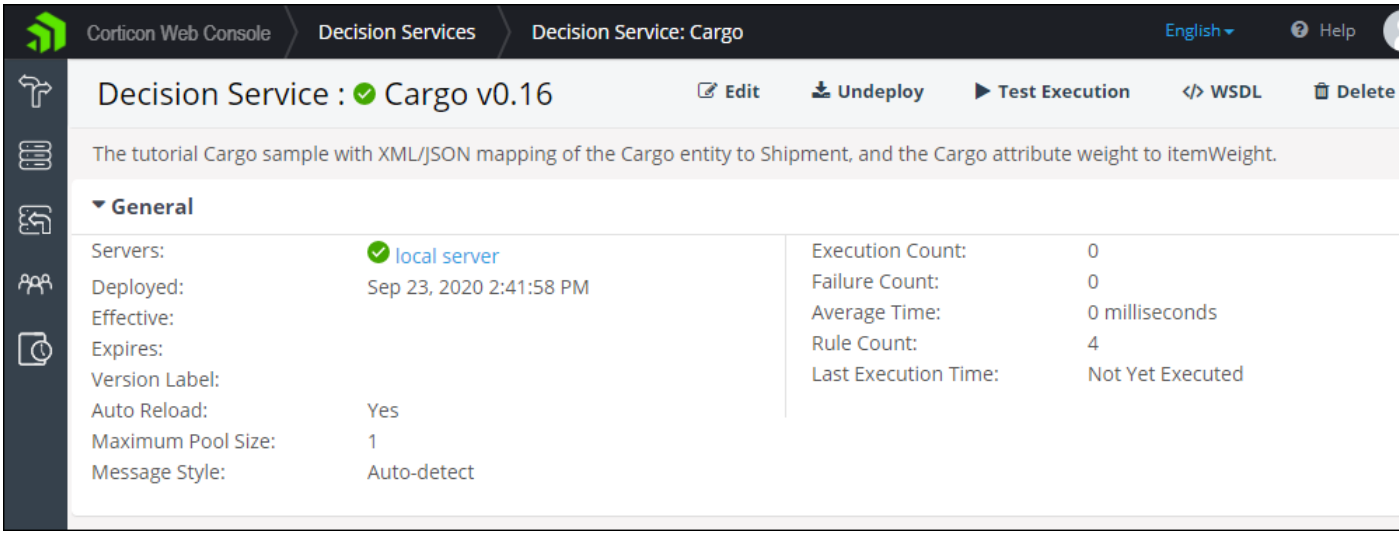
You have now successfully deployed the Ruleflow as a Decision Service on Corticon Server.

6

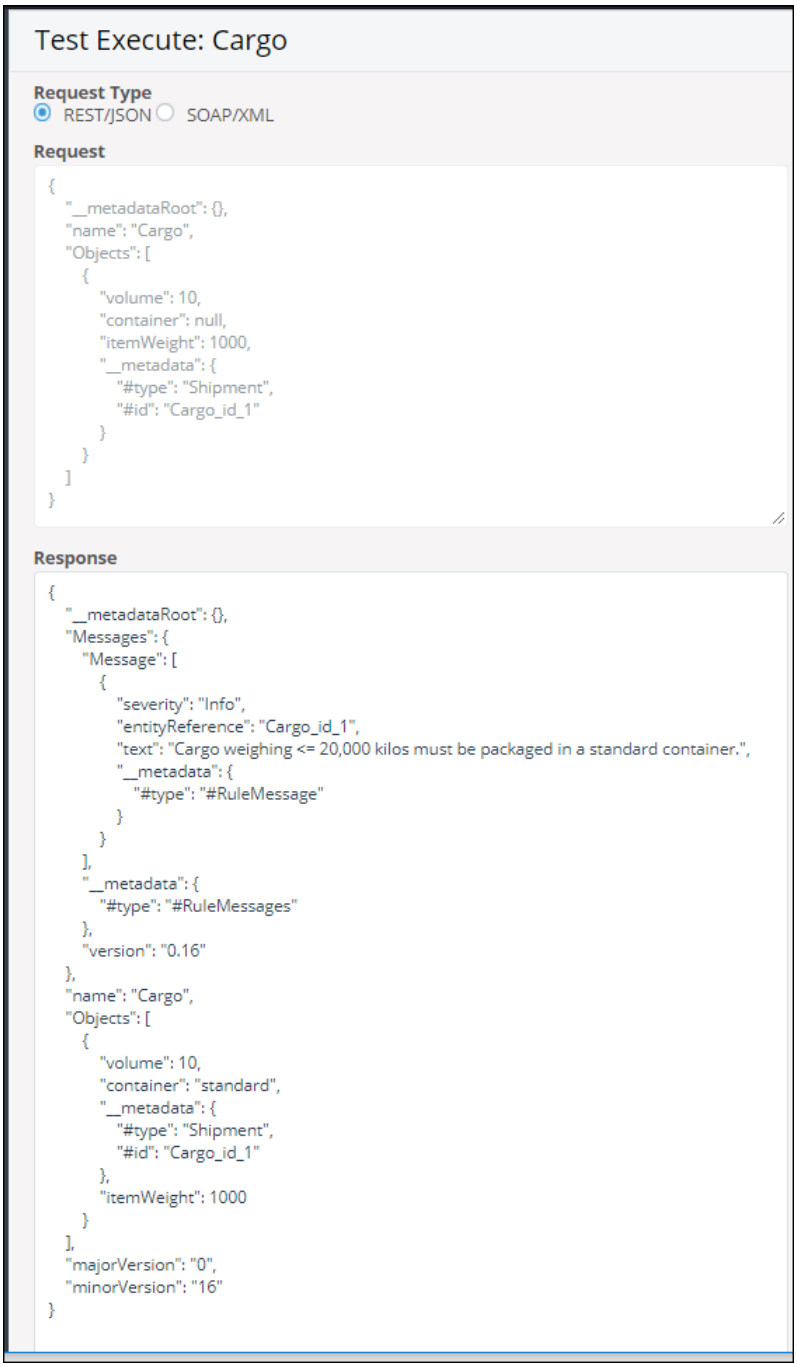
Test the Decision Service

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, click Test Execution.

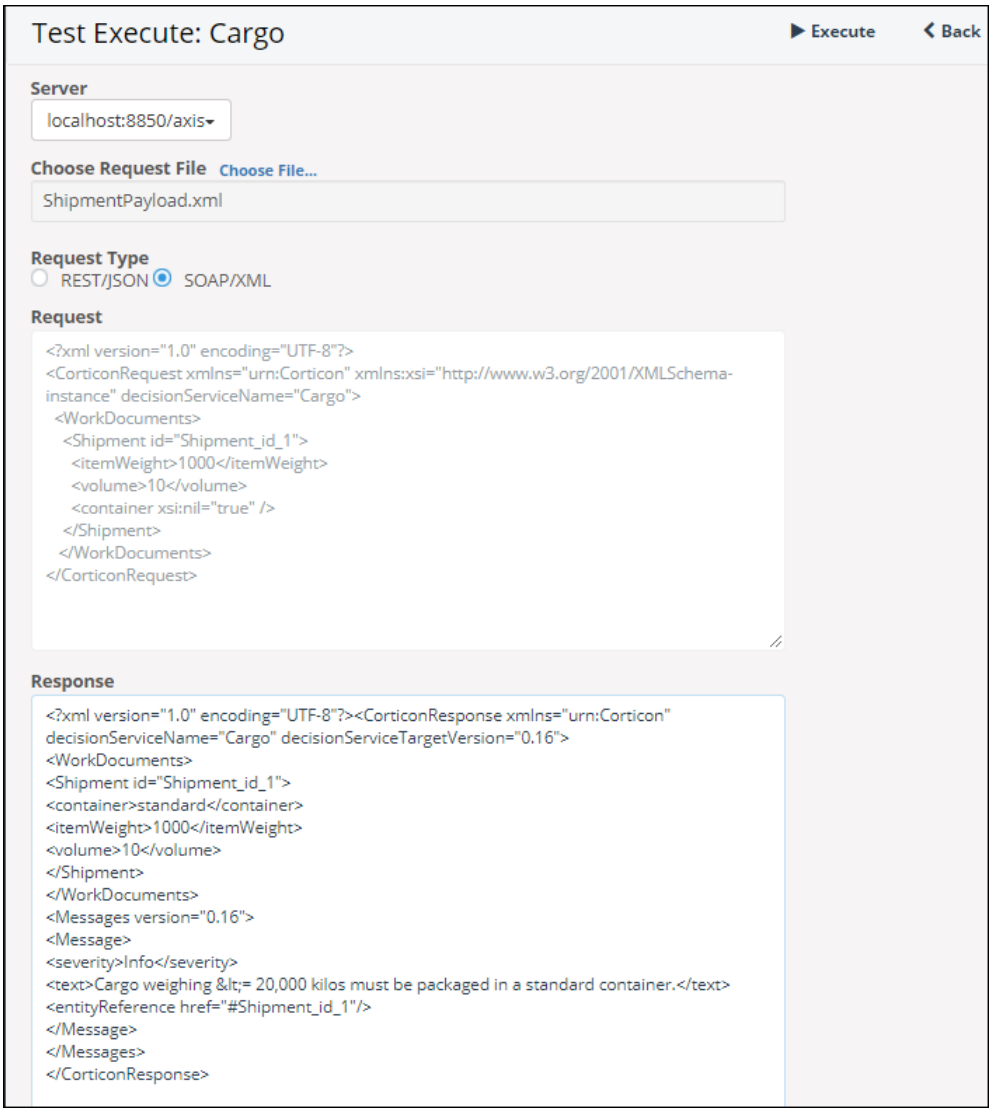


2. On the Test Execution page, click Choose File. Let's try the JSON request first.
3. Locate the JSON request file in the project. The Request Section is now populated.
4. Click Execute.



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

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



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.