



## What's new in Corticon



# Copyright

---

© 2020 Progress Software Corporation and/or 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, Defrag This, Deliver More Than Expected, Icenium, Ipswitch, iMacros, Kendo UI, Kinvey, MessageWay, MOVEit, NativeChat, NativeScript, OpenEdge, Powered by Progress, Progress, Progress Software Developers Network, SequeLink, Sitefinity (and Design), Sitefinity, SpeedScript, Stylus Studio, 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, DataDirect Autonomous REST Connector, DataDirect Spy, SupportLink, DevCraft, Fiddler, iMail, JustAssembly, JustDecompile, JustMock, 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.

**Last updated with new content:** Corticon 6.1.1

**Updated:** 2020/10/07



---

# Learn about Corticon releases

---

**Corticon.js** enables you to define rules and package them into fully, self-contained JavaScript bundles that can be deployed to any compatible JavaScript platform.

**Corticon.js 1.1** provides additional collection operators, and the ability to quickly export from a Ruletest to the clipboard.

**Corticon 6.1.2** is a service pack release of Corticon 6.1 that provides bug fixes and minor improvements.

**Corticon 6.1.1** is a service pack release of Corticon 6.1 that provides improvements, such as:

- Microsoft Dynamics CRM data source integration has been refined and is now presented as Microsoft Dynamics 365 for accessing Dynamics data.
- Simplified Corticon Studio interface that hides unused Eclipse menu commands to present a sleeker user experience.

**Corticon 6.1** is a point release of Corticon 6 that provides improvements and added features, such as:

- New Microsoft Dynamics CRM data source for accessing Dynamics CRM data.
- Ability to generate a Vocabulary from a REST or database source.
- New operators for random, replace, and matches.
- Ability to use third party database drivers for data access.

**Corticon 6.0** is a major release of Corticon. The changes in 6.0 include:

- New REST data source allowing you to access REST data from decision services.
- A new architecture for .NET server, one that provides remarkably faster throughput.
- The Deployment Console has been dropped. You can generate WSDL and XSD from new Ruleflow and Project menu actions.

For more information, see:

- *What is Corticon.js?*
- *What is new in Corticon 6.1.1*
- *What is new in Corticon 6.1*
- *What is new in Corticon 6*
- *What changed in Corticon 6*
- *What has been removed in Corticon 6*

## What is new in Corticon 6.1.1

---

This section summarizes changes in Progress® Corticon® 6.1.1.

- Microsoft Dynamics CRM data source integration has been refined and is now presented as Microsoft Dynamics 365 for accessing Dynamics data.
- The Corticon Studio user interface is now simplified with unused Eclipse menu and toolbar items removed. This is the default **Corticon Designer** perspective. The previous UI remains available as the **Corticon Classic** perspective if you require the full set of Eclipse Workbench commands.
- The SVN plugin is no longer included in the Studio installation. The plugin can be accessed and updated from [Eclipse](#).





---

## What is new in Corticon 6.1

---

This section summarizes new, and enhanced features in Progress® Corticon® 6.1.0.

### **Ability to generate a Vocabulary from a REST or database source.**

Accelerate project development by generating your rule vocabulary from existing data sources. The data models defined in your SQL databases or REST services can be used to create or update a rule vocabulary. This can save weeks in the development of new projects and helps ensure your rule vocabulary is in sync with your business data models. See the Rule Modeling topics under *"Populate a Vocabulary from a Datasource"* in the *Rule Modeling* guide for details.

### **Ability to use third party database drivers for data access**

When you need to connect Corticon to a database that does not have a DataDirect driver packaged with Corticon, you can add it. You need the driver JAR from the supplier. You define a configuration file for it, and then package it in a JAR. When both JARs are on the classpaths, Corticon Studio can use your driver in Vocabulary generation and in rules that you deploy to Corticon Servers. See the Data Integration topic *"Add an unlisted database server"* in the *Data Integration* guide for more information.

### **Database driver for Microsoft Dynamics and its authentication**

Corticon 6.1 adds a new Microsoft Dynamics 365 datasource, allowing you to utilize the power of Corticon rules in your Microsoft Dynamics 365 applications. Invoke rules from Dynamics 365 applications using the Corticon REST API and your rules now have full CRUD access to your Dynamics 365 data. With Corticon you can define, document, validate and automate the complex business logic for your Dynamics 365 applications.

---

**Note:** The authentication options for this driver are in a pulldown menu. That menu called for redesigning the user interface for authentication options for other drivers so that Keberos is now presented in the authentication pulldown menu, and Username/Password is available as appropriate. See *"Authentication on EDC and ADC connections"* in the *Data Integration* guide for more information.

---

## New operators

The release adds the following Attribute operators:

- **replaceString** - Returns a new string where the instances of the String to be replaced are replaced by the value of the replacement String. See [Replace String](#) in the *Rule Language* guide.
- **regexReplaceString** - Returns a new String where the strings matching the regular expression are replaced by the replacement string. See [Regular expression to replace String](#) in the *Rule Language* guide.
- **matches** - Returns true if the regular expression matches the String. See [Matches](#) in the *Rule Language* guide.
- **random** - Returns a random value within a range. See [Random](#) in the *Rule Language* guide.

## Ruleflow version identifier displays in Web Console

You can add a text descriptor to the Ruleflow by typing in the **Version Label** field. The description stays with the Ruleflow file, and is packaged in any Decision Services created from the Ruleflow. In the Web Console, every deployed instance of the Decision Service now lists the **Version Label** on its details page, as illustrated:

The screenshot shows the Corticon Web Console interface. The breadcrumb navigation at the top indicates the path: Corticon Web Console Technology Preview > Decision Services > Decision Service: Cargo... The main header displays 'Decision Service : ✔ Cargo 2019 v1.6' with action buttons for Edit, Undeploy, Test Execution, and WSDL. A left sidebar contains navigation icons. The 'General' tab is active, showing a table of properties. The 'Version Label' property is highlighted with an orange border and contains the text 'My label for this version of this Decision Service.'.

General		Performance	
Servers:	✔ local server	Execution Count:	0
Deployed:	Sep 30, 2019 2:49:01 PM	Failure Count:	0
Effective:		Average Time:	0 milliseconds
Expires:		Rule Count:	4
Version Label:	My label for this version of this Decision Service.	Last Execution Time:	Not Yet Executed
Auto Reload:	Yes		
Maximum Pool Size:	1		
Message Style:	Auto-detect		

## Streamlined Eclipse Environment

Corticon Studio has reduced the number of plugins loaded by Corticon Studio by no longer bundling the Eclipse Plug-in Development Environment (PDE). The PDE is used for creating Eclipse plugins, and is not applicable to Corticon rule development. If you need the PDE, you can add it to your Corticon Studio installation from the Eclipse download site.

If you open workspaces created in previous releases, you might see that the **Error Log** tab displays that it could not create LogView. The error is benign as Corticon did not use this view when the PDE was installed.

---

## What is new in Corticon 6

---

This section summarizes new, and enhanced features in Progress® Corticon® 6.0.0.

### Support for accessing REST services

Corticon 6.0 expands the data integration capabilities of Corticon with the introduction of a new REST Datasource. A REST Datasource lets you retrieve data from REST services to enrich the payloads being processed by your rules. Examples of usage include accessing an external credit bureau's REST service to retrieve a credit rating for an applicant, and accessing an internal REST service to retrieve supporting information for a healthcare claim. The new REST Datasource complements the existing EDC and ADC features, providing the ability to access both REST and database data from your rules.

Two sample projects, "REST Connectivity" and "Mixed Connectivity" are bundled with Corticon Studio to introduce how to use the new REST Datasource.

For more information, see the topic *"Getting Started with REST" in the Data Integration Guide*.

### New Corticon .NET Server architecture

Corticon 6.0 introduces a new architecture for Corticon .NET. The previous IKVM cross-compiling architecture has been replaced with a much simpler and faster architecture for bridging between .NET and Java. Corticon 6.0 .NET performance is at least 3x faster than previous versions.

Corticon 6.0 .NET supports both in-process and IIS deployment. See the Web Services and In-Process sections for information about deploying Corticon 6.0 decision services to .NET. The samples and supporting utilities have all been completely revised so that they are easier to use and easier to contrast to the Java implementation.

For more information, see the topics in *Web services on .NET*.



---

## What changed in Corticon 6

---

This section summarizes the features that have changed in Progress® Corticon® 6.0.0.

- **Documentation improvements** The Corticon documentation has been reorganized to be more focused and to remove duplications. Notable changes include:
  - Splitting the “Integration & Deployment Guide” into two separate guides; “Server Guide” and “Deployment Guide”. The “Server Guide” covers operation of the Corticon Server. The “Deployment Guide” covers the packaging and deployment of Decision Services.
  - Combining “Deploying Web Services with Java” and “Deploying Web Services with .NET” into a single “Web Services Guide”.
  - New “Deploy Corticon Server in an Application Guide”. This guide details how to use Corticon in-process in Java and .NET applications.
- **Support for Eclipse 4.9** - Corticon Studio has been updated to Eclipse 4.9. The previous release of Corticon Studio supported Eclipse 4.5. Any plugins you add to Corticon Studio must be compatible with Eclipse 4.9. See the Eclipse documentation for changes and new features in Eclipse 4.6 through 4.9.
- **Installation of Tomcat 9.0** - Corticon 6.0 now installs a standard distribution of Tomcat 9.0 with Corticon Server. The Progress Application Server (PAS) is no longer distributed with Corticon Server. See the Apache Tomcat documentation for more details on configuring Tomcat.
- **Project validation on demand** - Corticon Studio now allows the user to decide when to perform rule project validation. The automatic validation of rule projects is disabled by default. This eliminates excessive CPU usage performing validation when making multiple edits. To perform validation of a project, select **Validate Project** on the **Project** menu.
- **Vocabulary search and refactor performance** - Corticon Studio vocabulary search and refactor operations have been enhanced to not be dependent on the validation of your rule project. This allows search and refactor to function correctly when automatic validation is disabled.

- **WSDL and XSD export in Corticon Studio** - Corticon Studio now allows the export of Vocabulary and Ruleflow WSDL and XSD files. This simplifies publishing of service contracts for decision services. To export the WSDL or XSD, select `Export WSDL` or `Export XSD` from the Vocabulary or Ruleflow menu of the corresponding editor. See *"How to integrate Corticon Decision Services" in the Deployment Guide* for more information.
- **CDD command line generation** - The `corticonManagement` utility now supports generation of a CDD file. The CDD file generated can be used for deployment of decision services or manually edited and then used. See the topic *"Create a CDD file"* in the Deployment Guide.
- **Corticon Extension access to response data** - From your Corticon extensions you can now access the response data that is returned from a Decision Service execution. Extensions can retrieve this data to perform custom logging, or to raise alerts on Response or Rule Messages. For more information, see the topic *"Serialize Response and RuleMessages in Corticon extensions"* in the Extensions Guide.

## What has been removed in Corticon 6

---

This section summarizes the features that are no longer supported and no longer documented as of Progress® Corticon® 6.0.0:

- **Deployment of Ruleflows** - Corticon Server no longer supports deployment of ERF files, only compiled EDS files can be deployed. Previously you could deploy Ruleflows and associated assets to Corticon Server so that the Server would compile the assets into runnable Decision Services. That was a performance hit every time that Decision Service was loaded and could lead to error situations in production if the Decision Service failed to compile. You can generate EDS files with Corticon Studio, Corticon command line utilities, or Corticon ant macros. If deploying ERF files with Corticon Deployment Descriptors (CDD files), you will need to modify these to deploy EDS files. Eliminating ERF deployment helps enforce Continuous Integration/Continuous Deployment (CI/CD) best practices by encouraging the deployment of fully compiled and tested Decision Services. See the topics *"How to package and deploy Decision Services"* in the *Deployment Guide*.
- **Corticon Deployment Console** - The Corticon Deployment Console bundled with Corticon Server provided a GUI mechanism for creating CDD files, compiling Decision Services, and generating WSDL files for Decision Services. These tasks can each be performed in other ways.

*If you have been using the Deployment Console, its functionality is available as:*

- **Command line interface for creating CDD files** - The CDD file format is a simple text file manifest describing a Decision Service. The most common practice is to copy a sample CDD file bundled with Corticon, and then make modifications to it in a text editor. The new `corticonmanagement` command `-cdd` as described in *"Create a CDD file"* enables all the CDD options for a single Decision Service deployment description file in one command.
- **Compile Decision Services** - Multiple techniques in the Deployment guide perform compilation:
  - Using Studio to compile and deploy Decision Services
  - Automating packaging and testing of Decision Services
  - Using Server API to compile and deploy Decision Services
- **WSDLs are now generated from the Vocabulary and Ruleflow editors** - WSDL is bundled with a Decision Service on compilation. The WSDL for a Decision Service can be accessed via the Corticon Web Console or the Corticon Server REST API. To create WSDL, see *"Generate Service Contracts in Corticon Studio"*.
- **Download of Decision Services** - Corticon Studio no longer allows download of Decision Services. As rule assets can no longer be staged for compilation on the server, the best practice for managing your rule assets is to store them from Studio in a source code control system such as Git. Corticon Studio now includes plugins for using Git with your rule projects.
- **SOAP Management API (except execution)**- Corticon Studio no longer provides SOAP API to deploy or manage Decision Services. You will need to migrate to using the equivalent REST APIs. Note that the ability to execute a Decision Service with a SOAP request will not be removed – only the management APIs are affected. Corticon Server's REST APIs are accessible via Swagger on your deployed Corticon Server at <http://localhost:8850/axis/swagger> (use your appropriate host name and port number ).
- **Java Server Console** was previously removed from the Corticon Server install and documentation but was available in the server.zip download. It is no longer available. The Server Console has been replaced with the Corticon Web Console..
- **EAR file deployment** - Corticon Server no longer includes the `CcServer.ear` file as an option for deploying to an application server. Use the Corticon Server `axis.war` file for deployment to an application server. The `CcServer.ear` file was not included with the Corticon Server install. It was available only in the server.zip distribution.
- **Enterprise JavaBeans deployment** - With the removal of `CcServer.ear`, Corticon Server no longer provides support for EJB deployment.
- **.NET Business Objects** - Corticon no longer supports the use of .NET business objects for executing Decision Services in-process. The use of business objects for runtime execution was primarily done for performance. For performance critical in-process deployments, you should pass JSON to Corticon, not business objects. The performance of JSON is superior to that of business objects. If you have been using .NET business objects, contact your Progress representative for guidance in migration strategies.
- **Option useForQueryService in datasource.xml** - The `useForQueryService` option in a `datasource.xml` file for identifying the datasource to use for ADC queries has been removed. In its place, add a **Query** datasource to your Corticon vocabulary to identify the source of ADC and Batch queries. See *"Define and import queries for ADC" in the Data Integration topics*.
- **EDC database.properties file** - The option to specify EDC properties via a database.properties file has been removed. In its place, use a `datasource.xml` file. The `datasource.xml` file can define properties for multiple datasources. See *"Export the Datasource Configuration file" in the Data Integration topics* for more information.



- 
- **Passing REST API arguments in HTTP Header** - The Corticon Server REST API no longer supports passing arguments in the HTTP header. This mechanism did not fully support localization. In its place, pass arguments as query String parameters.
  - **Rule Execution Recording Service** - The Corticon Server rule execution recording service has been removed. In support of this, the Corticon Studio **Project > Create Execution Recording Schema** menu option has been removed. Similar functionality can be achieved using Corticon's ADC. Contact your Progress representative for guidance in migration strategies.
  - **WebConsole Monitoring of In-Process Corticon Servers** - The Corticon Server no longer provides an option to run an in-process Tomcat server for use when using Corticon in-process. This change results in in-process Corticon Servers not being monitorable with the Corticon Web Console. To monitor an in-process Corticon Server with the Web Console, your container application will need to proxy the Corticon REST API to the Corticon Server. Contact your Progress representative for guidance in migration strategies.
  - **ICcServer API Changes** - Previously deprecated methods in the ICcServer API have been removed. In addition, methods present for support of the Java Server Console have been removed. Users running Corticon in-process may need to use alternate method signatures with 6.0. See the [Corticon Server JavaDoc](#) for current API methods.
  - **Sample Extended Operators** - The SeMath and RandomGenerator extended operators have been removed.

