Progress®

What's new in Corticon

Progress® Corticon®

Copyright

Last updated with new content: Corticon 6.3.1

Updated: 2022/09/23

1

Learn about Corticon releases

Corticon 6.3 is a point release of Corticon 6 that provides improvements and added features, such as:

- Rule trace viewer
- Simplified JSON metadata requirements
- Updated REST Connector driver
  - REST data source OAUTH2 support
  - REST data source support of POST
- Updated bundled Eclipse
- Updated bundled OpenJDK to 11+
- Updated JNBridge
- Updated Tomcat
- Servers registered in the Web Console update for changed IPs
- Corticon on Docker
- **Corticon 6.3.1** provides ADC Insert Primary Key handling

Corticon 6.2 is a point release of Corticon 6 that provides improvements and added features, such as:

- Auto registration of Corticon servers in Web console.
- Improved refresh in Web Console.
- Enhanced automation enables reports for assets from a command line utility.
- Data source options are simplified to one for each brand.
- Projects no longer have to specify the addon Corticon Extensions ADCSco.jar or RESTSco.jar as those JARs are now embedded for every project.
- The REST API in static format is available in Java Server's install directory for Java and .NET, and at https://documentation.progress.com/output/Corticon/6.2.0/RESTDoc/.
- Certified for Java 11.
- Improvements to Ruletest reports.

Corticon 6.1 is a Long Term Supported (LTS) product release of Corticon 6 that provides improvements and added features, such as:

- New Microsoft Dynamics CRM data source for accessing Dynamics CRM data, presented as **Microsoft Dynamics 365** for accessing Dynamics 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 6.1.1** simplified the Corticon Studio interface to hide unused Eclipse menu commands to present a sleeker user experience.
- **Corticon 6.1.2** stopped including the WSDL and the report in the EDS file by default in Decision Service compilation.

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

- REST data source allowing you to access REST data from decision services.
- A new architecture for .NET server, one that provides remarkably faster throughput. For more information, see *What changed in Corticon 6*.
- The Deployment Console has been dropped. You can generate WSDL and XSD from new Ruleflow and Project menu actions. For more information, see *What was removed in Corticon 6*.

2

What's new and changed in Corticon 6.3

Corticon 6.3.1

Corticon 6.3.1 provides additional functionality to 6.3 with:

- **ADC Insert Primary Key handling**—When the primary key of an inserted record is generated by the connected database, Corticon retrieves this generated value and adds it to working memory for that Entity. This will allow follow-up database updates on that Entity to occur, and also allows associated Entities that are dependent on that primary key value to be stored as a foreign key value in the associated Entity.

For more information about ADC writes, see *"How to configure ADC writes" in the Corticon Data Integration Guide*.

Corticon 6.3

Progress® Corticon® 6.3 includes the following changes and new features:

- **Rule trace viewer**—Troubleshooting the execution of rules just got easier. When running Ruletests you now have the option to gather rule trace data. Trace data identifies the sequence of rules triggered and the actions performed. When execution of your Ruletest completes, trace data is displayed in the Rule Trace Viewer where you can see each rule triggered, changes to attribute values and associations, and more. From the Rule Trace Viewer you can easily navigate to the Rulesheet for any rule show in the Rule Trace Viewer. This makes it easy for you to quickly make changes to your rules and see the results in your Ruletests. You no longer need to add rule messages on every rule to trace rule execution. See *"Trace rule execution" in the Corticon Rule Modeling Guide*.

  Note: See how the rule trace view helped in a large project in the blog Fast Rules Diagnostics and Root Cause Analysis with the New Rule Trace Viewer.

- **Simplified JSON metadata requirements**—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. An example is shown in the Web Console Guide at *"Simplified JSON in requests" in the Web Console Guide*.

- **Updated REST Connection driver**—The Progress DataDirect Autonomous REST Connector driver has been updated and now supports OAuth2 Authentication and POST requests:
  - **REST data source support of OAUTH2**—Uses authorization tokens to prove an identity without giving away your password. See *"Authentication on REST Service connections" in the Corticon Data Integration Guide*.
  - **REST data source support of POST**—A **POST** request does not include parameters as part of the URL, instead the parameters are data in the request document. Selecting **Post** parameter type changes how the REST connector makes its requests to the endpoint, so any specified URL parameters may be ignored. See *"Parameters on REST Service connections" in the Corticon Data Integration Guide*.
- **Updated bundles for Eclipse, OpenJDK, Tomcat, and JNBridge**—For details, see Corticon Supported Platforms Matrix.
- **Self-registration update in Web Console**—When a Corticon Server self-registers with the Web Console, the Web Console now updates any existing registration for the server by updating the IP address of the registered server. Previously, a new registration would be added resulting in the Web Console having server registrations for both the old and new IP addresses. Updating an existing registration better supports deployments where the IP address of a Corticon Server might change on each restart. The Web Console can then provide continuity of trend data across restarts. See *"Server registration with Web Console" in the Corticon Server Guide*.
- **Corticon deployment on Docker**—The popular platform for building, sharing and running applications, Docker, now has detailed download and configuration instructions so that you can quickly deploy Corticon Servers onto Docker images. See *"How to deploy Corticon on Docker" in the Corticon Deployment Guide*.

3

What was new and changed in Corticon 6.2

Progress® Corticon® 6.2 included the following changes and new features:

- **Web Console server self-registration**—New in 6.2 is the ability to have Corticon Servers automatically register with the Corticon web-console. This eliminates the need to manually add Servers to the Web Console and better supports deployments where Servers have dynamic IP addresses or are started or stopped to meet elastic demands.
- **Web Console context**—In 6.2 the Web Console was enhanced to preserve the display context of tables on page refresh or redisplay. The context includes the page of data displayed in a table, state of what is expanded or collapsed, sort order, and more.
- **Command line report generation**—New in 6.2 is the ability to generate rule asset reports with command line utilities. This allows you to integrate report generation with your CI/CD practices.
- **Simplified datasource selection**—In Corticon Studio when configuring a datasource, the database server selection is no longer version-specific. For example, where you previously chose from Microsoft SQL Server 2012 through 2019, you now choose just Microsoft SQL Server. This change allows Corticon to certify new database versions as they are certified by Progress Data Direct without the need to re-release Corticon. This change is automatically applied when you upgrade assets after the product upgrade. The configuration file for deployment will be updated; however, previously generated configuration files in deployment need to be regenerated or manually adjusted to work with updated deployments.
- **Simplified ADC and REST datasource configuration**—In Corticon Studio the Corticon REST and ADC datasource jars are automatically available to your project. You no longer need to add the jars as project extensions. The process to upgrade assets that had explicit specifications of these JARs will not have those references removed. You will see warnings after upgrading a project. The condition is benign. The files can be removed from the Corticon Extensions section of the Project properties. However, the packaging and testing techniques in Studio will ignore the listed JARs.
- **Support for Java 11**—Corticon Server and Corticon Web Console have been certified on Java 8 and 11. Oracle JDK 8 is bundled with Corticon Server.

- **Bundled REST API documentation**—The Corticon REST API is available in Java Server's install directory for Java and .NET (but not propagated to the IIS Server). This provides an alternative to the Swagger REST API documentation bundled with Corticon Server for Java, The REST API documentation is also available online at https://documentation.progress.com/output/Corticon/6.2.0/RESTDoc/.
- **Removal of Support for OpenAccess 8.0 for OpenEdge**—Corticon support for OpenAccess 8.0 for OpenEdge is removed in this release. OpenAccess for OpenEdge was a database server option for Corticon EDC and ADC. Open Access 8.0 for OpenEdge is retired and no longer supported by Progress.

4

What was 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. Contact Progress support for details on the procedures. See the Data Integration topic *"Add your own database driver"* in the *Data Integration* guide for more information.

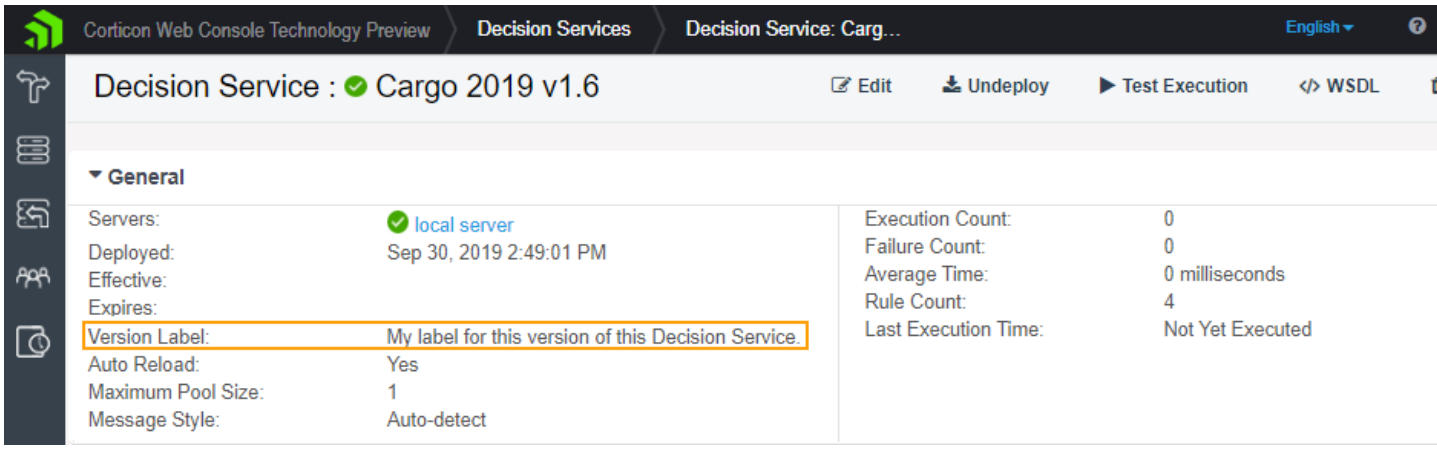
Database driver for Microsoft Dynamics and its authentication—Corticon 6.1 adds a new Microsoft Dynamics 365 datasource, allowing you to utilize the power of Corticon rules in your Microsoft Dynamics 365 applications. Invoke rules from Dynamics 365 applications using the Corticon REST API and your rules 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 Kerberos 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:



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.

6.1.1 update

This section summarizes changes in 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 plug-in is no longer included in the Studio installation. The plug-in can be accessed and updated from Eclipse.

6.1.2 update

This section describes a change in Corticon 6.1.2.

- Decision Service compilation no longer includes the WSDL and the report in the EDS file by default. If you use WSDL or reports in the EDS, add the following lines to the `brms.properties` file of the Studio that is producing the EDS:

  ```
  com.corticon.server.compile.add.wsdl=true
  com.corticon.server.compile.add.report=true
  ```

  For more information about this change, see the topic *"Properties that impact Decision Service compilation" in the Corticon Deployment Guide*.

5

What was 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.*

6

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.

7

What was 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 needed to be recompiled and reloaded due to changes to the rules, which could lead to error situations in production if the Decision Service failed to compile because all dependent files were not uploaded to the server. 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 -cdd` command as described in , 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" in the Corticon Deployment Guide .

Note: Decision Service compilation no longer includes the WSDL and the report in the EDS file by default. If you use WSDL or reports in the EDS, add the following lines to the `brms.properties` file of the Studio that is producing the EDS:

```
com.corticon.server.compile.add.wsdl=true
com.corticon.server.compile.add.report=true.
```

For more information, see the topic "Properties that impact Decision Service compilation" in the Corticon Deployment Guide.

- **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 Server 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**—Previously removed from the Corticon Server install and documentation yet 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 no longer provides sample EJBs that call into a running Corticon Server..
- **.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. Now that Corticon uses JNBridge for .NET executions your performance-critical in-process deployments can pass JSON or XML to Corticon, you should pass JSON or XML 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 Guide.
- **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 the section "Importing Datasource and Database Access configurations" in "Define the database connection for EDC" in the Data Integration Guide 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.
- **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 Corticon 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. The RandomGenerator was replaced by the Random operator for Decimal and Integer data types. Be sure that your rules replace the deprecated ones with the new operators.