Progress®

New in Corticon 6

Progress® Corticon®

Copyright

Updated: 2019/07/31

1

Key differences in Corticon 6

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

- Addition of REST APIs as a Datasource, transforming the REST sources into relational data.
- 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's new*
- *What's changed*
- *What's been removed*

2

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 as 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 *the Web Services section*.

3

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” from 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 *"Integrating 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 "Creating 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 "Serializing Response and RuleMessages in Corticon extensions" in the Extensions Guide.

4

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 *"Packaging and deploying 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.

Note:

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*
  - *Using command line utilities to compile 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 SVN or Git. Corticon Studio now includes plugins for using both SVN and Git with your rule projects.
- **SOAP Management API**- 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 topis.*

- **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 *"Exporting 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 JavaDoc for current API methods.