Progress®
Corticon®

What's new in Corticon

Copyright

Last updated: Corticon 7.2

Updated: 2025/08/27

1

Overview of Progress Corticon

Progress® Corticon® is the Business Rules Management System with the patented rules engine that enables you to automate sophisticated decision processes—without having to write code.

Progress Corticon products

Progress Corticon distinguishes its development toolsets from its server deployment environments.

- **Corticon Studio** is the Windows-based development environment for creating and testing business rules. Corticon Studio provides a complete Eclipse development environment for Corticon in the **Corticon Designer** perspective. You can use this Eclipse installation as the basis for adding other Eclipse tools.
- **Corticon deployment components** implement web services for deploying and managing business rules defined in Corticon Studio:
  - **Corticon Server for Java** is supported as in-process servers and on various application servers. After you install it on a supported Windows platform, its deployment artifacts can be redeployed on various UNIX and Linux web service platforms as Corticon Decision Services.
  - **Corticon Server for .NET** facilitates deployment of Corticon Decision Services on Windows .NET Framework and Microsoft Internet Information Services (IIS).
  - **Corticon Utilities** package scripts that maintain Corticon Java Server installations.
- **Management of multiple versions of Corticon Servers** The Corticon Web Console enables administration of multiple remote Corticon Servers of mixed versions. A Web Console server is accessed by users through authenticated web browser connections.

2

Learn about Corticon releases

Corticon 7.1.2

Corticon 7.1.2 in summary includes the following changes and new features:

- **Updated .NET Server prerequisites**- Removes dependencies on unsupported MS libraries. See the Corticon 7.1 Supported Platforms Matrix.
- **Corticon AI Assistant improvements** include:
  - Supports Azure Open AI.
  - Supports OpenAI (and Azure) 4.1 models.
- **Supports Windows Server 2025**.
- **Corticon Studio bundles Java 17.0.156**.

Corticon 7.2 in summary includes the following changes and new features:

- Azure OpenAI Integration
- AI Rule Generation from Plain text prompts
- AI Rule Generation from Code Segments
- Updated .NET Dependencies
- Upgrade to Jakarta Namespace
- Java 21 Support
- Updated to Eclipse 4.35
- Tomcat 10.1 & JBoss 8 Support
- Swagger Removed

- Database Support dropped for OpenEdge, HDP, and Snowflake

Corticon 7.1 in summary includes the following changes and new features:

- **AI Assistant**
  - Supports Azure Open AI.
  - Supports OpenAI (and Azure) 4.1 models.
- **Corticon Studio**
  - Ruletest Generation
  - Vocabulary Generation from JSON
  - Complexity Reports
  - Upgrade Wizard Report
  - Upgrade Wizard on Rule Projects
  - Simplified JSON and XML Mapping
- **Web Console**
  - One Web Console
  - License Visibility
  - Configuration Export and Import
  - Cached Entities for Batch Processing
- **Corticon Server**
  - JSON Array Support
  - EDS Version Enforcement
- **Corticon Installers**
  - Corticon Studio
  - Corticon Server
  - Corticon Server .NET
  - Corticon Web Console
  - Corticon Utilities
- **Java Requirement**
  - Java 17 Not Bundled with runtime components
  - Java 17.0.156 Bundled with Corticon Studio
- **Tomcat no longer bundled**

Corticon 6.3 in summary includes the following changes and new features:

- **Corticon 6.3.4** rolls up the various patch releases for 6.3.3 as well as:
  - Updated Web Console—The Web Console for 6.3.4 introduces the OneWebConsole, on par with the 7.1 Web Console, hardened against many CVEs, and offering new features for deployment visibility and management. The updated Web Console can manage deployed servers and their decision services from version 6.3 and version 7.1.
  - Log filter changes—PAYLOAD is new and INVOCATION has been dropped.
  - New server property—A new server property has been added to turn off disabling of thread underutilization allocation to control how execution threads are allocated when the Corticon Server is underutilized.
- **Corticon 6.3.3** rolls up the various patch releases for 6.3.2 as well as:
  - Decision service version enforced at deployment.
  - Swagger is optional.
  - Tomcat 8.5 not supported.
  - Support for multiple applications on .NET server.
- **Corticon 6.3.2** rolls up the various patch release for 6.3.1 as well as:
  - Updates the Corticon Web Console's LDAP authentication to support OpenLDAP.
  - Improved runtime performance of rulesheets containing filters that use entities not also used in a rule condition or action.
  - Improved decimal precision comparison in rule tests.
  - Modified the default logging configurations of the Web Console database to have trace logging disabled.
- **Corticon 6.3.1** provided ADC Insert Primary Key handling.
- **Corticon 6.3.0** 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 proxy for .NET
  - Updated Tomcat
  - Servers registered in the Web Console update for changed IPs
  - Corticon on Docker

Retired Releases

Corticon 7.0 is a major release of Corticon that provides improvements and added features, such as:

- Updated JNBridge
- Native JSON
- Swagger as an addon
- Support for Progress DataDirect Hybrid Data Pipeline and Snowflake as Database Servers
- Stronger licensing enforcement
- Generate Vocabulary from JSON
- Filter Vocabulary Similar to other filters in Studio, the filter can be a regular expression, and it is case sensitive.
- Filter and export Ruletest trace results
- Copy input from a Testsheet in a Ruletest and paste into another testsheet in the same Ruletest
- Deprecated features
- REST execute interface accepts `application/xml` as part of the request. SOAP/XML is still supported on deployment servers.
- Requires two third-party driver JARs.

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 retired 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 6.1.4 Included Swagger but requires steps to enable it. Corticon Server now enforces version check on EDS deployment. Support for multiple applications on .NET server.

- **Corticon 6.1.2** stopped including the WSDL and the report in the EDS file by default in Decision Service compilation.
- **Corticon 6.1.1** simplified the Corticon Studio interface to hide unused Eclipse menu commands to present a sleeker user experience.

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 6.0*.

3

What's new and changed in Corticon 7.2

Corticon 7.2.0

New in Corticon 7.2.0 are the following features and enhancements:

- **Azure OpenAI Integration –** New in 7.2, the Corticon AI Assistant also supports Microsoft Azure OpenAI as an LLM option. Securely configure your Azure API key and enjoy the latest advancements in AI-driven rule authoring.
- **AI Rule Generation from Plain text prompts** – New in 7.2, The Corticon AI Assistant helps you generate rules directly from plain-English prompts, making rule modeling faster and more intuitive. The Assistant automatically classifies prompts and generates rules in context, significantly reducing manual authoring effort.
- **AI Rule Generation from Code Segments**- New in 7.2, users can now paste code snippets (such as Java, JavaScript..) into the AI Assistant, which automatically translates the logic into Corticon rules. This speeds up migration from coded logic to no-code rules and bridges the gap between IT and business authors.
- **Updated .NET Dependencies**- New in 7.2, Corticon Server .NET dependencies have been upgraded to eliminate outdated components (like Microsoft C++ 2012 libraries), ensuring compliance.
- **Upgrade to Jakarta Namespace -** New in 7.2, the entire Corticon codebase and third-party libraries have migrated from javax.* to jakarta.* namespaces, keeping Corticon aligned with current Java EE standards. Relevant for customers who have custom extensions.
- **Java Support –** Corticon 7.2 Studio is now bundled with Java 21, and Corticon Server is fully compatible with Java 21, aligning with modern Java runtime requirements and long-term JVM improvements. Corticon WebConsole requires Java 17.
- **Corticon Web Console 7.1 Compatibility –** Corticon 7.2 includes Corticon Web Console 7.1 (latest release is 7.1.1). The 7.1 Web Console is fully compatible with Corticon 7.2

- **Updated to Eclipse 4.35** - Corticon Studio has been updated to Eclipse 4.35. Any plugins you add to Corticon Studio must be compatible with Eclipse 4.35. See the Eclipse documentation for changes and new features in Eclipse 4.25 through 4.35.
- **Tomcat 10.1 & JBoss 8 Support -** Corticon Server 7.2 is now certified and tested on Tomcat 10.1 and JBoss 8. See the Corticon Supported Platforms Matrix for full details on supported platforms.
- **Swagger Removed -**As part of our security hardening efforts, the Swagger UI component has been **removed** in 7.2. The **auto-generated REST API documentation** is still included with the Server distribution, we recommend customers use **Postman** or similar open-source tools for REST API testing going forward.
- **Database Support -** OpenEdge, HDP, and Snowflake are not currently supported.

  See the Corticon Supported Platforms Matrix for full details on supported platforms.

4

What's new and changed in Corticon 7.1

Corticon 7.1.2

The service pack release 7.1.2 extends AI development, and provides some customer-requested features:

- **Updated .NET Server prerequisites**- Removes dependencies on unsupported MS libraries. See the Corticon 7.1 Supported Platforms Matrix.
- **Log filter changes**—A new filter is available, PAYLOAD, that will put the request payload and the response payload in the log. The filter INVOCATION has been dropped.
- **Corticon AI Assistant improvements** include:
  - Supports Azure Open AI.
  - Supports OpenAI (and Azure) 4.1 models.
- **Supports Windows Server 2025**.
- **Corticon Studio bundles Java 17.0.156**.

Corticon 7.1.1

The service pack release 7.1.1 extends AI development, and provides some customer-requested features:

- **Corticon AI Assistant improvements** include:
  - AI Assistant has access to all constructs in a ruleflow including branches and service callouts. This allows you to use AI to explore any aspect of a ruleflow.
  - AI Assistant can be used on rule assets containing errors. This allows you to use AI to explore rule assets even if they have unresolved errors.
  - AI Assistant is visible by default in all new project workspaces. If you do not use the AI Assistant, you can close it.

- **New Utility for Custom Reporting**— The `corticonManagement` command now allows you to create a JSON representation of the contents of any ruleflow. The `genJson` utility is useful for customer reporting solutions capturing the specifics of a decision service to be deployed or other use cases where a simplified format detailing the contents of a decisionService is needed. See the Deployment Guide topic *"Export a decision service as a JSON file" in the section "Syntax of the compile and test commands," part of the "Automate packaging and testing of Decision Services topics under "How to package and deploy Decision Services."*
- **Updated REST Datasource**—The REST datasource has been updated to new version of the Progress Autonomous REST Connector. Existing deployments using the REST connector are unaffected. Rediscovering the schema of a REST datasource in Studio may introduce mapping errors that you will need to address.
- **Improved JSON Support in Tester**— The Tester now supports the copying native JSON from the clipboard into the input or expected sections of a ruletest. The prior Corticon annotated format is still supported. See *"How to import a JSON document to a testsheet" in the Quick Reference Guide*.
- **Log filter changes**—A new filter is available, `PAYLOAD`, that will put the request payload and the response payload in the log. The filter `INVOCATION` has been dropped. See *"Configure Logs" in the Server Guide*.
- **New Server Execution Property: Turn off disabling of thread underutilization allocation—** A new server property has been added to control how execution threads are allocated when the Corticon Server is underutilized:

  ```
  com.corticon.ccserver.allocation.disable.underutilization.algorithm =false
  ```

  By default, the allocation algorithm allows Corticon Server to temporarily exceed the configured maximum number of execution threads when the server is underutilized.

  Setting this property to **true** disables that behavior, ensuring that Decision Services will strictly adhere to the maximum thread limit, even if the server is not fully utilized.

  Default value: `false`

  Note: This property only takes effect if allocation is explicitly enabled.

  To enable thread allocation, the following property must be set to true. Default value is `false`:

  ```
  com.corticon.server.decisionservice.allocation.enabled=false
  ```

  See *"Server Execution Properties" in the Server Guide*.

Corticon 7.1.0

New in Corticon 7.1.0 are the following features:

- **AI Assistant** – New in 7.1, the Corticon AI Assistant enables the use of artificial intelligence (AI) in Corticon Studio to explore assets in your Corticon rule projects by asking natural language questions and viewing the AI-generated responses. See *"Corticon AI Assistant" in the Corticon Quick Reference Guide* for details.
- **Corticon Studio** – Corticon 7.1 Studio includes multiple new and modified features to enhance your rule development.
  - **Ruletest Generation** – New in 7.1, the ruletest generator provides the ability to automatically create unit tests for each rulesheet in a Ruleflow from an existing system test. This allows you to perform more targeted testing and more easily isolate issues. You can efficiently generate and run unit tests for individual rulesheets and ruleflow steps, improving their ability to isolate and fix problems. That simplifies the process of identifying issues in large decision services, helping both users and Progress support isolate changes or problems in behavior. Also reduces reliance on full system tests, making rule testing more efficient and allowing for targeted troubleshooting. See *"Generate Ruletests" in the Corticon Rule Modeling Guide*.
  - **Vocabulary Generation from JSON**– New in 7.1, the generation of a Corticon rule vocabulary from a JSON payload or JSON schema accelerates your rule project development and keeps your vocabulary

in sync with the payloads that will be processed. See *"Generate a Vocabulary" in the Corticon Rule Modeling Guide*.

- **Complexity Reports** – New in 7.1, complexity reports allow you to export CSV files with raw metrics about rule assets, such as the number of rules per rulesheet in a ruleflow. These reports can help you better understand and track complexity in your rule projects.
- **Upgrade Wizard Report**– New in 7.1, the upgrade wizard report provides a record of the files updated, or not updated, by the upgrade wizard.
- **Upgrade Wizard on Rule Projects** – Changed in 7.1, the upgrade wizard now operates only on Corticon rule projects. Previously, it operated on any file system folder with Corticon rule assets. This change addresses potential upgrade issues where project context is necessary to upgrade a rule asset.
- **Simplified JSON and XML Mapping**– Changed in 7.1, vocabulary elements now have separate fields for XML and JSON mappings. The mappings are used by Corticon Server to map an input payload to a decision service vocabulary.

- **Web Console** – Corticon 7.1 Web Console includes multiple new and modified features to better enable the management of Corticon Server deployments. See the *Corticon Web Console Guide* for details:
  - **One Web Console** – New in 7.1, you can manage multiple versions of Corticon Server from one instance of the Web Console. This is helpful with large Corticon deployments or during upgrades. The 7.1 Web Console can manage Corticon Server 6.0 and later instances.
  - **License Visibility** – New in 7.1, the Web Console provides greater visibility of the Corticon licenses deployed across your Corticon servers and alerts when a server's license is expiring.
  - **Configuration Export and Import**– New in 7.1, the ability to export the Web Console configuration and import it into a separate instance is useful during upgrades or rehosting of the Web Console.
  - **Cached Entities for Batch Processing** – New in 7.1, the option to define a set of entities to be cached during batched rule processing can accelerate batch rule execution by limiting the retrieval of redundant data.

- **Corticon Server**– Corticon 7.1 Server and Server .NET include new and modified features to simplify Corticon Server deployments. See “The Corticon Server Guide” for details.
  - **JSON Array Support** – New in 7.1, support for object arrays in JSON input payloads simplifies the integration of Corticon decision services with external services.
  - **EDS Version Enforcement**– New in 7.1, enforcement of an EDS version check when deploying a decision service will reject a decision service deployment if the Corticon version used to compile the decision service is not compatible with the version of Corticon Server.

- **Corticon Installers** – Changed in 7.1, Corticon now consists of 5 separate installers, providing you greater control over installations and updates. See the *Corticon Installation Guide* for details.
  - **Corticon Studio**– Provides a complete platform for modeling rules for Corticon.
  - **Corticon Server** – Provides the Corticon Server for Java for deployment to supported J2EE application servers and Java applications.
  - **Corticon Server .NET** – Provides the Corticon Server .NET for deployment to Microsoft IIS application server and .NET applications.
  - **Corticon Web Console** – Provides a Web UI for managing Corticon Server and Server .NET deployments.
  - **Corticon Utilities** – Provides command-line utilities for automating the packaging, testing, and deployment of Corticon decision services.

Note: A Corticon Server install provides the Docker configuration file, `Dockerfile`, and the `.war` files for Corticon Server and Corticon Web Console. See *"How to deploy Corticon on Docker" in the Deployment Guide*.

- **Java Requirement** – Changed in 7.1 is the requirement for Java 17 and the packaging of Java. See “The Corticon Supported Platforms” for details on supported Java distributions.
  - **Java 17 Not Bundled** – Java is no longer packaged with Corticon Server, Server .NET, Web Console, or Utilities. You need to download and install Java 17 to use any of these Corticon installations.
  - **Java 17 Bundled with Corticon Studio** – Java is bundled with Corticon Studio, enabling you to use Studio without a separate Java 17 installation. The distribution of Java bundled with Studio should not be used with any other Corticon components.

  - **Note: Cannot install Studio into an existing Eclipse**— Corticon Studio ships with Java 11 or higher, which introduces changes in how JAR files are handled: Java no longer supports the direct addition of custom JARs to the classpath as earlier versions did. As a result, the ability to drop extension JARs into the Eclipse plugin folder is no longer functional. The recommended method for adding extension JARs is in an installed Corticon Studio either:
    - For Eclipse tools, choose **Help > Install New Software**, and then typically choosing tools at `https://marketplace.eclipse.org/` .
    - For your extensions, such as extended operators and service callouts, in a Rule Project, choose **Properties > Corticon Extensions.**

- **Tomcat Bundling** – Changed in 7.1 is the removal of an Apache Tomcat distribution from Corticon Server and Corticon Web Console. If deploying to Tomcat, you need to download and install Tomcat 9 and manually deploy Corticon components. See Corticon Supported Platforms Matrix for details on supported application servers.

Note: The Corticon documentation refers to the prior default port value of 8850. Substitute the port you chose in your application server configuration.

5

What's new and changed in Corticon 6.3

Corticon 6.3.4

Corticon 6.3.4 rolls up the various patch releases for 6.3.3 as well as:

- **Updated Web Console**—The Web Console for 6.3.4 introduces the OneWebConsole, on par with the 7.1 Web Console, hardened against many CVEs, and offering new features for deployment visibility and management. The updated Web Console can manage deployed servers and their decision services from version 6.3 and version 7.1. Corticon 6.3.4 and later bundle the Web Console from Corticon 7.1. To keep documentation streamlined, all Web Console documentation lives in the 7.1 library—no separate “6.3.4” Web Console guide is available. The benefits are:
  - Updated UI with improved behavior and security
  - Ability to manage all 6.3 and later Corticon Servers from one console. Wherever you see “Web Console” in any 7.1 topic, it applies equally to your 6.3 environment.

  For Corticon 7.1 Web Console documentation, see the 7.1 archive.

  For Corticon 6.3.3 Web Console documentation, see the 6.3 archive item for 6.3.3
- **Log filter changes**—A new filter is available, PAYLOAD, that will put the request payload and the response payload in the log. The filter INVOCATION has been dropped.
- **New server property: Turn off disabling of thread underutilization allocation --** A new server property has been added to control how execution threads are allocated when the Corticon Server is underutilized:

  ```
  com.corticon.ccserver.allocation.disable.underutilization.algorithm =false
  ```

  See *"Server Execution Properties" in the Server Guide* for details.

Corticon 6.3.3

Corticon 6.3.3 rolls up the various patch releases for 6.3.2 as well as:

- **Corticon Server now enforces version check on EDS deployment**—When an EDS is deployed, Corticon Server will inspect the EDS to determine which version of Corticon it was generated with. If this version is not the same major.minor version of Corticon Server, the EDS will not be deployed. Examples:
  - - 6.3.0.0 EDS deployed to 6.3.3.0 Corticon Server: Allowed
  - - 6.1.0.0 EDS deployed to 6.3.3.0 Corticon Server: Disallowed

  Enforcing this version check ensures compatibility between the EDS and Corticon Server. The generated EDS code is dependent on internal Corticon Server APIs which may change across major.minor releases. This could result in an EDS failing to execute despite initially reporting successful deployment.
- **Swagger is optional**—Corticon Server no longer bundles Swagger for exploring and testing the Corticon Server REST API. Swagger is now optional and can be added to an instance of Corticon Server. See *"The REST API Swagger documentation" in the Server Guide* for details. The use of Swagger is best limited to development deployments of Corticon Server. It is typically not needed, or wanted, in production deployments.
- **Tomcat 8.5 not supported**—Tomcat 8.5 will reach end-of-life in March 2024. Customers are encouraged to update to Tomcat 9.0 to ensure access to Tomcat security updates. Corticon Server bundles Tomcat 9.0

Corticon 6.3.2

Corticon 6.3.2 rolls up the various patch releases for 6.3.1 as well as:

- **Updates the Corticon Web Console's LDAP authentication to support OpenLDAP**—The Web Console supports both Microsoft Active Directory and OpenLDAP.
- **Improved runtime performance of rulesheets**—Rulesheets containing filters that use entities not also used in a rule condition or action can expect improved runtime performance.
- **Improved decimal precision comparison in rule tests**—Previously, for Decimal data types, the output and expected values might not be flagged as different when they in fact are. As an example, the value 111.100002 would not be flagged as different from 111.100001. These values will now be flagged as differences in rule tests.
- **Logging configurations in Web Console for trace logging are disabled**—Modified the default logging configurations of the Web Console database to have trace logging disabled. This is a change from the previous configuration, in which trace logging was enabled by default.

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 .NET proxy**—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*.

Note: Cannot install Studio into an existing Eclipse— Corticon Studio ships with Java 11 or higher, which introduces changes in how JAR files are handled: Java no longer supports the direct addition of custom JARs to the classpath as earlier versions did. As a result, the ability to drop extension JARs into the Eclipse plugin folder is no longer functional. The recommended method for adding extension JARs is in an installed Corticon Studio either:

- For Eclipse tools, choose **Help > Install New Software**, and then typically choosing tools at `https://marketplace.eclipse.org/` .
- For your extensions, such as extended operators and service callouts, in a Rule Project, choose **Properties > Corticon Extensions.**

6

Retired Releases

When releases are retired, they can still be downloaded and their documentation can be accessed as PDFs through the documentation archives. However, any new features and functionality in active releases are not added to the retired releases, and not updated in the documentation.

You are encouraged to upgrade to the current release and, if necessary, to upgrade to an earlier active release that supports the platform constraints you require.

For details, see the following topics:

- What was new in Corticon 7.0
- What was new in Corticon 6.2
- What was new in Corticon 6.1
- What was new in Corticon 6.0
- What changed in Corticon 6.0
- What was removed in Corticon 6.0

What was new in Corticon 7.0

Added in Corticon 7.0.0.2

This section list a feature added in Progress Corticon 7.0.0.2:

- **Corticon Server now enforces version check on EDS deployment**—When an EDS is deployed, Corticon Server will inspect the EDS to determine which version of Corticon generated it. If the version is not the same major.minor version of Corticon Server, the EDS will not be deployed.

  For example:

  - 7.0.0.0 EDS deployed to 7.0.0.1 Corticon Server: Allowed

  - 6.3.4.0 EDS deployed to 7.0.0.1 Corticon Server: Not Allowed

  Enforcing version checks ensures compatibility between the EDS and Corticon Server. The generated EDS code is dependent on internal Corticon Server APIs which might change across major.minor releases which could result in an EDS failing to execute despite initially reporting successful deployment.

  When deploying an incompatible EDS, Corticon Server performs the validation check and, if it is a deployment error, it is logged.

Added or changed in Corticon 7.0

This section summarizes the features that were added or changed in Progress® Corticon® 7.0:

- **New Web Console**—The Web Console has been updated to provide a better user experience and improve security. The new Web Console will be readily familiar to existing users yet with an improved appearance and behavior. The third-party components used by the Web Console have been updated to address known CVEs. See the updated *Web Console Guide*.
- **Native JSON**—New in 7.0 is support for unannotated JSON payloads when calling a decision service. Previously, a JSON payload required Corticon specific annotations to allow Corticon to map a payload to the Corticon vocabulary. This complicated integration of Corticon with external applications by requiring passing JSON data in “Corticon’s format”, not the “native format” of the application. Eliminating the need for these annotations simplifies integrating Corticon with your applications. The annotations can still be provided but are optional.
- **REST/XML**—New in 7.0 is the ability to call decision services via REST with XML payloads. This benefits applications which utilize XML but want to call Corticon with the simplicity of REST, as opposed to SOAP. Both REST/JSON and SOAP/XML are still supported.
- **Embedded Analytics**—New in 7.0 is the ability to capture the details of rule execution. This enables auditing of individual decision service executions and analysis of overall decision service behavior. Corticon 7.0 provides a new extension point to Corticon Server where you can provide an analytics handler for persisting analytics data. This gives you the flexibility to store analytic data in the database and format of your choice. See *"How to use the Corticon Analytics Handler" in the Corticon Extensions Guide*. A reference example of an analytics handler for persisting and reporting on analytics data in a SQL database is available.
- **Snowflake Datasource**—New in 7.0 is support for the popular cloud Snowflake database. Corticon 7.0 bundles a Snowflake database driver allowing use of Snowflake as a datasource with Corticon’s EDC and ADC data integration options.
- **Progress Hybrid Data Pipeline**—New in 7.0 is support for the Progress Hybrid Data Pipeline (HDP). HDP provides secure, centralized access to datasources. It’s a great option for decision services deployed to the cloud that need access to on-premise datasources. HDP can be used as a datasource with Corticon’s EDC and ADC data integration options. See *"Supported Databases" in the Data Integration Guide*.

- **Export of Rule Trace Data**—New in 7.0 is the ability export rule trace data from a ruletest in Corticon Studio. After running tests in Corticon Studio with rule tracing enabled, you can now export the rule trace data for analysis with tools like Microsoft Excel. This allows you to perform deeper analysis of the data and even compare the data to previous executions.
- **Rule Trace Filters**—New in 7.0 is the ability to filter the Rule Trace view in Corticon Studio. Ruletests can produce a large amount of trace data, making it hard to find data of interest. The Rule Trace view now allows you to filter this data to just the data of interest.
- **Rule Tester Copy/Paste Clipboard**—New in 7.0 is the ability in copy ruletest data to and from the Windows clipboard. When working with ruletests in Corticon Studio, you can now copy/paste ruletest data to/from the Windows clipboard in both XML and JSON format. This simplifies the creation of ruletests.
- **Vocabulary Filters**—New in 7.0 is the ability to filter the vocabulary view. Vocabularies can be very large, the ability to filter the vocabulary view makes it easier to find and focus on the elements of interest. For more information about the Ruletest window, see *"Ruletest window" in the Corticon Quick Reference Guide*.
- **Expand and Collapse Vocabulary views**—You can extend the Vocabulary view to show all its Entities, their Attributes, and their Associations. You can also drop the Vocabulary view down to just listing Entities.
- **Vocabulary Generation Naming Conventions** - When generating a vocabulary from a datasource in Corticon Studio, Corticon will now force the generated vocabulary to comply with the Corticon vocabulary naming conventions.
- **Stronger Licensing Enforcement**—When your Corticon Studio license expires, you are alerted at startup. Acknowledgement of the alert will exit the startup. There is no "read-only" mode. Contact your Progress Corticon representative to obtain an updated license file.
- **Installer Changes**—The Corticon installers now prompt for a Corticon license during installation. This is optional. If provided, Corticon will copy the license to the necessary locations based on the type of installation. In addition, the default work directories have changed to have separate default work directories for Corticon Studio and Server.
- **Simplified .NET configuration**—Installation of Corticon Server to IIS server now provides a folder structure that makes it easy to create multiple IIS applications that share one .NET Server. See *"Set up Corticon Server for .NET for multiple applications" in the Corticon Web Services Guide*.
- **Command Line Utility Licenses**—The Corticon command line utility "corticonManagement" now provides full functionality with either a Corticon Studio or Server license.
- **Swagger Optional**—Corticon Server no longer bundles Swagger for exploring and testing the Corticon Server REST API. Swagger is now optional and can be added to an instance of Corticon Server. See *"The REST API Swagger documentation" in the Server Guide* for details. The use of Swagger is best limited to development deployments of Corticon Server. It is typically not needed, or wanted, in production deployments.
- **Optional Service Contracts and Reports in Decision Service** - Decision Service compilation no longer, by default, includes the decision service WSDL and reports in the EDS file. To include these, add the following lines to the brms.properties file:

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

- **Java 17** - Corticon 7.0 supports Java 17. See the Corticon Supported Platform Guide for full details on supported platforms.
- **Third Party Updates**—Corticon 7.0 includes updates to many third party components to provide support for the latest platforms and address known CVEs.

Removed in Corticon 7.0

This section summarizes the features that are no longer supported as of Progress® Corticon® 7.0:

- **Corticon Studio SOAP Execution**—Corticon Studio no longer supports the use of SOAP for execution of ruletests against remote Corticon Servers. The benefit of SOAP execution was the ability to pass the test payload as XML to the remote server. With the addition of REST/XML support in 7.0, you can now pass XML via REST when running ruletests in Corticon Studio. The default is JSON. Note, this change only impacts Corticon Studio. Corticon Server's support of SOAP execution is unchanged.
- **Monitored Attributes**—Corticon Web Console no longer supports the monitored attributes feature. The new 7.0 analytics feature provides greater flexibility for gaining insight into rule execution.
- **database.properties**—Earlier versions of Corticon Server supported the use of a database.properties file to define connection information for a database. This was superseded in Corticon 5.7 by the datasource.xml file, it provides the ability to define connection information for multiple datasources. The old database.properties option is no longer supported.
- **Pre Corticon 5.5.1 CDD format**—A CDD file is a manifest of decision services to be deployed to Corticon Server. It's one of multiple options for deploying decision services. The format of CDD files changed in Corticon 5.5.1. The old format is no longer support, all CDD files must be in the latest format.
- **Create Update DB Schema**—Corticon Studio no longer supports the generation of a database schema from a rule vocabulary. Best practice is to generate your rule vocabulary from a database schema, this feature is unchanged.
- **CDT Import**—Corticon Studio no longer supports the import of enumerations into custom data types in a rule vocabulary. The CDT editor in Corticon Studio supports the pasting of one or more values into an enumeration. This can be used to populate the labels and values in a CDT enumeration.
- **testServer.bat**—Corticon Server no longer includes the utility testServer.bat. This provided a mechanism to run Corticon Server with a command line interface for testing Corticon Server features.
- **Java 8**—Corticon 7.0 does not support Java 8. Corticon 7.0 supports both Java 11 and Java 17.
- **Other Platform Changes**—See the Corticon Supported Platform Guide for the list or supported operating systems, databases, application servers, and more.
- **Tomcat 8.5 not supported**—Tomcat 8.5 will reach end-of-life in March 2024. Customers are encouraged to update to Tomcat 9.0 to ensure access to Tomcat security updates. Corticon Server bundles Tomcat 9.0.
- **Installation of Corticon Studio into an existing Eclipse is no longer supported.**

What was new 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.

What was new in Corticon 6.1

6.1.4 update

This section describes changes in Corticon 6.1.4:

- **Simplified .NET configuration**—Installation of Corticon Server to IIS server now provides a folder structure that makes it easy to create multiple IIS applications that share one .NET Server. See *"Set up Corticon Server for .NET for multiple applications" in the Corticon Web Services Guide*.
- **Swagger Optional**—Corticon Server no longer bundles Swagger for exploring and testing the Corticon Server REST API. Swagger is now optional and can be added to an instance of Corticon Server. See *"The REST API Swagger documentation" in the Server Guide* for details. The use of Swagger is best limited to development deployments of Corticon Server. It is typically not needed, or wanted, in production deployments.
- **Corticon Server now enforces version check on EDS deployment**—When an EDS is deployed, Corticon Server will inspect the EDS to determine which version of Corticon generated it. If the version is not the same major.minor version of Corticon Server, the EDS will not be deployed.

  For example:

  - 5.7.4 EDS deployed to 6.1.0.0. Corticon Server: Not Allowed

  - 6.1.0.8 EDS deployed to 6.1.0.0 Corticon Server: Allowed

  Enforcing version checks ensures compatibility between the EDS and Corticon Server. The generated EDS code is dependent on internal Corticon Server APIs which might change across major.minor releases which could result in an EDS failing to execute despite initially reporting successful deployment.

  When deploying an incompatible EDS, Corticon Server performs the validation check and, if it is a deployment error, it is logged.

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*.

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.

Corticon 6.1.0

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.

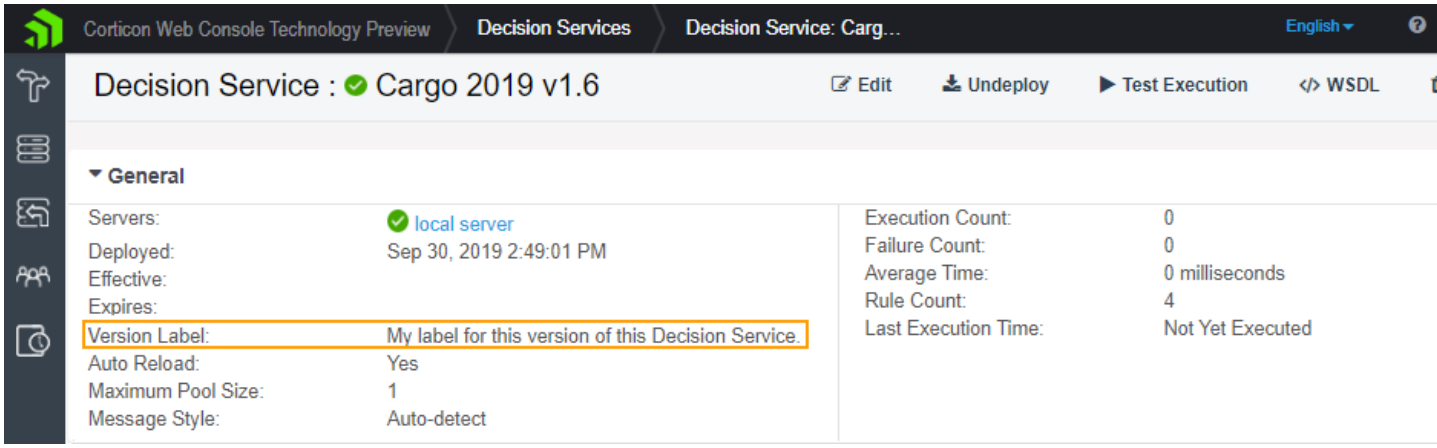
Database driver for Microsoft Dynamics and its authentication—Corticon 6.1 adds a new Microsoft Dynamics 365 datasource, allowing you to utilize the power of Corticon rules in your Microsoft Dynamics 365 applications. Invoke rules from Dynamics 365 applications using the Corticon REST API and your rules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

What was new in Corticon 6.0

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 6.0

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.

What was removed in Corticon 6.0

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:

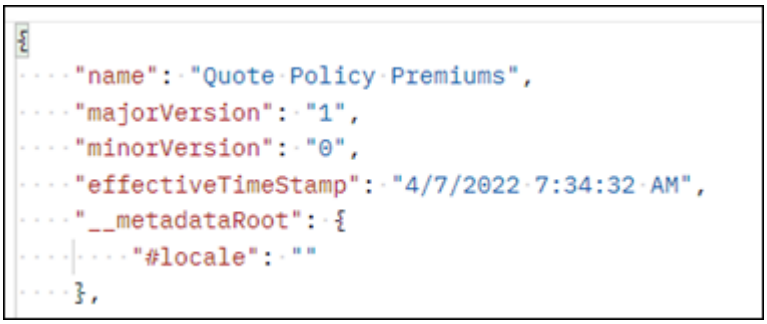
  - **Command line interface for creating CDD files**—The CDD file format is a simple text file manifest describing a Decision Service. The most common practice is to copy a sample CDD file bundled with Corticon, and then make modifications to it in a text editor. The new `corticonmanagement -cdd` command as described in "Create a CDD file" in the Corticon Deployment Guide, 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:
    - Use Studio to compile and deploy Decision Services
    - Use the `corticonManagement` utility to automate packaging and testing of Decision Services
    - Use Server API to package and deploy Decision Services
  - **WSDLs are now generated from the Vocabulary and Ruleflow editors**—To create WSDL, see "Generate Service Contracts in Corticon Studio" 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. See "Test the installed Corticon Server on Java" in the Web Services Guide. See

Note: Corticon Server's REST APIs are accessible:

- For versions prior to 7.2 on your deployed Corticon Server at http://localhost:8850/axis/swagger (use your appropriate host name and port number)
- For your Corticon Server for .NET running on IIS, Swagger is not available. Instead, see Corticon Server .NET APIs

- **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 technology for .NET executions your performance-critical in-process deployments can pass JSON or XML to Corticon, you should do that, 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, as illustrated:



- **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.