PROGRESS® Corticon®

What's New in Corticon

PROGRESS

Notices

Third party acknowledgements are listed in the Corticon Release Notes that accompany this release.

1

Overview of Progress Corticon

Progress® Corticon® is the Business Rules Management System with the patented "no-coding" rules engine that automates sophisticated decision processes.

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:
  - When installed as a standalone application, Corticon Studio provides the complete Eclipse development environment for Corticon as the **Corticon Designer** perspective. You can use this fresh Eclipse installation as the basis for adding other Eclipse toolsets.
  - When installed into an existing Eclipse such as the **Progress Developer Studio** (PDS), our industry-standard Eclipse and Java development environment, the PDS enables development of Corticon applications in the **Corticon Designer** perspective that integrate with other products, such as Progress OpenEdge.

  Note: Corticon installers are available for 64-bit and 32-bit platforms. Typically, you use the 64-bit installer on a 64-bit machine, where that installer is not valid on a 32-bit machine. When adding Corticon to an existing Eclipse, the target Eclipse must be an installation of the same bit width. Refer to the *Corticon Studio: Installation Guide* to access, prepare, and install Corticon Studio.

  Studio Licensing - Corticon embeds a time-delimited evaluation license that enables development of both rule modeling and Enterprise Data Connector (EDC) projects, as well as testing of the projects in an embedded Axis test server. You must obtain Studio development licenses from your Progress representative.

- **Corticon Servers** implement web services for business rules defined in Corticon Studios:
  - **Corticon Server for deploying web services with Java** is supported on various application servers, and client web browsers. After installation on a supported Windows platform, that server installation's deployment artifacts can be redeployed on various UNIX and Linux web service platforms as Corticon Decision Services. The guide *Corticon Server: Deploying web services with Java* provides details on the full set of platforms and web service software that it supports, as well as installation instructions in a tutorial format for typical usage.
  - **Corticon Server for deploying web services with .NET** facilitates deployment of Corticon Decision Services on Windows .NET Framework and Microsoft Internet Information Services (IIS). The guide *Corticon Server: Deploying web services with .NET* provides details on the platforms and web service software that it supports, as well as installation instructions in a tutorial format for typical usage.

  Server Licensing - Corticon embeds a time-delimited evaluation license that enables evaluation and testing of rule modeling projects on supported platform configurations. You must obtain server deployment licenses and server licenses that enable the Enterprise Data Connector (EDC) from your Progress representative.

2

What’s new and changed in Corticon 5.4.1

This chapter summarizes the new, enhanced, and changed features in Progress® Corticon® 5.4.1 -- a Service Pack that is applied to a corresponding product's 5.4.0 installation.

This Corticon release coordinates with other Progress Software releases:

- Progress OpenEdge is available as a database connection. You can read from and write to an OpenEdge database from Corticon Decision Services. OpenEdge applications can also call Corticon Decision Services, allowing business rules to be defined in Corticon. OpenEdge is a separately licensed Progress Software product.
- Progress DataDirect Cloud (DDC) enables simple, fast connections to cloud data regardless of source. DataDirect Cloud is a separately licensed Progress Software product.
- Progress RollBase enables Corticon rules to be called from Progress Rollbase. Rollbase is a separately licensed Progress Software product.

For details, see the following topics:

What's new in Service Pack 5.4.1

Corticon Studio and Servers provide several feature and performance improvements in this service pack release. Significant items include:

- **Expanded support of JSON/RESTful requests and responses:**
    - **Tests can generate JSON into logs and output files** - A new Corticon Studio property lets you specify the data format of the Tester Input Tree that is used for test executions. When you set the property `com.corticon.tester.ccserver.execute.format=JSON` in the Studio's `brms.properties` file, you replace its default value of `XML`. Then, `RULETRACE`-level logs will echo the JSON formatting of the test's input and output. Tester's import and export menu options will change to allow the import the import or export of JSON. For more information, see *"Ruletest menu commands" in the Corticon Studio: Quick Reference Guide*. For examples of the XML and JSON formatting of Corticon Requests and Responses, see the excerpts of Studio test server logs listed in *the topic "Content of Corticon server logs" in the Corticon Server: Integration and Deployment Guide*
    - **Corticon requests and responses are now supported in JSON/RESTful format on all Corticon Servers.**

      For more information, see the following topics in the *Integration and Deployment Guide*:

        - "Decision-service-level JSON/RESTful messaging style: Creating a JSON request message for a Decision Service" in the "Service contract and message samples" section.
        - "Examples" topic "Sample JSON request and response messages"
        - "REST call" topic in the "Invoking Corticon Server" section.

    - Specifics of running JSON requests are discussed in the Java and .NET Server Guides:
        - **Java Server** - See the topic "Path 4 - Using JSON/RESTful client to consume a Decision Service on Java server" in the "Consuming a Decision Service" section of the *Deploying Web Services with Java Guide*
        - **.NET Server** - See the topics "Path 4 - Using JSON/RESTful client to consume a Decision Service on .NET server" and "Path 5 - Using bundled JSON/REST sample code to consume a Decision Service" in the "Consuming a Decision Service" section of the *Deploying Web Services with .NET Guide*

          Note: If you intend to use JSON\RESTful services on the .NET Server, you must turn on all the **Windows Features** in the group **Common HTTP Features**, as shown:

          ⊟ ☑ Common HTTP Features

    - **New Server Property to handle JSON nulls** - A new server property, `com.corticon.server.execution.json.null`, provides the option to define what null means when processing a JSON payload. A JSON Object will not have null value for an Attribute. If the value is null when added to the JSON object, the Attribute will be removed from the JSON Object. This option tells the Server to treat null values in either of two ways:
        - **`JSON_NULL`** - The default value. When this is set, a user can pass in a string value of `"#null"` for an Attribute to signal a null value. This allows a user to pass in a name/value

pair and to have the value set to null when the JSON is converted into Corticon Data Objects.

- **`JAVA_NULL`** - When this is set, the Corticon Server will set a null value into the Attribute when a Rule sets the Attribute to null. This will cause the Attribute to be removed from the JSON Object. For more information, see *the detailed "Server Properties" in the "Integration and Deployment Guide".*

An example of cases where a rule sets a null value under each setting, see the topic "Passing null values in messages" in the "Integrating Corticon decision services" section of the *Corticon Server: Integration and Deployment Guide*.

- **Discussion of the handling of JSON formatting for datastore identity** -

  See the notes in *the topic "Metadata for Datastore Identity in XML and JSON Payloads" in the "Implementing EDC" section of the Corticon Server: Integration and Deployment Guide*

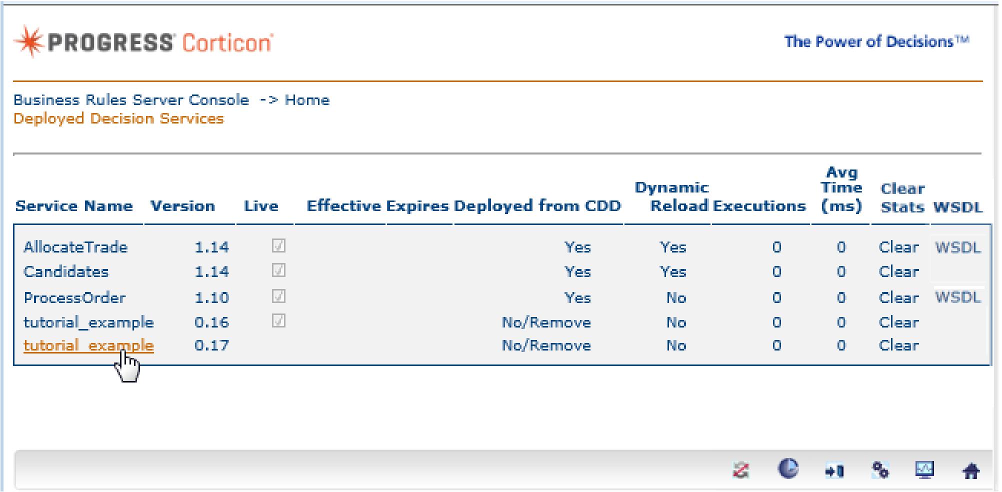
- **Enhancement to Enterprise Data Connections:**

  - **New Server property to handle nulls in EDC synchronization** - A new server property, `com.corticon.server.execution.sync.nulls`, handle nulls in EDC synchronization. When a record is retrieved from the database during a query that uses EDC to access a database, standard behavior is to synchronize the retrieved record to what was passed in the payload. If the payload value is null, this property will determine if that null value is set into the retrieved record (`true`) or if the null value is ignored (`false`), the default value, which means that the null value in the payload will be ignored, and the value from the database will be honored. For more information, see *the detailed "Server Properties" in the "Integration and Deployment Guide".*

- **Enhancements to deployment of Decision Services:**

  - **A Decision Service on .NET Server can be compiled into an Assembly.dll** - Compiling a Decision Service's `.eds` file into an `Assembly.dll` file provides a marked reduction in load time. The resulting `Assembly.dll` is added to the `.eds` file. Generated stubs of Business Objects can be included in the compilation. See *the topic "Compiling a Decision Service into an Assembly DLL" in the "Deploying Web Services with .NET Guide">*

  - **Every Decision Service on Java Server embeds its WSDL** - When you compile assets to create a Decision Service, the resulting `.eds` file will embed its Decision Service-level WSDL service contract. This greatly simplifies interface communications and ensures synchronization. The Corticon Server Console now provides a link on each Decision Service version that lets you open the WSDL on Decision Services that have one attached, as shown:



PROGRESS Corticon

The Power of Decisions™

Business Rules Server Console -> Home
Deployed Decision Services

Service Name	Version	Live	Effective	Expires	Deployed from CDD	Dynamic Reload	Executions	Avg Time (ms)	Clear Stats	WSDL
AllocateTrade	1.14	☑			Yes	Yes	0	0	Clear	WSDL
Candidates	1.14	☑			Yes	Yes	0	0	Clear	
ProcessOrder	1.10	☑			Yes	No	0	0	Clear	WSDL
tutorial_example	0.16	☑			No/Remove	No	0	0	Clear	
tutorial_example	0.17				No/Remove	No	0	0	Clear	

Existing decision Services will need to be recompiled to generate their embedded WSDL. This feature is an option, enabled by default, that can be turned off by setting the Server Property `com.corticon.server.compile.add.wsdl = false` in the server's `brms.properties` file. If you turn it off, subsequent Decision Services will not embed WSDL, while any `.eds` files that do have it will retain it. You can still generate Service Contracts from the Deployment Console as separate files. For more information, see *the topic "Decision Services page" in the "Using the Server Console" section of the "Deploying Web Services with Java Guide".*

- **Guidance on enabling HTTPS in Corticon Server for Java** - When you install and use the default application server, Pacific Application Server, you specify an HTTPS port. A section in the Java Server guide helps you set up SSL and HTTPS. For more information, see *the topic "Enabling HTTPS" in the "Deploying Web Services with Java Guide".*

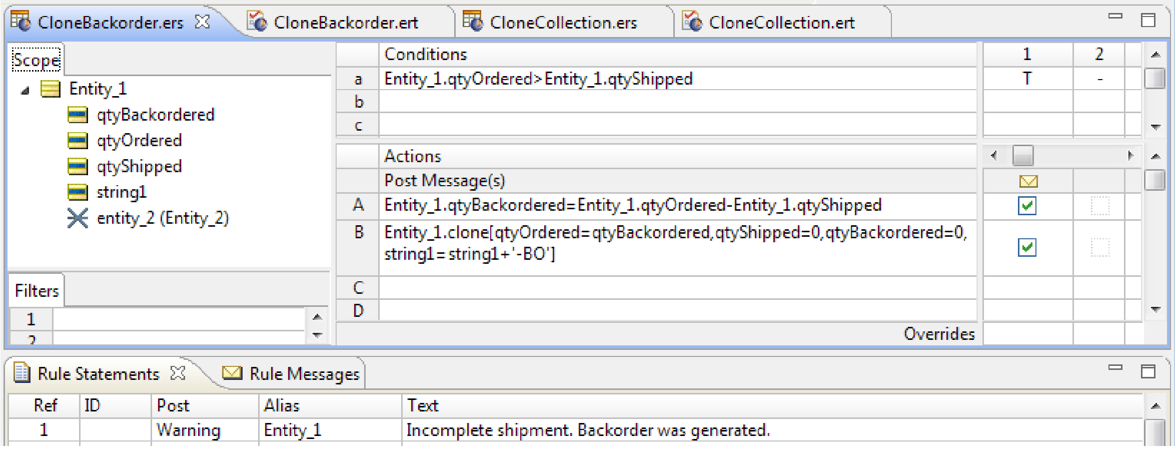
- **Addition to the Corticon Rule Language:**

  - **Entity.clone operator** - A Rule Operator has been added to extend the `.new` function to a provide a complete **clone** of the original entity. Now, instead of *adding a list of attributes* to a *new* entity, you can *create a clone* -- a complete copy of the entity with all its attribute values -- and then override just the attribute values that need to be different from those on the original entity. The syntax of the `.clone` operator is:

```
<Entity>.clone[<Expression1>,<Expression2>…]
```

Note: The implementation is a *shallow clone* -- associations are not duplicated.

The following Rulesheet uses **.clone** to create a new `Entity2` element when the value of `qtyOrdered` in `Entity1` is greater than the `qtyShipped` value. An alias is not required by the **.clone** operator, because it is possible to create a new entity at the root level, without inserting it into a collection.



	Conditions	1	2
a	Entity_1.qtyOrdered>Entity_1.qtyShipped	T	-
b			
c			

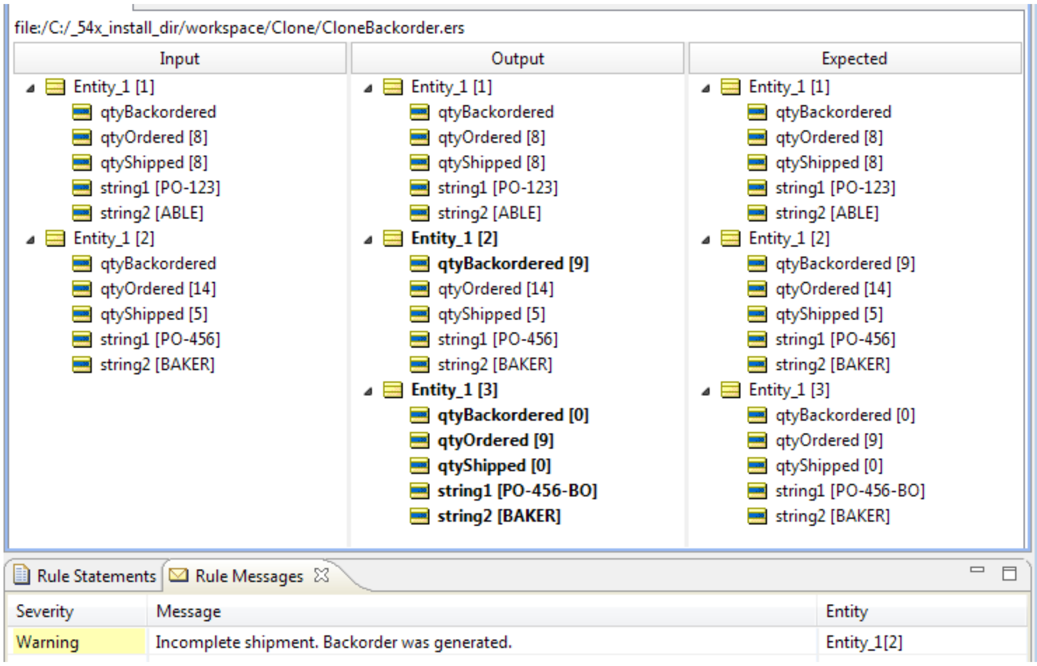
	Actions		
	Post Message(s)	✉	
A	Entity_1.qtyBackordered=Entity_1.qtyOrdered-Entity_1.qtyShipped	☑	□

B	Entity_1.clone[qtyOrdered=qtyBackordered,qtyShipped=0,qtyBackordered=0,
	string1=string1+'-BO']
		☑	□

C			
D			
	Overrides		

Ref	ID	Post	Alias	Text
1		Warning	Entity_1	Incomplete shipment. Backorder was generated.

A Ruletest against this Rulesheet produces the following output:



file:/C:/_54x_install_dir/workspace/Clone/CloneBackorder.ers

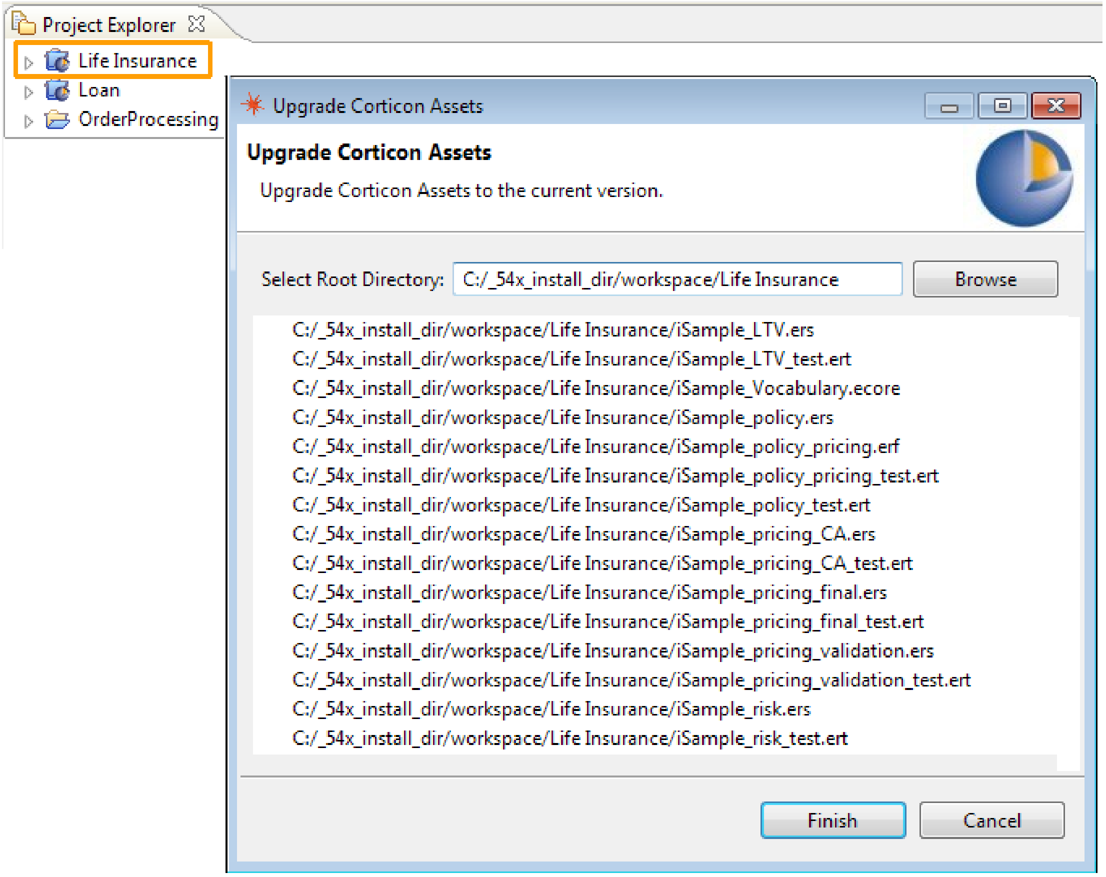
Input	Output	Expected
Entity_1 [1]	Entity_1 [1]	Entity_1 [1]
qtyBackordered	qtyBackordered	qtyBackordered
qtyOrdered [8]	qtyOrdered [8]	qtyOrdered [8]
qtyShipped [8]	qtyShipped [8]	qtyShipped [8]
string1 [PO-123]	string1 [PO-123]	string1 [PO-123]
string2 [ABLE]	string2 [ABLE]	string2 [ABLE]
Entity_1 [2]	**Entity_1 [2]**	Entity_1 [2]
qtyBackordered	**qtyBackordered [9]**	qtyBackordered [9]
qtyOrdered [14]	qtyOrdered [14]	qtyOrdered [14]
qtyShipped [5]	qtyShipped [5]	qtyShipped [5]
string1 [PO-456]	string1 [PO-456]	string1 [PO-456]
string2 [BAKER]	string2 [BAKER]	string2 [BAKER]
	Entity_1 [3]	Entity_1 [3]
	qtyBackordered [0]	qtyBackordered [0]
	qtyOrdered [9]	qtyOrdered [9]
	qtyShipped [0]	qtyShipped [0]
	string1 [PO-456-BO]	string1 [PO-456-BO]
	string2 [BAKER]	string2 [BAKER]

Severity	Message	Entity
Warning	Incomplete shipment. Backorder was generated.	Entity_1[2]

- See the "Clone" topic in the "Rule operator details and examples" section of the *Rule Language Guide*.

- **Improved installation and upgrade flexibility:**
  - **Option to install Corticon Studio as plugins** - A plugins-only distribution of Corticon Studio's Eclipse plugins has been added which lets you install the Corticon plugins into an existing Eclipse 4.3.x installation. The Corticon plugins distribution is always a complete version of the Corticon Eclipse plugins; there is no need to install a major or minor release and then a service pack. The download package includes a script to integrate all the Corticon Designer and Corticon Help plugins into your target 4.3.x Eclipse. A Java Software Development Kit (JDK) is required for Corticon, not just a Java Runtime Environment (JRE). If you do not have one installed locally, download and install a 1.6 or higher (1.7 preferred) JDK of the same bit-width as your existing Eclipse. The JDK can be downloaded from the Oracle Java website.
    - For details, see the topic "Installing Corticon plugins into an existing Eclipse" in the *Corticon Studio: Installation Guide*.
  - **Project utility for upgrading assets** - A new **Upgrade** utility lets you upgrade all the Corticon development assets in selected Rule Projects (Vocabularies, Rulesheets, Ruleflows, and Ruletests) for use in the current Corticon Studio. Upgrading your rule assets using this upgrade feature prevents upgrade messages when opening individual assets.

    The **Upgrade** command, Upgrade..., on the **Project** menu (and on the dropdown menu of a selected project in the **Project Explorer** tabs) opens, for example, the following dialog box:



Upgrade Corticon Assets

Upgrade Corticon Assets to the current version.

Select Root Directory: C:/_54x_install_dir/workspace/Life Insurance

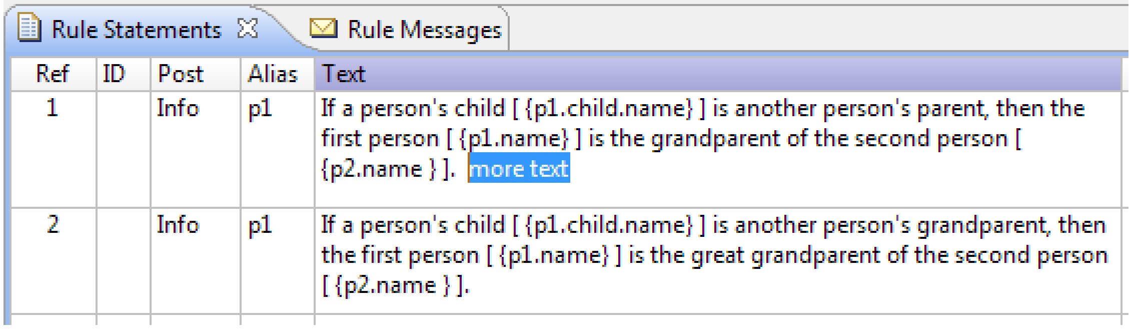
C:/_54x_install_dir/workspace/Life Insurance/iSample_LTV.ers
C:/_54x_install_dir/workspace/Life Insurance/iSample_LTV_test.ert
C:/_54x_install_dir/workspace/Life Insurance/iSample_Vocabulary.ecore
C:/_54x_install_dir/workspace/Life Insurance/iSample_policy.ers
C:/_54x_install_dir/workspace/Life Insurance/iSample_policy_pricing.erf
C:/_54x_install_dir/workspace/Life Insurance/iSample_policy_pricing_test.ert
C:/_54x_install_dir/workspace/Life Insurance/iSample_policy_test.ert
C:/_54x_install_dir/workspace/Life Insurance/iSample_pricing_CA.ers
C:/_54x_install_dir/workspace/Life Insurance/iSample_pricing_CA_test.ert
C:/_54x_install_dir/workspace/Life Insurance/iSample_pricing_final.ers
C:/_54x_install_dir/workspace/Life Insurance/iSample_pricing_final_test.ert
C:/_54x_install_dir/workspace/Life Insurance/iSample_pricing_validation.ers
C:/_54x_install_dir/workspace/Life Insurance/iSample_pricing_validation_test.ert
C:/_54x_install_dir/workspace/Life Insurance/iSample_risk.ers
C:/_54x_install_dir/workspace/Life Insurance/iSample_risk_test.ert

In this example, the **LifeInsurance** project was selected so all the assets in that project are selected for upgrade. You can browse to any folder hierarchy to upgrade all the discovered project assets, but you are advised to keep each batch to a moderate size.

- See the topic "Upgrading projects coming forward from a prior release" in the *Corticon Studio: Installation Guide*.

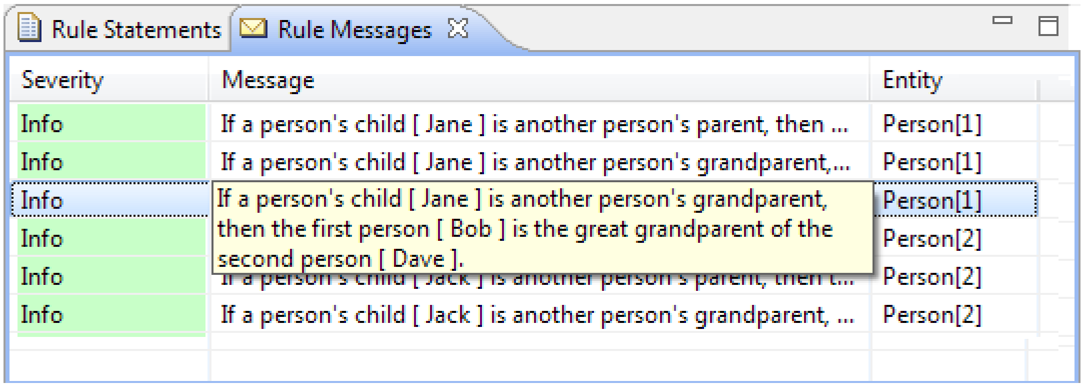
- **Enhancements to User Interface:**

  - **Line wrapping on Rule Statement tab and Rule Message tabs -** To handle cases where a Rule Statement's text gets wider than the entry line on the Rulesheet, Corticon introduced the ability to drag the line width down so that you can edit the text in its wrapped format:



Ref	ID	Post	Alias	Text
1		Info	p1	If a person's child [{p1.child.name}] is another person's parent, then the first person [{p1.name}] is the grandparent of the second person [{p2.name }]. more text
2		Info	p1	If a person's child [{p1.child.name}] is another person's grandparent, then the first person [{p1.name}] is the great grandparent of the second person [{p2.name }].

Now, Rule Messages on a Ruletest can expose wrapped text when the available line width is inadequate. As there is no edit function in the output, simply hovering over a message line displays wrapped hover help for the entire message, as illustrated:



Severity	Message	Entity
Info	If a person's child [Jane] is another person's parent, then ...	Person[1]
Info	If a person's child [Jane] is another person's grandparent,...	Person[1]
Info	If a person's child [Jane] is another person's grandparent, then the first person [Bob] is the great grandparent of the second person [Dave].	Person[1]
Info		Person[2]
Info	If a person's child [Jack] is another person's parent, then t...	Person[2]
Info	If a person's child [Jack] is another person's grandparent, ...	Person[2]

What's changed in Service Pack 5.4.1

Some features were dropped or changed their behaviors in Corticon 5.4.1. Significant items include:

- Ruleflow context menu no longer provides **Download** and **Publish** commands.

3

What was new and changed in Corticon 5.4.0

This section summarizes the new, enhanced, and changed features in Progress® Corticon® 5.4.0.

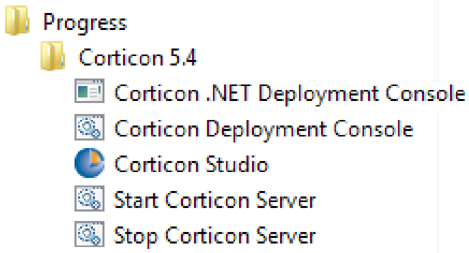
Note: When you upgrade from 5.3, a separate installation location will be specified and you will migrate assets from the 5.3 installation.

For details, see the following topics:

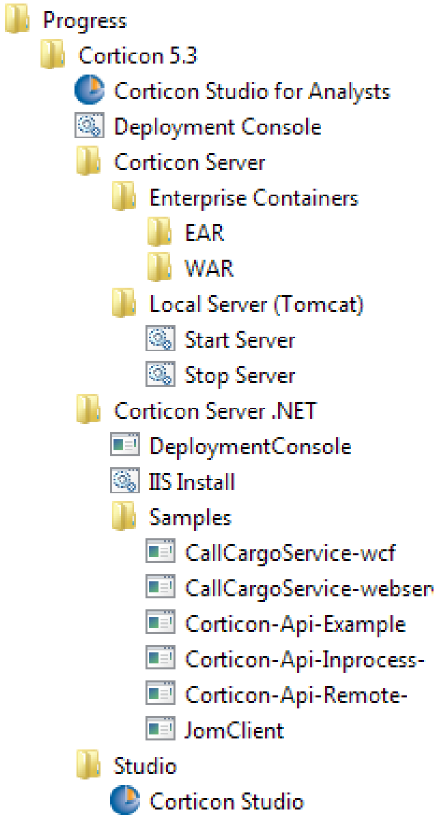
Simplified Installation and User Interface in 5.4

Several improvements were made to Corticon installation procedures and the Corticon User Interface. Significant items include:

- **Discovery of PDS-OE and tailored installation** - The Progress Software Development environment for Progress OpenEdge (PDS-OE) combines with Corticon Studio in Eclipse to enable development of Business Rules in OpenEdge. In earlier releases, you had to choose to install Corticon into the external Eclipse of PDS-OE, and then run the `integrateCorticon` script. In this release, the installer will discover an existing PDS-OE on the target machine, confirm that its version and bit width are compatible with the Corticon installer that is running, and then offer to install and tailor Corticon without any further action required. Note that the tailoring script and the opportunity to tailor Corticon into a different external Eclipse continue to be offered in the installer
- **Unified Studio** - There is only one 'flavor' of Corticon Studio in 5.4. The Studio formerly known as Corticon Studio for Analysts is no longer available. Projects created with previous versions of Corticon Studio for Analysts are fully compatible with Corticon Studio.
- **Standard location of scripts, samples, and source files** - Where scripts (`.bat` files), samples, and source files (`.java` files) were previously in various locations within an installation, these files are now all located in common locations:
  - Scripts in the installation's `[CORTICON_HOME\bin` directory
  - Samples in the installation's `[CORTICON_WORK_DIR\samples` directory
- **Simplified Start menu** - The **Start** menus have been refined. The Corticon 5.4 **Start** menu when all products have been installed looks like this:



Previously, in Corticon 5.3, the **Start** menu when all products were installed looked like this:



- Progress
  - Corticon 5.3
    - Corticon Studio for Analysts
    - Deployment Console
    - Corticon Server
      - Enterprise Containers
        - EAR
        - WAR
      - Local Server (Tomcat)
        - Start Server
        - Stop Server
    - Corticon Server .NET
      - DeploymentConsole
      - IIS Install
      - Samples
        - CallCargoService-wcf
        - CallCargoService-webser
        - Corticon-Api-Example
        - Corticon-Api-Inprocess-
        - Corticon-Api-Remote-
        - JomClient
    - Studio
      - Corticon Studio

Other than general simplification of the structure, some item links were dropped:

- For Server, the Enterprise Containers can be accessed at their install location, `[CORTICON_HOME]\Server\Containers`.
- For .NET Server:
  - The JOM client samples can be accessed at their install location, `[CORTICON_WORK_DIR]\Samples\rule-assets\JOM`
  - The API example and test scripts can be accessed at their install location, `[CORTICON_HOME]\Server .NET\samples\bin`
  - The IIS install script can be accessed at its install location `[CORTICON_HOME]\Server .NET\IIS`

Enhancements to Corticon Studio in 5.4

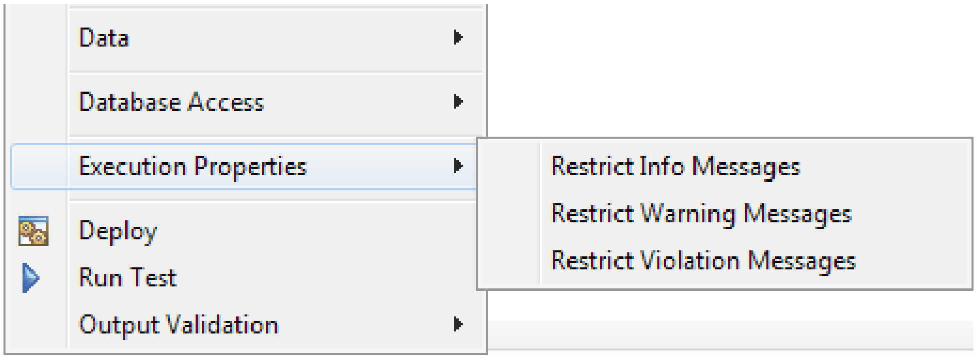
Several improvements were made to Corticon Studio in Corticon 5.4. Significant items include:

- **Option to exclude selected attributes in rule test output from comparison** - You can now designate specific attributes in the Ruletest **Output** to not compare automatically with **Expected** results. Some attributes, such as dates and time, can cause errors in tests when **Input** values are provided. Selecting an attribute in the **Expected** column, right-clicking and then choosing **Ignore Validation** enables suppression and, when the focus is off the attribute, shows it greyed out. Once set, each feature can be toggled back by choosing **Enable Validation** on each specific attribute on which it was set. For more information, see the topic *"Using the expected panel" in the Quick Reference Guide.*

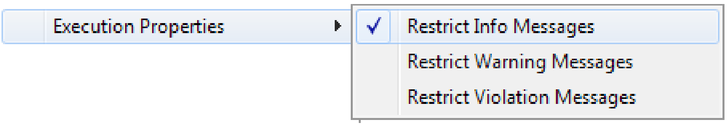
- **Option to suppress messages by severity in Ruletests -** When you choose to suppress selected level of messages in server output and logs, you might want to see that same behavior in your Ruletests first. These settings restrict each of the three types of Rule Messages (info, warning, and violation) from being posted to the output of an execution. They are test server settings that correspond to the Corticon Server properties:

  ```
  com.corticon.server.restrict.rulemessages.info
  com.corticon.server.restrict.rulemessages.warning
  com.corticon.server.restrict.rulemessages.violation
  ```

  You can see the impact of these restrictions in testsheets by engaging each of the toggles under **Execution Properties** on the **Ruletest > Testsheet** menu as shown:

  When the option is checked, messages of that type are suppressed:

  Clicking an option again clears it, and that message type is again produced.

  This information was added to *"Ruletest menu commands" in the Quick Reference Guide*.

- **Corticon Studio available as 64-bit -** The download packages available for Corticon Studio offer 32-bit and 64-bit installers. The 64-bit Studio installer is preferred for all 64-bit platforms. Note that for integration with PDS-OE, the OpenEdge development environment, the installer validates that the bit widths are compatible, so check the OpenEdge Eclipse environment before proceeding with integration.
- **Eclipse 4.3 -** The Corticon development environment advances from 3.7.1 (Indigo) and moves to 4.3 (Kepler). The changes have little impact on Corticon functionality.

Enhancements to Corticon Servers and Deployments in 5.4

Several improvements were made to the Java and .NET servers and deployments in Corticon 5.4. Significant items include:

- **Pacific Application Server (PAS)** - This release replaces the bundled generic Tomcat application server with our Pacific Application Server. PAS tailors Apache Tomcat as a production server by replacing the default Tomcat ROOT Web application with a ROOT application that:

  - Returns no content if a client accesses the root URL of the server, so that information about server configuration and deployed Web applications is not accessible
  - Adds an implementation of the Spring Security framework to support user authentication and authorization (HTTPS on default port `8851`)
  - Removes functionality that could present a security risk, such as remote administration

  A key feature of PAS is that you can easily create and run more than one instance of the core server, and configure the instances separately to function either in a development or in a production environment. Each instance of the core PAS shares the executables and libraries of a common Tomcat server, but each instance is a separate process, running in a separate JVM, with its own configuration of ports, security framework, and Web applications.

- **JSON RESTful interface for execution of decision services** - You can create Corticon requests in JavaScript Object Notation (JSON), a text format that you can use as an alternative to XML.A JSON RESTful interface is one that follows the REST architectural style and uses JSON as its data representation format. Specifically, a standardized `JSONObject` with name/value pairs of `“Objects”:<JSONArray>`can be passed in to Corticon Server's `ICcServer.execute(…)` to process the request and return a JSON-formatted reply.

  For details, see *"Path 4 - Using JSON/RESTful client to consume a Decision Service*" in the "Deploying Web Services with Java Guide".

  Note: JSON requests can also be used with an in-process server (without REST), similar to XML requests that run on an in-process server.

  New Server pool timeout property, `com.corticon.server.serverpool.timeout` - Timeout set on an execution thread waiting for an available Reactor from the Decision Service pool. When the thread's wait time exceeds this property's value, then a `CcServerTimeoutException` is thrown for that thread. Default value is `180000` milliseconds -- 3 minutes. You can override this property's value in your `brms.properties` file.

  ```
  com.corticon.server.serverpool.timeout=180000
  ```

  This information was added to *"Corticon Server properties*" in the "Integration and Deployment Guide".

- **Additional .NET infrastructure versions** - .NET now supports IIS 8.0 and 8.5 as well as .NET Framework 4.5. The guide *Deploying Web Services with .NET* details setting up Windows Internet Information Services (IIS) on:

  - Windows Server 2008 or Windows 7 (IIS 7.5)
  - Windows Server 2012, R1 and R2 (IIS 8.0 and 8.5)

Enhancements to Corticon Enterprise Data Connector in 5.4

Several improvements were made to database connnectivity in Corticon 5.4. Significant items include:

- Upgraded drivers for DataDirect Cloud

Enhancements to Server logging in 5.4

Several improvements were made to Corticon Server logs. Corticon logs continue to offer Server level and Decision Service level logging and thread-by-thread logging. Significant improvements include:

- **Environment and license information in logs -** In their logs, the Servers (including the Studio's embedded Server that is launched when you run Ruletests) record the current license and environment information whenever a new log file is started. The information is similar to the following:

  ```
  LICENSE_OEM_NAME = Evaluation
  LICENSE_DATE_GRANTED = null
  LICENSE_MAX_POOLS = 20
  LICENSE_MAX_REACTORS = 1
  LICENSE_MAX_NUMBER_OF_RULES = 500
  LICENSE_DEACTIVATE_DATE = mm/dd/yyyy
  LICENSE_DEACTIVATE_DATE_OVERRIDE = NO
  LICENSE_SINGLE_EXECUTION_ENABLED = YES
  LICENSE_SINGLE_EXECUTION_DB_ACCESS_ENABLED = NO
  LICENSE_BATCHPROCESSING_ENABLED = NO
  LICENSE_USAGE_ENFORCED = false
  LICENSE_USAGE_NAME = Corticon
  LICENSE_SERVER_IP = 0.0.0.0
  LICENSE_INCREMENT_IP = 0
  LICENSE_IP_OVERRIDE = YES
  LICENSE_SERVER_TRANSACTIONS_TIME_PERIOD = 60000
  LICENSE_SERVER_TRANSACTIONS_MAX_EXECUTIONS = 100
  LICENSE_SERVER_TRANSACTION_OVERRIDE = YES
  LICENSE_NAMED_USERS = 3

  Available processors (cores): 4
  Free memory (bytes): 45157776
  Maximum memory (bytes): 389283840
  Total memory available to JVM (bytes): 79757312
  ```

  This information records runtime conditions and license constraints which are useful when diagnosing problems or getting assistance from Progress support personnel.

  This material has been added to *"About Corticon Logs" in the Integration and Deployment Guide.*

- **Performance monitoring** - Corticon now provides a `DIAGNOSTIC` log level that emits key metrics into the log at defined intervals. The data captured at regular intervals creates log entries that can be extracted through a bundled utility that extracts and transforms data in standard comma-separated-value format. You can then import the data to a data analysis product such as Tableau or Excel to create visualizations. For more information, see the topic *"Diagnosing server runtime performance" in the "Performance and Tuning Guide" section of the Integration and Deployment Guide.*

Enhancements to override properties in 5.4

Several improvements were made to user access to setting Studio and Server properties. Significant improvements include:

- **Easier access to override properties** - While functionally is unchanged, the technique for specifying and maintaining override properties is improved across all installation types. The file `brms.properties`, inserted at the work directory root, lists the properties that Studio and Server users most often change. You can uncomment and set the overrides you want for these properties, as well add the name=value pairs that define any other Corticon property overrides you want. *"Configuring Corticon properties and settings" in the Integration and Deployment Guide.*
- **Ability to use multiple custom override files** - In Studio, you can set up several properties files that you might use in certain circumstances where you want to set several parameters for one run, and then change to other parameters for another run. This feature makes moving in and out of various log paths and log levels an easy task. For more information, see the topic *"Applying override properties to Corticon Studio and its built-in Server" in the Rule Modeling Guide.*

Note: Preferred location of the override properties file - In prior releases, the `brms.properties` file was either placed into the installation folder, or updated directly into the `CcConfig.jar`. Both of those practices -- as well directly modifying the values in the default properties files -- are now discouraged, and any such placements of that file when upgrading should be removed so that it replaces the file installed at the work directory root, or -- for Studio -- the location specified in Eclipse preferences. For backward compatibility, the deprecated locations will be checked, and will continue to be supported; however, if you use those classic techniques, overrides in the `brms.properties` at the work directory root will trump all others.

Enhancements to Corticon documentation in 5.4

Several requests for enhancements to Corticon documentation are resolved in Corticon 5.4. Significant items include:

- **Transactions in the Enterprise Data Connector (EDC)** -- Clarification was added to note that:
  - Each decision service call is one database transaction. Transactions are not per operation, per rulesheet, or per ruleflow. Corticon does not currently provide for configuration of transaction management.
  - The default transaction isolation level in Corticon EDC is the same as the default transaction isolation level of the database to which it is connected.
  - When an exception occurs, the database transaction is rolled back, and the database reverts to the same state as before the decision service was called.

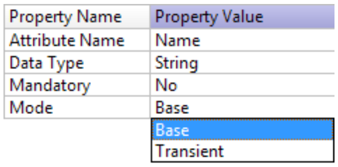
This information is now documented in *How EDC handles transactions and exceptions* in the *Integration and Deployment Guide*.

- **Example of how to de-duplicate retrieved associations** -- The problem is: *For a Customer->Address association (one-to-many), each address must be unique*. The recommended technique compares each Address to the rest of the collection to check if it is a duplicate, and if it is, then removes it. The example compares all pairs of addresses without requiring an explicit loop. For details, see *Removing duplicated children in an association* in the *Rule Modeling Guide*.
- **Example of how to handle nulls in compare operations** - Unless the application that formed the request ensured that a value was provided before submission, one (or both) of the attributes used in a comparison test might have a null value. You might need to define rules to handle such cases. An example describes the workaround for these cases. For details, see *Handling nulls in compare operations* in the *Rule Modeling Guide*.
- **How to achieve sorted results in collections across associations** - The `SortedBy->first` and `SortedBy->last` constructs work as expected for any first-level collection regardless of datatype, determining the value of the first or last element in a sequence that was derived from a collection. When associations are involved, you have to take care that the collection operator is not working at a grandchild level. For more about this clarification, see *Finding first or last in grandchild collections* in the *Rule Modeling Guide* discusses...

What changed in 5.4

Some features were dropped or changed their behaviors in Corticon 5.4. Significant items include:

- **Application server ports have changed** - With the transition to Pacific Application Server, the default port for HTTP connection is `8850` instead of `8082`. Additional ports are added for HTTPS (default value is `8851`) and server shutdown (default value is `8852`). These ports can be set to preferred values (any integer from `1024` to `49151`) in the Server installation wizard.
- **Work Document Entity (WDE) has been dropped** - Client requests that specify WDE will now ignore the WDE setting when they are processed. This does not impact any user functions or existing assets.
- **Behavior of the `.new` operator has changed -** A `.new` operator will always fire when explicit conditions are met. In other words, the `.new` operator will not consider implied conditions of non-mandatory attributes from the initialize expressions during execution. Each initialize expression within a `.new`… expression will be executed (or not) depending upon implied conditions: if any input to the expression is null, the target attribute remains null. Another case where an implied condition would prevent a `.new` operator for executing is where the new entity is a target to an association assignment and the parent of that association does not exist. This information and examples have been added to *'New' operator in the "Rule operator details and examples" section of the Rule Language Guide.*
- **"Extended transient" mode is now referred to as just "Transient"**



Property Name	Property Value
Attribute Name	Name
Data Type	String
Mandatory	No
Mode	Base

Base
Transient

This has no impact on any user functions or existing assets.

- **Log level defaults to `DIAGNOSTIC`** - With the introduction of server performance diagnostics, the added log level, `DIAGNOSTIC`, is now the default log level. One level of detail above the previous default, `VIOLATION`, the new default level produces a log entry with diagnostic data every five minutes. You can adjust the diagnostic interval in the `brms.properties` file.
- **Some Java source files have been dropped** - Java code samples for the sample SOAP container located in the Java Server 5.3 at `[CORTICON_HOME]\ServerJava\Server\Tomcat\CcServer\src\` are no longer installed.
- **Changes to supported platforms**
  - *Operating systems, Server*:
    - Added: Microsoft Windows Server 2008 R2 SP2 64-bit
    - Dropped: Microsoft Windows Server 2003
    - Dropped: Microsoft Windows Server 2008 R2 SP1
    - Dropped: Microsoft Windows 7 32-bit
  - *Operating systems, Studio*:
    - Added: Microsoft Windows Server 2008 R2 SP2 64-bit
    - Dropped: Microsoft Windows Server 2008 R2 SP1
    - Dropped: Microsoft Windows Server 2003 R2 (SP2) 64-bit
  - *Application Servers*:
    - Dropped: Apache Tomcat 6.0
  - *Databases*:
    - Added: Oracle Database 12c
    - Added: IBM DB2 10.5
    - Added: Progress OpenEdge 11.4
  - *Microsoft Internet Information Services (IIS)*:
    - Added: 8.5
    - Dropped: 7.0
    - Dropped: 6.0

Note: Refer to the Progress Software web page Progress Corticon 5.4 - Supported Platforms Matrix for the latest updates to the supported platforms.

A

Progress Corticon documentation

The following documentation, as well as a *What's New in Corticon* document, is included with this Progress Corticon release:

Corticon Tutorials	
Corticon Studio Tutorial: Basic Rule Modeling	Introduces modeling, analyzing, and testing rules and decisions in Corticon Studio. Recommended for evaluators and users getting started. *See also the PowerPoint-as-PDF version of this document that is accessed from the Studio's **Help** menu.*
Corticon Studio Tutorial: Advanced Rule Modeling	Provides a deeper look into Corticon Studio's capabilities by defining and testing vocabularies, scope, collections, messages, filters, conditions, transient data, and calculations in multiple rulesheets that are assembled into a Ruleflow. *See also the PowerPoint-as-PDF version of this document that is accessed from the Studio's **Help** menu.*
Corticon Tutorial: Using Enterprise Data Connector (EDC)	Introduces Corticon's direct database access with a detailed walkthrough from development in Studio to deployment on Server. Uses Microsoft SQL Server to demonstrate database read-only and read-update functions.
Corticon Studio Documentation: Defining and Modeling Business Rules	

Corticon Studio: Installation Guide	Step-by-step procedures for installing Corticon Studio on computers running Microsoft Windows as a standalone installation and as a part of an existing Eclipse installation such as Progress Developer Studio for OpenEdge. Shows how to enable internationalization on Windows.
Corticon Studio: Rule Modeling Guide	Presents the concepts and purposes the Corticon Vocabulary, then shows how to work with it in Rulesheets by using scope, filters, conditions, collections, and calculations. Discusses chaining, looping, dependencies, filters and preconditions in rules. Presents the Enterprise Data Connector from a rules viewpoint, and then shows how database queries work. Provides information on versioning, natural language, reporting, and localizing. Provides troubleshooting and many *Test Yourself* exercises.
Corticon Studio: Quick Reference Guide	Reference guide to the Corticon Studio user interface and its mechanics, including descriptions of all menu options, buttons, and actions.
Corticon Studio: Rule Language Guide	Reference information for all operators available in the Corticon Studio Vocabulary. A Rulesheet example is provided for many of the operators. Includes special syntax issues, handling arithmetic and character precedence issues.
Corticon Studio: Extensions Guide	Detailed technical information about the Corticon extension framework for extended operators and service call-outs. Describes several types of operator extensions, and how to create a custom extension plug-in.
Corticon Server Documentation: Deploying Rules as Decision Services	
Corticon Server: Integration & Deployment Guide	An in-depth, technical description of Corticon Server deployment methods, including preparation and deployment of Decision Services and Service Contracts through the Deployment Console tool. Describes JSON request syntax and REST calls. Discusses relational database concepts and implementation of the Enterprise Data Connector. Goes deep into the server to discuss state, persistence, and invocations by version or effective date. Includes server monitoring techniques, performance diagnostics, and recommendations for performance tuning.

Corticon Server: Deploying Web Services with Java	Details installing the Corticon Server as a Web Services Server, and then deploying and exposing Decision Services as Web Services on the Pacific Application Server (PAS) and other Java-based servers. Includes samples of XML and JSON requests. Presents the features and functions of the browser-based Server Console. Provides administrative instructions for the Pacific Application Server.
Corticon Server: Deploying Web Services with .NET	Details installing the Corticon Server as a Web Services Server, and then deploying and exposing decisions as Web Services with .NET. Includes samples of XML and JSON requests. Provides installation and configuration information for the .NET Framework and Internet Information Services (IIS) on various supported Windows platforms.