



Corticon.js Product Notes

Table of Contents

Known Issues in Corticon.js 1.4.....	7
Issues fixed in Corticon.js 1.4.....	9
Known Issues in Corticon.js 1.3.....	11
Issues fixed in Corticon.js 1.3.....	15
Known Issues in Corticon.js 1.2.....	17
Issues fixed in Corticon.js 1.2.....	21
Known Issues in Corticon.js 1.1.....	23

Copyright

© 2022 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.

These materials and all Progress® software products are copyrighted and all rights are reserved by Progress Software Corporation. The information in these materials is subject to change without notice, and Progress Software Corporation assumes no responsibility for any errors that may appear therein. The references in these materials to specific platforms supported are subject to change.

#1 Load Balancer in Price/Performance, 360 Central, 360 Vision, Chef, Chef (and design), Chef Habitat, Chef Infra, Code Can (and design), Compliance at Velocity, Corticon, Corticon.js, DataDirect (and design), DataDirect Cloud, DataDirect Connect, DataDirect Connect64, DataDirect XML Converters, DataDirect XQuery, DataRPM, Defrag This, Deliver More Than Expected, DevReach (and design), Driving Network Visibility, Flowmon, Inspec, Ipswitch, iMacros, K (stylized), Kemp, Kemp (and design), Kendo UI, Kinvey, LoadMaster, MessageWay, MOVEit, NativeChat, OpenEdge, Powered by Chef, Powered by Progress, Progress, Progress Software Developers Network, SequeLink, Sitefinity (and Design), Sitefinity, Sitefinity (and design), Sitefinity Insight, SpeedScript, Stylized Design (Arrow/3D Box logo), Stylized Design (C Chef logo), Stylized Design of Samurai, TeamPulse, Telerik, Telerik (and design), Test Studio, WebSpeed, WhatsConfigured, WhatsConnected, WhatsUp, and WS_FTP are registered trademarks of Progress Software Corporation or one of its affiliates or subsidiaries in the U.S. and/or other countries.

Analytics360, AppServer, BusinessEdge, Chef Automate, Chef Compliance, Chef Desktop, Chef Workstation, Corticon Rules, Data Access, DataDirect Autonomous REST Connector, DataDirect Spy, DevCraft, Fiddler, Fiddler Classic, Fiddler Everywhere, Fiddler Jam, FiddlerCap, FiddlerCore, FiddlerScript, Hybrid Data Pipeline, iMail, InstaRelinker, JustAssembly, JustDecompile, JustMock, KendoReact, OpenAccess, PASOE, Pro2, ProDataSet, Progress Results, Progress Software, ProVision, PSE Pro, Push Jobs, SafeSpaceVR, Sitefinity Cloud, Sitefinity CMS, Sitefinity Digital Experience Cloud, Sitefinity Feather, Sitefinity Thunder, SmartBrowser, SmartComponent, SmartDataBrowser, SmartDataObjects, SmartDataView, SmartDialog, SmartFolder, SmartFrame, SmartObjects, SmartPanel, SmartQuery, SmartViewer, SmartWindow, Supermarket, SupportLink, Unite UX, and WebClient are trademarks or service marks of Progress Software Corporation and/or its subsidiaries or affiliates in the U.S. and other countries. Java is a registered trademark of Oracle and/or its affiliates. Apache and Kafka are either registered trademarks or trademarks of the Apache Software Foundation in the United States and/or other countries. Any other marks contained herein may be trademarks of their respective owners.

Please refer to the NOTICE.txt or Release Notes – Third-Party Acknowledgements file applicable to a particular Progress product/hosted service offering release for any related required third-party acknowledgements.

Last updated with new content: Corticon.js 1.4

Known Issues in Corticon.js 1.4

Release Notes for Corticon.js 1.4.0.0

Item	Details	Knowledge Base
COR-11515	Importing invalid JSON into a ruletest in Corticon Studio may generate an "Index out of bounds" exception. This can occur when the JSON document is not properly structured JSON. Once corrected, the document can be imported.	
COR-11502	Using the "now" operator in conjunction with the "isSameDate" operator in a rulesheet returns false even if the date component of the DateTime compared to is the current (now) date.	
COR-11499	<p>Within an expression in a rulesheet, combining the "getMilliseconds" and "toString" operators on a DateTime attribute will give erroneous behavior. An example expression would be:</p> <pre>MyEntity.string1 = MyEntity.dateTime1.getMilliseconds.toString()</pre> <p>A workaround for this would be to split it into two expressions such as:</p> <pre>MyEntity.millis =MyEntity.dateTime1.getMilliseconds MyEntity.string1 =MyEntity.millis.toString()</pre>	
COR-9093	Given a vocabulary with CDTs defined of data type Decimal or DateTime, Corticon.js ruletests may indicate a warning when run due to the precision of attribute values. Suppose you defined a CDT of Decimal type to have "1.2" in its enumerated value set. A ruletest that sets this attribute may set the value to "1.200000". This will be flagged as a warning indicating the value is not in the enumerated value set.	
COR-9073	Corticon.js does not support removing the Root entity by rules. Attempting to remove the Root entity will generate a "Circular reference" error when executed.	
COR-8667	Rule statements containing double quotes characters such as "My Text" are displayed with the characters escaped such as \\ "My Text\\" in rule messages in the Corticon.js Tester.	
COR-8413	<p>When deploying Corticon.js rules to Microsoft Azure as Azure Functions, setting the property</p> <pre>logRuleMessages = true</pre> <p>is not logging Corticon rule messages to the Azure console. The rule messages are only being returned as part of the response payload.</p>	
COR-7867	<p>The installation of Corticon.js Studio will write the exception:</p> <pre>"org.eclipse.core.runtime.CoreException: Plug-in com.corticon.eclipse.studio.product was unable to load class com.corticon.eclipse.studio.product.CorticonProductInfo"</pre> <p>to the post_install.log file. This exception can be ignored.</p>	

Issues fixed in Corticon.js 1.4

Customer reported defects fixed in Corticon.js 1.4.0.0
--

Item	Details	Knowledge Base
COR-11480	Improved Corticon.js Studio performance when validating ruleflows. Previously redundant validations could occur when using nested ruleflows. The performance improvement can have significant benefits when working on large rule projects in Corticon.js Studio.	
COR-11202	Fixed a problem where rules using 2 filter aliases in 2 different actions would not fire. If the rule conditions were true, the actions would not execute.	
COR-10983	Fixed a code generation issue which could lead to a runtime exception due to faulty null check analysis.	
COR-10982	Fixed a code generation issue which could lead to a runtime exception due to faulty null check analysis.	
COR-8678	Corticon.js now supports entering decimal values in rulesheets in your preferred locale. For example in a German locale, you can now enter decimal values using comma as the decimal separator in rulesheet expressions and ruletest attribute values.	
COR-7849	Corticon.js Studio now bundles Node.js v16. This addresses the problem when executing ruletests in Corticon.JS Studio, DateTime values earlier than 2007 may be off by 1 hour due to a daylight savings time error.	

Known Issues in Corticon.js 1.3

Release Notes for Corticon.js 1.3.0.0

Item	Details	Knowledge Base
COR-9691	Custom functions are not automatically reloaded by the Corticon.js Studio Tester when the JavaScript file is modified. When running ruletests in Corticon.js Studio, the JavaScript files for any custom functions are loaded into memory. If these files are modified, they are not automatically re-loaded into memory. This results in the prior implementation of the function being used in the rule test. To force the Tester to reload the custom function either close and reopen the Tester or close and reopen the Project.	
COR-9093	Given a vocabulary with CDTs defined of data type Decimal or DateTime, Corticon.js ruletests may indicate a warning when run due to the precision of attribute values. Suppose you defined a CDT of Decimal type to have "1.2" in its enumerated value set. A ruletest that sets this attribute may set the value to "1.200000". This will be flagged as a warning indicating the value is not in the enumerated value set.	
COR-9073	Corticon.js does not support removing the Root entity by rules. Attempting to remove the Root entity will generate a "Circular reference" error when executed.	
COR-8678	Corticon.js does not support entering decimal values in rulesheets in your preferred locale. Specifically decimal values must use the period character "." as the decimal separator.	
COR-8667	Rule statements containing double quotes characters such as "My Text" are displayed with the characters escaped such as \"My Text\" in rule messages in the Corticon.js Tester.	
COR-8413	When deploying Corticon.js rules to Microsoft Azure as Azure Functions, setting the property: <code>logRuleMessages = true</code> is not logging Corticon rule messages to the Azure console. The rule messages are only being returned as part of the response payload.	
COR-7867	The installation of Corticon.js Studio will write the exception: <code>org.eclipse.core.runtime.CoreException: Plug-in com.corticon.eclipse.studio.product was unable to load class com.corticon.eclipse.studio.product.CorticonProductInfo</code> to the <code>post_install.log</code> file. This exception can be ignored.	

Item	Details	Knowledge Base
COR-7849	When executing ruletests in Corticon.js Studio, DateTime values earlier than 2007 may be off by 1 hour due to a daylight savings time error. This issue is specific to the Corticon.js Studio tester and does not affect any deployed decision services in the supported platforms.	
COR-7796	Translations are missing for some of the operator descriptions in the Corticon Studio operator tree.	

Issues fixed in Corticon.js 1.3

Customer reported defects fixed in Corticon.js 1.3.0.0

Item	Details	Knowledge Base
COR-10016	Modified the Corticon.js sample browser wrapper to set the encoding to UTF-8. When package rules for browser deployment the file <code>browser.sample.html</code> is generated. This is sample code demonstrating how to include Corticon.js packaged rules in a browser application. If the encoding is not set to UTF-8, you may encounter errors using the packaged rules if any of the rule assets contained non-ASCII characters.	
COR-9695	Optimized Corticon.js code generation to reuse code for redundant conditions. This reduces the generated code size for very large rulesheets with high numbers of redundant conditions. Previously such rulesheets could not be packaged by Corticon.js. This would prevent running them in Corticon.js Studio Tester or packaging them for deployment.	

Known Issues in Corticon.js 1.2

Known Issues in Corticon.js 1.2.0.0

Item	Details	Knowledge Base
COR-9157	<p>Corticon.js does not support JSON arrays mixing scalar values and objects. For example:</p> <pre>A: [1,2,3, {B: {"color" : "red"}}]</pre> <p>This JSON snippet defines an array A containing the scalar values 1, 2, 3 and the object B. In Corticon.js, an array must be either all scalar values or all objects.</p>	
COR-9093	<p>Given a vocabulary with CDTs defined of data type Decimal or DateTime, Corticon.js ruletests may indicate a warning when run due to the precision of attribute values. Suppose you defined a CDT of Decimal type to have 1.2 in its enumerated value set. A ruletest that sets this attribute may set the value to 1.200000. This will be flagged as a warning indicating the value is not in the enumerated value set.</p>	
COR-9073	<p>Corticon.js does not support removing the Root entity by rules. Attempting to remove the Root entity will generate a "Circular reference" error when executed.</p>	
COR-8926	<p>Corticon.js does not support JSON consisting of an array of scalar values at root. For example JSON containing only an array such as:</p> <pre>[1, 2, 3]</pre> <p>is not supported. Arrays of scalars within objects in JSON is supported.</p>	
COR-8678	<p>Corticon.js does not support entering decimal values in rulesheets in your preferred locale. Specifically decimal values must use the period character, ., as the decimal separator.</p>	
COR-8667	<p>Rule statements containing double quotes characters such as "My Text" are displayed with the characters escaped such as "My Text" in rule messages in the Corticon.js Tester.</p>	
COR-8413	<p>When deploying Corticon.js rules to Microsoft Azure as Azure Functions, setting the property</p> <pre>logRuleMessages = true</pre> <p>is not logging Corticon rule messages to the Azure console. The rule messages are only being returned as part of the response payload.</p>	
COR-7867	<p>The installation of Corticon.js Studio will write the exception:</p> <pre>org.eclipse.core.runtime.CoreException: Plug-in com.corticon.eclipse.studio.product was unable to load class com.corticon.eclipse.studio.product.CorticonProductInfo to the post_install.log file. This exception can be ignored.</pre>	

Item	Details	Knowledge Base
COR-7849	When executing ruletests in Corticon.JS Studio, DateTime values earlier than 2007 may be off by 1 hour due to a daylight savings time error. This issue is specific to the Corticon studio tester and does not affect any deployed decision services in the supported platforms.	
COR-7796	Translations are missing for some of the operator descriptions in the Corticon.js Studio operator tree.	

Issues fixed in Corticon.js 1.2

No issues were fixed in Corticon.js 1.2

Known Issues in Corticon.js 1.1

Known Issues in Corticon.js 1.1.0.0

Item	Details	Knowledge Base
COR-7849	When executing ruletests in Corticon.js Studio, DateTime values earlier than 1990 may be off by 1 hour due to a daylight savings time error. This is bug in Node.js v8, which Corticon.js uses for executing Ruletests. The problem does not occur in Node.js v10 or later. Only Node.js v10 or later is supported for production deployment.	-
COR-7867	The installation of Corticon.js Studio will write the exception: <pre>org.eclipse.core.runtime.CoreException: Plug-in com.corticon.eclipse.studio.product was unable to load class com.corticon.eclipse.studio.product.CorticonProductInfo</pre> to the post_install.log file. This exception can be ignored.	-
COR-8387	Corticon.js does not generate rule messages for rulesheet warnings. This differs from Corticon where, for example, rule messages are generated for potential errors like using null in the right hand of an assignment. Corticon.js only generates rule messages for the rule statements a rule author specifies.	-
COR-8413	When deploying Corticon.js rules to Microsoft Azure as Azure Functions, setting the property <code>logRuleMessages = true</code> is not logging Corticon rule messages to the Azure console. The rule messages are only being returned as part of the response payload.	-
COR-8601	The Corticon.js Studio UI is not fully translated for other supported languages. Not translated elements include the rulesheet operators and descriptions, the package dialog and several other UI components.	-