



Corticon.js Product Notes

Table of Contents

Known Issues in Corticon.js 2.3.....	7
Issues fixed in Corticon.js 2.3.....	9
Known Issues in Corticon.js 2.2.....	11
Issues fixed in Corticon.js 2.2.....	13
Known Issues in Corticon.js 2.1.....	15
Issues fixed in Corticon.js 2.1.....	17
Known Issues in Corticon.js 2.0.....	19
Issues fixed in Corticon.js 2.0.....	23
Known Issues in Corticon.js 1.4.....	25
Issues fixed in Corticon.js 1.4.....	27
Known Issues in Corticon.js 1.3.....	29
Issues fixed in Corticon.js 1.3.....	33
Known Issues in Corticon.js 1.2.....	35
Issues fixed in Corticon.js 1.2.....	39
Known Issues in Corticon.js 1.1.....	41

Copyright

Visit the following page online to see Progress Software Corporation's current Product Documentation Copyright Notice/Trademark Legend: <https://www.progress.com/legal/documentation-copyright>.

Last updated with new content: Corticon.js 2.3

Known Issues in Corticon.js 2.3

Release Notes for Corticon.js 2.3.0.0

Item	Details
COR-15884	When using Corticon Studio, if the Windows Display Scale is set to a value other than 100%, the bottom row of prompt suggestions in the AI Assistant may not be visible even with scrolling.
COR-12939	<p>When packaging a JavaScript decisions service, Corticon.js both obfuscates and minimizes the generated JavaScript code. The minification process reduces the size of the generated JavaScript bundle. The minification process fails when the line count of generated code exceeds 500,000 lines. This will result in an error in the Corticon.js log file:</p> <pre>The code generator has deoptimised the styling of as it exceeds the max of 500KB.</pre> <p>This is not a fatal error. It indicates that the generated code was not minimized. The generated decision is still valid and can be deployed. The generated code is still obfuscated.</p>
COR-12654	Using promises and .then asynchronous programming pattern in service callouts is not supported. Running a ruletest from Corticon Studio that calls a service callout which uses this pattern, will hang Corticon Studio.

Issues fixed in Corticon.js 2.3

Customer reported defects fixed in Corticon.js 2.3.0.0

Item	Details
COR-16160	Fixed an issue in Corticon Studio where rule tracing in ruletests failed to correctly disambiguate between rulesheets with identical names located in different filesystem folders within a ruleflow. While the ruletest would execute and generate a rule trace, attempting to open a rulesheet from a trace line could lead to the incorrect rulesheet being opened. Corticon Studio now correctly identifies and opens the appropriate rulesheet.
COR-12991	Fixed an issue in Corticon.js rule tests where specifying Date attributes using numeric values (e.g., 1883952000 as time since epoch) would incorrectly trigger a data type mismatch warning. Although the runtime has always supported numeric date formats, the warning was misleading. This has now been corrected to prevent unnecessary warnings in Corticon.js Studio Tester.
COR-12982	Fixed an issue in Corticon.js rule tests where specifying {{DateTime}} attributes as epoch values (e.g., {{1757782800000}} instead of {{2025-09-13T13:00:00-0400}}) would incorrectly trigger a data type mismatch warning. Although the numeric epoch format is supported by the Corticon.js runtime, the warning was misleading. This has now been corrected to prevent unnecessary warnings in Corticon.js Studio Tester.

Known Issues in Corticon.js 2.2

Release Notes for Corticon.js 2.2.0.0

Item	Details
COR-15975	The Corticon Studio installer log file may contain the error "java.io.IOException: The system cannot find the path specified". This error does not cause an installation failure and can be ignored
COR-15884	When using Corticon Studio, if the Windows Display Scale is set to a value other than 100%, the bottom row of prompt suggestions in the AI Assistant may not be visible even with scrolling
COR-12991	Within a Corticon.js rule test, when you specify Date attributes as numeric values (time since epoch) such as 1883952000 instead of 2025-09-13 it is reported as a data type mismatch even though the format is supported. This warning can be ignored, the Corticon.js runtime will accept the numeric format.
COR-12982	Within a Corticon.js rule test, when you specify DateTime attributes as epoch values such as 1757782800000 instead of 2025-09-13T13:00:00-0400 it is reported as a data type mismatch even though the format is supported. This warning can be ignored, the Corticon.js runtime will accept the numeric format
COR-12939	When packaging a JavaScript decision service, Corticon.js both obfuscates and minimizes the generated JavaScript code. The minification process reduces the size of the generated JavaScript bundle. The minification process fails when the line count of generated code exceeds 500,000 lines. This will result in an error in the Corticon.js log file: The code generator has deoptimised the styling of as it exceeds the max of 500KB. This is not a fatal error. It indicates that the generated code was not minimized. The generated decision is still valid and can be deployed. The generated code is still obfuscated.
COR-12654	Using promises and .then asynchronous programming pattern in service callouts is not supported. Running a ruletest from Corticon Studio that calls a service callout which uses this pattern, will hang Corticon Studio.

Issues fixed in Corticon.js 2.2

Customer reported defects fixed in Corticon.js 2.2.0.0
--

Item	Details
COR-16153	Fixed an issue where the Vocabulary Complexity Report was inaccurate when running in supported locales. The report failed to correctly account for attributes when executed in locales other than English, resulting in discrepancies.
COR-15957	Fixed the issue where Rule tracing in ruletests in Corticon Studio could not disambiguate rulesheets with identical names in a ruleflow. Corticon Studio will now open the correct rulesheet link in the trace results.
COR-10068	Fixed an issue where collection operators could sometimes trigger a Corticon Runtime Exception. This was due to improper handling of context changes across rules.

Known Issues in Corticon.js 2.1

Release Notes for Corticon.js 2.1

Item	Details
COR-12991	Within a Corticon.js rule test, when you specify Date attributes as numeric values (time since epoch) such as 1883952000 instead of 2025-09-13 it is reported as a data type mismatch even though the format is supported. This warning can be ignored, the Corticon.js runtime will accept the numeric format.
COR-12982	Within a Corticon.js rule test, when you specify DateTime attributes as epoch values such as 1757782800000 instead of 2025-09-13T13:00:00-0400 it is reported as a data type mismatch even though the format is supported. This warning can be ignored, the Corticon.js runtime will accept the numeric format.
COR-12939	<p>When packaging a JavaScript decisions service, Corticon.js both obfuscates and minimizes the generated JavaScript code. The minification process reduces the size of the generated JavaScript bundle. The minification process fails when the line count of generated code exceeds 500,000 lines. This will result in an error in the Corticon.js log file:</p> <p>The code generator has de-optimised the styling of as it exceeds the max of 500KB.</p> <p>This is not a fatal error. It indicates that the generated code was not minimized. The generated decision is still valid and can be deployed. The generated code is still obfuscated.</p>
COR-12654	Using promises and .then asynchronous programming pattern in service callouts is not supported. Running a ruletest from Corticon Studio that calls a service callout which uses this pattern, will hang Corticon Studio.
COR-12435	The rule trace view in Corticon Studio does not properly distinguish between rulesheets with identical names. If the rule trace statements from running a ruletest has entries for different rulesheets with identical names, the go to source navigation may not navigate to the correct rulesheet.

Issues fixed in Corticon.js 2.1

Customer reported defects fixed in Corticon.js 2.1
--

Item	Details
COR-13216	Fixed an issue where using an OR operator combined with a postfix operator caused an execution failure.
COR-13193	Fixed an issue where using a Boolean literal in an expression with the OR operator caused an execution failure.
COR-12880	Added missing sample link to documentation: [https://docs.progress.com/ja-JP/bundle/corticon-js-integration/page/Customized-database-operations.html]
COR-12879	Added missing sample link to documentation: [https://docs.progress.com/ja-JP/bundle/corticon-js-integration/page/Customized-database-operations.html]
COR-12857	Fixed a problem where changing "JSON Element Name" for an attribute in a Corticon vocabulary would introduce errors when running Corticon rule tests. Changes made to the JSON mapping in a vocabulary were not propagating to existing rule tests resulting in Corticon using the prior mapping in the JSON payload when running the rule test. The changes are now propagated and the updated mapping used when running the rule test.
COR-12660	Fixed a problem where the files names generated for service contracts and reports created from Corticon Studio would have "%20" in the generated file names in place of any spaces in the rule flow name. The generated file names will now contain spaces.
COR-12146	Fixed an issue where a rule project would not be automatically revalidated when removing a service callout from the project in Corticon.js Studio. Any ruleflow using the service callout would not be flagged as having an error. If there were any ruleflows using the removed callout, ruletests using the ruleflows would not run and the ruleflows would not be able to be packaged for deployment. Any references to the removed callout would need to be manually removed from ruleflows. This has been fixed.
COR-11516	Fixed an issue where files names generated for reports created from Corticon Studio have '%20' in the generated file names in place of any spaces in the asset name. Now the generated file names contain spaces correctly.
COR-9691	Fixed an issue where custom functions were not automatically reloaded by Corticon.js Studio Tester when the JavaScript files were modified. When running ruletests in Corticon.js Studio, the JavaScript files for any custom functions were not loaded into memory, when the files were modified. This resulted in the prior implementation of the function being used in the rule test. This has been fixed, now the files are correctly detected when modified and reloaded by Studio tester for rule test execution.

Known Issues in Corticon.js 2.0

Release Notes for Corticon.js 2.0.0.0

Item	Details	Knowledge Base
COR-12654	Using <code>promise</code> and <code>.then</code> asynchronous programming pattern in service callouts is not supported. Running a ruletest from Corticon Studio that calls a service callout which uses this pattern, will hang Corticon Studio.	
COR-12652	<p>Using <code>promise</code> and <code>.then</code> asynchronous programming pattern in service callout is not supported. Instead, you should use the <code>await</code> asynchronous programming pattern.</p> <p>For example, this code using <code>.then</code>:</p> <pre>const promise = getSomeDataAsynchronously(); promise.then((data) => { // Got the data, now update the payload with it. });</pre> <p>Should be refactored to using <code>await</code> like this:</p> <pre>const data = await getSomeDataAsynchronously(); // Got the data, now update the payload with it.</pre> <p>For a full SCO working example using the <code>await</code> keyword, access the Corticon.js GitHub samples repository at ServiceCallOut/BasicAsyncOperation</p>	
COR-12435	The rule trace view in Corticon Studio does not properly distinguish between rulesheets with identical names. If the rule trace statements from running a ruletest has entries for different rulesheets with identical names, the go to source navigation may not navigate to the correct rulesheet.	
COR-12146	When removing a service callout from a project in Corticon.js Studio, the project is not automatically revalidated. Any ruleflow using the service callout will not be flagged as having an error. If there are any ruleflows using the removed callout, ruletests using the ruleflow will not run and the ruleflow will not be able to be packaged for deployment. Any references to the removed callout will need to be removed from ruleflows.	
COR-11516	The files names generated for reports created from Corticon Studio have "%20" in the generated file names in place of any spaces in the asset name. The generated file names should contain spaces.	
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.	

Item	Details	Knowledge Base
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.	

Issues fixed in Corticon.js 2.0

Customer reported defects fixed in Corticon.js 2.0.0.0
--

Item	Details	Knowledge Base
COR-12395	Modified the Corticon.js Studio ruletest editor export feature to export decimal values as quoted strings in the exported JSON. This is to handle differences in the representations of decimals in Corticon and numbers in JavaScript. At runtime, Corticon.js accepts decimal values as number or quoted strings.	
COR-12393	Fixed a problem where Corticon.js did not properly handle two rulesheets with identical names but different folder locations within a rule project. At runtime, the first rulesheet instance matching the name was executed. The correct rulesheet is now executed.	
COR-12216	Fixed a problem in Corticon.js Studio where rulesheet validation messages could be erroneously duplicated in the Problems View.	
COR-11515	Fixed a problem where 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-8413	<p>Fixed a problem when a Corticon.js decision service is deployed as a Microsoft Azure Function. The property:</p> <pre>logRuleMessages = true</pre> <p>is now supported. Previously rule messages were not logged to the Azure console; they were only returned in the response payload.</p>	

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.	-