Info Namespace Reference

Access info state and operations via scripts.

The **Info Namespace** exposes runtime info state to .NET scripts. Use **`@Info`** to reference the namespace, followed by the accessors listed below.

Child Namespaces
Info Members

Child Namespaces

Sub-namespaces exposing collections and grouped configuration. Reach each via its qualified path.

Path	Description
`@Info.categoryItem` (1 children)	Gets the collection of solution categories, each exposing the per-category settings (name, color, icon, enabled flag) that drive category-scoped runtime behavior.
`@Info.license` (32 members)	Gets the license-information object: serial number, product family / model, allowed-capacity numbers (web clients, devices, tag elements, protocols), expiration / subscription dates and remote-licensing status. Use to license-gate features in scripts (e.g., disable a feature when AllowedDataHubStation is false). Per-machine.
`@Info.module` (15 children)	Gets the collection of runtime modules (Alarm, Device, Historian, Script, etc.), each exposing its current execution status, enabled flag and module-specific diagnostics.
`@Info.product` (4 members)	Gets the product-level information object: product name, company, full version string and build, independent of the specific solution being run.
`@Info.solution` (15 members)	Gets the solution-level information object: product family / model, version, execution folders, project category, and everything else authored at the .dbsln level.

Info Members

Direct properties and methods on `@Info`.

Member	Kind	Type	Description
`evaluationMessage`	Property	`String`	Gets the human-readable status message describing the current evaluation-mode state (e.g., reason the runtime is in evaluation, time-remaining hint, expiration warning). Useful as a banner on operator screens when EvaluationMode is true.
`evaluationMode`	Property	`Boolean`	Gets whether the solution is running in evaluation mode (no valid license, or a license error has been reported). True when no production license is active and the runtime is operating on the time-limited evaluation grace period; pair with EvaluationTimeRemaining and EvaluationMessage to surface remaining time and reason on operator screens.
`evaluationTimeRemaining`	Property	`TimeSpan`	Gets the remaining wall-clock time before the current evaluation-mode runtime session expires. Meaningful only while EvaluationMode is true; otherwise returns TimeSpan.Zero. Counts down as the session runs and reaches zero when the evaluation grace period ends, at which point the runtime stops accepting new work.
`executionPath`	Property	`String`	Gets the runtime executable's working folder for the current process — the folder that hosts FactoryStudio's bin output (or the OEM-equivalent install path) — with a trailing path separator. Per-process; on a typical install matches the FactoryStudio\bin folder. Use as a base when building paths to runtime artifacts; use GetExecutionFolder when the trailing separator is unwanted.
`getExecutionFolder`	Method	`String`	*(no description)*
`getExecutionPath`	Method	`String`	*(no description)*
`getTypeDefinition`	Method	`T.Library.TTypeTable`	*(no description)*

`lastInvalidSetValue`	Property	`String`	Gets the last value the runtime rejected as out-of-range / invalid when a script tried to write it to a tag. Set by the runtime whenever a SetValue is rejected by the LimitValuesMinMax / type validation; useful for diagnosing why an expected write did not take effect. Cleared on next successful write attempt.
`onlineConfig`	Property	`Boolean`	Gets whether the runtime is running in online-configuration mode — solution edits made in the Designer apply to the running runtime without restart. True on a Designer-attached debug session or any runtime configured for online edits; false on a production runtime where configuration is locked.
`profile`	Property	`Int32`	Gets the integer code of the active Execution Profile (0=Production, 1=Development, 2=Validation, 3=Custom). The profile is selected when the runtime starts and determines per-environment overrides such as database routing and AI-safety gates. Use ProfileName when a human-readable label is needed.
`profileName`	Property	`String`	Gets the human-readable name of the active Execution Profile ("Production", "Development", "Validation", "Custom"). Useful for branching script logic per environment (e.g., write to a test report folder when ProfileName == "Development"). Pairs with the Profile integer code.
`trace`	Method	`Int32`	*(no description)*
`traceInformation`	Method	`Int32`	*(no description)*
`traceMessageAndType`	Method	`Int32`	*(no description)*

Member set generated from FrameworX 10.1.5. For the full .NET API surface see the external Info Namespace .NET API Reference.

In this section...