

iCanScript Release Notes



iCanScript v2.0.24

July 22, 2015

V2.0.24 introduces the iCanScript packages!

A package allows to configure common traits for a group of visual scripts. Packages can be independent or nested. The package controls the following visual script traits:

- engine & editor namespaces;
- engine & editor code generation folders;

- engine & editor visual script folders (future use).

[WARNING] EXAMPLES FILES prior to v2.0.24:

Each example created prior to V2.0.24 uses a separate namespace but a common code generation folder for all of the examples. With packages, the namespaces and code generation folders should be separate. Therefore, these examples must be adapted to work with V2.0.24.

List of Changes

- **[NEW]** **iCanScript Packages:**
 - A package allows to configure common traits for a group of visual scripts. Packages can be independent or nested. The package controls the following visual script traits:
 - engine & editor namespaces;
 - engine & editor code generation folders;
 - engine & editor visual script folders (future use).
 - A default package with the name of the Unity project is automatically generated. This package may be sufficient for small projects.
- **[NEW]** Added ***iCanScript->Packages...*** menu item to access the package settings.
- **[IMPROVED]** Support preprocessor directives in inline code node.
- **[FIXED]** Fix sporadic index out of range exception when generating code.
- **[FIXED]** Fix some conditions that generated unnecessary ***“if(trigger)”*** code.
- **[FIXED]** Fix code generated for ***Color*** using RGBA.

iCanScript v2.0.23

V2.0.23 was internally used for testing features in development and was not released to the user community.

iCanScript v2.0.22

June 26, 2015

V2.0.22 is a minor bug fix release.

List of Changes

- **[FIXED]** Variables created for Unity Objects are no longer allocated.
 - **[FIXED]** Invalid generated code for specific enable port conditions.
-

iCanScript v2.0.21

June 20, 2015

V2.0.21 now allows to configure the function access and scope specification. In addition, it includes the ability for the user to define an **Inline** code node to inject textual code inside the generated code.

[WARNING] V2.0.20 requires Unity V5.1.0 or later. Please upgrade Unity before using this version of iCanScript.

List of Changes

- **[NEW]** The function access (*public, private, protected*) and scope (*static, virtual, override, new*) specifications can now be configured using *Function Definition Editor*.
- **[NEW]** A new node type **Inline Code*** has been added.
 - The **Inline Code** node can be used to injecting your own source code into the generated code.
 - The **Inline Code** can be added to your graph from the contextual menu.
- **[NEW]** Operator *Is Equal* (==) and *Is Not Equal* (!=) as been added for generic *Object* type in the *CSharp.Primitives.Object* type in the *Library*.
- **[FIXED]** The *Owner* port type is allowed on the *Target* port of a function.
- **[FIXED]** Invalid code generated when a trigger port is added to a package.
- **[FIXED]** Fix parameter index errors when generating source code.

iCanScript v2.0.20

June 8, 2015

V2.0.20 add the ability to define the generated code variable specification associated with the ports.

With V2.0.20, the user can create public or private ports as well as determine if the port is static (class scope) or not (instance scope). Support for Constant, Parameters, and Local Variables is also provided.

[WARNING] V2.0.20 requires Unity V5.0.2 or later. Please upgrade Unity before using this version of iCanScript.

List of Changes

- **[NEW]** Added support for selecting the type of variable to be generated in the port dialog editor:
 - Constant;
 - Public & Private Variables;
 - Static Public & Private Variables;
 - Local Variables;
 - Function Definition Parameters.
- **[NEW]** New port visuals to easily identify the variable type associated with the port.
- **[IMPROVED]** Improve contrast of the selected node.
- **[IMPROVED]** Increase port size to improve variable type visibility.

- **[FIXED]** Remove unneeded casts in the generated code.
 - **[FIXED]** Visual script data upgrade in now perform on import.
-

iCanScript v2.0.19

May 28, 2015

V2.0.19 adds the support for C# primitive variables **Bool**, **Int**, **Float**, and **String**.

List of Changes

- **[NEW]** Added support for the creation and usage of C# primitive types: *bool*, *int*, *float*, *string*. The C# primitive types are located in namespace **CSharp.Primitives**.
 - **IMPORTANT:** The variables in namespace **iCanScript.Variables** should be replaced by their C# equivalent. The iCanScript variables are now obsoleted and will be removed in a future release.
 - **[IMPROVED]** Partial conversion of the API that changed between Unity 5.0.1 and 5.0.2.
 - **NOTE:** To remain compatible with Unity 5.0.1, not all of the API changes have been applied.
 - **[IMPROVED]** Avoid generating variable and function names that collide with C# reserved keywords.
 - **[FIXED]** Null exception when generating comments for a function parameter that has no description field.
-

iCanScript v2.0.18

May 25, 2015

V2.0.18 is a minor bug fix and code generation optimization release.

List of Changes

- **[NEW]** Added support for **operator +(string, Object)** in the C# primitive namespace.
 - **[IMPROVED]** The length of the generated comments have been readjusted to match the forum post width.
 - **[IMPROVED]** Removed unnecessary *"using CSharp.Primitive;"* generated for each file.
 - **[IMPROVED]** Code generation optimization when target port of a function is connected to the output of a property or field.
 - **[FIXED]** Invalid generated code for the **Get** of a *Field* variable.
-

iCanScript v2.0.17

May 20, 2015

V2.0.17 adds the operators for simple primitive types such as Boolean, Int, and Float.

[WARNING] The library *iCanScript.Logic.Boolean* has been replaced by *CSharp.Primitives.Bool*. Note that you may have to regenerate the source code from the visual script if you see errors after you upgrade.

List of Changes

- **[NEW]** Added support for C# primitive variable operators. The new operators can be access from the library under namespace: *CSharp.Primitives*. The primitive types that are affected are:
 - *Bool*, *Int*, and *Float*.
 - **[NEW]** An option to *Show / Hide the Unity Editor library* has been added to the library toolbar.
 - **[IMPROVED]** Properly indent the generated code for the first line of the class declaration.
 - **[IMPROVED]** The Unity event handlers in the dynamic menu are now sorted.
 - **[IMPROVED]** The library options *Show Inherited* is now preserved after a recompile or a restart of Unity.
 - **[CHANGED]** The library options *Show Protected* has been removed from the library toolbar.
 - **[REMOVED]** The library node *iCanScript.Logic.Boolean* has been removed. Use *CSharp.Primitives.Bool* instead.
-

iCanScript v2.0.16

May 20, 2015

V2.0.16 is a code generation bug fix release.

List of Changes

- **[NEW]** Support code generation for function definition input parameters.
 - **NOTE:** Output parameters and return value is not yet supported.
-

iCanScript v2.0.15

May 15, 2015

V2.0.15 is a library database bug fix release.

List of Changes

- **[IMPROVED]** Don't show the inherited members by default in the library window.
 - **[IMPROVED]** Rename library branches for element without a namespace to “– *no namespace* –” instead of *<empty>*.
-

iCanScript v2.0.14

May 15, 2015

V2.0.14 is a critical bug fix release.

List of Changes

- **[FIXED]** Fix issue introduced in v2.0.12 where new visual script could not be created.
 - **[FIXED]** Fix “Unable to find suitable parent” error message when drawing from the library.
-

iCanScript v2.0.13

May 15, 2015

V2.0.13 was rejected due to critical bugs found.

iCanScript v2.0.12

May 13, 2015

V2.0.12 introduces a redesigned library capable of supporting all features of an object-oriented programming language (ex: C# or C++).

The new library component automatically imports any library added to the project and therefore includes all functionality available in the Unity Engine, Unity Editor, and .NET libraries.

WARNING: The option to display in the library the functions with a protected accessibility scope is not functional in this release.

List of Changes

- **[NEW]** The library database and its selection window has been fully designed to support all features of an object-oriented language (C# or C++).
 - **[NEW]** Imports all public libraries and source code available in the Unity project;
 - **[NEW]** Library database hierarchy is structured based on the namespaces and types imported from the libraries.
 - **[NEW]** Three search fields allow to filter the library database content based on the **namespace**, the **type**, and the **member (field / property / function)** names.
- **[CHANGED]** The iCanScript nodes were relocated under the iCanScript namespace to conform with the new library database structure. A data upgrade will be performed the first a scene created with an earlier version of iCanScript.
- **[REMOVED]** The ability to programatically import external libraries into the iCanScript library database has been removed (since all libraries are automatically imported).

- **[REMOVED]** The option to include/exclude the Unity editor library using the *Global Configuration panel* has been removed (the Unity Editor library is always imported).
-

iCanScript v2.0.11

May 8, 2015

V2.0.11 was rejected and therefore not released.

iCanScript v2.0.10

May 1st, 2015

V2.0.10 was rejected and therefore not released.

iCanScript v2.0.9

April 24, 2015

V2.0.9 is a development release of iCanScript2.

List of Changes

- **[NEW]** The **Type Name** of the visual script can now be configured in the *Visual Script Configuration* panel.
- **[NEW]** iCanScript2 can now be used to create *Editor Scripts*.
 - Enable/Disable the *Editor Script* option in the *Visual Script Configuration* to switch between engine and editor scripts.
 - The *Unity Editor Library* must be enabled to create *Editor Visual Scripts*.
- **[NEW]** An option to include the **Unity Editor Library** is now available in the *Code Generation* section of the *Global Preferences*.
 - **NOTE:** A restart is required for this option to take effect.
- **[NEW]** The user can now customize the **code generation folders** for the *Editor* and *Engine* visual scripts from the *Global Preferences*.
- **[NEW]** The user can now customize the **namespace** for code generation:
 - **[NEW]** Separate *Namespaces* for *Editor* & *Engine* code generation can be customized from the *Global Preferences*;
 - **[NEW]** The *Namespace* can be overridden on a visual script basis using the *Visual Script Configuration Panel*.
- **[IMPROVED]** The **“Add Unity Event Handler”** menu item is no longer displayed if the script does not inherit from *MonoBehaviour*.
- **[IMPROVED]** Warning provided to help simplify situation where data and control flow overlap.
- **[FIXED]** The name of the type can now be configured when a port required a Type value. The type name format is: *“namespace.type”*.

- **[FIXED]** Invalid code generated for static functions of “*GameObject*” and “*Transform*”.
-

iCanScript v2.0.8

April 22, 2015

V2.0.8 is a development release of iCanScript2.

List of Changes

- **[NEW]** Added a *Visual Script Configuration Panel* accessible from the visual script toolbar.
 - **[NEW]** The user can now configure the **base type (inheritance)** for the type defined by the visual script.
 - **[NEW]** The global *Base Type* can be configured in the *Global Preferences*;
 - **[NEW]** The *Base Type* can be overridden for each visual script in the new *Visual Script Configuration Panel*.
 - **[FIXED]** Null exception when attempting to import a field with its value set *null*.
 - **[FIXED]** Fields defined in a base class not properly imported in derived classes resulting in a type not found error.
-

iCanScript v2.0.7

April 20, 2015

V2.0.7 resolves two major issues introduced in release v2.0.6. Sorry for the inconvenience.

List of Changes

- **[FIXED]** Enumeration null exception on data upgrade of visual script.
 - **[FIXED]** Invalid error generated on *get* accessor of fields.
-

iCanScript v2.0.6

April 20, 2015

V2.0.6 is a development release of iCanScript2.

List of Changes

- **[IMPROVED]** Add the ‘*Self*’ port on all function nodes.
 - **[FIXED]** Disabled nodes are now displayed with half intensity in editor mode.
-

iCanScript v2.0.5

April 19, 2015

V2.0.5 is a development release of iCanScript2.

List of Changes

- **[IMPROVED]** Automatically generated type-cast now uses the most specialized type instead of the common base type.
 - **[FIXED]** Fix code generation when multiple enables exists on a single node.
-

iCanScript v2.0.4

April 17, 2015

V2.0.4 is a development release of iCanScript2.

List of Changes

- **[IMPROVED]** Performance of panning canvas.
 - **[IMPROVED]** Improve name of trigger port in generated code.
 - **[IMPROVED]** Restructuring of the dynamic menu for future support of variable & nested type creation.
 - **[IMPROVED]** Remove illegal option to add *enable/trigger* ports on a Unity event handler.
 - **[IMPROVED]** Remove unused menu item *+ Iterator* from contextual menu.
 - **[FIXED]** Fix code generation when *Target* port is connected to a package input port that is initialized to *Owner*.
-

iCanScript v2.0.3

April 16, 2015

V2.0.3 is a development release of iCanScript2.

V2.0.3 introduces the concept of the script *Owner* which is used to configure input ports to designate the game object on which the script is installed.

List of Changes

- **[NEW]** Default *Target* port value to script *Owner* if not connected and of type *Transform* or *GameObject*.
- **[IMPROVED]** Prefix all constructors with the keyword *New* in the library.
- **[FIXED]** Opening the port/node editor now requires a double click.
- **[FIXED]** Remove generation of extra function parameters when input port is connected to a variable defined outside of the

function.

- **[FIXED]** Generate upcast when the *Target* port when needed.
-

iCanScript v2.0.2

April 15, 2015

V2.0.2 is a development release of iCanScript2.

List of Changes

- **[NEW]** Error display when a function node cannot find its runtime code.
 - **[IMPROVED]** In the contextual menu, rename menu item “**+ Out Instance Port**” to “**+ Add Self Port**”.
 - **[IMPROVED]** In the contextual menu, rename menu item “**+ Enable Port**” to “**+ Add Enable Port**”.
 - **[IMPROVED]** In the contextual menu, rename menu item “**+ Trigger Port**” to “**+ Add Trigger Port**”.
 - **[FIXED]** Prepend namespace to generated type names when type exists in more than one namespace.
-

iCanScript v2.0.1

April 14, 2015

V2.0.1 is a development release of iCanScript2.

List of Changes

- **[FIXED]** Generation of the “*using*” directive is now dependent on the content of the visual script.
 - **[IMPROVED]** Addition of the Unity “*Awake()*” event handler in the dynamic menu.
-

iCanScript v2.0.0

March 20, 2015

V2.0.0 is the initial development release of iCanScript2.