



ViSi-Genie Magic Slide to Unlock with a Host

DOCUMENT DATE: **8th MAY 2020**
DOCUMENT REVISION: **1.1**



Description

This application note serves as a quick reference manual for beginners who are unfamiliar with the Genie objects under the **Magic** pane and the procedure for adding them to a project.

Before getting started, the following are required:

- [Workshop 4 IDE](#) (installed according to the installation document)
- When downloading an application note, a list of recommended application notes is shown. It is assumed that the user has read or has a working knowledge of the topics presented in these recommended application notes.

Content

Description	2
Content	2
Application Overview	3
The Magic Pane.....	4
<i>1.1 How to Add a Magic Event Object</i>	<i>5</i>
1.1.1 Select the Object Icon	5
1.1.2 Open the 4DGL Code	5
1.1.3 Relevant Application Notes and Examples	6
<i>1.2 How to Add a Magic Touch Object</i>	<i>6</i>
1.2.1 Select the Object Icon	6
1.2.2 Open the 4DGL Code	7
1.2.3 Relevant Application Notes and Examples	7
<i>1.3 How to Add a Magic Move Object</i>	<i>8</i>
1.3.1 Select the Object Icon	8
1.3.2 Open the 4DGL Code	8
1.3.3 Relevant Application Notes and Examples	9
<i>1.4 How to Add a Magic Release Object</i>	<i>9</i>
1.4.1 Select the Object Icon	9
1.4.2 Open the 4DGL Code	10
1.4.3 Relevant Application Notes and Examples	10
<i>1.5 How to Add a Magic Keyboard + Color Picker Event.....</i>	<i>11</i>

1.5.1 Select the Object Icon	11
1.5.2 Open the 4DGL Code	11
1.5.3 Relevant Application Notes and Examples	12
<i>1.6 How to Add a Magic Code Object.....</i>	<i>12</i>
1.6.1 Select the Object Icon	12
1.6.2 Magic Code Insertion Point	13
1.6.3 Open the 4DGL Code	13
1.6.4 Relevant Application Notes and Examples	14
<i>1.7 How to Add a Magic Object.....</i>	<i>14</i>
1.7.1 Select the Object Icon	14
1.7.2 Open the 4DGL Code	15
1.7.3 Relevant Application Notes and Examples	15
Proprietary Information	16
Disclaimer of Warranties & Limitation of Liability.....	16

Application Overview

Genie Magic is an extension to Genie that requires Workshop PRO to function. It allows for the addition of any standard 4DGL code at various points in the Genie environment. Extra functionality could include almost anything the user can imagine. Examples are:

- Support R/W access to uSD from a host
- Produce sounds when objects are touched
- Manipulate multiple objects in different ways by a single write command
- Read data from I/O ports and write this data to Genie Objects
- Collect and edit keyboard input sending result to host
- Perform multiple updates with a single write command
- 'Join' multiple objects together to get them to behave as a single object
- Allow Genie to perform many more functions in a 'hostless' environment

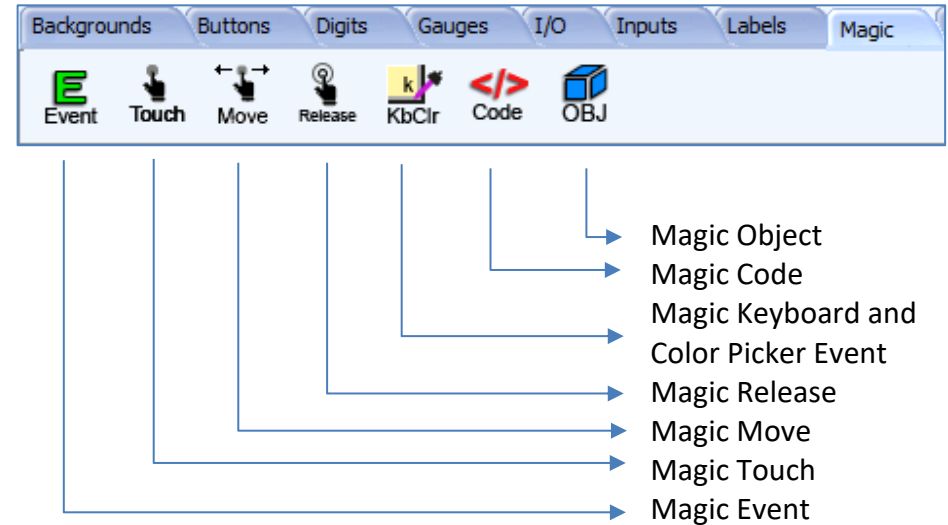
Custom 4DGL code, in the form of a Genie Magic object, can be added to a Genie project at the following points or locations.

Genie Magic Object	Location
Magic Event Magic Keyboard + Color Picker Event	Event Handlers
Magic Touch Magic Move Magic Release	Touch press action point Touch move action point Touch release action point
Magic Code	Specific code positions Constant / Global / Data definition Main Loop Pre/Post Activate Form Pre/Post Genie Initialization
Magic Object	An object to interface to the host

This application note shows the procedures for adding the Genie Magic objects to a project and the procedure for opening the code for the objects. The relevant examples and application notes provided at the end of each section will demonstrate possible uses and applications of the Genie Magic objects.

The Magic Pane

There are seven objects under the Magic pane.

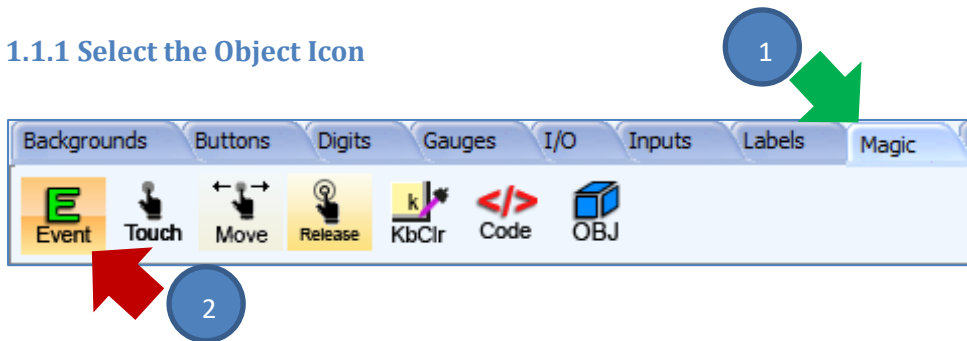


Note: When added to a project, these objects will reside in **Form0**.

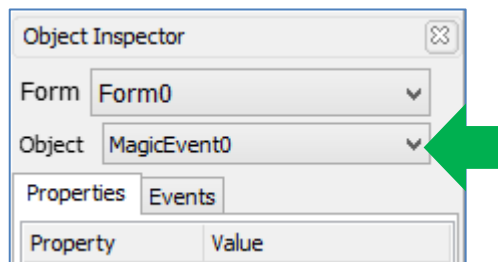
1.1 How to Add a Magic Event Object

Events are usually triggered when input objects such as winbutton objects on the display are pressed. In the Genie environment of the standard Workshop IDE, the OnChanged event property of a winbutton, for example, can be configured to toggle the state of a LED object. In Workshop Pro, it is now possible for the user to create a custom event, through 4DGL coding, in the form of a Magic Event object. For instance, a winbutton, when pressed, can make a LED object blink ten times.

1.1.1 Select the Object Icon

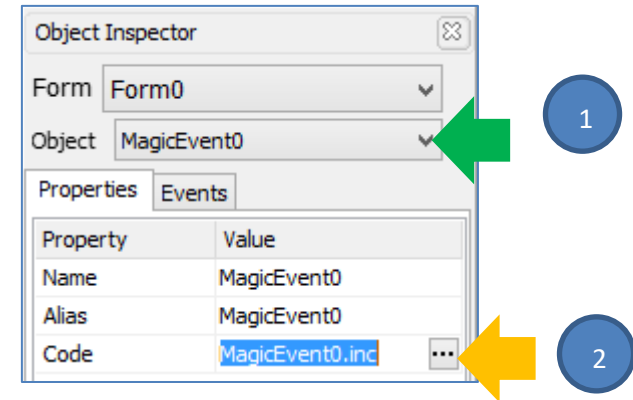


After selecting the object icon, the object is automatically added to **Form0**. This is **MagicEvent0**.

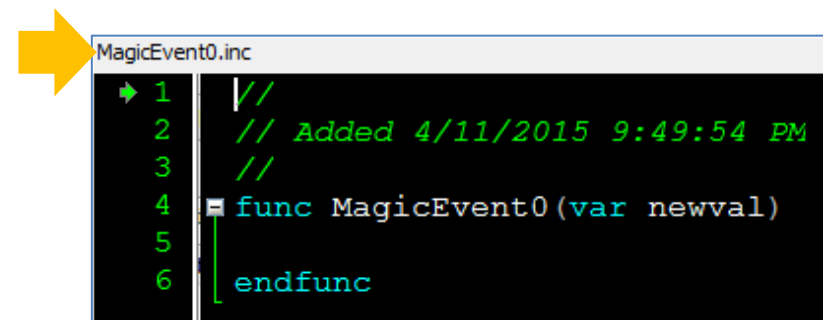


1.1.2 Open the 4DGL Code

To open the code for **MagicEvent0**, click on the ellipsis symbol indicated below.



The code for **MagicEvent0** opens.



The user can now insert custom 4DGL code into the empty function.

1.1.3 Relevant Application Notes and Examples

For an actual usage or application of this object, please see the following application note(s) and/or example(s).

[ViSi-Genie Magic Slide to Unlock](#)

[ViSi-Genie Magic Main Loop Counter](#)

[ViSi-Genie Magic Button Counters](#)

Workshop -> File -> Samples -> ViSi Genie Magic (Picaso/Diablo):

SlideToUnlock.4DGenie

MainLoopCounter.4DGenie

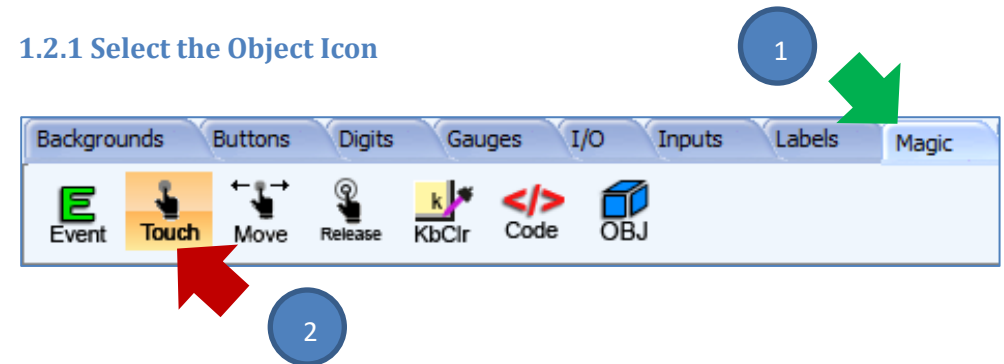
UpDownRepeat.4DGenie

SliderMult.4DGenie

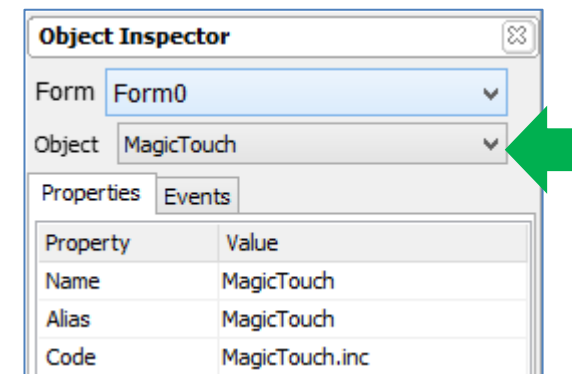
1.2 How to Add a Magic Touch Object

A Magic Touch object contains a 4DGL routine that is executed the moment that a "TOUCH_PRESSED" action is detected on the display. For example, the description for a button with an icon can be printed the moment that it is pressed.

1.2.1 Select the Object Icon



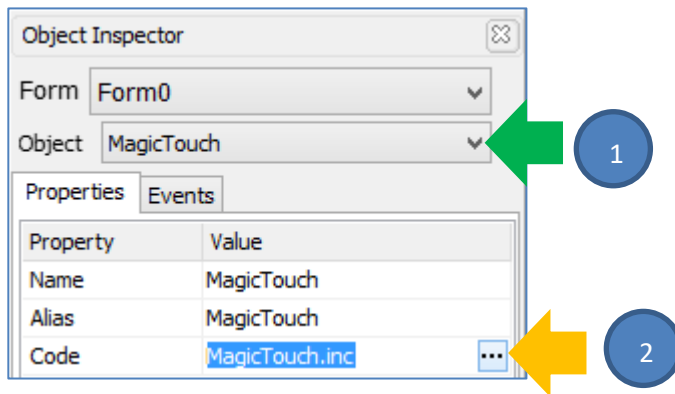
After selecting the object icon, the object is automatically added to Form0. This is **MagicTouch**.



Note: Only one instance of the **Magic Touch** object is possible inside a project.

1.2.2 Open the 4DGL Code

To open the code for **MagicTouch**, click on the ellipsis symbol indicated below.



The code for **MagicTouch** opens.

```
MagicTouch.inc
1 //
2 // Added 4/11/2015 10:57:05 PM
3 //
4 // Use 'ImageTouched' to detect 'current' object
5 // to determine the object for which a touch has occurred
6 //
```

The user can now insert custom 4DGL code following the instructions in the comments.

1.2.3 Relevant Application Notes and Examples

For an actual usage or application of this object, please see the following application note(s) and/or example(s).

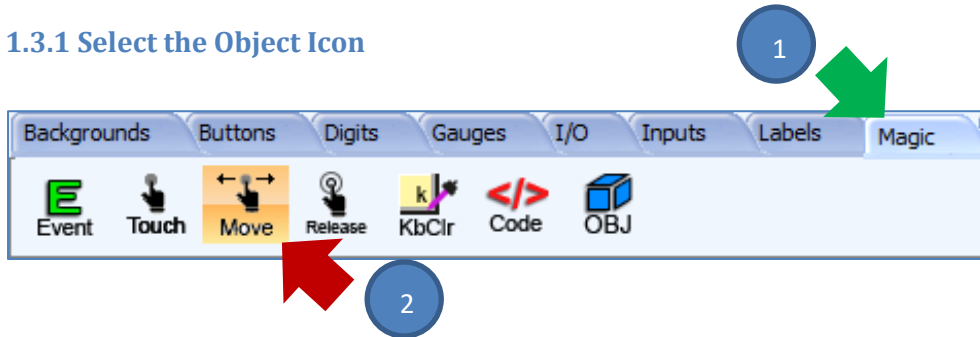
[ViSi-Genie Magic Beep Bop](#)

Workshop -> File -> Samples -> ViSi Genie Magic (Picaso/Diablo):
BeepBop.4DGenie

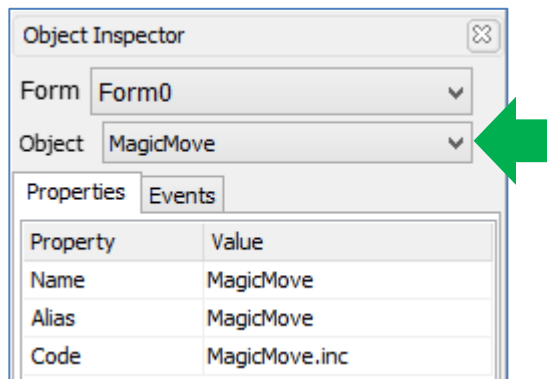
1.3 How to Add a Magic Move Object

A Magic Move object contains a 4DGL routine that is executed the moment that a "TOUCH_MOVING" action is detected on the display. For example, a line can be drawn along the path of a stylus as it is being dragged across a large button.

1.3.1 Select the Object Icon



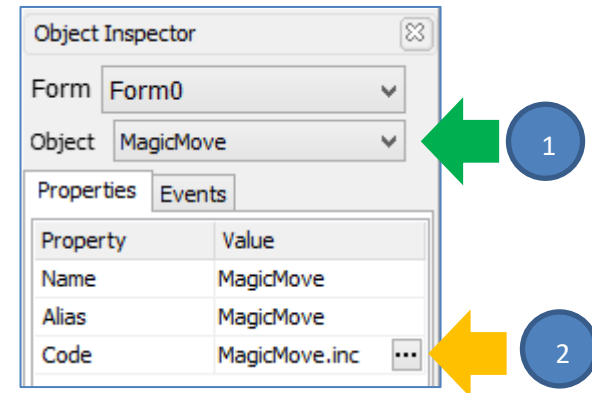
After selecting the object icon, the object is automatically added to Form0. This is **MagicMove**.



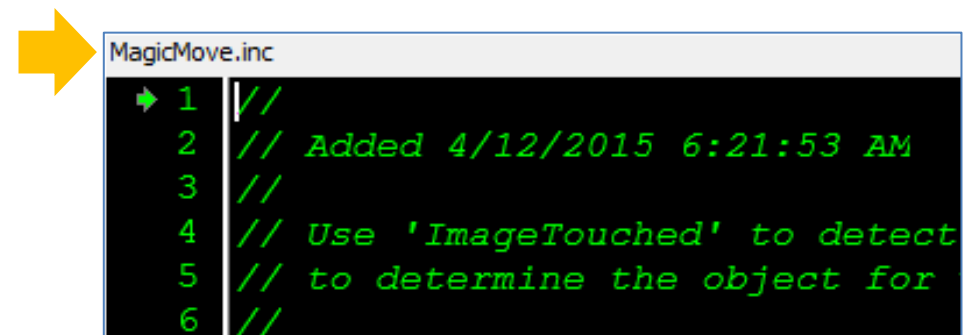
Note: Only one instance of the **Magic Move** object is possible inside a project.

1.3.2 Open the 4DGL Code

To open the code for **MagicMove**, click on the ellipsis symbol indicated below.



The code for **MagicMove** opens.



The user can now insert custom 4DGL code following the instructions in the comments.

1.3.3 Relevant Application Notes and Examples

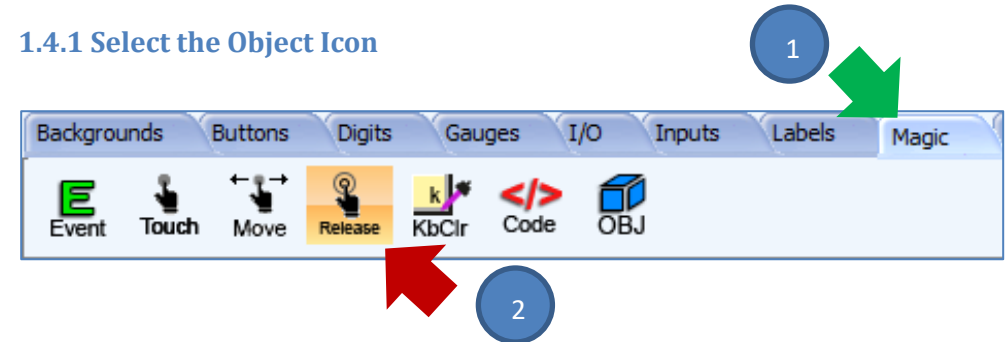
For an actual usage or application of this object, please see the following application note(s) and/or example(s).

[ViSi-Genie Magic The Magic Move Object](#)

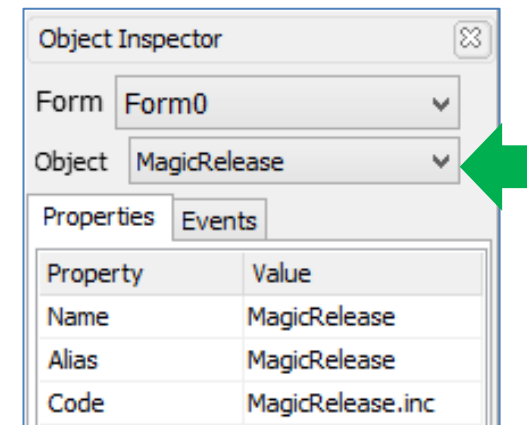
1.4 How to Add a Magic Release Object

A Magic Release object contains a 4DGL routine that is executed the moment that a “TOUCH_RELEASED” action is detected on the display. For example, the printed description for a button with an icon can be “erased” the moment that the button is pressed.

1.4.1 Select the Object Icon



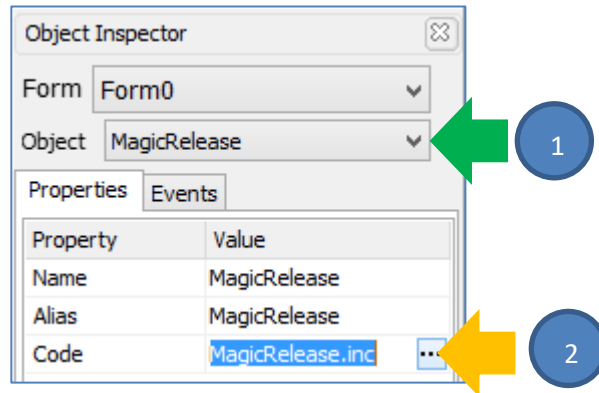
After selecting the object icon, the object is automatically added to Form0. This is **MagicRelease**.



Note: Only one instance of the **Magic Release** object is possible inside a project.

1.4.2 Open the 4DGL Code

To open the code for **MagicRelease**, click on the ellipsis symbol indicated below.



The code for **MagicRelease** opens.

```

1 //
2 // Added 4/12/2015 6:36:44 AM
3 //
4 // Use 'ImageTouched' to detect 'cu
5 // to determine the object for whic
6 //

```

The user can now insert custom 4DGL code following the instructions in the comments.

1.4.3 Relevant Application Notes and Examples

For an actual usage or application of this object, please see the following application note(s) and/or example(s).

[ViSi-Genie Magic Beep Bop](#)

[ViSi-Genie Magic Exit Genie](#)

Workshop -> File -> Samples -> ViSi Genie Magic (Picaso/Diablo):

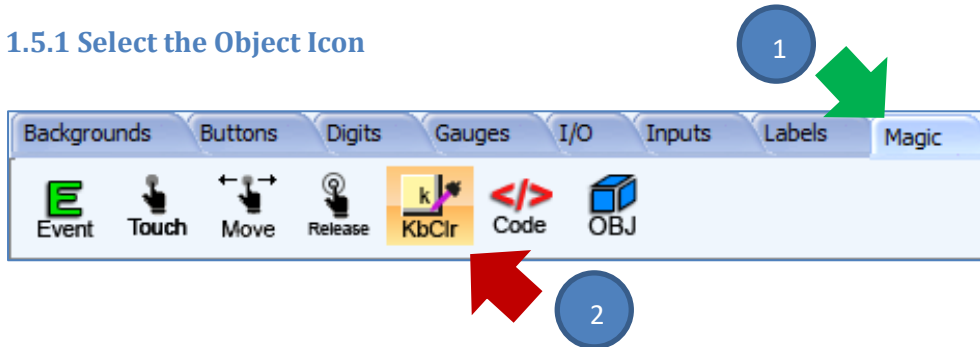
BeepBop.4DGenie

ExitGenie.4DGenie

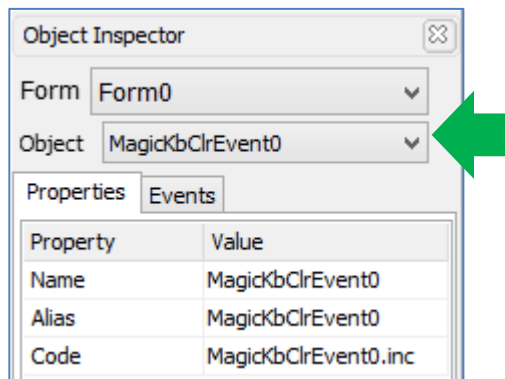
1.5 How to Add a Magic Keyboard + Color Picker Event

Similar to the Magic Event object, the Magic Keyboard + Color Picker Event object is a custom event. More specifically, the Magic Keyboard + Color Picker Event is for handling events from keyboard and color picker objects.

1.5.1 Select the Object Icon

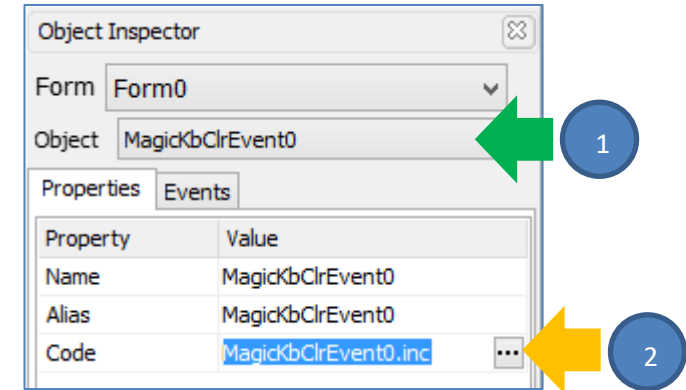


After selecting the object icon, the object is automatically added to Form0. This is **MagicKbClrEvent0**.

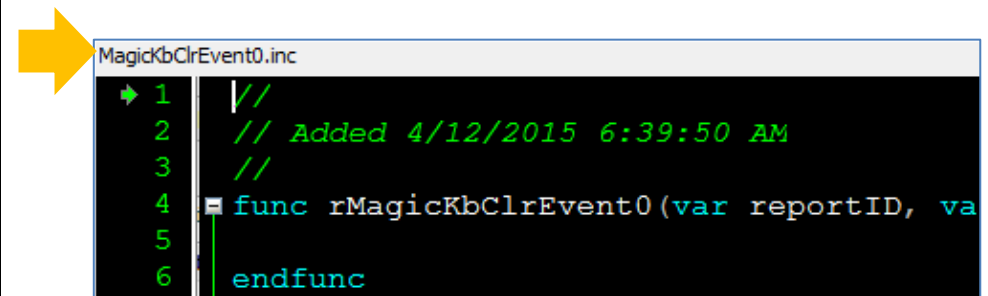


1.5.2 Open the 4DGL Code

To open the code for **MagicKbClrEvent0**, click on the ellipsis symbol indicated below.



The code for **MagicKbClrEvent0** opens.



The user can now insert custom 4DGL code into the empty function.

1.5.3 Relevant Application Notes and Examples

For an actual usage or application of this object, please see the following application note(s) and/or example(s).

[ViSi-Genie Magic Color Picker Blob](#)

[ViSi-Genie Magic Keyboard Edit](#)

Workshop -> File -> Samples -> ViSi Genie Magic (Picaso/Diablo):

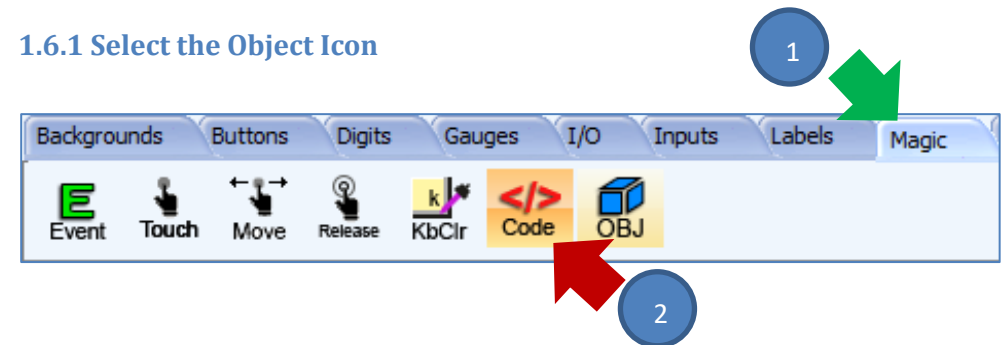
ColorPickerBlob.4DGenie

KeyboardEdit.4DGenie

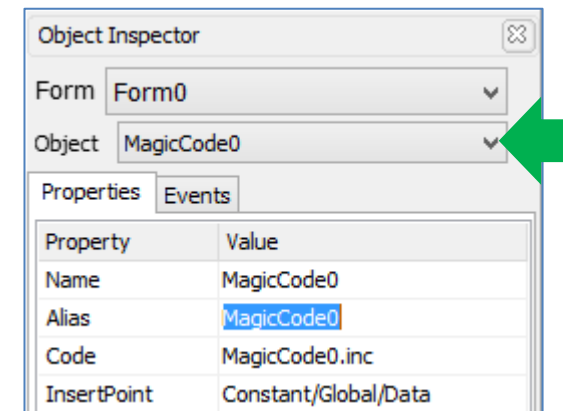
1.6 How to Add a Magic Code Object

A Magic Code object allows the user to insert custom 4DGL code into specific locations inside the Genie project. For example, a counter variable can be declared and initialized in a Magic Code object inserted to the location “**Constant/Global/Data**”. This variable can then be accessed and used by another Magic Code object inserted at the location “**MainLoop**”.

1.6.1 Select the Object Icon

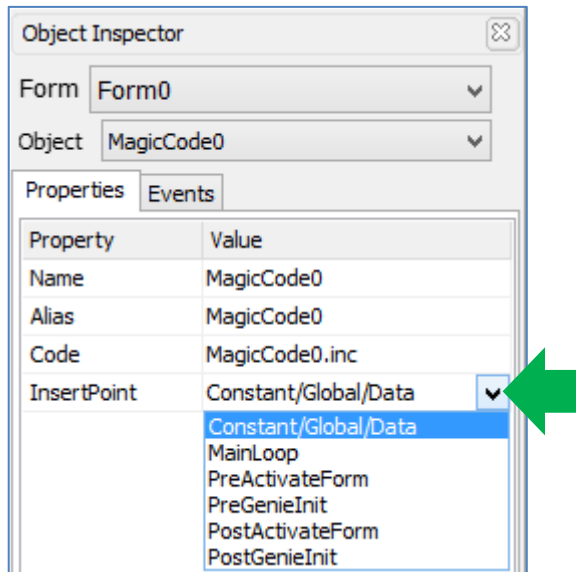


After selecting the object icon, the object is automatically added to Form0. This is **MagicCode0**.



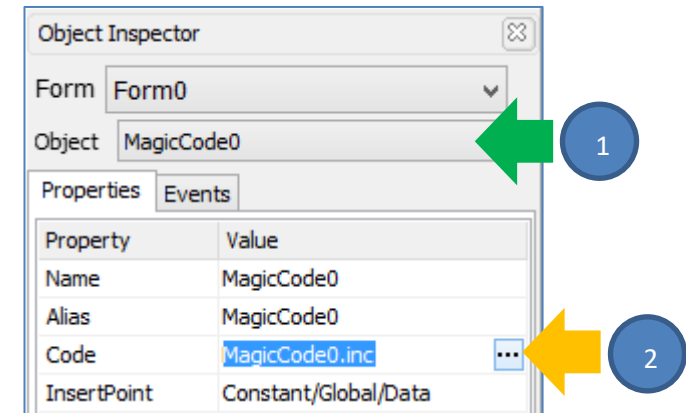
1.6.2 Magic Code Insertion Point

A Genie project is a complex program. Choose where to insert the Magic Code object by selecting a location from the drop-down menu indicated below. The default insertion point or location is “Constant/Global/Data”.

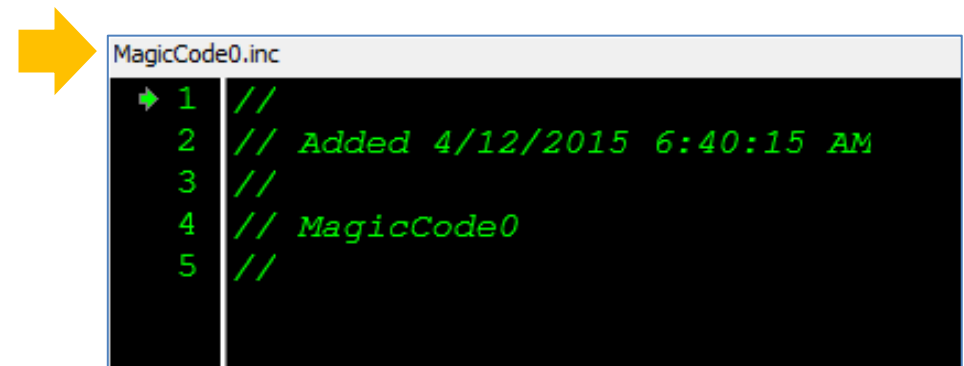


1.6.3 Open the 4DGL Code

To open the code for **MagicCode0**, click on the ellipsis symbol indicated below.



The code for **MagicCode0** opens.



The user can now insert custom 4DGL code into the empty code.

1.6.4 Relevant Application Notes and Examples

For an actual usage or application of this object, please see the following application note(s) and/or example(s).

[ViSi-Genie Magic Main Loop Counter](#)

[ViSi-Genie Magic Keyboard Edit](#)

Workshop -> File -> Samples -> ViSi Genie Magic (Picaso/Diablo):

MainLoopCounter.4DGenie

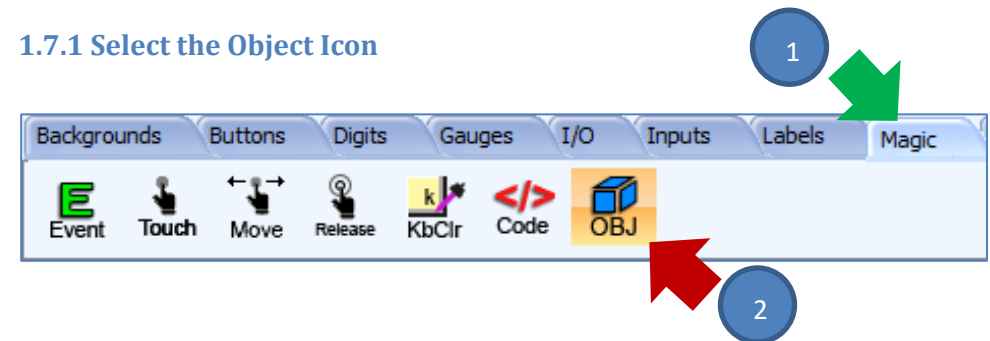
KeyboardEdit.4DGenie

AutoScope.4DGenie

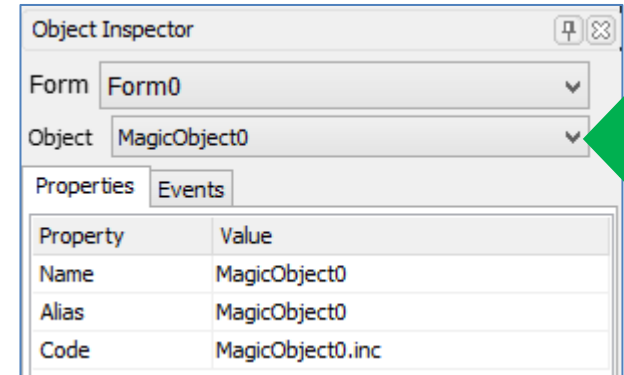
1.7 How to Add a Magic Object

A Magic Object allows the user to write a custom handler for messages received from the serial port.

1.7.1 Select the Object Icon

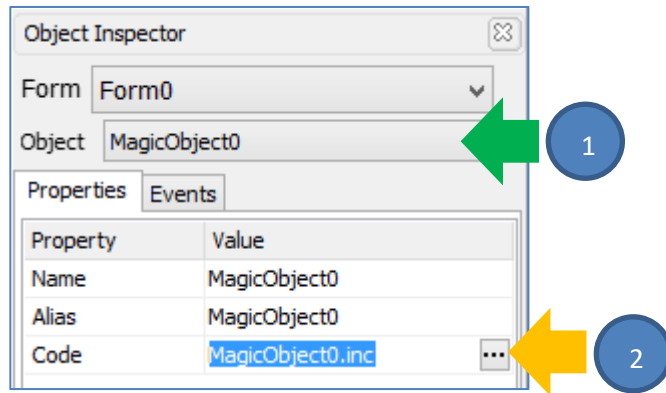


After selecting the object icon, the object is automatically added to Form0. This is **MagicObject0**.

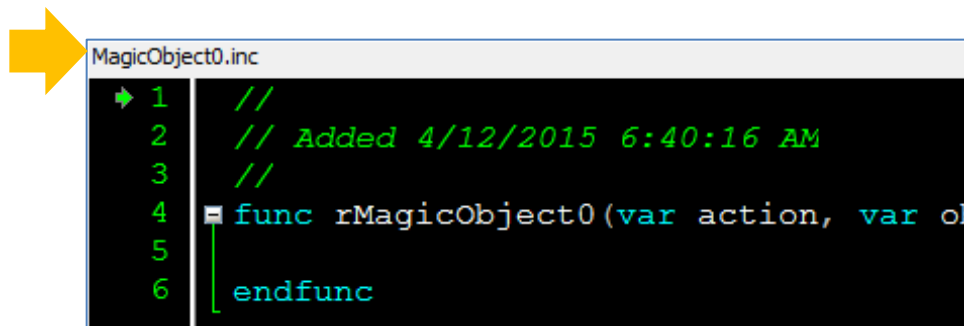


1.7.2 Open the 4DGL Code

To open the code for **MagicObject0**, click on the ellipsis symbol indicated below.



The code for **MagicObject0** opens.



The user can now insert custom 4DGL code into the empty function.

1.7.3 Relevant Application Notes and Examples

For an actual usage or application of this object, please see the following application note(s) and/or example(s).

[ViSi-Genie Magic 32bit LED Digits](#)

[ViSi-Genie Magic How to Read a File](#)

[ViSi-Genie Magic How to Append to a File](#)

[ViSi-Genie Magic File Size](#)

[ViSi-Genie Magic Spectrum](#)

[ViSi-Genie Magic How to Write a File](#)

Workshop -> File -> Samples -> ViSi Genie Magic (Picaso/Diablo):

FileAccess.4DGenie

MagicScope.4DGenie

MagicSpectrum.4DGenie

Proprietary Information

The information contained in this document is the property of 4D Systems Pty. Ltd. and may be the subject of patents pending or granted, and must not be copied or disclosed without prior written permission.

4D Systems endeavours to ensure that the information in this document is correct and fairly stated but does not accept liability for any error or omission. The development of 4D Systems products and services is continuous and published information may not be up to date. It is important to check the current position with 4D Systems.

All trademarks belong to their respective owners and are recognised and acknowledged.

Disclaimer of Warranties & Limitation of Liability

4D Systems makes no warranty, either expresses or implied with respect to any product, and specifically disclaims all other warranties, including, without limitation, warranties for merchantability, non-infringement and fitness for any particular purpose.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications.

In no event shall 4D Systems be liable to the buyer or to any third party for any indirect, incidental, special, consequential, punitive or exemplary damages (including without limitation lost profits, lost savings, or loss of business opportunity) arising out of or relating to any product or service provided or to be provided by 4D Systems, or the use or inability to use the same, even if 4D Systems has been advised of the possibility of such damages.

4D Systems products are not fault tolerant nor designed, manufactured or intended for use or resale as on line control equipment in hazardous environments requiring fail – safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines or weapons systems in which the failure of the product could lead directly to death, personal injury or severe physical or environmental damage ('High Risk Activities'). 4D Systems and its suppliers specifically disclaim any expressed or implied warranty of fitness for High Risk Activities.

Use of 4D Systems' products and devices in 'High Risk Activities' and in any other application is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless 4D Systems from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any 4D Systems intellectual property rights.