

Getting Started with Girder

This guide is based upon Girder 3.3.1

GETTING STARTED WITH GIRDER.....	1
INTRODUCTION.....	3
<i>What can Girder do ?</i>	3
INSTALLATION	4
SETTINGS	4
<i>General Tab</i>	5
<i>User Interface Tab</i>	6
<i>Plugins Tab</i>	7
<i>On Screen Display (OSD) Tab</i>	8
GIRDER CONCEPTS.....	9
COMMANDS.....	9
<i>Group</i>	9
<i>Multigroup</i>	9
EVENTS AND EVENTSTRINGS.....	10
<i>Event Types</i>	10
ACTIONS	11
MAIN WINDOW EXPLAINED	12
THE MENU BAR.....	13
<i>File Menu</i>	13
<i>Edit Menu</i>	13
<i>Commands Menu</i>	14
<i>Help Menu</i>	14
THE COMMAND TREE	15
THE EVENTS VIEW	16
ACTION TABS.....	17
STATUS BAR	18
ACTIONS EXPLAINED.....	19
<i>Window Actions</i>	19
<i>OS Actions</i>	21
<i>Command Actions</i>	23
<i>Girder Actions</i>	24
<i>Mouse Actions</i>	27
<i>Keyboard Actions</i>	28
<i>Plugins Actions</i>	29
<i>Volume actions</i>	30
SPECIAL FUNCTIONS.....	32
<i>States</i>	32
<i>Registers</i>	33
<i>Targetting</i>	<i>Error! Bookmark not defined.</i>

<i>The keyboard tab</i>	<i>Error! Bookmark not defined.</i>
<i>Target window</i>	35
<i>Event Processing</i>	37

Introduction

Girder is the most powerful and feature rich Windows automation tool available! Girder controls programs like WinDVD, WinAmp and Windows Media Player using infrared and wireless remote controls so you can sit back and enjoy your digital library from your couch. In Home Theater PC's, Girder controls Windows programs (media/DVD players) and hardware devices (sound cards, X10 and serial devices such as projectors, and receivers). Girder is the behind the scenes glue providing full control of your multimedia experience.



What can Girder do?

Girder can do most of the stuff that you can do with the mouse and the keyboard. Girder can type, move the mouse, click, start programs, shutdown the computer and then some. All within the limitation of Windows of course.

Girder works by linking a wide variety of input devices (handheld remotes, keyboards, etc.) to commands, which can control almost any Windows program or hardware device. With a hand held remote, you can have complete control over your computer. Extensive libraries of controls (plug-ins) are available for download on the Girder website. Girder works with many IR receivers including the USB-UIRT, TIRA, IRA-2, and IRTrans.

For more advanced users, Girder's event driven programming environment allows for sophisticated automation applications. Girder's scripting language, Lua, provides a powerful (yet simple) way to create programs that can perform advanced tasks including database access, file operations, XML processing, Windows Management and Instrumentation, and internet data retrieval. Girder receives events from an unlimited number sources including the Window's operating system, Window's applications (even across networks and the internet) and hardware devices. See the Girder website for a complete list of plugins.

Since 1998, Girder has been at the forefront of Windows automation and remote control. Girder has a large and active online community making it the most widely used and versatile automation tool of its kind. Girder will work on Windows 98, ME, NT, 2000, XP and 2003 platforms and requires minimal system resources. Some plugins are only supported under certain versions of Windows.

This manual is intended to teach the basics of using Girder. More advanced topics are better suited for the online forum. So, we are trying to keep it as simple as possible. If you do find things that are not clear please drop me us email.

Installation

Simply download the installer, “Girder331x.exe” and run the installer to create the desired directories and install the application. Girder comes with several plugins, which help Girder control your PC, and many more are available on the Girder website. We will discuss these later, after we get the basic system working.

Read the included “readme.txt” files for updated information before you begin.

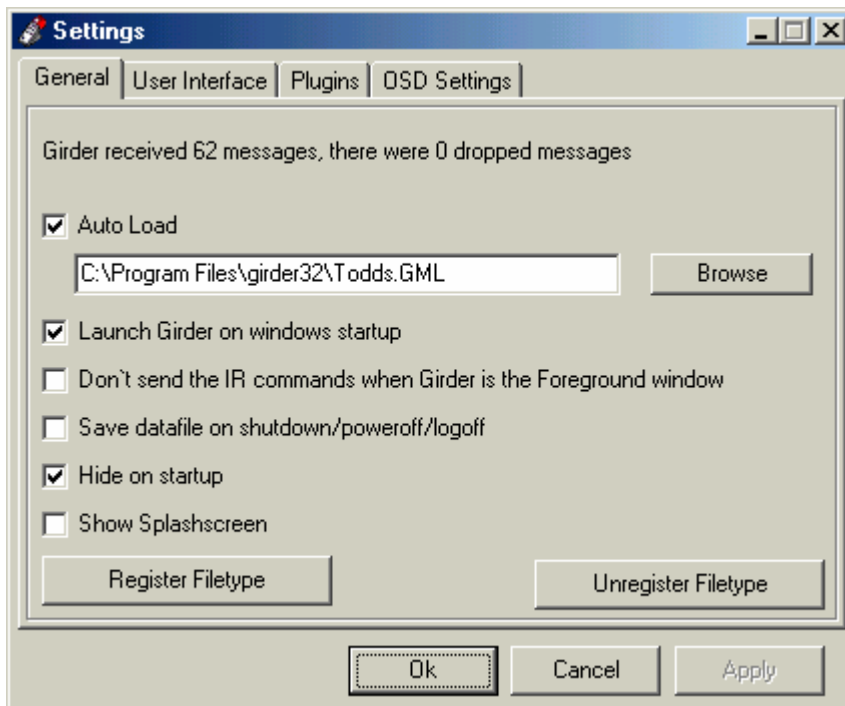
Start Girder, and select “File/Settings...” to configure your hardware and plugins. Then, you’re ready to go! You will notice that the little dot is green on your lower right hand side of Girder (indicating that the hardware is enabled and will send the events to girder if needed)

Settings

Here you can find the explanation of all those options in the settings dialog. This controls how Girder operates on your system. Click on the tabs to go to the section of your choice.

General Tab

General program control settings.



Auto load - Load the specified .gml file when Girder starts. This is very useful if you have Girder start when Windows loads.

Launch Girder on Windows startup - Starts Girder when Windows starts.

Don't send IR commands when Girder is the foreground - window Girder doesn't actually trigger any of the commands if it is the foreground window and this is checked, can be useful when working on the .gml file.

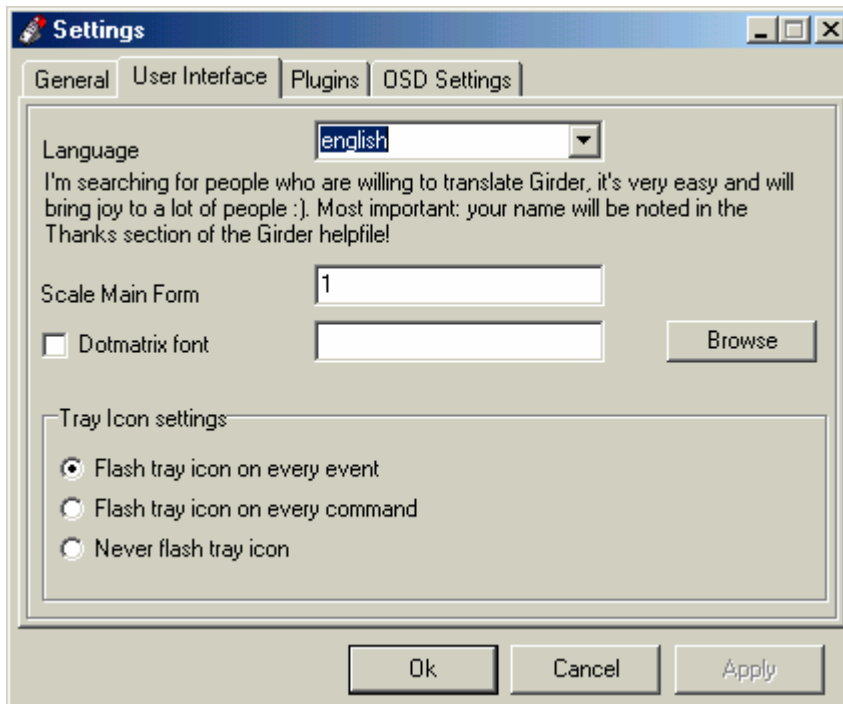
Save data file on shutdown - Saves the .gml file when Girder closes.

Hide on startup - This minimizes Girder to the tray bar when starting without display the main form.

Show Splash screen - Hey don't disable this, took me long enough to make it :-)

User Interface Tab

How Girder interfaces with the user, language and display preferences.



Language - this is the language that Girder uses, if your language isn't available you can translate Girder yourself and earn the gratitude of thousands of your fellow countrymen.

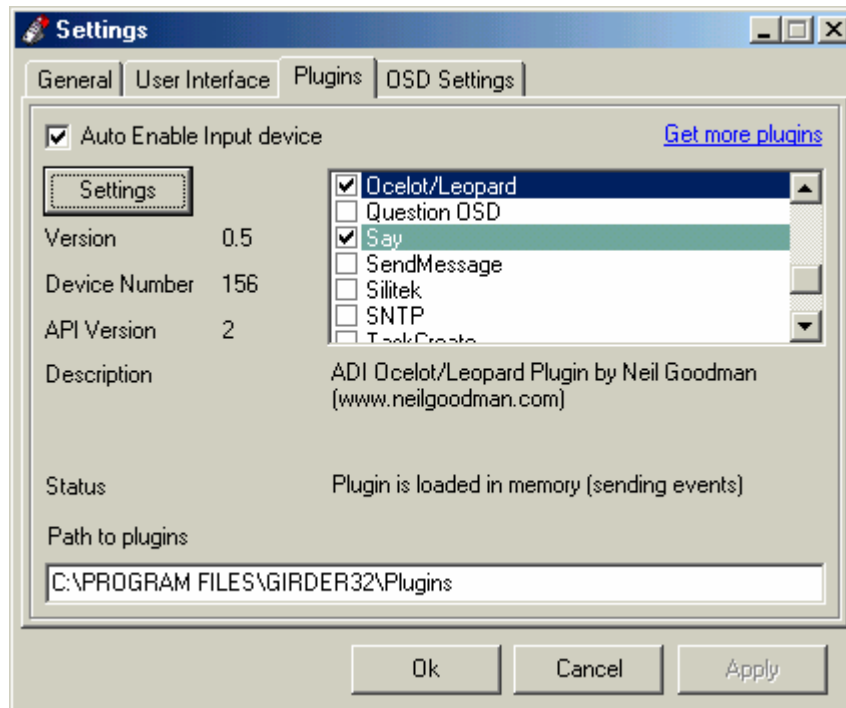
Scale Main form - If your window is setup to display big fonts you might have to adjust this to something like 1.25.

Dot-matrix font - The default dot-matrix font does not have any special characters in it, if you need them you can download special fonts from the girder download page.

Tray Icon Settings - Normally Girder flashes the tray icon on every incoming event, you can also check 'on every command' and then Girder will only flash if there is an action assigned to the event. Or, check 'never flash' to disable this feature.

Plugins Tab

All of the cool add-on utilities that really make Girder powerful and customized to you installation needs.



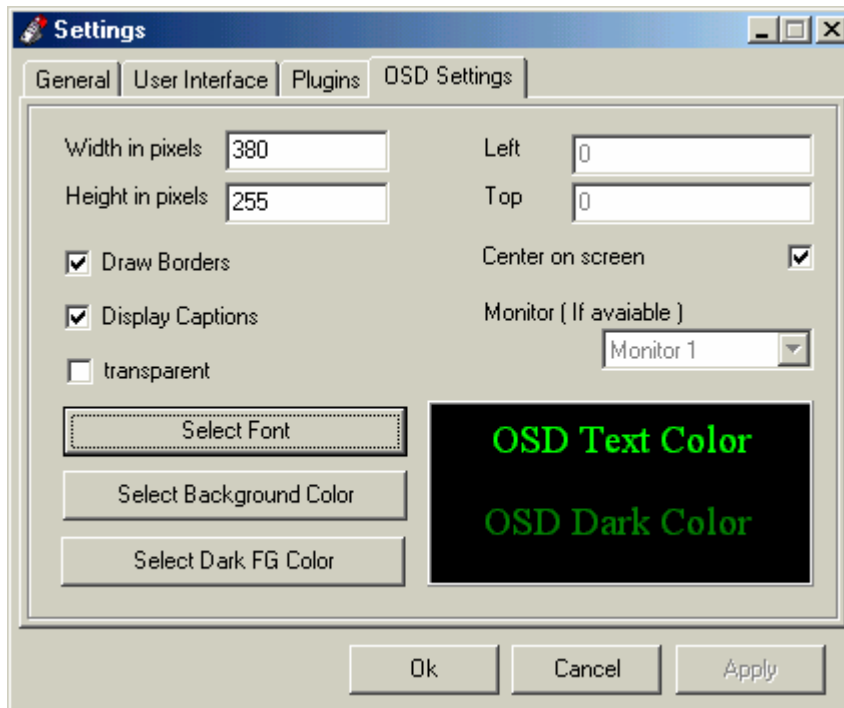
Auto Enable input device - This will enable the plugins when Girder starts, this is always a good idea :-)

Settings - After you select a plugin from the list, this button will be available (if the plugin is loaded! this means the checkmark is placed)

Plugin list - If you want to load a plugin, check it and press apply, Girder shows the status of the loaded plugin and the Settings button will available.

On Screen Display (OSD) Tab

Pop-up display settings. Used by the included OSD plugin to display messages on the screen.



Width, Height in pixels - This is the size of the on screen display.

Left, Top - This is the position of the OSD.

Center on screen - This centers the OSD on the screen.

Monitor - If you have multiple monitors (displays) connected to your computer you can specify where the OSD should appear.

Draw Borders - This enable/disables the border that Girder draws

Display Captions - With some actions Girder displays a caption in the OSD (for example the volume actions), you can disable this text here. Note: the captions can be customized by putting text in the "comment" field.

Transparent - Makes the On screen display transparent.

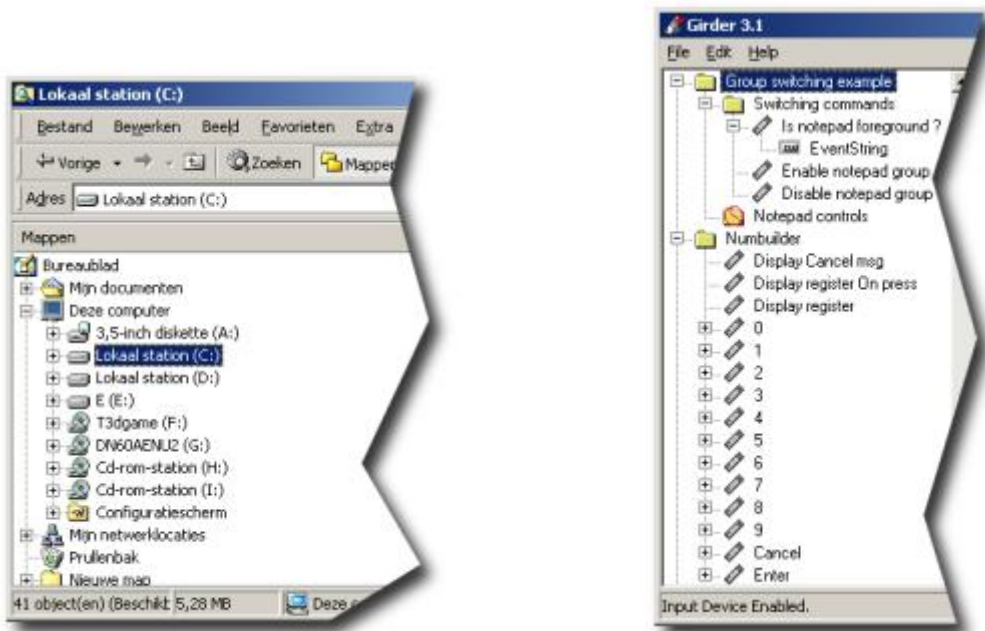
Select Font, Color - Selects the font and the color of the font.

Girder Concepts

Girder is a control software that manages a series of commands that are based around the “cause and effect” concept. A command is activated by a ‘trigger’ and does an ‘action’. These commands are stored and managed in a tree like structure. See below.

Commands

Everything that Girder does is encapsulated in a command. That means a command can run Netscape or Close windows, just to name 2 things. These commands are stored in a so-called tree. You have seen these things before, compare the Windows Explorer with the Girder window.



Windows Explorer tree

Girder Tree

The “commands” must always be stored in a Group, or a Multigroup. If you want to add a command, you can select "Add command" from the edit menu. The difference between a group and a multigroup is very important.

Group

A Group is a set of commands used to control a common device or application, for example, a set of commands to control WinAmp. A command that is placed in a group is triggered by a single event. So, a group has single commands, each with a single trigger event.

Multigroup

A multigroup is a set of commands, all activated by a single event “trigger”, kind of like a macro! So, a single trigger can activate several commands like, Open app, select song, increase volume, and then play!

For now use a "group", this is a yellow folder. See example below.



Group – “DVD Station 1.8”
Multi-Group – “Power switch DVD”
Commands – “Stop, Play”

So, you need to decide what you want to control, what commands you’ll need, then organize them by group.

Events and Eventstrings

Okay now we have commands that are stored in a tree, but how on earth are they triggered?

Girder has the possibility to receive events from a large number of devices ranging from your keyboard, your IR remote, or even the internet. These events, for example a key press, all have eventstrings attached. This eventstring is the unique identifier of the event in the Girder universe.

But how does Girder know what command to trigger?

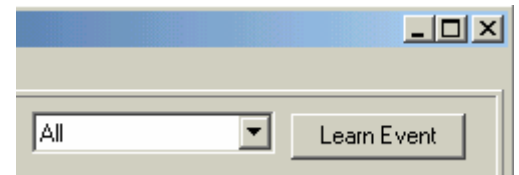
We tell Girder what command to trigger by clicking on that command in the tree and pressing the "Learn Event" button. Now Girder goes into learning mode, the next event that comes in will be assigned to the command. Girder now knows exactly what event should trigger this command and the next time that event happens Girder will trigger the command.

Event Types

Girder comes with several input plugins that generate ‘Events’ or ‘Eventstrings’. They include the Keyboard and IR plugins, and the special ‘Girder Event’ plugin. They may or may not show up in the ‘Type’ pull down next to the ‘Learn Event’ button. See note below. These plugins are what connects the ‘Commands’ or actions to the input device.

Important Note!

The drop down box next to the "learn event" button makes Girder learn from a specified plugin. **However, not all plugins support this so usually you'll only find "All" and "Girder Event" there.** For most plugins you can leave this on "All".



"All" simply means that after pressing learn Girder will wait for the next event to come in. Sometimes you want to assign a special event. This can be done by selecting the appropriate plugin from the dropdown box. For example the "Girder Event" option pops up a dialog box giving you a few options. So if you get this box and don't want it, simply close it and select "All" from the dropdown box.

Wait a moment!. After I pressed "learn event" a new thing appeared on the tree, it is called "Eventstring"! What is up with this?

This is the way that Girder attaches the event (again for example the key press) to the action; also you can have as many events attached to the action as you like!

Conclusion!

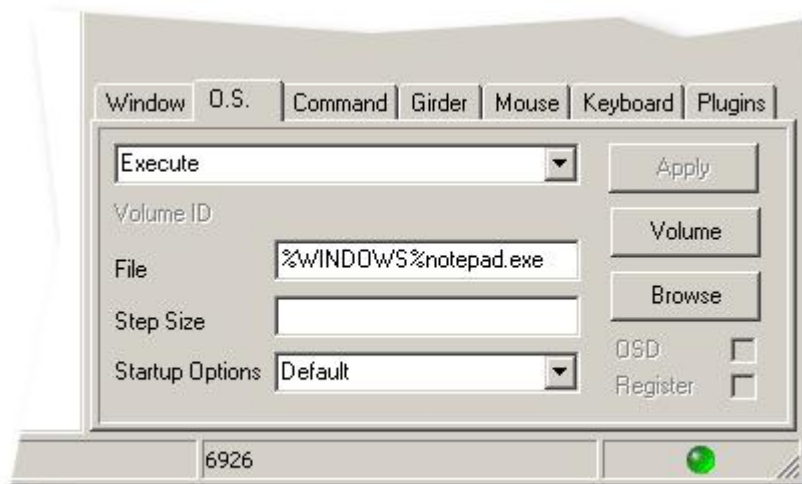
Commands execute 'Actions' and 'Event Strings' trigger these Commands.

Great! I now have a command called "New" with an eventstring attached to it in the tree but it does nothing, how do I make it do something?

Let's start really basic. We want to open notepad. First thing I should point out is the "action tabs"; this is the place where you actually tell Girder what the specific command should do.

Actions

Actions are what happens when a command is triggered.

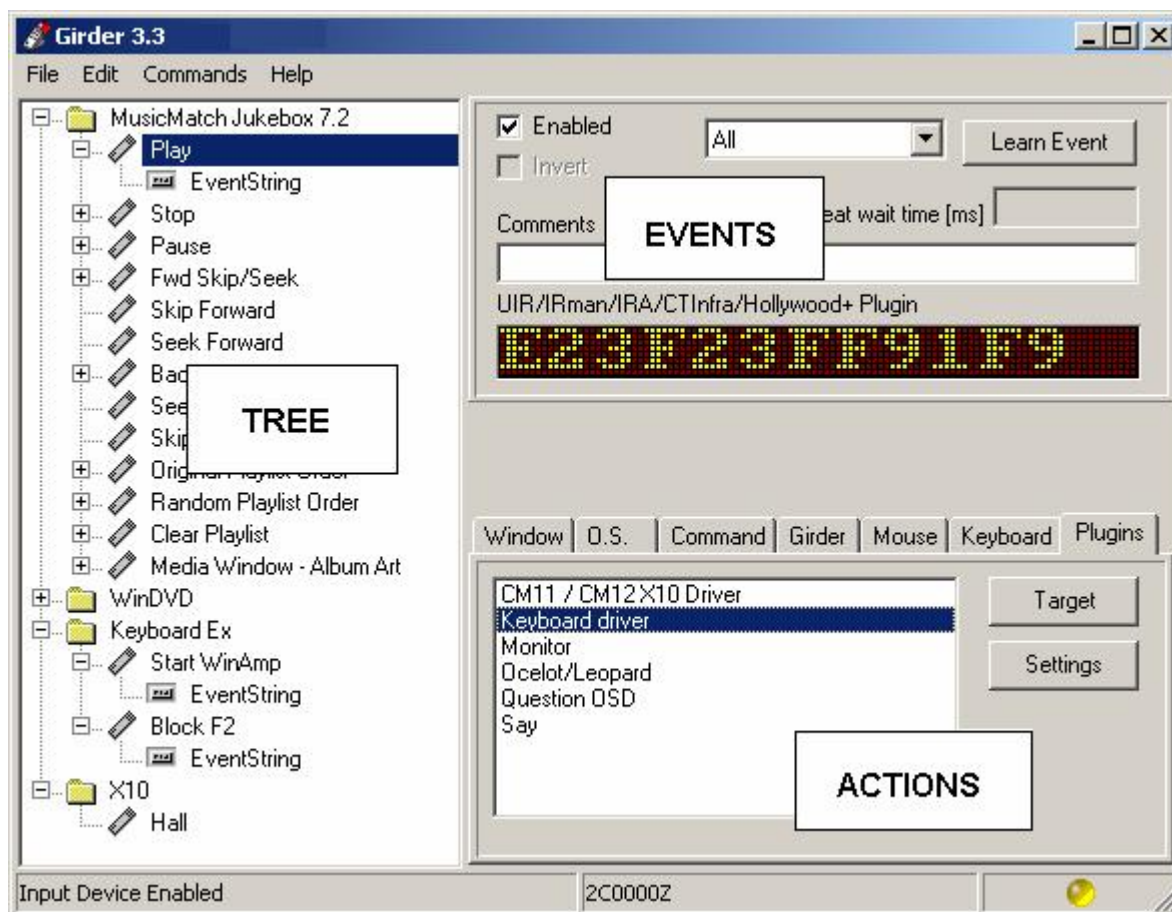


Go to the "OS" tab as seen above and select "Execute" from the pull down list. The text box "file" becomes active (amongst others, but we only care about this right now). In that box enter the path and the command to execute, such as "C:\WINNT\system32\notepad.exe". Leave the quotes out, and type it as you see it in the image. Or, you may hit 'Browse' to find the correct path to your desired application. Finally, press 'Apply' to save the changes. This apply pressing is of utmost importance, because it tells Girder that it should store your input. To test if you actually did this correctly press "F5", and the Notepad application should now magically start. Also pressing the button on your remote that you assigned (learn event) to it will also start notepad way from you couch!

Cool! But what the heck should I do with notepad?

Well nothing but remember this is a getting started document! More on Actions later!

Main Window Explained



File Menu – pull down menus

Tree – command storage and management area

Events – area used to set trigger for input events

Actions – area to set what the command actually does

Status Bar – program status, 'eventstring' code area, and LED display

The Menu Bar



File Menu

New - Removes all actions from the tree view

Open... - Open a new .gml file

Save - Saves the current .gml file

Save as... - Saves the current .gml file under another name

Lock – password the file to prevent unauthorized changes

Import Group... - Imports a group into the current .gml file, you can find these at: Group Exports

Export Group... - Exports the selected group into a .gml file

Settings... - Opens the settings window, to configure Girder

Enable/Disable input devices - You need to explicitly start the input devices, you can do that with this menu.

Close Window – close Girder window and minimize to task bar, and leave Girder active.

Exit - Exits Girder completely.

Edit Menu

Copy - Copy the currently selected command or group.

Paste - Pastes the previously copied command or group.

Delete - Deletes the currently selected command or group.

Rename - Renames the currently selected command or group.

Duplicate – create copy of selected command.

Add command - Adds a new command to the tree at the current position

Add multigroup - Adds a new multigroup (sometimes also called a multicommand) to the tree.

Add group - Adds a group to the tree.

Add top level group – Create new group to top or 'root' of tree

Add eventstring – to selected command

Collapse Node – hide subtasks

Expand Node – show subtasks

Collapse All – show only groups

Collapse all eventstrings - hide all eventstrings

Commands Menu

Test Command - Executes the currently selected command.

Clear Links – Clear all ‘go to’ links and group Disable/enable

Set OSD Timeout – set time for OSD to remain on screen

Learn Event – create trigger or eventstring for selected command

State Settings – set initial command state (see other documentation)

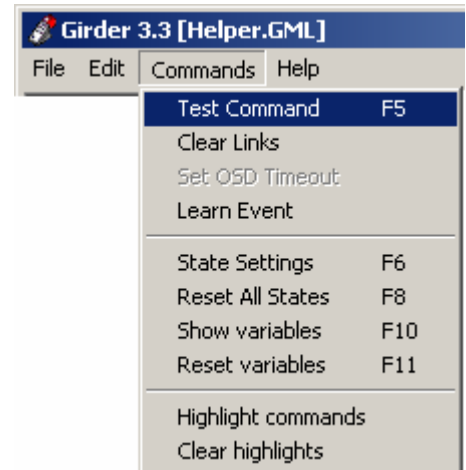
Reset all States - Resets the states to the "Begin State"

Show Variables – List Girder variables (see other documentation)

Reset Variables – Resets the values of variables

Highlight commands - ???

Clear highlights - ???



Help Menu

Online Help – Link to Girder web page for online help documents

Discussion board – Link to online forum, VERY HELPFUL!

About – about screen with version and registration info

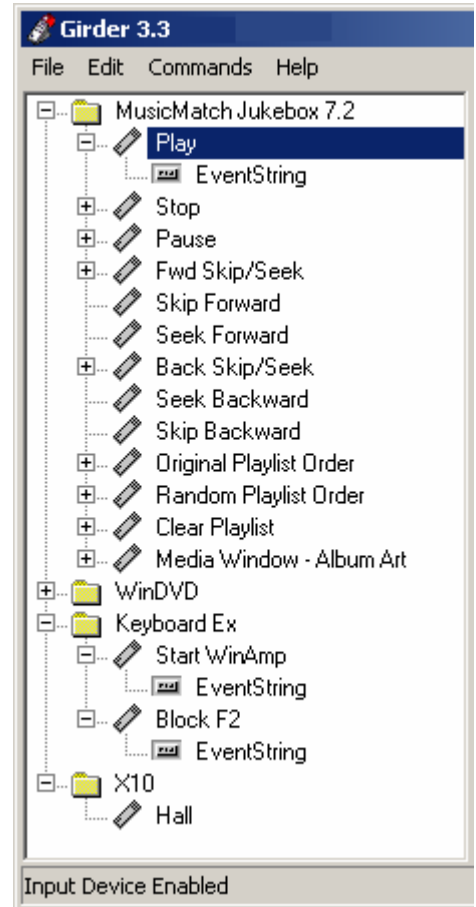
The Command Tree

This is where you make Girder do its magic!

When Girder receives an event it starts checking all actions and multigroups in a linear fashion, thus it starts at the top and stops at the bottom. If a command is disabled it won't be executed, if a group is disabled Girder will not search in this group for a command.

If you want Girder do to more things on 1 event you can assign that event to multiple commands or use a multigroup, aka multicommand. A multigroup doesn't execute anything; it can only get an event string assigned to it. You can now add commands to the multicommand. Thus when a multicommand has an eventstring assigned and that event happens, it will execute all commands that where added to the multicommand.

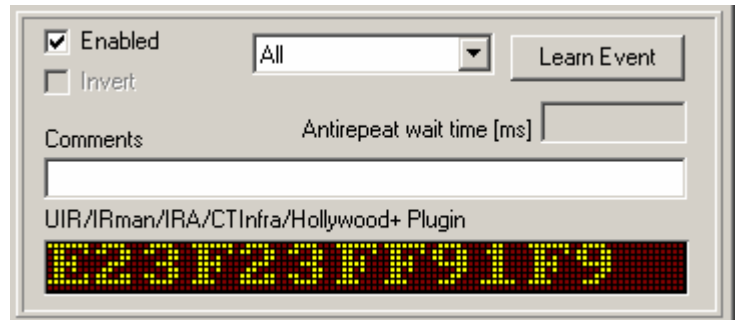
You can make all kinds of jumps, conditional (for example; if window exists) or unconditional (for example; go to).



The Events View

Enabled - This enables or disables a command or a whole group. This allows you to selectively use commands that you have defined.

Invert - Girder executes a command when the event string matches the learnt event string. When you check the invert checkbox Girder will execute this command whenever then event string does not match.



The screenshot shows a dialog box for configuring an event. It has a 'Learn Event' button, a dropdown menu set to 'All', and checkboxes for 'Enabled' (checked) and 'Invert'. Below these is a text input field for 'Comments' and another for 'Antirepeat wait time [ms]'. At the bottom, it displays the plugin name 'UIR/IRman/IRA/CTInfra/Hollywood+ Plugin' and an LED display bar showing the hex event string 'E23F23FF91F9'.

Type (pull down) - Use to learn the event from a specific plugin only. Leave “All” for most commands.

Learn Event - Girder works event based, this means that Girder reacts to an outside event, this can be a key press or an IR signal that was received or something completely different. To teach Girder on what event which command should be executed you first select the command you wish to execute and then click on this button. Now Girder waits for the event that should trigger the command. At this point you should generate the event, this could be pressing a key on the keyboard, sending IR from the remote control, or sending an X10 command. It can be virtually anything.

Anti repeat wait time - Sometimes events repeat very fast, but you only want Girder to react once a second, which is where anti-repeat comes in.

Comments - User notes area, great for documenting your commands.

Plugin Status (text display) - Displays which plugin learned the event. Will list ‘Girder event’, Ocelot, or any plugin name. This example shows the built in UIR/IRman plugin received the ‘Learn’.

LED Display Bar - Displays the ‘Event string’ (text or code) that was assigned to the command or the IR code or hex key code from the plugin. Example shows the IR code received from the UIR/IRMan plugin “E23F23FF91F9”

Action Tabs

In this part of the window you tell Girder what you want done. The actions are divided semi-logical in 7 categories.

Window - Here you'll find commands that do things to windows, like minimize hide, show, close, move etc.

O.S. - On this tab the Operating related actions can be found like shutdown, reboot, monitor on/off.

Command - Girder has the ability to capture certain messages that are commonly sent to a window. Also it can capture mouse clicks, more specifically; it detects the mouse click and the position.

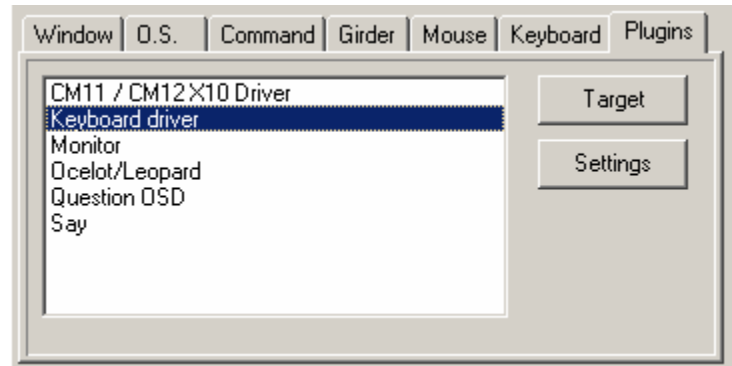
Girder - Here you'll find actions that do things with Girder, for example disable and enable groups or input devices.

Mouse - Action like moving the mouse

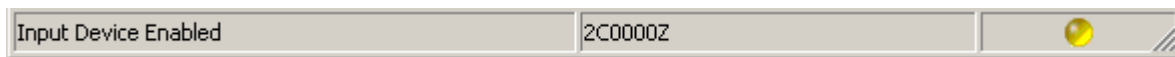
Keyboard - Keyboard simulation

Plugins - Here all the Action Plugins appear these you can download and can be written by anyone.






These tab sections will be explained more in detail later.



Status Bar



Here the status of Girder can be found. The actions also return some information on if they completed their action correctly or what went wrong. If some event happens in the hardware plugins, Girder will give a visual feedback with a color LED and the eventstring. The center box is the eventstring. The right hand side gives the current condition via LED as listed below.

-  All input devices are disabled.
-  All is okay!
-  An event is received
-  Girder is in standby mode
-  An error occurred.

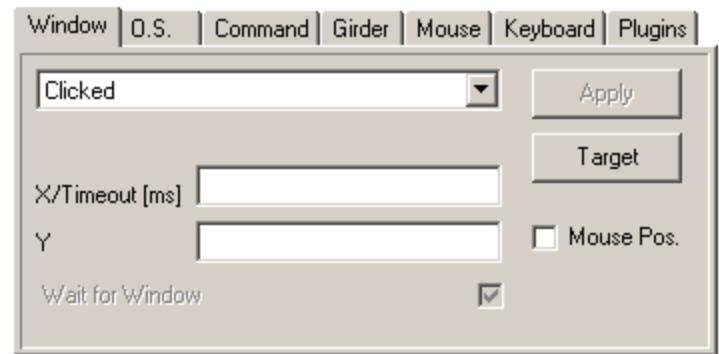
Actions Explained

Actions are what the commands actually do. They are grouped into several categories as indicated by the tabs located in the Action view area.

Most actions require that you select the desired action, click the "Apply" button to set the action and click the "Target" button to tell Girder what window or control that action will be performed on. **If you fail to click the "Apply" button your changes will not be saved.** If you do not specify which window Girder should target, Girder will perform the action on the window or application in the foreground when the action is invoked.

Window Actions

The Window tab consists of an Action Drop Down List (combo box) which selects which type of action will be performed, and the following fields and buttons. "X/Timeout [ms]" and "Y" text boxes, and "Mouse Position" and "Wait for Window" check boxes along with "Apply" and "Target" buttons.



The Action Drop Down List includes the following Actions:

Clicked - Sends a left mouse button click to the control specified in the target options. This action can be used to simulate clicking a button or control.

Double Clicked - Sends a left mouse button double-click to the control specified in the target options. This action can be used to simulate double-clicking a button or control.

Show - Shows the previously hidden window or control specified in the target options. NOTE: in order to restore a previously hidden object, the "Match invisible Tasks" option must be selected (checked) in Girder's "Target Selector"

Hide - Hides the window or control specified in the target options.

Close - Closes the window specified in the target options. If a target is not specified Girder closes the foreground window (window that has focus at the time the action is invoked).

Set focus - Sets the focus to the window specified in the target options.

Wait - Forces Girder to wait for the time specified in the "X/Timeout [ms]" field before moving on to the next command. The wait action is typically used in Commands within MultiGroups to force Girder to halt processing for a specified period of time. Once the timeout has elapsed, Girder will continue processing with the next Command in the MultiGroup.

"Wait for Window" Checkbox - Forces Girder to stop processing and wait for the window specified in the target options until the timeout set in the "X/Timeout [ms]" field has elapsed. The Wait for Window action ensures that Girder will not continue with the next Command in the MultiGroup until the application window exists OR the timeout specified in "X/Timeout [ms]" field has elapsed, whichever occurs first. This action can be useful in cases where a MultiGroup includes a Command that starts an application and is followed by additional Commands that manipulate a control within this application such as clicking a button, filling in a text field, etc.

NOTE: If 0 (zero) is entered in the "X/Timeout [ms]", Girder will timeout immediately and continue on to the next command regardless of whether or not it sees the targeted window.

Maximize - maximizes the window specified in the target options.

Minimize - minimizes the window specified in the target options.

Absolute Mouse Move - Moves the mouse to the given coordinates referenced from X=0, Y=0 (upper left corner of Windows desktop).

Move window - Moves the window specified in the target options to the absolute X (left to right) and Y (up and down) coordinates entered in "X/Timeout [ms]" and "Y" fields.

Resize window - Resizes the window specified in the target options to the X and Y dimensions entered in "X/Timeout [ms]" and "Y" fields.

Center and Resize Window - Resizes the window specified in the target options to the X and Y dimensions entered in "X/Timeout [ms]" and "Y" fields and centers it on the display.

Restore - Restores the window specified in the target options window (only functions if the window is currently maximised).

Move window relative - Moves the window specified in the target options window relative to its previous position (e.g. if the "Y" field is -100 the window will be moved up the screen by 100 pixels).

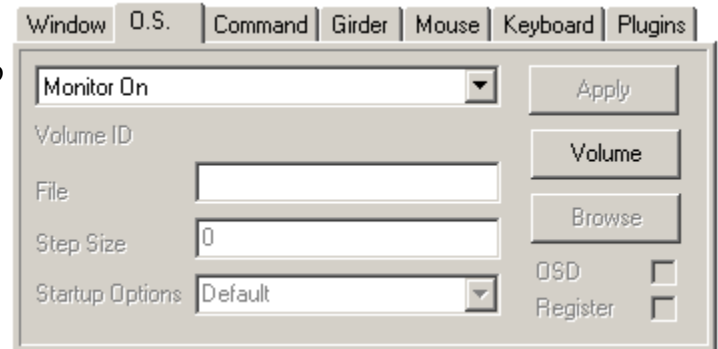
Get Caption - Retrieve window caption from the window specified in the target options window (for example the WinAMP window caption contains the current song name).

Relative Mouse move - Moves mouse cursor relative to the window specified in the target options window (Ensure that only one window is targeted).

OS Actions

The O.S. tab consists of an "Action Drop Down List", "File" and "Step Size" text boxes, a "Startup Options" Drop Down List and an "OSD" check box along with "Apply", "Volume" and "Browse" buttons.

The O.S. Action Tab's controls may serve more than one purpose depending on the particular action being used.



The O.S. tab includes the following Actions:

Monitor On - Used to wake up the monitor after running a Command, and includes a Girder "Monitor Off" action.

Monitor Off - Puts the monitor into power save mode.

NOTE: Be sure to create a Girder Command that includes a "Monitor On" action as well. Otherwise you'll need to navigate windows in the dark or reboot to turn the monitor back on (keyboard/mouse will still wake the monitor up as normal).

Screen Saver - Launches the normal Windows screen saver.

NOTE: You must select a screen saver from the Windows' Display Properties for this action to function properly. If no screen saver is configured this action will have no effect.

Play Wav - Plays the .wav file indicated in the "File" text box's path and file name.

Shutdown - Shuts down the computer.

Power off - Shuts down and powers off the computer (you must have a compatible computer for it to completely power off - most newer computers will do this).

Logoff - Log off the current user. Since Girder cannot (yet) run as a service girder will also be closed and you must use keyboard/mouse at the logon screen.

Reboot - Shuts down and restarts the computer.

Volume Change - Type the amount to increase/decrease the volume level each time this action is executed into the "Step Size" text box. *See volume actions for more information.*

Volume Right Change - Type the amount to increase/decrease the right channel volume level each time this action is executed into the "Step Size" text box. *See volume actions for more information.*

Volume Left Change - Type the amount to increase/decrease the left channel volume level each time this action is executed into the "Step Size" text box. *See volume actions for more information.*

Volume Balance Change - Type the amount to move balance to left or right (left is -ve, right is +ve) each time this action is executed into the "Step Size" text box. *See volume actions for more information.*

Volume Mute Toggle - Toggle Mute on/off. 0 = Toggle mute. *See volume actions for more information.*
> 0 = Unmute.
< 0 = Mute.

Execute - This action executes a file specified by the "file" text box.
Type any parameters in the "Step Size" text box and specify the startup mode in the "Startup Options" drop down box.

Talking Clock - As it says, this action simply reads out loud the windows time. You need the sound files for this to work, available on the Girder website, Downloads section. Unzip these into the sound subdirectory of Girder (you may need to create the sound subdirectory as it probably won't exist).
NOTE: You can create you own "Talking Clock" by using the "Time2Reg" and the "Say" plugins (both included with Girder).

Volume Display - Display windows current volume level using Girder's OSD function.
See volume actions for more information.

Mute Display - Displays Windows current mute status using Girder's OSD function.
See volume actions for more information.

Task Switcher - This action does the same as pressing the "alt-tab" key combination.

Change Resolution - Allows you to change the windows screen resolution.
Specify resolution by entering the resolution into the "file" text box: WxHxC@Hz
w = width in pixels
h = height in pixels
C = color depth in bits (8 = 256 colors, 16 = 16bits etc)
Hz = refresh For example: 800x600x8@75 is 800 by 600 pixels at 256 colors at 75 hertz refresh.

Eject Media - Eject removable disk or tape media from drive specified in the "File" text field.

Load Media - Load removable disk or tape media into drive specified in the "File" text field.

Set Volume - Type the volume level into the "Step Size" text box.
See volume actions for more information.

Power Management - Allows Girder to trigger Hibernate and Standby modes. Specify mode in the dropdown box:

- hibernate polite - asks running applications for permission before entering hibernation.
- hibernate forced - Goes straight to hibernation without notifying running applications.
- ram-alive polite - asks running applications for permission before entering standby.
- ram-alive forced - Goes straight to standby without notifying running applications.

Get resolution - Stores the current screen resolution in a string variable specified in the "file" textbox.

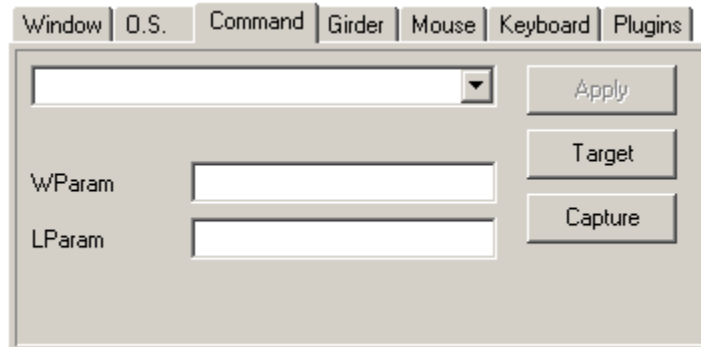
Get Volume - Retrieves the volume level for the specified line and stores it in the variable specified in the "Step Size" box. *See volume actions for more information.*

Command Actions

The Command tab consists of an Action Drop Down List; "WParam" and "LParam" text boxes; "Apply", "Target" and "Capture" buttons.

As with all other actions, clicking the "Apply" button saves the action selection and optional values.

Use the "Capture" button to find out what parameters/messages to use by capturing real-time events - e.g. left click on a button somewhere: you will see the corresponding message(s) appear in the list.



Note: To send other windows messages you can use the SendMessage plugin (included with girder distribution) which also allows saving of the message result to a register.

The Command tab includes the following Actions:

Command - Sends a WM_COMMAND message to a window specified by the "Target" button.

SysCommand - Sends a WM_SYSCOMMAND message to a window specified by the "Target" button.

Left Click - Sends a Left Click to a window specified by the "Target" button.

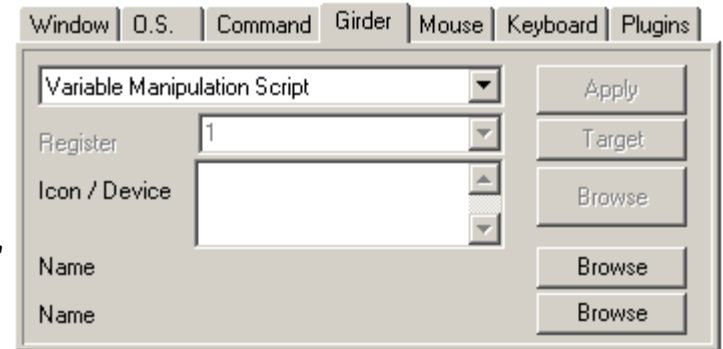
Right Click - Sends a Right Click to a window specified by the "Target" button.

Left Double Click - Sends a Left Double Click to a window specified by the "Target" button.

Right Double Click - Sends a Right Double Click to a window specified by the "Target" button.

Girder Actions

The Girder tab consists of an Action Drop Down List; a Register Drop Down List; an "Icon / Device" text box; two "Name" labels; "Apply", "Target" and 3 "Browse" buttons.



- 1) The top browse button launches the script editor when "Variable Manipulation Script" action is selected. It is also used to specify an icon file / Hardware Plugin Device number.
- 2) The second and third browse buttons are used to specify commands for the first/second "name" boxes.

* Actions in **red** are superseded by the Variable Manipulation script and are only there for compatibility. Therefore, these actions should not be used in new girder scripts.

The Girder tab includes the following Actions:

Group Enable - Enable a group by remote control. Use the second "Browse" button to specify group.

Group Disable - Disable a group by remote control. Use the second "Browse" button to specify group.

Set On-Tray icon - (Set [XX]-Tray icon) - Sets the icon that Girder displays in the Tray bar. Use the first "Browse" button to search for an icon or type a filename in the "Icon / Device" text box.

Set Off-Tray icon - See "Set On-Tray icon" above.

Set Active-Tray icon - See "Set On-Tray icon" above.

Set Standby-Tray icon - See "Set On-Tray icon" above.

Minimize Girder - Minimizes Girder. This command gets executed even when Girder is in the foreground.

Girder Standby Toggle - Usually Girder searches through all the commands to find the pressed button. When Girder is in standby mode it only checks if the pressed button is the Girder standby toggle button, so Girder doesn't react to anything else. Also the Led is orange.

Simple OSD - This displays Girder's simple built-in OSD with the text specified by the "Icon/Device" text box. The timeout can be specified by right-clicking on the command and selecting 'Set OSD Timeout' (note: timeout is per command - i.e. each command can have a different timeout set). For a more advanced OSD you can use Mark Fietchner's Popup OSD plugin, or for an OSD menu you can use Matt Wire's OSDMenu plugin.

Go to - This action jumps to another command in the tree (specified by the first "name" box), and then returns to continue executing any further commands.

Enable Device - With this function you can selectively enable plugins: you must enter the plugin number in the "Icon / Device" box. This number can be found in the settings dialog on the hardware tab.

Disable Device - With this function you can selectively disable plugins: you must enter the plugin number in the "Icon / Device" box. This number can be found in the settings dialog on the hardware tab.

Window Exists - With this function you can make a decision based on whether a window exists or not. Specify the target window using the "Target" button. Then specify the command to execute if the window exists in the first "name" box; specify the command to execute if the window doesn't exist in the second "name" box.

Stop Processing - Sometimes you need Girder to stop any further processing of commands (to avoid loops etc) use this command at the place where you want girder to stop.

Reset State - Resets states for the group specified in the first "name" box. See states for more information about states.

Is Checked - If the checkbox window specified by the "Target" button is checked then the command specified by the first "name" box is executed. Otherwise the command specified by the second "name" box is executed.

Is Foreground - If the window specified by the "Target" button is the current foreground window then the command specified by the first "name" box is executed. Otherwise the command specified by the second "name" box is executed.

Put Val in Reg. - Set the register specified in the "Register Drop Down List" to the value in "Icon / Device".

Add Val to Reg. - Add the value in "Icon / Device" to the register specified in the "Register Drop Down Box".

Mult Reg. with Val. - Multiply the value in register specified in the "Register Drop Down List" with the value in "Icon / Device".

Div Reg. through Val. - Divide the value in register specified in the "Register Drop Down List" by the value in "Icon / Device".

Add Reg. to Reg. - Add the value in register specified in the "Register Drop Down List" to the register specified in "Icon / Device". Result is stored in register specified in the "Register Drop Down List".

Sub Reg. from Reg. - Subtract the value in register specified in the "Register Drop Down List" from the register specified in "Icon / Device". Result is stored in register specified in the "Register Drop Down List".

Mult Reg. with Reg. - Multiply the value in register specified in the "Register Drop Down List" with the register specified in "Icon / Device". Result is stored in register specified in the "Register Drop Down List".

Div Reg. Through Reg. - Divide the value in register specified in the "Register Drop Down List" by the register specified in "Icon / Device". Result is stored in register specified in the "Register Drop Down List".

Check Reg. - Select a register in the "Register Drop Down List". Enter the condition you want to test in "Icon / Device", which should be one of the following operators plus an integer value or a register: e.g. "=="5"; ">=[reg3]". Operator Meaning:

- == Equal to
- >= Greater than or equal to
- > Greater than
- <= Less than or equal to
- < Less than

If the condition is true then the command specified by the first "name" box is executed. Otherwise the command specified by the second "name" box is executed.

Save Reg. - Specify a file in "Icon / Device" to save the register values in.

Load Reg. - Specify a file in "Icon / Device" to load the register values from.

Set Text Reg. - Set the text register specified in the "Register Drop Down List" to the value in "Icon / Device".

File Exists - If the file specified in "Icon / Device" exists then the command specified by the first "name" box is executed. Otherwise the command specified by the second "name" box is executed.

Explicit Return - The "explicit return" is like the 'exit' from Delphi or the 'return' from C. It exits the current procedure, even if there are more commands after it.

Get Tick Count - This action retrieves the number of milliseconds that have elapsed since the system was started, and stores it in the register specified by the register drop down box.

Reset Tick Count - This action resets the system 'tick count' which can be retrieved using "Get Tick Count" above.

Variable Manipulation Script - Powerful variable manipulation system and scripting language (Powered by LUA). See Girder Lua Scripting quick guide and the LUA Reference Manual.

Variable To Clipboard - Stores the variable specified in "Icon/Device" on the windows clipboard.

Clipboard To Variable - Stores the windows clipboard contents in the variable specified in "Icon/Device".

Reset all states - Resets all states to their initial values. See states for more information about states.

Mouse Actions

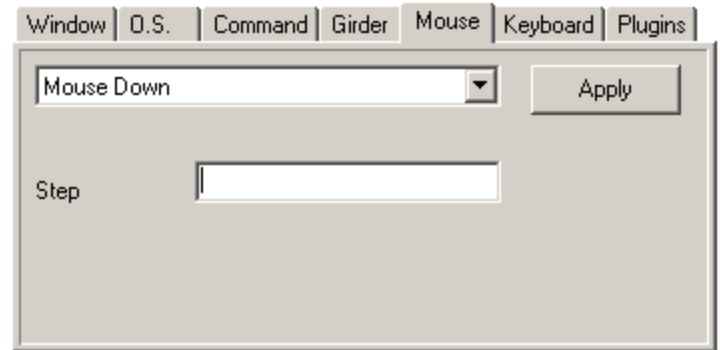
The Mouse tab consists of an Action Drop Down List; a "Step" text box; and an "Apply" button. As with all other actions, Clicking the "Apply" button saves the action selection and optional values.

Enter a value in the "Step" text box to specify the amount by which the mouse should be moved.

The Mouse tab includes the following Actions:

- Mouse Up -**
- Mouse Down -**
- Mouse Left -**
- Mouse Right -**
- Mouse Left Click -**
- Mouse Left Double Click -**
- Mouse Right Click -**
- Mouse Right Double Click -**
- Mouse Middle Click -**
- Mouse Middle Double Click -**
- Mouse Up Left -**
- Mouse Up Right -**
- Mouse Down Left -**
- Mouse Down Right -**
- Mouse Wheel Up -**
- Mouse Wheel Down -**

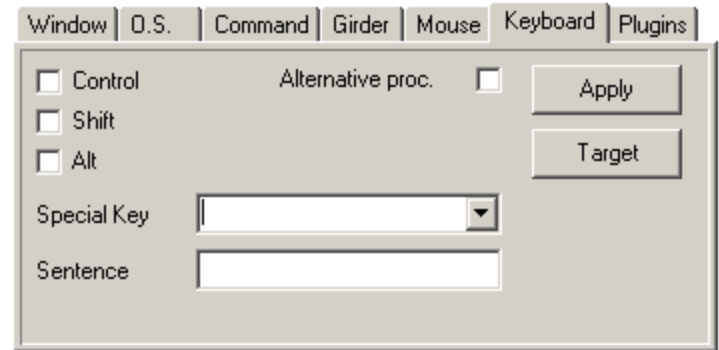
These functions emulate a mouse and are self-explainable.



Keyboard Actions

The Keyboard tab consists of a "Sentence" text box; "Special Key" Drop Down Box; "Apply" and "Target" buttons; "Control", "Shift", "Alt" and "Alternative proc." checkboxes.

If you do not specify which window Girder should target, Girder will perform the action on the window or application in the foreground when the action is invoked.



The Keyboard tab includes the following items:

Control, Shift, Alt checkboxes - Specifies whether control, shift, or alt keys are held down for the key press action.

Alternative proc. checkbox - Try this if the key press action doesn't work.

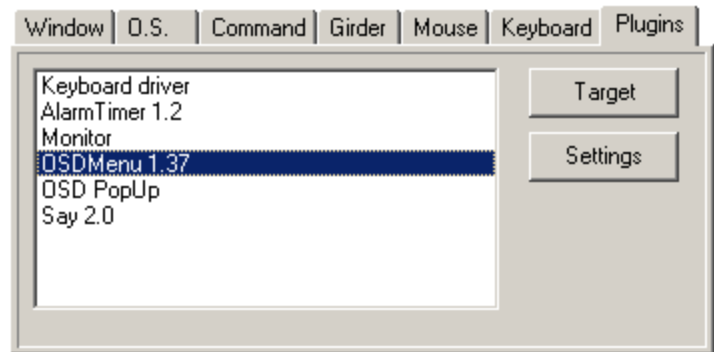
Special Key Drop Down Box - Select a special key from this box to send.

Sentence Text Box - Type keys into this box that you want to send.

Plugins Actions

The Plugins tab consists of a List Box; "Target" and "Settings" buttons.

Click the "Settings" button to configure the selected action plugin. If you create a valid configuration with that action plugin the action selection will be saved.



If applicable to the selected action plugin: click the "Target" button to tell Girder what window or control that action will be performed on. If you fail to click the "Apply" button your changes will not be saved. If you do not specify which window Girder should target, Girder will perform the action on the window or application in the foreground when the action is invoked.

See Girder's Plugins download page to find all the currently available action plugins.

Volume actions

The following Volume actions are currently available:

Volume Change

Volume Right Change

Volume Left Change

Volume Balance Change

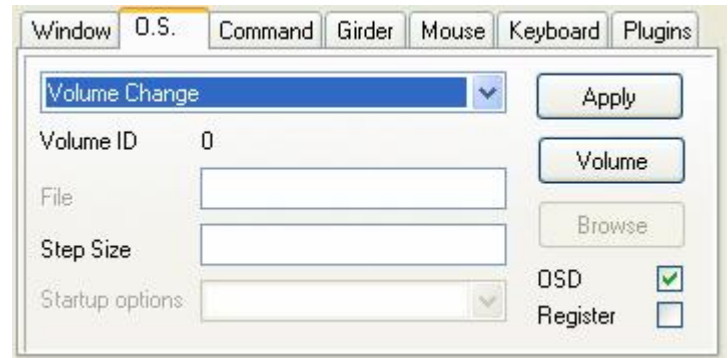
Volume Mute Toggle

Volume Display

Mute Display

Set Volume

Get Volume



Specifying Volume increment

Specify the amount you want to change volume by in the "Step Size" box. This should be in the range 0-65535(min-max). Use a '-' sign to decrease level.

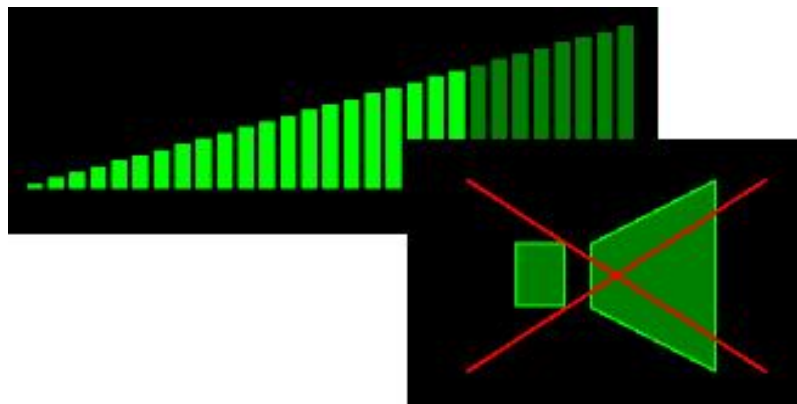
Example

- Create a command to increase the volume
- Enter a value of 4000 to 6000 in the step size box
- Assign or 'Learn' a trigger

Then, simply activate the trigger to raise the volume!

Built-in OSD

Girder has a built-in volume OSD which can be enabled by checking the 'OSD' box on the OS Action Tab.



This OSD can be used with all the volume actions, though the OSD will vary depending on the action selected (e.g. a mute action displays a muted/unmuted speaker; a volume action displays an incremental bar).

Selecting the correct Volume "Line"

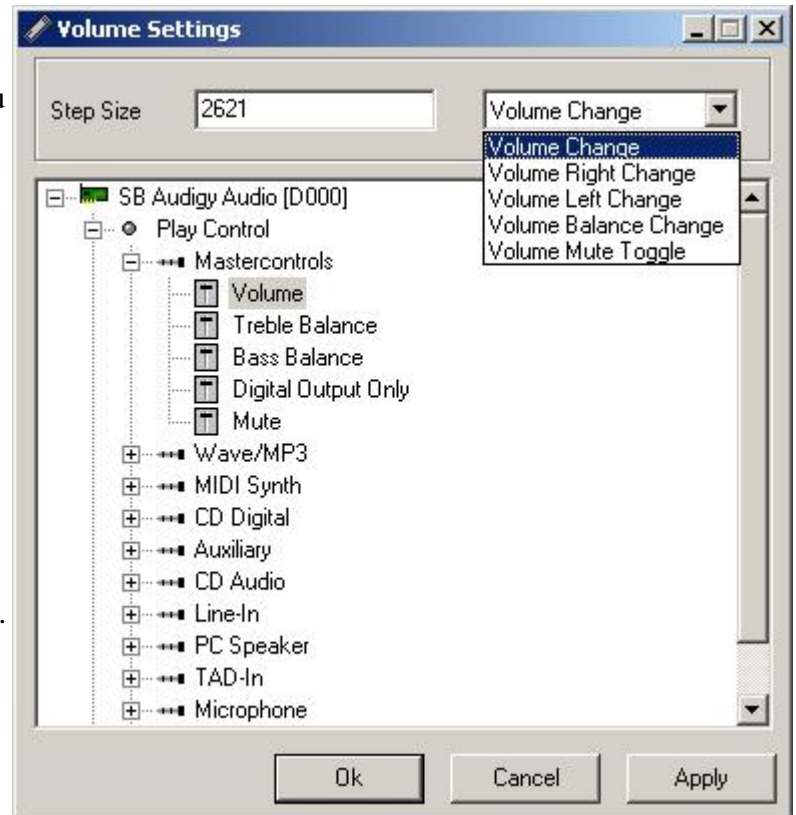
In order to use the various volume actions in Girder you need to select the correct 'line'. This is done by clicking on the 'Volume' button on the OS Action Tab. A new window will appear, as shown on the right.

Remember that each soundcard is slightly different, each has a different variety/number of lines available, and if you change soundcard you need to update the line ID (reselect it).

For 'Mute' actions you should select the appropriate line, (e.g. in this case: "Play Control->Mastercontrols->Mute" would be selected to affect the master mute).

If you select a volume line (e.g. Mastercontrols->Volume), then mute actions will set it's level to 0.

You can also use the mute actions for other lines which have a simple on/off action (e.g. "Mastercontrols->Digital Output Only" can be toggled on/off with the mute toggle.).



Retrieving Volume Level

You can retrieve the level of a line by checking the "Register" box on the OS Actions Tab. It will be stored in the variable "Reg25".

Volume Change - Returns volume level BEFORE change.

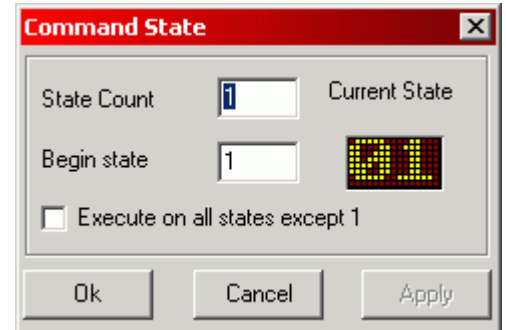
Get Volume - Returns volume level of selected line.

Special Functions

States

What are states?

Girder uses states to determine if an IR command should be executed. When you give an IR command by pressing an assigned button on your remote Girder increases the state by one, except if the state has reached state count, in which case it is reset to 1. When the current state is one (1) the command with state 1 gets executed.



Set "State count" to the total number of different states you want. Then set "Begin State" to the state number you wish the command to be executed on (if you want a command to execute on every command except 1, make sure "Execute on all states except 1" is checked - specify the "Begin state" as the state number you do not wish the command to be executed on.

Where do I find state settings?

State settings can be found by right-clicking on a command/multigroup and selecting "state settings" from the popup menu.

Example:

State Count: 3

This is a simulation of one IR command being pressed seven times on the remote:

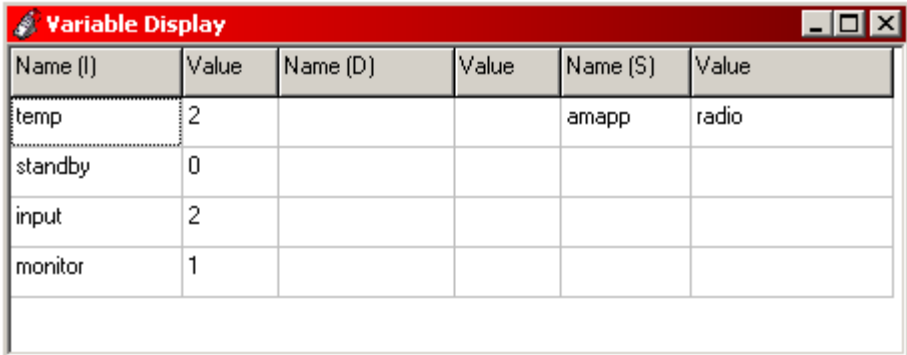
State before	State after	Command Executed?	Command Executed (on all states except 1)?
1	2	Yes	No
2	3	No	Yes
3	1	No	Yes

Thus, on the first key press the command gets executed, then two times nothing appears to happen. The fourth time the command gets executed again.

Registers

What are Registers?

Girder now has an infinite number of variables, which can be manipulated with the **Girder Variable Manipulation Script (GVMS)**.



Name (I)	Value	Name (D)	Value	Name (S)	Value
temp	2			amapp	radio
standby	0				
input	2				
monitor	1				

Previously girder had 25 integer variables and 25 text variables which were known as registers. Many functions/actions still use/refer to these 25 integer/text registers, although they are internally translated to standard variables with the names "regX" and "tregX". This means they can be used with the GVMS and the old functions.

NOTE: "Load Reg" and "Save Reg" load/save all of the variables, not just the original 25.

Variables (registers) are used to store values which can be used by various commands to pass information/display data etc.

Many of the action plugins have the facility for setting/reading registers. For example the OSDPopup plugin can read display text registers on the OSD by specifying the text as: e.g. "My name is [TReg5]". Another example, the "Numbers in Reg 10" exported group uses numeric register 10 to store a value based on user input.

Integer Registers

These registers store numeric values. They can be referenced by [reg1], [reg2]...[reg25].

An excellent example of register usage can be found in this girder group: [register_examples.gir](#).

Text Registers

These are a parallel set of registers which store text strings. They can be referenced by [TReg1], [TReg2]...[TReg25].

Viewing/Editing the Variables

View/reset the variables by selecting Edit => Show Variables/Reset Variables

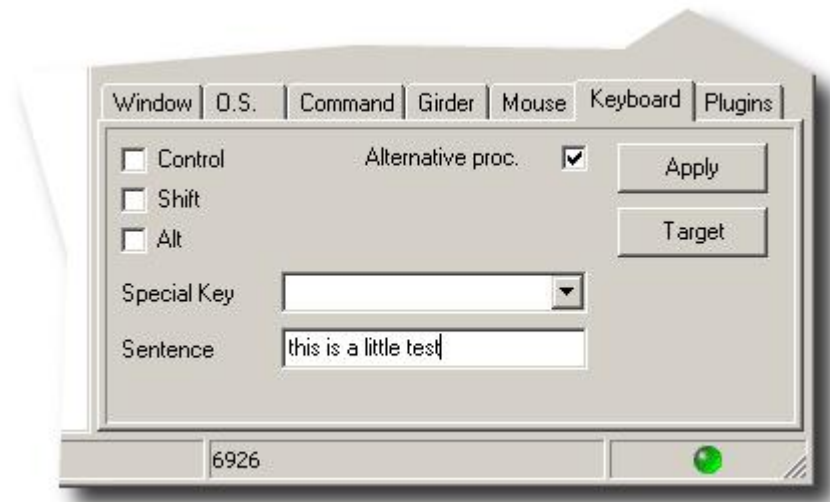
Targeting

Some of the Girder actions can be told to work on a specific window, this is called a 'target' in Girder-lingo. I'll show you a brief example of this using the notepad application.

Add another command to the tree. You know, edit menu, and "add command". Select this newly created command and go to the action tabs. We are going to make Girder type something in notepad without us even near the keyboard!

Select the command and press learn event, now press a button on your remote, Girder will add the "eventstring" and assign the remote control event to it.

Next thing that we need to do is actually assign text-typing to the command. So you'll go to the action tabs and choose the keyboard tab.



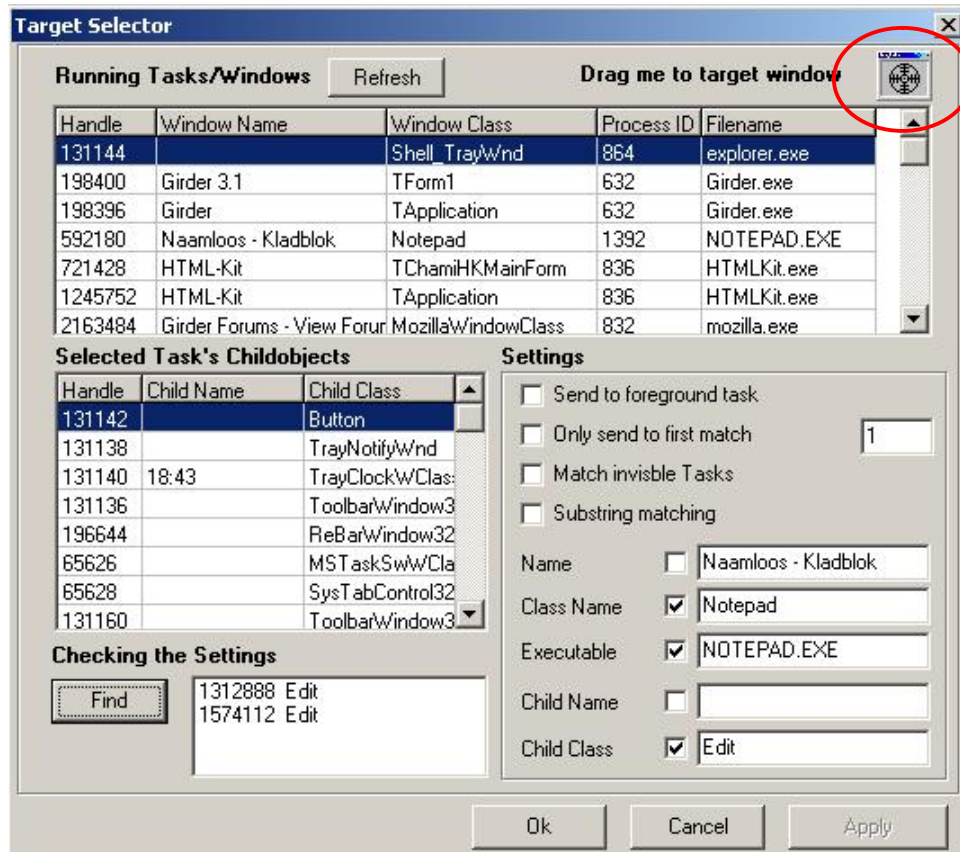
Match the stuff you see above in your own action, thus check the "alternate proc" and type some text, doesn't really matter. Okay don't forget to press apply, remember? We are now going to test this action. Fire up notepad with whatever event that you assigned to it. Make sure that notepad has the focus (title bar is blue or whatever color you chose). Now press the button for the keyboard typing.

Way Cool ! Text is magically appearing in the window! But now I want to be able to type in the notepad window when it is not in the foreground.

Good question, the default target of Girder is the foreground window, but often you'll want to control a program that is not the foreground window. That is where the targeting comes in. Go back to Girder and select the text typing action. Got it ? Good. Do you see the "Target" button, okay now press it. Girder will show you a new window packed with options, text and buttons. Take a few moments to let it sink in what you are seeing. It basically has 3 parts, the 2 lists, these contain the currently running windows, the settings on your right hand side. And a test section on your lower left hand side.

Target window

A really cool feature of the target window is the little crosshair at the top right corner of the window. Arrange the windows on your screen in such a way that you can see Girder and notepad. Now click and hold the crosshair and drag it over the notepad window. Notice how the notepad window gets a black rectangle around it. Aim the crosshair at the big white plane and let go. If you look back at the target window you'll notice that the "Name", "ClassName", "Executable" and "Child Class" have been filled with values, resembling the ones in the image above.



Okay read it over again or try targeting some other windows just to get a feel for it.

We now have the criteria by which Girder seeks out the window on which Girder will unleash hell... well the command to be more precise. But wait, why not check to see if Girder can actually find the window that we just targeted. Okay so press "Find" on the lower left-hand side. As you can see in the image above it listed 2 matches on my computer, I clearly had 2 notepads open while I took the screenshot, so you see Girder will find any open windows that match the criteria. This is an important point to grasp, if you have 2 notepads open and have the targeting set to match notepad, it will match for both notepads thus triggering the action for both of the notepad windows. (Don't forget to press OK or Apply when you are done targeting).

Enough talking, lets try this! Open some other program like calculator or something just to make sure that notepad isn't the foreground window. Press the button on your remote again. Now you'll miraculously see the text appear in the notepad window on the background.

This method can be applied to a lot of the actions that deal with a window, for example if you want to control WinAMP, you'd set the target to WinAMP and then WinAMP doesn't need to be the foreground window to control it.

Can you give me another example of these targets ?

No problem, how about learning how to close a program?

Again, add a new command to the tree, give it a descriptive name (you should always do this by the way, so you know what it does). We now want to close notepad, so we go to the "Window" tab, and select "close" from the list. Don't forget to press apply! Of course Girder needs a target for this action because if you don't set a target Girder will use the window that currently has the focus (remember?). Thus set the target to notepad as we learned before. The last thing we do is attach the event to it (press "learn event").

Now start notepad with the button that you previously assigned to it. Now comes the fun part! Close it again with the button that you assigned to the close action we just made. See how it magically disappears.

Event Processing

But, I want to automate stuff!

Ah you want to make life easier hey, well that's where the multigroup comes into the picture. Think of this as a sort of macro command. The multigroup has commands (actions) nested in it. These commands may have eventstrings assigned to them but Girder ignores them! Girder only looks at the eventstring that you assigned to the multigroup. If the incoming event and the eventstring or the multigroup match all of the commands in the multigroup will be triggered! So you get the automation! Press one button and a whole lotta stuff starts happening.

Groups are quite different , you can compare groups to "Directories" or "Folders" in windows, and they just make your tree look nice and tidy. They 'group' commands that do similar stuff, so for example you'd pack all command that have to do with WinAMP in 1 group. But this is by no means necessary, you can also fill one group with everything you'll ever make. It's just a matter of taste.

You might wonder if 2 actions have the same eventstring assigned to them which of them gets triggered first. This is easy to understand, Girder processes the tree top-down. So its starts at the top looking for a command that matches the incoming eventstring and walks al the way to the bottom of the tree. If it finds a match it will immediately trigger that command. Thus the command closest to the top of the tree will be first.

Conclusion

I hope you now have a little feeling for the way that Girder works, please browse the online help and the help file that comes with Girder to find out more about the virtually limitless possibilities of Girder.