



Welcome to the Girder 3.0 Helpfile

Use this Help for more Information about Girder, the Windows Infrared Automation Tool for everybody.

- It's free and it will always stay that way.
- No stupid nag screens or advertisements.
- Available in many different languages



See the [index page](#) for the topics covered in this helpfile. Or read the about the history of Girder in the [Preface](#).



Preface

About 2 years ago I created my own infrared receiver and transmitter, it didn't work to good but it was a nice experiment. At that point not much software was available for this type of hardware and most certainly not for that pile of scrap I build.

Then last year while doing my practical work for the University on a setup called Nano Magnetical Structures I came across the Universal Infrared Receiver. I was in love. I immediately build this very nice piece of hardware.

Then I went out to find some software to go along with it. But what did I see ? It was all shareware and not quite right. So I created Girder 1.0. It was a *real mess*. My first experiments with Threaded applications, horrible. And most important of all it barely did it's job. But I learned from this program.

So came the day January 11 2000. I started work with ambitious plans. Version 2.0 was born. This took me exactly 10 days (full time that is) to get to version 2.0.0.4. This was nice but it still lacked some features, most importantly the multicommands. So came the day that 2.1 was born. 23 January 2000

Then after a lot of enthusiastic reactions from all of you that I thought: "Ok the features are good, the interface needs work and we need plugins" So came May 5 2000, a Dutch holiday, "Bevrijdingsdag" (Liberation day). The day Girder 3.0 was freed onto the internet, how appropriate.

Thanks for all of your positive reactions and have fun with Girder !

Ron Bessems

<support@girder.nl>

[Back to index](#)



Index

This helpfile is a little out of date, best thing is to check the online help at <http://www.girder.nl/help>. You'll also find a getting started guide there.

General

- [Welcome](#)
- [Preface](#)
- [What's new in version 3.0](#)
- [Features](#)
- [Troubleshooting](#)
- [Group Exports Internet Site](#)
- [Thanks](#)
- [License and Contact information](#)
- [Getting Started](#)

How-to **Information (Frequently asked questions)**

- [Send a Keysequence](#)
- [Close a program](#)
- [Start and stop a program with only one button on your remote](#)
- [Use one remote with many profiles, how to switch groups on and off.](#)
- [What is a State ?](#)
- [Short introduction to Windows Message handling](#)
- [What are multi commands](#)
- [How do I hide and show the taskbar ?](#)
- [How can I make a shutdown that asks before it really shutdown my computer ?](#)
- [Using Special Folders](#)

Interface

- [Main window](#)
- [Actions](#)
- [Settings window](#)
- [Target window](#)
- [Command Capture window](#)
- [State window](#)
- [Volume window](#)

Supported Hardware

Plugins



License and Contact Information

Girder was written by Ron Bessems. He does this in his spare time for the fun of it, and the challenge to get this kind of a thing to work. The copyright license reflects this. The license means something like it's free but you can't make money selling Girder! Also if you use Girder for commercial purposes you have to buy a license. For the exact text see the file copying.txt that came with Girder

My Email address is:

support@girder.nl

The homepage of Girder is:

<http://www.girder.nl>

There is also a mailing list for questions, you can find it at my homepage

[Back to index](#)



What's New



Things that changed from 2.1 to 3.0

- User interface was rewritten
- Actions divided logically.
- Drag and Copy within the treeview is possible
- Copying whole trees is possible
- Almost 100% translatable
- Hardware plug-ins, you can write a plug-in for your own hardware if you like. If you send it to me I'll include it in the distribution and add your name to the thanks section!
- Multiple hardware plug-ins can be enabled at the same time, for example: you can use Girder as a shortcut key manager as well as a IR receiver program!
- Action plug-ins, again you can write your own plug-ins for Girder. If you send it to me I'll include it in the distribution and add your name to the thanks section!
- Better handling of the large font displays. (Scalable interface)
- Timer plugin
- Question action.
- TCP/IP support
- Import / Export groups

[Back to index](#)



Features

Actions

- Recording events in your system
- Mouse moves/clicks
- Window/Object close
- Window/Object hide
- Window/Object show
- Window/Object click
- Window/Object Double Click
- Monitor Power management
- Screensaver activation
- Playing wave files
- Shutdown/Poweroff/ Logoff/ Reboot
- Volume Change
- Execute program
- Girder Group Enable/Disable
- Change the Girder trayicon to whatever you like
- Automatically execute commands on girder startup
- Automatically start Girder on windows start
- Anything you like via Plugins

And much more just experiment with Girder !

[Back to index](#)



Group Exports Internet site

If you have good groups for common (or less common) programs please send them to me so I can include them on my homepage.

<http://www.girder.nl/exportgr.php>

[Back to index](#)



Thanks

I would like to thank the following people for translating Girder 2.1

- Croatian (Hrvatski) - Radovan Kanizaj <Radovan.Kanizaj@crosco.tel.hr>
- Slovak - Miroslav Hlina <miroslav.hlina@ovp.sk>
- Czech - David Pribyl <pribyl@zlin.vatbr.cz> and Tomas Najbert <tox@phil.muni.cz>
- French – Mike Savaria <Mike_Savaria@hotmail.com>
- German (Deutsch) - Uwe Heidrich <uweheidrich@hotmail.com>
- Dutch - Ron Bessems <support@girder.nl>
- English - Ron Bessems <support@girder.nl>
- (no I'm not thanking myself it's also supposed to be a list of people to contact in case of language emergency)

And the Translators of Girder 3.0!!

- Brazilian Portuguese – Gustavo Ambrozio <gustavo@engineer.com>
- Czech - David Pribyl <pribyl@zlin.vatbr.cz> and Tomas Najbert <tox@phil.muni.cz>
- Chinese (by EDIY Studio)
- Danish - Anders Jorsal <anders@jorsal.dk>
- Dutch – Ron Bessems <support@girder.nl>
- French - Mike Savaria <Mike_Savaria@hotmail.com> and Erik Damiano <damiano@cls.fr>
- German (Deutsch) - Uwe Heidrich <uweheidrich@hotmail.com> and Stefan Strehse <Stefan@Strehse.de>
- Greek - Anestis Georgiadis <mranest@iname.com>
- Italian - Alessandro Oddera <aoddera@unige.it>
- Norwegian - Olav Haugan <olav@haugan.as>
- Polish - Jacek Zarnowiecki <zarnow@cksr.ac.bialystok.pl>
- New Polish - Fabiszak Mariusz <M.Fabiszak@wsisiz.edu.pl>
- Russian - Ilja V. Levinson <lev@oduurl.ru>
- Slovak - Miroslav Hlina <miroslav.hlina@ovp.sk> (original version) Lukas Catman <catman@atlas.sk> and Slavo Sebik <sebik@pobox.sk> (updated version)
- Slovenian - Gregor Gliha <gregor.gliha@anni.si>
- Svensk - Kristoffer Lidman <kristoffer-lidman@swipnet.se>
- Spanisch - by Xavi Rodriguez <mailto:x_nobita@hotmail.com>

Helpfile Translators

- Uwe Heidrich German Language
- David Pribyl Czech language
- Radoslaw Rybinski Polish language

Plugin-Writers

- Silitek driver - Erik Damiano <damiano@cls.fr>
- SFH-56 driver - Igor Cesko
- Creative iNFRA receiver by Dan Grainger
- WinLirc by Alexander Seitz
- PCTV remote driver written by Docs <docs@hhc-online.de>
- X10 MouseRemote - written by Alessandro Angeli
- Slavo Sebik <sebik@pobox.sk>
- Mark Fiechtner

Miscellaneous

- Adriaan Boertjes <A.G.Boertjes@stud.tue.nl> for making the splashscreen for Girder, which I brutally abused into 16 colors.
- John Paton for the wave samples for the talking clock.
- Teemu Hummer <teemu.hummer@kolumbus.fi >for maintaining the Exported Group Database

Thanks a million guys (if I forgot to mention you on this page don't feel insulted and just drop me an email)

[Back to index](#)



Gettings Started

This section briefly describes how to start working with Girder.

Installing from 'Installer'

This is very simple just tell the installer where everything goes and you're ready.

Installing from 'zip-file'

Create a sub-directory called girder3 (or something else, this is just an example) and unzip Girder there. If you use the dos pkunzip, it won't work. Girder uses long filenames. Please use unzip32 or Winzip.

Choosing your hardware

Because Girder supports multiple hardware plugins (drivers for the UIR/Irman IrDA) you must tell girder which hardware you have. You do this by opening the settings window, and switching to the Hardware plugins tab.

File->Settings->Hardware Plugins

Here you should see a list with hardware plugins for example the UIR/Irman, IrDA, Keyboard, TCP/IP. To enable one or more of these drivers check the checkbox in front of the name in the list. When you press apply Girder loads the drivers.

If you now want to change the settings of the drivers click on the name of the driver in the list, the button 'settings' should now light up. (if it doesn't make sure that you've checked the item and pressed apply).

When you press the settings button the driver displays a settings dialog. Look at the [Supported Hardware](#) page on info about how to set them up.

After you configured the driver close the settings window and activate the driver by pressing 'F9' or 'enable input devices' on the file-menu, the blue led should turn green. With the Infrared drivers like UIR/Irman or the IrDA adapter Girder should blink the green led to yellow when you press a button on the remote control.

Enabling Action plugins

Girder comes with some extra action plugins. To enable them open the settings dialog, and switch to the action tab.

File->Settings...->Action tab

Select the action plugins that you want to enable by checking the checkbox in front of the action plugin name. After you pressed apply they should appear in the main window on the settings tab.

Setting Girder to your language

Girder as of this writing supports 10 additional languages, and more are being written every month ! To select your language again open the settings dialog, and switch to the user interface tab. Select your language from the dropdown box and press apply. Girder should now speak your language.

Making your first Girder action !

I'll show you now how to quickly make an action that launches notepad.

Right click the treeview (the big empty white plane) and select add command. Girder now added 2 item to the treeview. A group, and a command. The commands in girder must be divided into groups. Select the command. Now on the bottom right you'll see a lot of tabs select the O.S. tab (the second). In the dropdown box select execute. Press apply. Then press the browse button and find notepad.exe (usually in the windows directory : c:\windows\notepad.exe). Then learn a IR code by pressing learn IR, and pressing a button on your remote control. You're done.

It's possible that Girder doesn't react when you press the remote control button because Girder has a option not to execute the commands when it is in the foreground. So either disable this option or click on the desktop so that Girder loses the focus.

Also take a look at the sample.gir in the Girder installation directory. Also the [Group Exports Internet Site](#) is a great source of pre-made actions.

[Back to index](#)



HOWTO (FAQ)

In this section I will cover some techniques and common problems.

These examples are based on Notepad and Winamp, naturally you can do this to any other program.

[Send a keysequence](#)

[Close a program](#)

[Start and stop a program with only one button on your remote](#)

[Use one remote controls with few buttons but with many groups, how to switch groups on and off.](#)

[What is a state ?](#)

[Short Introduction to Windows Message Handling](#)

[What are Multi commands](#)

[How do I Hide or Show the windows taskbar ?](#)

[How can I make a shutdown that asks before it really shutdown my computer ?](#)

[Using Special Folders](#)

[Back to index](#)



Sending a Keysequence

Problem: I want to send a sentence to Notepad.

Answer:

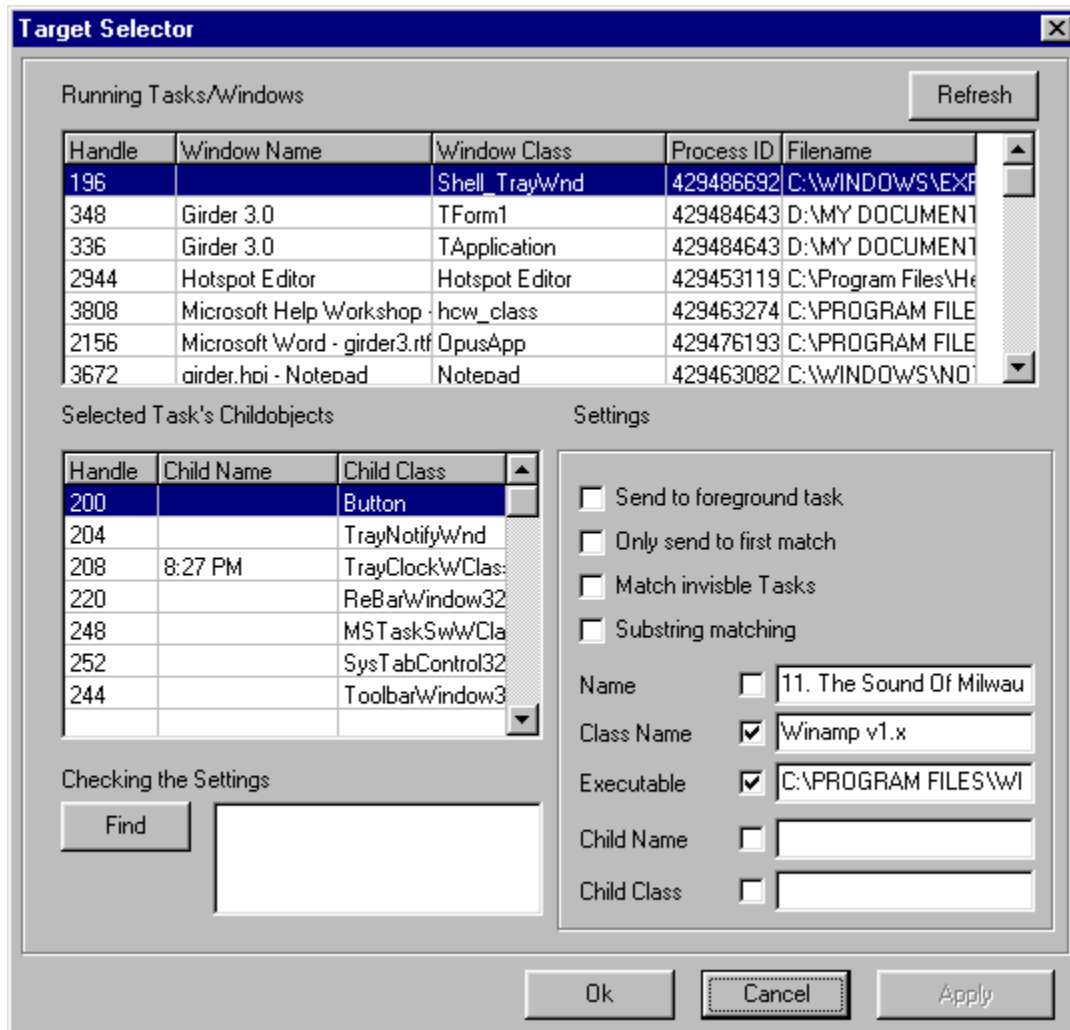
The first thing you should do is to add a command, enable it and set the target, and set the keysequence. Now all of this with explanations:

Add a command:

right click the treeview, select 'add command', and select the command by left clicking it.

Set the target:

this is the most important and difficult part of all of this. Make sure that notepad is running. And press the 'Target' button in the main window, you should now see the following: (press refresh if you started notepad after opening the target window)



Because Girder needs to know where to send the sentence you should specify the destination window, that would be Notepad.exe, and the child object on the destination window. We want to type some text into the textwindow. We see that Notepad has 2 child objects: Edit and TrayNotifyWnd. You could try both to see which one is the one we want or think a little bit logical and select the Edit child object.

Notepad has the name of the text file in the caption, thus this can change if you tell Girder to match the WindowName it won't find notepad after you saved the file, because notepad then changes it's caption. So we only match ClassName and Executable.

You tell girder to only match a certain type of information by checking the checkboxes before the textboxes.

As you can see the child object doesn't have a name, only a class name (Edit). Make sure you tell girder in the target settings that it should match the Child Class only.

This should do it for the target settings!

Set the Sentence:

In the main window click on the keyboard tab, and type your sentence into the Key sequence textbox. You should see that the apply button is enable, whenever this happens you made a change that should be confirmed with apply.

Testing:

Now teach Girder some IR code by pressing Learn IR and off you go.

[Back to how-to index](#)



Close a program

Problem: I want to Close Notepad

Answer:

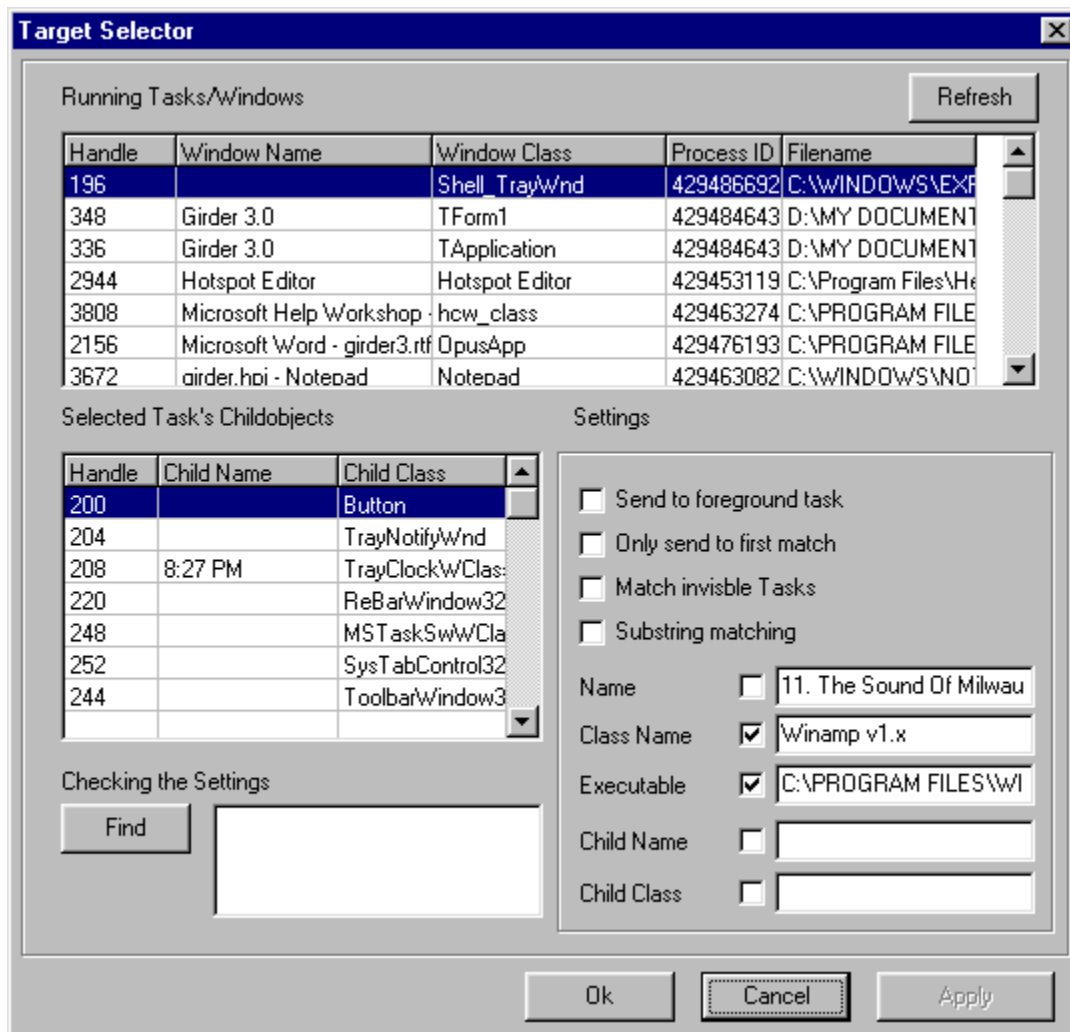
The first thing you should do is to add a command, enable it and set the target, and set the 'close' action in the action tab. Now all of this with explanations:

Add a command:

right click the treeview, select 'add command', and select the command by left clicking it.

Set the target:

this is the most important and difficult part of all of this. Make sure that notepad is running. And press the 'Target' button in the main window, you should now see the following: (press refresh if you started notepad after opening the target window)



Because Girder needs to know where to send the close command you should specify the destination window, that would be Notepad.exe. Because notepad changes its caption depending on the filename of the textfile you should tell Girder to only match the ClassName and Executable, unless you only want to close a specific notepad.

Now go back to the main window and click on the action tab. Select 'Close' from the message type dropdown box. And apply your settings.

Testing:

Now teach Girder some IR code by pressing Learn IR and off you go.

[Back to how-to index](#)



Start and Stop a program with one button

Problem: I want to start and stop winamp with one button

Answer:

Use a multicommand and add 2 commands to this. Set the begin state of the first to 1 and the begin state of the second to 2. Set both the state counts to 2. Execute winamp in the first command and close winamp in the second command.

Long answer:

- First you'll have to add a multi command to some group, right click the group to which you want to add the multi command, and select add multi command.
- Teach Girder the IR string. Press learn IR
- Enable the Multi command.
- Add two commands to the multi command. Right click the multi command and select add command (twice)
- Enable both the commands
- Rename the first to 'Start', rename the second to 'Stop' or something like that.
- Select the 'start' (first) command and open the state window (press 'state')
- Set the State count to 2 and the begin state to 1
- Select the 'Stop' (second) command and open the state window
- Set the state count to 2 and the begin state to 2
- Select the 'Start' (first) command and set the command option to execute, and browse to the winamp filename. (press apply afterwards!)
- Set the second command to close winamp (see [Close a Program](#))

This should do it!

An example is included in the sample.gir called 'WinAmp\Power switch Winamp'

Mind, some remote controls keep sending the IR string, thus starting winamp and then immediately sending the next IR string and thus closing winamp. This is usually the case for volume controls but this can also happen on other buttons. You can counteract this by setting the Anti-Repeat Wait time, to something like 500, this gives you have a second to let go of the remote control button.

[Back to how-to index](#)



How to switch groups on and off

Problem: I haven't got enough buttons on my remote to put all commands in that I want to use

I have 3 groups: 'DVD', 'Winamp', 'Windows function'

Answer: Girder can enable and disable groups. You can select a button on the remote that switches between groups.

Add a multi command and 6 commands to that multi command. Teach Girder the IR string that should switch between groups.

Fill the commands with the following information

Name	DVD_on	DVD_off	Winamp_on	WinAmp_off	Win_on	Win_off
State count	3	3	3	3	3	3
Begin state	1	1	2	2	3	3
All except 1	no	yes	no	yes	no	Yes
Action	enable group	disable group	enable group	disable group	enable group	Disable group
value1	DVD	DVD	Winamp	Winamp	Windows function	Windows function

Be sure to get the Case right in the value1 option Girder is case sensitive.

Now the following happens:

First button press :

'DVD' group gets enabled

'Winamp' group gets disabled

'Windows Function' gets disabled

Second button press :

'DVD' group gets disabled

'Winamp' group gets enabled

'Windows Function' gets disabled

Third button press :

'DVD' group gets disabled

'Winamp' group gets disabled

'Windows Function' gets enabled

An example is included in the sample.gir called 'Group Controls' in in the group "Misc"

[Back to how-to index](#)



Ask before shutdown

Problem: I want Girder to ask before it shuts down my computer

Short Solution: Create 3 commands, the first one should be of the type question, link the first group to the second command, en the second group to the third command. Make the second command shutdown the computer and the third should make some kind of a sound. Assign a IR code to the first one and your are done.

Long Solution:

- Create three commands.
- Rename the first command to “Question”
- Rename the second to “shutdown”
- Rename the third to “abort”
- Select the Girder tab, and the Question type.
- Type in the name box the following: “Should I Really shutdown the computer ?”
- Press apply.
- Assign a IR code to the first command, by pressing Learn IR
- On the lower right there are 2 small browse button, press the top one (not the big browse!). There now should popup a treeview of all of your commands similar to the main treeview. Pick the second command that you added.the name “Shutdown” should now appear next to the button
- Now press the lower browse button and select the third button that you created. The name “abort” should now appear next to the lower browse button.
- Select the “shutdown” command in the main treeview, then select the O.S. tab and set the type to shutdown or poweroff. Press Apply.
- Select the “abort” command in the main treeview then select the O.S. tab and set the type to “Play wav” browse for a nice wav to play and press apply.

When you now press the assigned button on your remote Girder should popup a OSD (On Screen Display) with the text that you typed. (Should I really shutdown the computer?) When you press the same button again within 5 seconds girder executes the “shutdown” command, and when you press any other button Girder executes the “abort” command. When you don’t press any button at all, nothing will happen.

[Back to how-to index](#)



What is a state ?

See :

[State Window](#)

Examples:

[How to switch groups on and off](#)

[Start and Stop a program with one button \(Toggle button\)](#)

[Back to how-to index](#)



Windows Message handling.

Windows communicates with application via Messages. These message have 4 parameters. Firstly the destination window. This can be a window, a textbox, a checkbox, a menu, a editfield etc... To find the destination window Girder uses the WindowName, WindowClass, Filename, Childname, ChildClass. The second parameter is the message type, like WM_CLOSE (closes a window) or WM_COMMAND (menu commands etc..). The third and fourth parameter are the message type parameters.

Some of the Message are the same for all applications like WM_CLOSE. But when you push a button windows generates a WM_COMMAND or WM_SYSCOMMAND message. The parameters of this message varies from program to program. There is no way to know what this will be. That is why Girder has the System Wide Capture facility.

The System wide capture facility hooks into the main message loop and filters all messages out except the WM_COMMAND and WM_SYSCOMMAND. These then get displayed in the capture window. Girder is now able to fake these messages, to the applications. This way the applications think that they had a buttonpress or something.

[Back to how-to index](#)



Multi Commands

This is a sort of a group but this group has a IR string assigned to it, and all commands in the multicommand don't have a IR string. The commands in the multi command all get executed when the IR string is received.

[Back to how-to index](#)



Hide and Show the Taskbar

Problem: I want to hide the taskbar !

Answer:

Add a command that uses the 'hide' action on the window with WindowClass 'Shell_TrayWnd', this is the taskbar.

Long Answer:

- Add a command
- Open the Target Window
- Select the Task with the WindowClass 'Shell_TrayWnd'
- Be sure to set the matchby options: only WindowClass should be selected.
- Apply these settings
- In the main window select the actions tab.
- For the message type select : Hide when you want to hide the taskbar, or show when you want to show the taskbar.
- Press Apply
- Teach the IRstring
- Enable the command

That's all folks.

[Back to how-to index](#)



Special Folders

•

If you have to do multiple installations of Girder you sometimes want to have access to the special windows directories like %DESKTOP% or %MYDOCUMENTS% before you actually know where they are. By using these variables you let Girder figure out where this directory is. Also it can be useful in exported groups!

Currently the following inputs accept these special variables

- Hardware plugin directory
- Action plugin directory
- Auto load
- Play Way Action on the OS tab
- Execute Action on the OS tab

The following variables are defined.

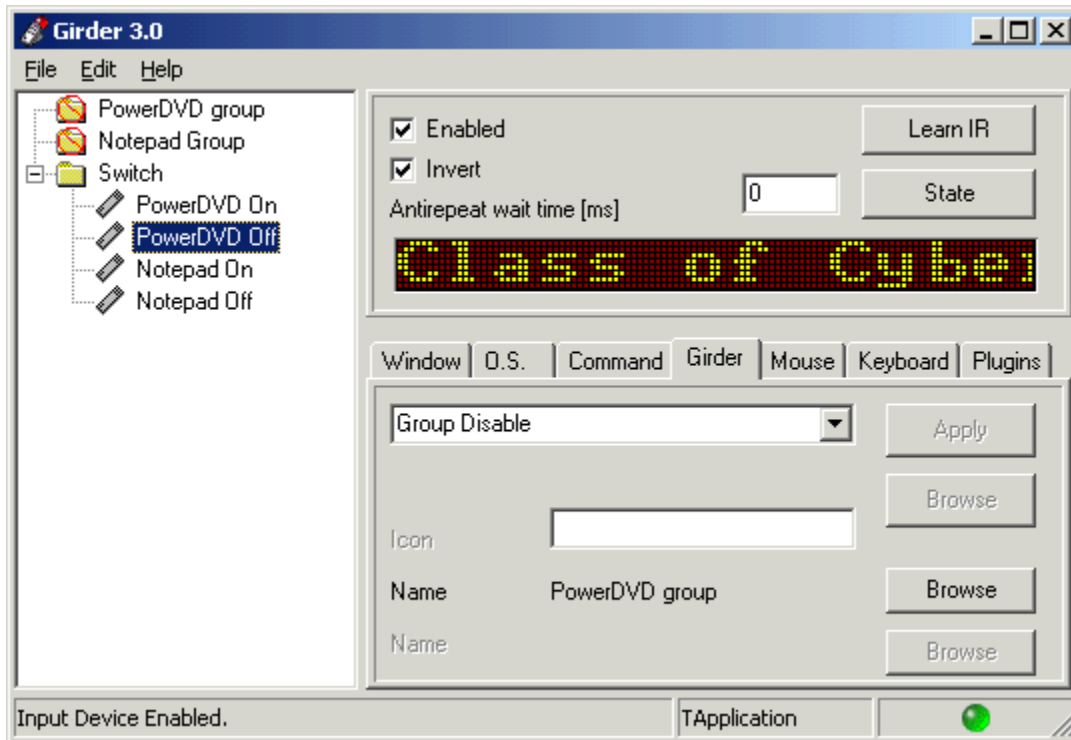
- DESKTOP
- MYDOCUMENTS
- MYPICTURES
- PROGRAMFILES
- WINDOWS
- SYSTEM
- APPDATA

For example you saved the autoloading file on the desktop and called it runme.gir. Now to open it you can type the following into the autoloading path control: “%DESKTOP%\runme.gir” Girder will now replace %DESKTOP% with the appropriate directory.

[Back to how-to index](#)



Main Window



This is the main window of Girder click on the picture to get more help

[Back to index](#)

This is the main treeview of Girder. Here all the commands are stored in a logical way.

Invert makes girder execute this command when the input signal doesn't match the stored signal.

This checkbox enables or disables a (multi)group or a command.

The anitrepeat wait time make Girder wait for the specified amount of milliseconds before accepting a new IR command. This is useful if your remote continuously sends IR command when you press a button, but you only want to execute one action.

This button make Girder assign the next IR string to the selected command. Some plugins need to know if they should be in learn mode, they popup somekind of a dialog. To choose from which plugin to learn press the right mouse button on this button and choose the plugin form the list. If the list only says "none" than you don't have plugins that need this feature and you can safely leave it at none, the learn will work from any input device.

Displays the [state window](#) of the current command.

The action tabsheet

The led indicator



An error has occurred



Input devices are enabled and Girder is waiting for a command



Input devices are turned off.



Girder is in standby mode. In this mode Girder only checks if the incoming IR string is the Girder standby toggle IR string, nothing else.

The Ledbar displays the IR string and sometime the name of the command



Actions

[Window Actions](#)

[OS \(Operating System actions\)](#)

[Command Actions](#)

[Girder Actions](#)

[Mouse Actions](#)

[Keyboard Actions](#)

[Plugin Actions](#)

[Back to index](#)



Window Actions

Clicked

Sends a mousebutton click to the control specified in the target options. Use this if you want to press a button (or something similar) on a window.

Double Clicked

Sends a mousebutton doubleclick to the control specified in the target options. Use this if you want to press a button (or something similar) on a window.

Show

Shows a previously hidden window or control.

Hide

Hides a window or control.

Close

Closes a window.

Set focus

Sets the focus to a window. (might not work)

Wait

Make Girder wait for a Window or a specified time. This can be usefull when you have a multicommand that starts a program and then has to click on a control. Put the wait inbetween te commands to make it work.

Maximize/Minimize

Maximize / minimizes a window

Absolute Mouse Move

This moves the mouse to the given coordinates

Move window

Moves a window to the given coordinates

Resize window

Resizes a window to the given size

Center and Resize

Resizes the window to the given size and centers it on screen

[Back to index](#)



OS Actions

Monitor On/Off

Puts the monitor into power save mode. Warning be sure to set both the Monitor on and off. Else you might need to navigate windows in the dark.

Screensaver

Starts the screensaver.

Play wav

Plays a wav file.

Shutdown/Poweroff/LogOff/Reboot

Shutsdown windows.

Volume changes

Changes the volume. If the step size is positive this volume increases if it's negative it decreases.

Volume Mute toggle

Toggles the mute switch of the soundcard. (This doesn't need to be the mute option it can be anything)

Execute

Run a program with the specified options. (like minimize to hide it on start) to add parameters enter these in the second box labeled "step size".

Talking Clock

This makes Girder say the current time in English. If you want to use this feature you'll need the speech pack from my homepage.

Volume/Mute Display

Sometimes you have keyboards that do the actual changing of the volume for girder, if girder would also change the volume strange things could happen. To be able to see the volume anyway use this action.

Taskswitcher

This action emulates a "Alt-Tab" so you can cycle through the desktop apps

Change Resolution

This action allows you to change the resolution of the display (including refresh rates), the available modes are listen in the bottom dropdown box.

Eject/Load Media

You can eject and load for example a CDROM with this command. (Load not supported under win9x)

[Back to index](#)



Command Actions

Command/ Syscommand

This sends the WM_COMMAND/WM_SYSCOMMAND to the application specified by target. You want to capture the Wparam/Lparam via the Command Capture facility.

[Back to index](#)



Girder Actions

Group Enable/Disable

Gives you the possibility to disable/enable a group by remote control.

Set (XX)Trayicon

Sets the icon that Girder displays in the Traybar.

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.

Question

This is a very nifty features, lets say you wanted Girder to shutdown your computer but you also wanted to be sure that you don't press the shutdown button by mistake. This action displays a On Screen Display with the sentence that you type in the name box. If you press the same button twice Girder will execute the command that you selected in the top box. If you press another button Girder will execute the command that you selected in the bottom box. When you don't press any button Girder won't do anything.

Note: If you have an question action in a multicommand there is ofcourse no IRstring (eventstring) coupled to the question action, however the question action needs an IRstring eventstring the way you do this is to move the command out of the multicommand and "learn" an event to it there and then move it back into the multicommand.

Goto

This action jumps to another command in the tree.

Enable/Disable Device

With this function you can selectively disable and 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

Window Exists

With this function you can make a decision based on if a window exists or not

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.

[Back to index](#)



Mouse Actions

This makes Girder move the mouse.

[Back to index](#)



Keyboard Actions

This make Girder send a sentence or a special key to a window. Girder now has 3 methods of sending keys

1. To the foreground select this by setting the target to foreground
2. PostMessages (Select Alternative proc)
3. SendMessage

You don't need to know what the last 2 things mean just try them both if it doesn't work.

[Back to index](#)



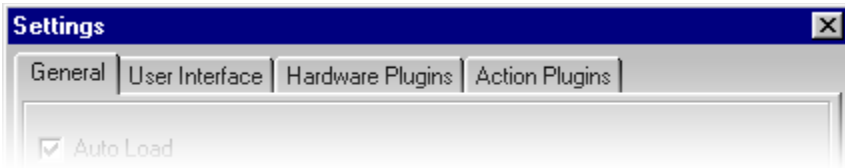
Plugin Actions

These are the plug-ins that are available. Double click them to configure them.

[Back to index](#)



Settings Window



Click on the tabs to select the page of interest.

[Back to index](#)



General Settings



Autoload: This automatically load the specified file when Girder starts.

Autoexecute: This command or multigroup will automatically be executed when Girder starts.

Launch Girder when windows starts: Does what it says. When windows starts Girder automagically starts also.

Don't send IR commands when Girder is in the Foreground: This one is good for some confusion. When Girder is in the foreground and you press a button on your remote it shows you which command is assigned to that signal. By default however it will not execute the action assigned to that command. So it looks as though Girder doesn't work. If you want Girder to execute the action even if it is the foreground window, uncheck to box.

(this does not apply to some 'Girder' actions like 'Minimize Girder')

Save data on shutdown: When you changed the datafile and quit Girder, you normally get prompted with a 'File has changed' dialog. When you check this box Girder will automatically save the datafile when Windows shutdown.

Register Filetype: This one couples the Filetype to Girder, so you can double click the .Gir files.

[Back to Settings](#)



User interface settings



Language: Here you set the language that Girder speaks. If your language isn't in there write it and send it to me. I will then add you name to the [thanks](#) section of Girder. And off course a lot of people will be very happy with you language file

Dotmatrix font: This is the font used by the dotmatrix display on the main window. Girder load PSF fonts, the kind containing 256 characters of width 8 and height 8 and no unicode mapping. This is a common type of PSF font. These fonts can be found on major ftp servers like metalab.unc.edu. When you specify no font Girder uses a internal font. I'll also release a fonteditor, just check my homepage to see if it is there allready.

Scale Main Form: This scales the main form to the factor you specify here. It can be handy when you use large fonts.

Traybar mode: This puts Girder's icon into the traybar. When Girder is minimized the main icon is also remove from the taskbar.

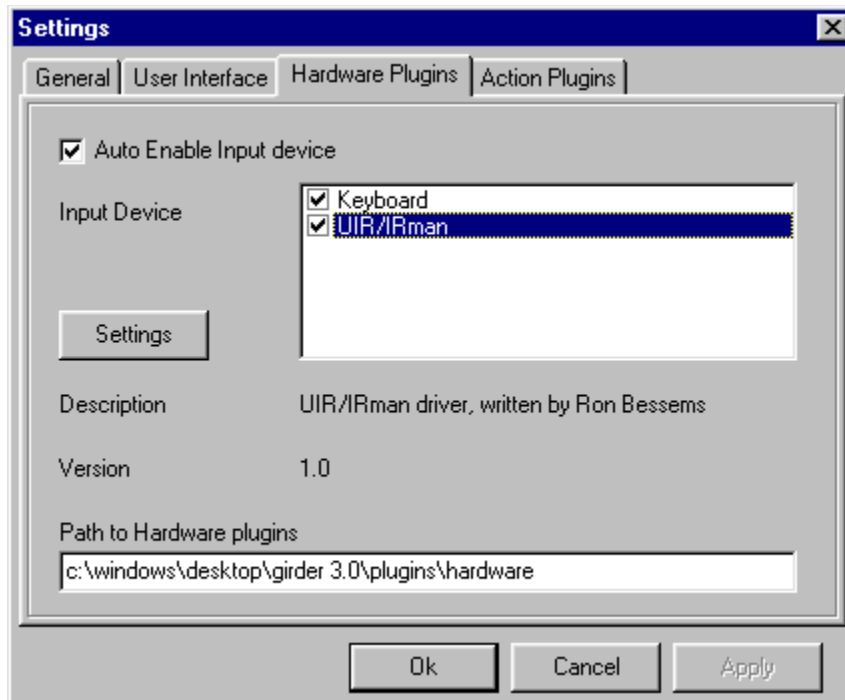
Hide on startup: If you don't want Girder popping up at windows startup but instead to immediately go to the traybar check this.

Display Splash screen: Disable the beautiful splash screen in Girder.

[Back to Settings](#)



Hardware Plugins



Autoenable Input device: This automatically enables the input devices (e.g. UIR/IRman) when Girder starts.

Input device: These are the available input devices [see supported hardware page](#).

Description: The description of the Plug-in usually the function and author is mentioned here.

Version: The version of the plug-in.

Settings: The settings of the plug-in will be displayed when you push this button

Path to Hardware DLLs: This should be correct but if its not put the path to your hardware dlls here.

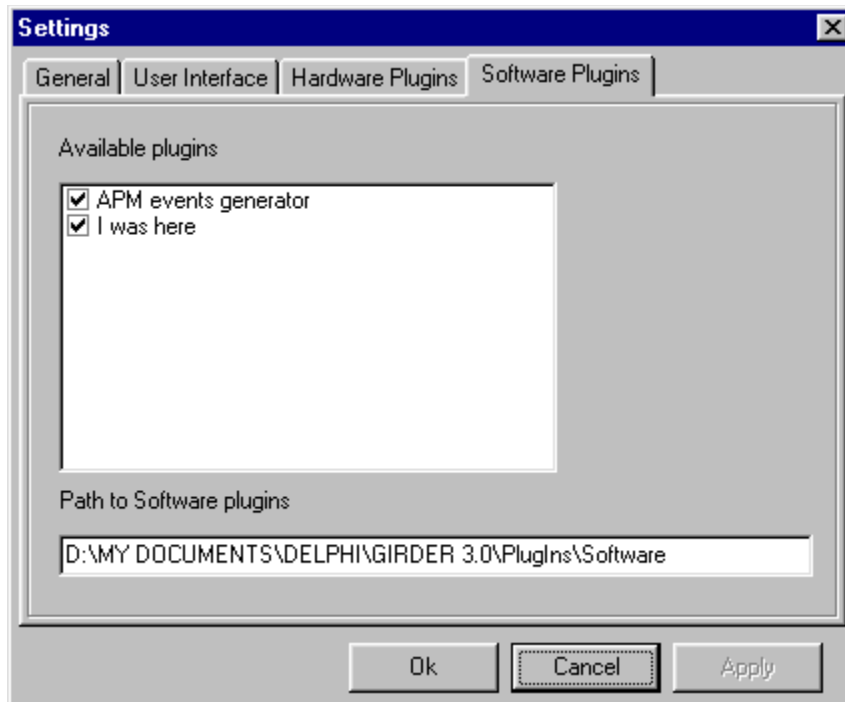
When you check a new input device the driver isn't loaded immediately. Only after you press Apply or Ok the driver loads.

If the input devices are enabled and you check another driver, this doesn't enable the drivers immediately, disable and enable the input device from the main menu to enable all input drivers.

[Back to Settings](#)



Software Plugins



This box lists all the available software plug-ins. If the plug-ins are checked then they will appear in the Plug-in tabsheet of the main window. This way you can save memory when you have a lot of plug-ins on your disk but only use a few.

[Back to Settings](#)



Target Window



Click on the window to find out more.

Girder now has the possibility for drag & drop targetting,... just drag the crosshair from the target window to the window or button of choice!!

[Back to index](#)

This is the list of running tasks/ programs / windows. In this list you can find the Handle, WindowName, WindowClass, ProcessID and the Filename (of executable). The Handle, ProcessID change with every start of the program, so these aren't useable for the identification of a program. It's possible that a window doesn't have a windowname or windowclass so make sure to set the match by checkboxes accordingly.

This are the child objects of the selected program. It's possible that there aren't any child objects.

When you're finished settings the target settings you can do a check. With this check you can determine if your settings yield one window or more, if this yields more windows, this means that Girder will send the command to every window specified here. If this isn't what you want then make the matchby options more strict. If it yields no matches than change the windowname and select submatch, or lessen the matchby options.

These are the Matchby options. When you check one of these options Girder will search for windows matching this criteria. If you check multiple options Girder will search for a program that has all of the characteristics.

Send to foreground window disables the window searching routine and simply sends the command to the window that currently is in the foreground.

When your matchby criteria yields more than one window you could choose to only send to the first match.

Windows also has programs that aren't visible you can also send commands to them. To see the invisible task in the task list first select this checkbox then press Refresh.

Substring matching does exactly what it say. For example Winamp. Winamp changes it's caption according to the mp3 file that it is playing, but the text 'winamp' is always in the caption so you could set the '*windowName*' to 'winamp' and check this button.

Window Name, this usually is the text in the titlebar of the window. Girder also uses this to identify a window.

All windows have a classname, Girder also uses this to identify a window.

The executable file of the window. Girder also uses this to identify a window.

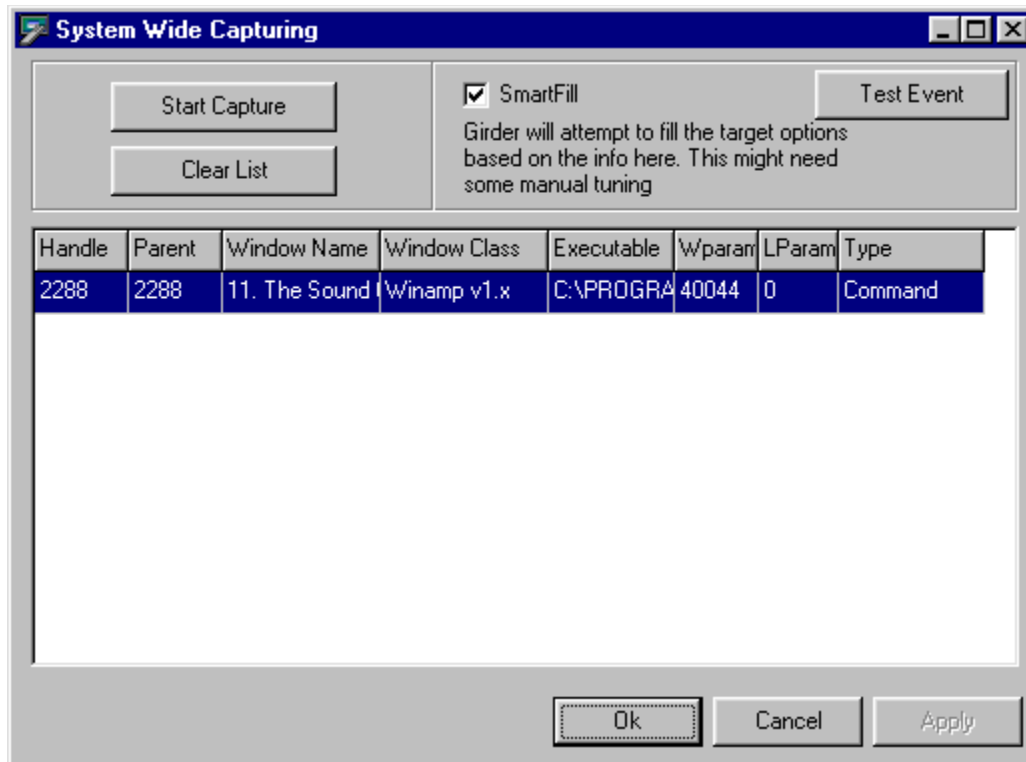
The Name of the child object Girder also uses this to identify a child object.

The `ClassName` of the child object. Girder also uses this to identify a child object.

The Refresh button refreshes the Task list, use this if you started the program you want to control after the target window appeared.



Command Capture Window



Under windows almost everything happens with messages. There is no way to know what message is being generated by a button or a shortcut. At least if you haven't got a spy tool and some knowledge of those events. That is where Girder system wide capturing comes in.

Girder Hooks itself into the Windows message loop and filters all the rubbish out and displays the interesting events. WM_COMMAND, WM_SYSCOMMAND and mouse clicks!!

How to find the event your looking for.

- First enable the capturing system, press start capture
- Then go to you application of choice and execute the command you want automated.
- Go back to Girder system wide capturing, you should see some information in the boxes.
- To see which of those is the events you are looking for, select the message that you think is the right one, and press Test event.
- Now the same thing should happen automatically.
- Press apply to enter this information into the command

If this didn't work you should try to generate the command on a different way, so if you type control-v the first time now select it from the menu with the mouse. Maybe the right command

was captured.

Example: Winamp, play a file.

- Add a command to some group in the main window
- Start Winamp
- Click on messages, then click on the '*capture event*' button
- Click on '*start capture*'
- Switch to winamp.
- Press 'x' (this is the shortcut to play a file)
- Now you should see at least one event in the Girder system wide capture window. Something like '1.Demo – Winamp' '40045' '65536' 'WM_SYSCOMMAND'
- Select that one.
- Press '*test event*', winamp should now start playing the file.
- Now press '*Ok*'.
- Girder now sets the '*message type*' to WM_SYSCOMMAND and the values to 40045 and 65536.
- Girder also sets the target window options, this however is incorrect for winamp because winamp always changes it's title.
- Open the Target window, and check the '*substring matching*' option, press '*Ok*'
- Change the windowname target to Winamp.
- Apply, and assign a IR command.

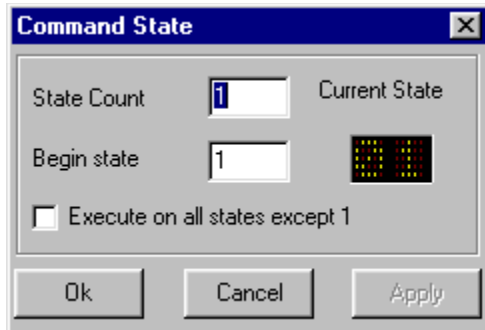
Put Girder to the background and press the button on your remote you just assigned. Winamp should now

Start playing!

[Back to index](#)



State window



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 with one, except if the state has reached state count than it is set to 1. When the current state is one (1) (before increasing happens) than the command gets executed. When Girder starts or when you add an command, the state is set to begin state. Confused ? Example :

State Count : 3
Begin state : 1
Execute on all states except 1 : not checked.

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

State	Executed ?
1	Yes
2	No
3	No
1	Yes
2	No
3	No
1	Yes

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

State Count : 4
Begin state : 3
Execute on all states except 1 : checked.

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

State	Executed ?
3	yes
4	yes
1	no
2	yes
3	yes
4	yes

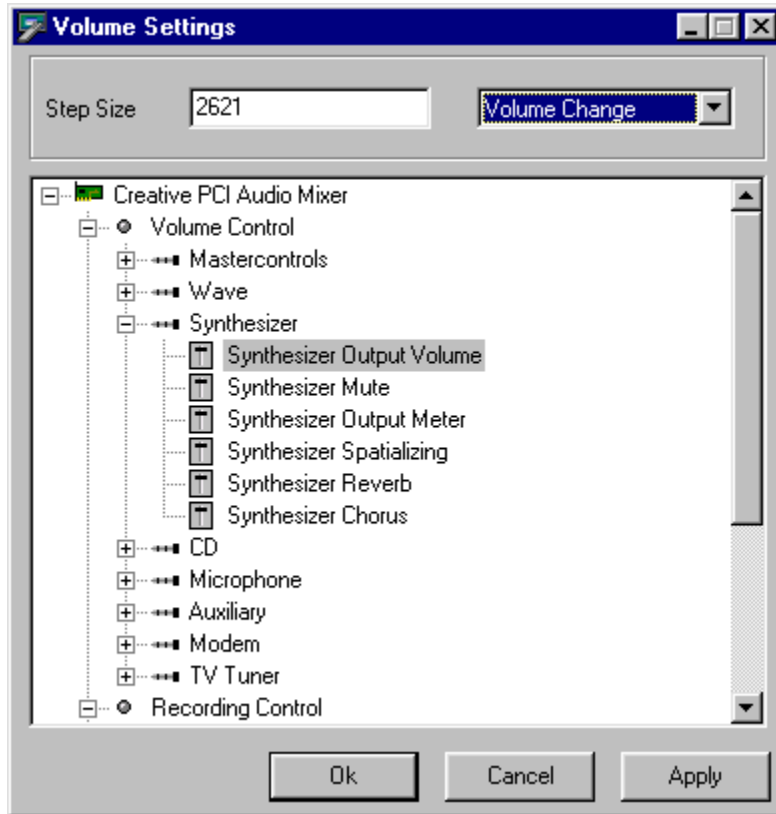
1 no

See the [on/off button example](#) in the FAQ section.

[Back to index](#)



Volume Settings



I've tried to make the volume settings as general as possible so that you can set any control that your soundcard offers, without me ever knowing anything about that. That's why at first glance this tree looks somewhat overdone.

If you want to be able to control the output volume of the wave device on your sound card, you can do the following:

- Select your sound card.
- Select the volume settings
- Select the wave device
- Select the volume control

Girder now gives you a default step size, positive means increase the volume, negative means decrease. Windows allows the user to change the volume in 65000 steps. But not every soundcard supports that many step, and it would take very long for you to set the volume.

For example you want to set the maximum volume from zero volume. One remote command takes 0.1 second that means that you would have to hold the button for 2 hours ! So that's why there is a step size. This way you can specify how exact you want to control the volume, but most of us want to use the default step size.

If you want to control the mute settings of a device be sure to select the Volume Toggle Mute setting.

Common pitfall: If you want to mute a channel be sure to select the correct line from the Volume window!!! If you select the normal volume line girder will turn te volume down to 0 thus appearing to mute but actually the volume changed!!!

[Back to index](#)



Troubleshooting

[Girder does not react to my command](#)

[The UIR/Irman does not initialize](#)

[My Cat got stuck in a tree](#)

[Back to index](#)



The program does not react to my command

- Make sure that Girder isn't the foreground window, by default Girder doesn't send the IR commands when it is the foreground window.
- Make sure that the command is enabled
- Make sure that the right IR string is assigned (check this by looking at the IR string that Girder detects, in the statusbar)
- Make sure that the group is enabled.
- Check the Target settings (Use the check button)

[Back to index](#)



The UIR / Irman does not initialize

Try the following

- Check the comport Settings.
- Set the initialization speed to low.
- Try skip initialization check
- Check the hardware
- [Mail me](#) with ALL details of your problem.

[Back to index](#)



My cat got stuck in a tree

Well,... hmm,... follow these instructions:

Make a multicommand

Set all options to 42

This should solve your problem, actually it should solve any and all problems all over the world, theoretically though.

[Back to index](#)



Supported Hardware

UIR/Irman:

This is the plug-in for the UIR/Irman.

Keyboard:

This is the keyboard plug-in that allows you to use Girder from the keyboard.

TCP/IP Server:

This plugin allows you to control girder from anywhere on the (inter)network. To be able to control Girder you must know the password and the port specified in the settings dialog of the Plugin. Be sure to add the IP address of the computer from where you want to control Girder. You can use the TCP/IP action plugin to control it, a simple terminal emulator or the commandline tool tcpclient.exe in the girder directory.

When you use a telnet client the first thing you should type is "pass <youpass>" girder should than

Respond with "

"2: execute <string> = execute a command in girder"

"Close = close connection"

When you want to make girder do something on a TCP event, press the learn IR button and send a IR string via the telnet session. Eg "execute teststring" Girder should now have learned the "teststring" into your command.

The next time you execute teststring girder should run the command you learned.

Mind! There is no echo of the text that you type, so maybe you can turn on the local echo function of your terminal emulator.

Silitek driver:

This is a driver for the silitek remotes, it was written by: Erik Damiano <damiano@cls.fr>

SFH-56 driver:

This is a driver for the SFH-56 (see the settings dialog of the plugin) written by Igor Cesko

It possible that this driver isn't included in the main distribution because of size constraints, you can find the driver in the download section of the Girder homepage.

IrDA driver: (NOT SUPPORTED, DO NOT CONTACT ME WITH QUESTIONS ABOUT THIS PLUGIN, this plugin has to be downloaded separately from the homepage)

Girder has been written with the Actisys IR device, so I don't know if any other devices work. Report please. Since the IrDA adapter wasn't built to support the remote control protocol the recognition might not be to good, you might even need 2 or more presses before girder recognizes a command. For the Actisys IR device a baud rate of 4800 baud seems to work o.k., the repeat tolerance should be something like 60 (out of 100) and the recognition tolerance should be something like 80 (out of 100). To get these thing to work you'll need to turn off the IrDA drivers of windows. Windows 95 should work fine, for Windows 98 you should check if the IrDA drivers where installed. You can find this out by checking the windows device manager for a Virtual comport. If that's in there you'll need to do the following:

In the file: C:\Windows\inf\Msports.inf add the following lines:

```
...
[Std]
%*PNP0510.DeviceDesc% = ComPort, *PNP0510           ; Generic IR device/Comm
...
[Strings]
```

*PNP0510.DeviceDesc = "Generic Ir Serial Port"

After you've added these two lines go to the Device manager and remove the device associated with the IrDA stuff:

- IrDa Network Adapter
- Virtual COM/PRT ports
- Infrared device

Reboot.

After the reboot windows will redetect the IrDA adapter but will install the drivers like Windows 95 did. And Girder should work.

If anyone wants to improve this driver please let me know, I haven't got the IrDA adapter anymore, so I can't work on the driver.

Task Switch detection.

This plugin allow you to switch groups on and off, depending on the active application.

Build you own plugin:

It reasonably easy to make your own plug-in. I've include the UIR/Irman and the Keyboard plug-ins in source form so you can start from there.

The UIR/Irman plug-in is polling based.

The Keyboard plug-in is callback based.

(Be sure to set the compiler option so that the functions are STDCALL's! and use a C/C++ compiler, Delphi just doesn't seem to work to good with dlls.)

If you think that the plug-in that you've written is useful to other people, please share your work and send the code/binary to me. Of course you will be added to the thanks section of the helpfile !!!!

Always check the homepage for the most uptodate information on plugins

[Back to index](#)



Software Plugins

APM:

And plug-in that sends a Advanced Power Management event to your computer, like suspend or hibernate.

I Was here:

This is a example plug-in it set the text of the controls that match the Target options to the value you specified.

TCP/IP

This plugin interacts with the TCP/IP server hardware plugin. You can make a Girder on one computer control a Girder on a totally different computer connected via the network. Make sure that you have Winsock2 (comes with Win98). Mind: this is the first release of this piece of code so it could be buggy, let me know !

Timer

This plugin waits for the amount of time that you specify and then executes the command that was chosen by you. You can see if the timer is running by the clock symbol in the traybar. The amount of time that girder waits is set by the number of times you press the button on the remote... not clear ? Just try it

Build Your Own:

The 'APM' and 'I was here' plug-in are supposed to be example plug-ins the source is included so look it up! Again be sure to set the compiler options to STDCALLs but you can use Delphi for these dlls. They are much simpler than their hardware counterparts.

If you think that the plug-in that you've written is useful to other people, please share your work and send the code/binary to me. Of course you will be added to the thanks section of the helpfile !!!!

Always check the homepage for the most uptodate information on plugins

[Back to index](#)

