



AppleScript Support for Growl



- [About](#)
- [Usage](#)
- [Basics](#)
- [Notifications Using Images](#)
- [Notes on File Paths](#)

About AppleScript Support

AppleScript support is built into Growl; nothing special needs to be done to enable it.

A Note on AppleScript changes in Growl 0.6

You must update your existing AppleScripts for Growl 0.6 In Growl 0.6 AppleScripts must register their notifications every time the script is run — unregistered notifications will *not* be displayed. See below for an example of registration.

Usage

Basics

To send a Growl notification via AppleScript, you talk to GrowlHelperApp using two commands: **register** and **notify**. A script (like an application) must register itself (once) with growl before sending any notifications. The full syntax of the **register** and **notify** commands can be viewed by opening Script Editor, choosing File → Open Dictionary, and selecting GrowlHelperApp from the list.

Sample AppleScript Notification

```
tell application "GrowlHelperApp"
    -- Make a list of all the notification types
    -- that this script will ever send:
    set the allNotificationsList to ~
        {"Test Notification" , "Another Test Notification"}

    -- Make a list of the notifications
    -- that will be enabled by default.
    -- Those not enabled by default can be enabled later
    -- in the 'Applications' tab of the growl prefpane.
    set the enabledNotificationsList to ~
        {"Test Notification"}

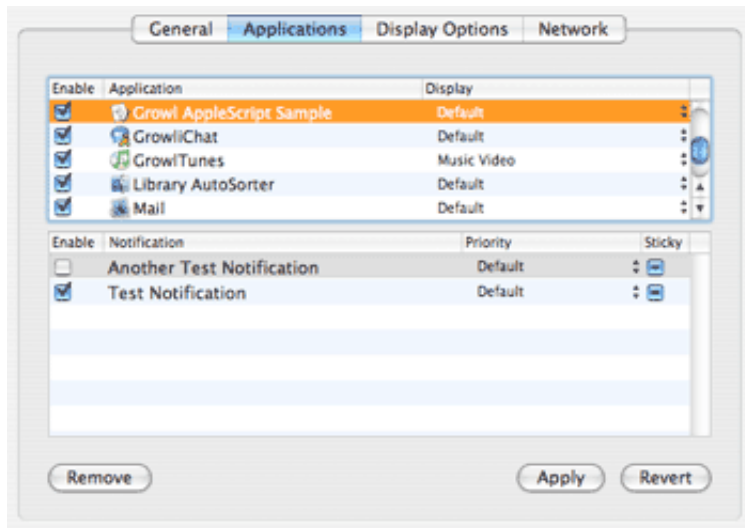
    -- Register our script with growl.
    -- You can optionally (as here) set a default icon
    -- for this script's notifications.
    register as application "Growl AppleScript Sample" ~
        all notifications allNotificationsList ~
        default notifications enabledNotificationsList ~
        icon of application "Script Editor"
```

```
--      Send a Notification...
notify with name "Test Notification" ~
    title "Test Notification" ~
    description "This is a test AppleScript notification." ~
    application name "Growl AppleScript Sample"

notify with name "Another Test Notification" ~
    title "Another Test Notification :)" ~
    description "Alas - you won't see me until you enable me..." ~
    application name "Growl AppleScript Sample"

end tell
```

Once you send the **Registration** command - your script will appear in the 'applications' tab of the Growl prefpane.



[Download Sample Script](#)

Notifications Using Images

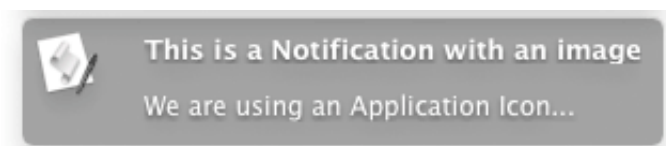
Growl's notify command supports four types of images for notification when using AppleScript:

- **Application Icons**

Example:

```
notify with ~
    name "Some Notification" ~
    title "This is a Notification with an App Icon" ~
    description "We are using an Application Icon..." ~
    application name "Growl AppleScript Sample" ~
    icon of application "Script Editor.app"
```

Note that the ".app" at the end is optional.

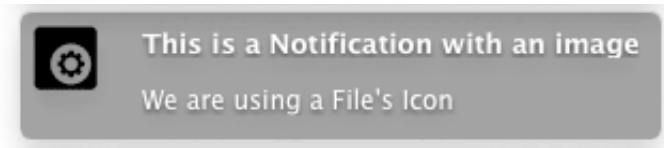


- **File Icons**

```
notify with ~
```

```
name "Some Notification" ↵
title "This is a Notification with an File Icon" ↵
description "We are using a File's Icon..." ↵
application name "Growl AppleScript Sample" ↵
icon of file "file:///Users/someone/Growl"
```

See [Notes on File Paths](#) for details on how to specify the path to a file icon.

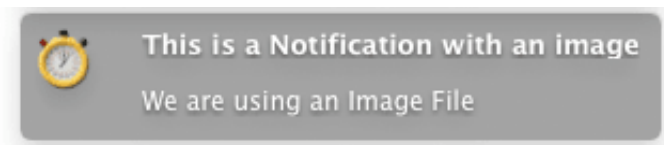


• Image Files

Supported Types: BMP, GIF, ICNS, ICO, JPEG, JPEG 2000, PNG, PSD, TGA, TIFF

Example:

```
notify with ↵
name "Some Notification" ↵
title "This is a Notification with an Image File" ↵
description "We are using an Image File..." ↵
application name "Growl AppleScript Sample" ↵
image from location ↵
    "file:///Users/someone/pictures/stopWatch.png"
```



• Image Data

Supported Types: PICT, TIFF

When you are dealing with raw image data you should use the **notify ... Image** or **notify ... pictImage** commands.

Examples:

Carbon apps tend to return PICT

```
set the PICTdata to my getArtworkFromiTunesRoutine()

notify with ↵
name "Some Notification" ↵
title "This is a Notification with PICT Image Data" ↵
description "We are using PICT data..." ↵
application name "Growl AppleScript Sample" ↵
PICTimage the PICTdata
```

Cocoa apps tend to return TIFF

```
set the TIFFdata to my getPhotoFromAddressBookRoutine()

notify with ↵
name "Some Notification" ↵
title "This is a Notification with TIFF Image Data" ↵
```

```
description "We are using TIFF data..." ↵
application name "Growl AppleScript Sample" ↵
image the TIFFdata
```

If you're not sure which type you are dealing with then you can look in Script Editor's *Event Log*. The first four chars of the data will show the type:

```
get data of artwork 1 of file track id 896 of library playlist id 854 of source id 37  
«data  
    PICT6B3C00000000012A012C001102FF0C00FFFE000000480000C  
    3000001C47C00000001000000000000000000000000000000
```

Notes on File Paths

For the "notify ... image from location" and "notify ... icon of file" commands Growl accepts any of the following types as 'locations':

- Aliases - the default file reference type in AppleScript.
- Local file:// URLs (as text) - e.g. "file:///Applications/" — *n.b. you must have three slashes after the colon — the third represents the root of the filesystem.*
- Paths - e.g. "~/Pictures"

Authors

v0.6 (update) — Diggory Laycock — www.monkeyfood.com

v0.5 — Adam Nolley — nanovivid.com