

Author: Jon Barnett
Date: 14th October 2003

Gnome 2.4 and file associations

I haven't been a frequent Gnome user. My last flirtation with it was in Red Hat 7.2. That was Gnome 1.4. While it operated well enough, it didn't seem to have the same responsiveness of KDE, occasionally crashed and the operation of the Nautilus file manager was a bit foreign.

With my recent plunge into Linux laptop computing with Slackware 9.1, I decided to give Gnome another try since it was now at version 2.4. It seemed simple enough and most programs installed well enough. I had my first file association problem with Real Player but this wasn't a major concern. I installed Open Office 1.1 and file associations worked perfectly.

It was then that I learned about Dropline Gnome, an optimized Gnome installation for Slackware. It boasted an optimized Gnome 2.4 system for 686-based computers. Since most graphical desktop environments are resource intensive, I thought it seemed prudent to get the best available performance out of my system, particularly as I want most of the CPU for compilation and other "real" development tasks. I'll do a review of Dropline Gnome next article.

Anyway, the upshot was that my Gnome system was upgraded, my personal settings were retained but my Open Office file associations were broken. I could no longer open documents in Nautilus with an Open Office application. The association editor with the distributions didn't solve the problem and would not accept the "Program to run" defined associations. Defining the association and then going back and clicking on the file would still result in a "no action associated" error message. Reviewing the definition would show that the "Program to run" had the full path removed.

Getting help

I had thought that this was not an atypical situation and that there would be help either in the electronic help documentation or on-line. Unfortunately, the help assumes that you are trying to perform the basic tasks and doesn't explain how the associations relate to tasks or how the tasks are defined. The "Edit File Types/File Types and Programs" management wasn't intuitive enough to allow me to understand how applications were defined and how the associations were managed. The Gnome site didn't help much either, possibly because I wasn't quite sure what I was asking. I also searched through Google and could only find vague references to the configuration files or to threads that ended with the suggestion that some of the Gnome configuration was a broken concept in the system.

Since I had switched my business system over to Open Office 1.1, these answers were really not satisfactory enough. I did a lot of investigation, using some of the references I did find, poking about the internals of the myriad of Gnome files and file locations. In some respects, it was easier than working out the JBoss application server configuration options. For a start, the configuration files in the area of file associations are not written in XML (eXtensible Markup Language) and you don't need to inspect a DTD (Document Type Definition) to work out the acceptable assembly syntax. In other respects, it is harder because the files are spread into several areas.

General organization

The Nautilus configurations for file associations are split into two categories. The first is the global definition for all users. The second is the local definition for a user. Both definitions make use of three files that manage the associations and actions.

For those familiar with web servers and their operation, you will know that a web server defines the content type for a file based on the MIME types definition file. This defines the mapping between a file's extension and the defined MIME type. MIME (Multi-purpose Internet Mail Exchange) and MIME types were originally defined for the purpose of shipping binary information compliant with SMTP-based mail content delivery, since the SMTP standard only specified delivery of text-based information. The MIME encoding and MIME type would allow the receiver to re-assemble and view the binary contained in the e-mail body.

However, the MIME type mapping, because of the thoroughness of the content definition including vendor hierarchies, has been widely adopted as the standard for content identification and its use has been extended beyond the original Internet Mail related use. Nautilus uses MIME type definitions to identify file content.

The second file associates content type with an application. It allows multiple applications to be associated with a particular type of content. This completes the requirements for Nautilus to determine the applications available for use when viewing or running a particular file.

There is a third file that provides some hierarchical and descriptive information about the identified MIME type for a file as well as readily available actions. This is of some importance in terms of associating files with default actions in the Nautilus file browser.

Global definitions

For most commercial Gnome installations, the global definitions exist in the /usr/share directories. This makes it easy for custom Gnome installations to easily replace existing installations. However, as was the case with my upgrade from Slackware's native Gnome installation to the Dropline system, this can have other consequences such as destroying previously working file associations.

The MIME mappings for files exist in /usr/share/mime-info. There are the actual MIME definitions and these are stored in files with a file extension of .mime. Most standard Nautilus and Gnome definitions are stored in gnome-vfs.mime. VFS is the Virtual File System and associated API that Gnome provides for file system operations.

It is similar to Midnight Commander's VFS but designed from scratch to be extensible and useable by any application.

Typical content would be:

```
image/tiff
    ext: tif tiff
```

This defines that files with an extension .tif or .tiff are most likely to contain information described by image/tiff.

The corresponding .keys file, gnome-vfs.keys contains some information for the same entry.

```
image/tiff
    description=TIFF image
    default_action_type=component
    short_list_component_ids=OAFIID:GNOME_EOG_Control
    short_list_application_ids_for_novice_user_level=eog,gimp
    category=Images
```

The real file contains more information than I've provided here as there are many MIME definitions, and there are also categories and descriptions available for most languages. You can read the file for yourself as the format itself is very simple. This makes it easy to understand without requiring another document that explicitly describes the layout.

For ease of administration, you can extract definitions or create new definitions in separate files as long as they have an extension describing whether they are keys or MIME information. For example, the lists for the Totem multimedia player supplied with Dropline Gnome are kept in a separate file called /usr/share/mime-info/totem.keys. The actual MIME types are already defined in gnome-vfs.mime so Totem has no separate .mime file for this.

The application registry for applications and the content with which these applications are associated is located in the directory /usr/share/application-registry. The files have a .applications extension. While most of the Nautilus definitions are contained in one file, you can also create associations in a separate file for any applications you introduce into your Gnome environment. A registry entry for an application defines the application reference as used in the .keys directory, the execution command and related environment, and the MIME types with which the application is associated.

A definition from `gnome-vfs.applications` that links with the previous example is given below.

```
eog
    command=/usr/bin/eog
    name=Eye of Gnome
    can_open_multiple_files=true
    expects_uris=true
    uses_gnomevfs=true
    requires_terminal=false
    mime_types=image/bmp,image/gif,image/jpeg,image/png,image/tif
f,image/x-xpixmap,image/x-bmp,image/x-png,image/x-portable-
anymap,image/x-portable-bitmap,image/x-portable-graymap,image/x-
portable-pixmap
```

These files all have very simple formats and you can edit them with `vi` or a similar editor. Since these relate to global definitions for an installation of Gnome on the system, it requires root access to edit and save these files. It also means that you cannot use the Gnome configuration managers in an individual's Gnome session to modify these settings.

The application definitions and keys also prove to be very important because they are used in the “File Types and Programs” management in Advanced Desktop Preferences for Gnome. The “Program to Run” option does not work well as it does not preserve the full path, at least for Dropline Gnome and the standard Slackware bundle. By defining global information for your applications and file associations, it makes it easier for the user to customize their local preferences from the drop down selections.

Local definitions

For each user, the corresponding local definitions provide customizations based on the global options available. The local definitions can also override the global definitions.

When you install a local copy of Open Office, it creates a `user.keys` file and a `user.mime` file in `~/gnome/mime-info` if these files do not already exist. These files provide much the same functionality as the global definitions.

An example of the MIME definitions in `user.mime` is:

```
application/vnd.sun.xml.writer
    ext: sxw
    description: OpenOffice.org Text Document
    deleted:
```

The corresponding entry in the `user.keys` defines a custom description, the linked application and can even contain a custom icon reference for display in Nautilus.

The user.keys file entry is:

```
application/vnd.sun.xml.writer
    default_application_id=OpenOffice.org11
    description=OpenOffice.org Text Document
    icon_filename=/opt/share/document-icons/002_text_document.png
    default_action_type=application
    short_list_application_user_removals=
    short_list_application_user_additions=OpenOffice.org11
```

The application customizations are contained in ~/.gnome/application-info. These customizations usually consist of additional content associations. For example:

```
xine
    mime_types=video/x-ms-asf,video/mpeg,audio/x-pn-realaudio
    mime_types=video/x-ms-asf,video/mpeg,audio/x-pn-realaudio
```

These can all be defined using the “File Types and Programs” applet of Gnome. However, if you want to add a local application, you may need to hand code it to define the full path to the application.

Such a definition would look like this:

```
gmpayer
    requires_terminal=false
    command=/usr/local/bin/gmpayer
    can_open_multiple_files=false
    mime_types=audio/x-pn-realaudio
    name=gmpayer
    mime_types=audio/x-pn-realaudio
```

With the applet, you can easily customize associations to an application that is defined in the global .applications file.

Solving the Open Office problem

The Dropline Gnome installation wrote over existing global definitions and it appears they did not account for a separate installation of Open Office 1.1. This left me with Microsoft Word documents associated with AbiWord and no associations that functioned for any of the Open Office supported documents. In order to fix the problem, I edited the global application registry, /usr/share/application-registry/gnome-vfs.applications.

The entry, was modified to refer to the core application for the Open Office 1.1 network installation, located in /opt/OpenOffice.org1.1.0/program. The reference key was also re-labelled openoffice.

openoffice

```
command=/opt/OpenOffice.org1.1.0/program/soffice
name=OpenOffice 1.1
can_open_multiple_files=true
expects_uris=false
requires_terminal=false

mime_types=text/abiword,application/x-abiword,application/vnd.ms-excel,application/rtf,application/msword,application/vnd.ms-powerpoint,application/vnd.stardivision.calc,application/vnd.stardivision.chart,application/vnd.stardivision.draw,application/vnd.stardivision.impress,application/vnd.stardivision.mail,application/vnd.stardivision.math,application/vnd.stardivision.writer,application/vnd.sun.xml.calc,application/vnd.sun.xml.calc.template,application/vnd.sun.xml.draw,application/vnd.sun.xml.draw.template,application/vnd.sun.xml.impress,application/vnd.sun.xml.impress.template,application/vnd.sun.xml.math,application/vnd.sun.xml.writer,application/vnd.sun.xml.writer.global,application/vnd.sun.xml.writer.template,image/x-emf,image/x-pcx,image/x-photo-cd,image/x-pict,text/html
```

Note the MIME types that are related to the Open Office suite. Also take note of the fact that the one run-time instance can handle multiple content files at once and that there are many different types of files it can handle.

All appropriate entries in /usr/share/mime-info/gnome-vfs.keys were then confirmed to refer to the key, 'openoffice' from gnome-vfs.applications. For example,

```
application/msword
    description=Microsoft Word document
    default_action_type=application
    short_list_application_ids_for_novice_user_level=abiword,openoffice,staroffice,applix
    short_list_application_ids_for_intermediate_user_level=abiword,openoffice,staroffice,applix
    short_list_application_ids_for_advanced_user_level=abiword,openoffice,staroffice,applix
    category=Documents/Word Processor
```

You could also tidy up your local keys, MIME types and applications definitions, to ensure structural integrity. However, unlike Windows, it is improbable that getting things wrong in these files will destroy Gnome and prevent you from correctly starting it up.

You might still have problems with Excel spreadsheet associations. Steve Roper of Open Europe suggested the following fix:

```
mv /usr/share/mime-info/gnumeric.mime /usr/share
mv /usr/share/mime-info/gnumeric.keys /usr/share
```

This removes the gnumeric mappings, leaving sole control to Open Office.

Final notes

The actual operation of the mapping is self-evident once you can establish the locations of the files and their relevance. The examples in a working Gnome installation such Dropline should provide enough information on possible modes of operation. Getting to that point was the main obstacle.

Armed with this knowledge you should be able to repair any breakages, or create your own program definitions and file associations. In terms of organizing your configurations, remember to balance the modularity of splitting configurations into separate files with the performance impact of reading multiple files when configurations are updated or during the Gnome boot process.

This guide provides a pragmatic approach to fixing any problems with file associations with Gnome 2.4. While there may exist more elegant solutions and explanations, this gives a starting point for understanding what is happening in your system.