

pgftp is a pGina plugin which allows users to have roaming profiles stored on an FTP server. Although pGina itself contains roaming profile functionality, it is somewhat limited since it requires that the account used to login has access to the profile server. In many configurations of pGina, the authentication is done via a plugin (e.g. LDAP) and the user logs in using a temporary local account on the local computer. Depending on your environment, it can be difficult or impossible to get your profile from the server using this account.

In addition to bypassing these issues, the pgftp plugin gives you more flexibility by allowing roaming over any TCP/IP connection (including the Internet). Also, by compressing the profile, pgftp can reduce central profile size and provide performance increases over slow connections.

## 1. Requirements

=====

- Windows 2000 or XP client machines with pGina version 1.7.7.4 or higher (NT 4.0 may work but it is completely untested). Clients MUST have at least one NTFS partition, the plugin unfortunately won't work for FAT32-only machines (see Known Issues below)
- pGina Chaining Plugin (pgftp alone doesn't authenticate users)
- pGina Authentication Plugin of some sort

## 2. Compiling

=====

If you obtained pgftp.dll as part of a binary distribution you can skip to the next section.

To compile pgftp, you will need:

- Microsoft Visual C++ 6.0 (or later) or the Microsoft C++ command line compiler / linker (cl.exe)
- The latest version of the Platform SDK

The Platform SDK is available for download from <http://www.microsoft.com/msdownload/platformsdk/sdkupdate/> (only the Core SDK is required). If you don't have a copy of Visual Studio, you can still obtain the command line compiler/linker cl.exe for free as part of the Platform SDK. To use it, make sure you register environment variables when the Platform SDK setup prompts you to.

IMPORTANT: If you are using Visual Studio, make sure that your library and include paths include the Platform SDK directories BEFORE the Visual Studio ones. Otherwise, you will probably receive errors during compilation and linking due to out-of-date Win32 API libraries in the Visual Studio directory. You can set these options in the Visual C++ GUI under Tools -> Options -> (Directories tab) or by changing the values of the \$INCLUDE and \$LIB environment variables.

You can compile pgftp by loading pgftp.dsw in Visual C++ or on the command line by typing "nmake" while in the pgftp source directory. Once the compilation is finished you will have <target>\pgftp.dll where

<target> is either Debug or Unicode Release.

### 3. Installation

=====

3.1 Install PGina as per the pGina instructions.

3.2 Install the Chaining plugin.

3.3 Install an authentication plugin.

3.3 Install the pgftp plugin. The recommended location is

C:\PGina\plugins\pgftp. The installation will place pgftp and 7za.exe in this directory. Due to unicode filename incompatibility with regular zip format, pgftp uses the unicode compatible archiving tool (7za.exe) from 7-zip ([www.7-zip.org](http://www.7-zip.org)).

### 4. Configuring

=====

4.1 Setup an FTP Server to act as your profile repository. The ftp server will need to give incoming users access to write and read their own profile. Profiles can either be stored within a user's file space if the user account has access to the ftp server, or all profiles can be stored in one location. Pgftp does not require a particular configuration, except when using a default profile which only requires that each user has read access to the default profile.

4.2 Setup pGina to use the chaining plugin.

4.3 Setup the Chaining Plugin to include your desired authentication plugin(s).

4.4 Setup the Chaining Plugin to include the pgftp.dll plugin. Most likely you will want pgftp to be last in the chain, after any authentication plugins.

4.5 Open the pGina FTP Profile Plugin configuration dialog by selecting pgftp.dll in the Plugin List and then clicking the Configure Plugin button on the Chaining Plugin Configuration dialog.

4.6 Fill in the fields according to the following:

- Server: enter the name of the FTP server used for storing profiles. Type the host name or IP address with no ftp:// prefix. (eg ftp.myhost.com)

- Temp Profile Path: enter the path to where the profile will be downloaded to be decompressed. This path must reside on an ntfs partition, and be accessible by pgina (eg C:\PGina\plugins\pgftp\profiletemp). Pgftp will create this directory as required.

- Username: Select "use login name" for pgftp to use the username of the user logging in. If all connections to the server are to be made using a particular account, select "Specify" and enter the account name.

- Password: Select "Use login password" for pgftp to use the password of the user logging in. If all connections to the server are to be made using a particular account, select "Specify" and enter the account's password.

- Compression: To enable compression, select "Enable profile compression" and set the compression option. When enabled, pgftp will compress the profile before uploading to the server which is useful for fast cpus on slower networks. If you do enable compression, you will likely see the best performance overall (compression and transferring) at the "Fastest" setting.

4.7 Set the profile path and (optionally) the default profile path in the profile tab of the pGina configuration tool. Normally these paths are UNC paths to the profile, however with pgftp they are used as FTP paths relative to the FTP root, so the paths should be Unix-style paths.

The profile path is a path to each individual user's profile. Since all users of the same machine use the same pGina configuration, you will need to somehow ensure that different users get different profiles. The easiest way to do this is to configure the FTP root to be the user's home directory. The profile path should then be relative to each user's home directory (eg for "Profile Path" enter `.windows_settings` to store the user profile in `/home/username/.windows_settings`). Another option is to use the `%USERNAME%` variable in the path, which will be expanded to the pGina login name (not the FTP login name, if they are different).

The default profile path is a path to a single, central profile that will be assigned to any new user who logs in for the first time. If not specified, the local Default User profile is used as normal. This option is useful if you want to have a customized default user profile maintained in a central place. (eg for "Default Profile" enter `/opt/skel/windows_profile`)

## 5. Profiles

=====

In Windows there are (at least for our purposes) 4 definitions of profiles:

- roaming profile: The profile type that is uploaded/downloaded to/from a server.
- local profile: A profile for a local user, or the copy of the roaming profile when it has been downloaded
- default roaming profile: A generic profile that is used in the case that windows cannot find the copy of the roaming profile that stored on the server.
- local default profile: the default profile of the computer which will be copied to the users local profile if pgftp cannot find either the roaming or default profile on the server.

!!NOTE!! Installations that require unicode filename compatibility will need to compress the profile into `profile.7z` (instead of `profile.zip`) using the 7-zip application at [www.7-zip.org](http://www.7-zip.org). The regular zip function is not unicode filename compatible !!NOTE!!

### 5.1 Default profile.

In order to have account profiles roam automatically (ie without first configuring a `profile.zip` file for each user), you will need a default profile. The steps to create a default profile follow:

- 5.1.1 Create an empty directory somewhere on your filesystem. This will be used to store a copy of the profile.
- 5.1.2 Create a new local user on a windows system.
- 5.1.3 Log on as the new user, and make any desired profile changes.
- 5.1.4 Log out from that user and log back in as administrator.
- 5.1.5 In the Windows (XP) System Properties, select the Advanced tab, and click the Settings button.

5.1.6 Select the profile of your new user, and click "Copy To"  
5.1.7 Select the empty directory from above, click OK  
5.1.8 Click the Change button in the "Permitted to use" frame.  
5.1.9 Add "Everyone" to the list, Click OK  
5.1.10 Click OK on the Copy To dialog to close it  
5.1.11 Open the newly filled profile directory in explorer  
5.1.12 Add the contents of the directory to a zip file called profile.zip (get hidden files too!).  
5.1.13 copy this zip file to the default profile location on the ftp server. Ensure that all users have read only access to this file and directory.

## 5.2 Migrating Existing Profiles

5.2.1 Create an empty directory somewhere on your filesystem. This will be used to store a copy of the profile.  
5.2.2 In the Windows (XP) System Properties, select the Advanced tab, and click the Settings button.  
5.2.3 Select the profile of your user, and click "Copy To"  
5.2.4 Select the empty directory from above, click OK  
5.2.5 Click the Change button in the "Permitted to use" frame.  
5.2.6 Add "Everyone" to the list, Click OK  
5.2.7 Click OK on the Copy To dialog to close it  
5.2.8 Open the newly filled profile directory in explorer  
5.2.9 Add the contents of the directory to a zip file called profile.zip  
5.2.10 copy this zip file to the profile location for the account on the ftp server. Ensure that the account has read/write access to this file and directory.

## 6. How It Works

=====

6.1 Connects to the ftp Server (defined in pgftp)  
6.2 Looks for profile.7z in the Profile Path (defined in pGina)  
6.3 Failing that, it looks for profile.zip in the Profile Path  
6.4 Failing that, it looks for profile.7z in the Default Profile Path (defined in pGina)  
6.5 Failing that, it looks for profile.zip in the Default Profile Path  
6.6 If it found a profile, it will download the file to the Temp Profile Path (defined in pgftp)  
6.7 The profile will be decompressed in the Temp Profile Path  
6.8 Control is returned to windows, with a pointer to the profile in the Temp Profile Path

## 7. Known Issues

=====

- Windows cannot accept a profile from an insecure location, such as a fat32 file system which has no security attributes. Thus pgftp must download the profile to an ntfs partition before it hands the profile off to windows .
- Installations that require unicode compatibility will need to compress the profile into profile.7z using the 7-zip application at [www.7-zip.org](http://www.7-zip.org). The regular zip function is not unicode compatible.

## 8. Possible Future Additions

=====

- scp/sftp
- authentication based on ssh success/failure
- gui indicator for profile storage/retrieval progress
- biological storage to store your profile within the unused portion of your DNA
- quantum mechanical profile storage to enable instantaneous profile storage/retrieval across any distance

Contributions anyone?

## 9. Known Bugs

=====

- Profile permissions are set world readable because on each login, an account can get a new SID.
- The roaming profile mechanism needs to first check for the existence of a local copy of the roaming profile before getting the default profile