

RESguru.com's Quick-Reference to Variables, Tags and Functions in RES PowerFuse

This document describes the various environment variables, specific tags and special functions at your disposal within RES PowerFuse. All the information provided below is available through the regular RES PowerFuse documentation, but has been combined here for quick reference and easy overview. At the end of the document, a detailed reference is provided for some of the described items.

Note: From PowerFuse 2008 SR3, the use of environment variables in registry key names, value names and profile folder and file names, are supported. For example, "HKEY_CURRENT_USER\Software\Microsoft\Office\%OFFICE_VERSION%". When environment variables are used, they will be dynamically processed both while saving and loading the User Preferences. This enables an administrator to load User Preferences to a location which is different from the location from which they were saved.

Variables

`%deskpic%`

Use this variable to dynamically specify a desktop logo or wallpaper for given ou's, groups, etc of your choice.

`%clientipaddress%`

Returns the client IP address from where the session originated (if applicable). This variable is set by RES PowerFuse.

`%fullusername%`

This variable returns the full displayname of the currently logged in user in a PowerFuse session. Usefull for setting up usernames in Office and such.

`%lcid%`

In multi-language PowerFuse environment, the LCID variable will contain the numerical Windows language-country ID, for the language selected by the user in the PowerPanel. For example, US English will yield the value 1033

`%password%`

Contains the cached copy of the users logon password. Usefull to passing on to single-signon appliations.

`%publishedappname%`

This variable can be used when editing the template ica file for instant passthrough in PowerFuse. Use this predefined variable to insert the name(s) of the actual published application(s) that should be launched by RES PowerFuse. You can use other environment variables too.

`%rescustomresources%`

Contains the path to where custom resources are stored on the PowerFuse enabled client. Typically this path will point to C:\Program Files\RES PowerFuse\Data\DBCACHE\Resources\custom_resources

`%reshomedrive%`

As default users will have H: as their homedrive. Set this variable through PowerLaunch to override the home directory for ou's, groups, etc as you choose.

`%saverpic%`

Use this variable to dynamically specify a picture for use in the default RES static screensaver.

`%sgdrive%`

This variable points to the SoftGrid mount-drive. The default is Q, but this variable can be used to override the default on a given machine. Tested on SR1 but doesnt seem to contain anything?

Tags:

`PF_IFA`

Specifying this tag in a IFA commandline, inserts the doubleclicked file/mimetype

`$PF_IFA_REPLACE$`

If this tag is used, the original main commandline from the app is not used. Use to specify an alternate app to launch the file extension.

`$PASSWORD$`

Does the same as `%password%` but is used for subscribed applications (where is it specified, in pf console or subscriber??)

Functions

`$adinfo()`

The `adinfo` function can return any AD attribute for the user. See examples in further ahead. Also check <http://msdn2.microsoft.com/en-us/library/ms683980.aspx> for additional attributes which can be specified

`$autocount()`

Generates a unique value when called. Syntax is: `$autocount(<minimum value>-<maximum value>)`. Se details below for more information.

`$substring()`

The `substring()` function can be used in the value to obtain a specified number of characters from an environment variable: Syntax: `$substring(%var%, start, length)`.

Example: `$substring(%username%, 3, 4)` with `%username%="Robert"` returns "bert".

This can be used on both PowerFuse generated and system variables. See reference for these below

`$lowercase()`

`$uppercase()`

Use `$lowercase(<variable name>)` and `$uppercase(<variable name>)` to retrieve and convert values from other environment variables. If an existing environment variable `%username%` has the value "testuser", then `$uppercase(%username%)` would return the string "TESTUSER". Quotation marks are only used to exemplify. Do not use.

Detailed References

The following section provides more in-depth information for certain items described above. All information provided here is available through the RES PowerFuse documentation.

\$PF_IFA\$ + \$PF_IFA_REPLACE\$

Configuring special File Association commandlines for an Application

Use the File Associations tab in the Edit application window > Configuration section to associate Windows file types and commands with an application.

How to configure File Associations for an application

Click the File Association tab. To add a File Association to the application, click Add. This will open the Add/Change Instant File Association window.

To change existing Instant File Associations, click Edit or Delete.

If more applications are associated with the same file types and commands, click the arrows to assign a priority. For example, WordPad can open simple .doc files, but Microsoft Word also supports this file type. If an application is at the top of the priority list for a certain file type and command, it will be used to handle the file type association. If this application is not accessible to the user, the second application will be used, etc. You can also edit this priority in the Instant File Associations node of Configuration Management.

Notes

You can use "special file types" when configuring Instant File Associations. These file types (mailto, http, ftp, nntp and news) are not really file types, but determine what should happen, for example, when the user clicks a mail link in Internet Explorer or an Internet URL in Microsoft Outlook.

You can indicate where the file name should be inserted in the command line when command line parameters are used, by using \$PF_IFA\$ in the application's command line.

For example, if you have configured Microsoft PowerPoint's command line as /splash to suppress its splash screen and the user double-clicks a .ppt file (assuming you configured an Instant File Association for this application), PowerPoint will attempt to start with the command line /splash h:\<file name>.ppt. This will fail, because PowerPoint requires the file name as first parameter.

To resolve this, enter \$PF_IFA\$ /splash as the application's command line option. If the application is launched through an Instant File Association, the file name will no longer be appended to the application's command line options, but instead will be inserted in the position indicated with \$PF_IFA\$. When the application is launched directly from the menu, the \$PF_IFA\$ text will be removed.

You can also use \$PF_IFA_REPLACE\$ to indicate that when an application is launched through an Instant File Association, the command line options that are configured for the application should not be used at all, but only the file name that is passed to the application through the Instant File Association.

Adinfo()

Extracting Active Directory and NDS attributes for current user

Use the `$adinfo()` function to obtain specific information about the current user that is stored in Active Directory. You can use this function to obtain any Active Directory property. The table below lists the most frequently used attributes. You can find more examples on e.g. Microsoft MSDN here: <http://msdn2.microsoft.com/en-us/library/ms683980.aspx>

Syntax: `$adinfo(<property>)`. Below are examples:

`$adinfo(ouname)`

Returns the name of the OU the current user directly belongs to.

`$adinfo(firstname), $adinfo(lastname)`

Returns the user's first and last name.

`$adinfo(fullname)`

Returns the user's full name.

`$adinfo(displayname)`

Returns the user's Windows display name.

`$adinfo(division)`

Returns the user's division.

`$adinfo(email)`

Returns the user's e-mail address.

`$adinfo(manager)`

Returns the name of the user's manager.

`$adinfo(title), $adinfo(office), $adinfo(department), $adinfo(homephone), $adinfo(fax)`

Returns these user properties from AD/NDS.

`$adinfo(employeeid)`

Returns the user's employee ID.

`$adinfo(homedirectory)`

Returns the user's home directory.

`$adinfo(homedirdrive)`

Returns the drive that contains the user's home directory.

`$adinfo(tshomedirectory)`

Returns the user's home directory on the Terminal Server.

`$adinfo(company)`

Returns the user's company name.

`$adinfo(street)`

Returns the user's street name.

`$adinfo(pobox)`

Returns the user's P.O. box number.

`$adinfo(city)`
Returns the user's city.

`$adinfo(state)`
Returns the user's state.

`$adinfo(postalcode)`
Returns the user's postal code.

`$adinfo(country)`
Returns the user's country.

`$adinfo(telephonehome)`
Returns the user's home telephone number

`$adinfo(telephonemobile)`
Returns the user's mobile telephone number

`$adinfo(telephonepager)`
Returns the user's pager telephone number

`$adinfo(webaddress)`
Returns the user's web address.

Notes

Some AD properties such as "homedirdrive", "division", "employeeid", "fullname", "faxnumber", "manager", "company", "country", "telephonehome", "telephonemobile", "telephonepager" and "webaddress" do not exist in NDS. They will return an empty string if used.

The share name of the drive mapping requires the presence of two backslashes, which are not present in the Ainfo results. You therefore need to use the Substring function that deletes the slashes that are added by the Drive & Port Mapping.

When a (Microsoft AD or Novell DS) property is not available in RES PowerFuse, you can add it in PowerLaunch: Click Add in PowerLaunch > Environment Variables and add the variable "CacheADProperties". In the Value field, enter one or more properties, separated by a semicolon (e.g. "Homepage;OtherName").

Autocount()

Generate a unique number

Use this function to specify a value for an environment variable in PowerLaunch, e.g. if an application needs a unique identifier in its .ini file for all current sessions. If you use the name of the environment variable in the .ini file, PowerLaunch will copy this file from the model directory.

Syntax: `$autocount(<minimum value>-<maximum value>)`

Example: `$autocount(1-10000)` would generate a value that counts from 1 to 10000 for every time the variable is set by PowerLaunch. When 10000 is reached, the counter will restart at one. A new counter will be used for every uniquely named variable.

Notes

For complete random numbers, one can use the os-generated `%random%` variable, available on Windows XP and Windows server 2003 and up.

System Environment variables

A complete reference

Below is an overview of available variables on Windows systems. These can be used directly in PowerFuse or in powerfuse functions, such as `$substring()`, `$lowercase()`, etc.

`%ALLUSERSPROFILE%`

Returns the path to the shared system profile.

`%APPDATA%`

Returns the path to the user's application data folder.

`%CLIENTNAME%`

Name of the Terminal Server client desktop.

`%COMMONPROGRAMFILES%`

Returns the path to the shared programs folder.

`%COMPUTERNAME%`

Returns the name of this computer.

`%HOMEDRIVE%`

Returns the drive letter of the user's home directory (if set).

`%HOMEPATH%`

Returns the path to the user's home directory (if set).

`%LOGONSERVER%`

Returns the UNC name of the network logon server.

`%PROGRAMFILES%`

Returns the UNC name of the network logon server.

`%RANDOM%`

Returns a random number between 0 and 32768

`%SESSIONNAME%`

Returns the name of this Terminal Server session.

`%SYSTEMDRIVE%`

Returns drive letter of the Windows system drive.

`%SYSTEMROOT%`

Returns complete path to the Windows system folder.

`%TEMP%`

Returns path to the user's temporary folder.

`%USERDNSDOMAIN%`

Returns DNS suffix used by this host.

`%USERDOMAIN%`

Returns name of the Windows domain.

`%USERNAME%`

Returns logon name of the current user.

`%USERPROFILE%`

Returns the path to the user's programs folder.

`%WINDIR%`

Returns complete path to the Windows system folder. Identical to `%SYSTEMROOT%`.

The `%LCID%` variable

Using RES PowerFuse in Multilingual Organizations

Use the node RES PowerFuse Setup > Settings > Multi Language to enable multilingual support in your RES PowerFuse environment. This allows you to change the language used in the RES PowerFuse Management Console, but also to provide multilingual support to your users: they can select different languages for their PowerPanel and Start Menu. Multilingual support also allows you to configure applications (such as Microsoft Office) to use the preferred language of the user.

Multilingual support in RES PowerFuse is primarily based on LanguagePacks. By installing a LanguagePack, you can provide support in RES PowerFuse for the following languages: Danish, Dutch, English, Spanish, Finnish, French, German, Italian, Norwegian and Swedish. By mapping languages to a LanguagePack, you can present the end user with a choice of additional languages to those available in LanguagePacks.

How to enable multilingual support in RES PowerFuse

- Download the LanguagePack from the RES Portal, at <http://portal.ressoftware.com>
- Install the LanguagePack on all RES PowerFuse Agents.
- Select Enable mapping of language to LanguagePack. This will show a list of available languages.
- Select a LanguagePack in the Installed LanguagePacks area.
- Select the language(s) that you want to map to this LanguagePack. The LCID code represents the identifier of the language in Microsoft Windows. All selected languages (LCID's) will be available for the user.

- If applicable, change the MLS initial default language.
- To view an audit trail of changes, click .
- To create a Building Block of the settings, click .
- To create an Instant Report of the settings, click .
- Click Apply to apply the changes.

Example

If you highlight LanguagePack English and select the language Norwegian (Nynorsk), the variable %LCID% will contain the Windows identifier of Norwegian (Nynorsk). You can then use this variable to make changes in the registry on language level using PowerLaunch. For example, if you change the registry using this variable, you can change the display language of Microsoft Office to Norwegian (Nynorsk).

After you have mapped Norwegian (Nynorsk), the user can select Norwegian (Nynorsk) in his PowerPanel. After refreshing PowerMenu, RES PowerFuse will show an English interface, whereas Microsoft Office will show the Norwegian (Nynorsk) interface.

Notes

When enabling or disabling language mappings to a LanguagePack, the current language selection of the user in RES PowerFuse will be reset to the MLS initial default language.

When configuring registry settings in PowerLaunch at language level, you can select the option Any language in the Language field. Language registry settings will be enforced when the RES PowerFuse session is refreshed. All other registry settings will only be enforced when the user logs on. By selecting this option, you can enforce registry settings that are not related to a specific language or are based on the %LCID% variable during a refresh.

Custom Resources

What are they and how to manage them

Custom Resources are resources that are referenced by other parts of RES PowerFuse, but which cannot easily be placed on a centrally available fileserver or which must also be available on a computer when it is offline. For example, these can be image files used in Instant Mail Outlook signatures, scripts used in PowerLaunch External tasks, etc.

Custom Resources are locally cached on all computers running RES PowerFuse. When referencing these resources, use the `%rescustomresources%` variable, which will be replaced by RES PowerFuse to point to the actual location of the file. The `%rescustomresources%` variable will always point to the root of the folder structure.

Use the Custom Resources node in the RES PowerFuse Setup > Datastore node to manage custom resources in the datastore.

Example

You have added an image file named corpimage.gif to the root of your Custom Resources (e.g. you have not created a folder structure for your custom resources). To use this image in you corporate e-mail signature, refer to this image as %rescustomresources%\corpimage.gif.

How to configure custom resources

- To add a custom resource, click Add. This will open the Browse for File or Folder window.
- To change existing files or folders, click Edit, Delete or Rename.
- Select a file or folder and click OK. This will show the selected files and folders in the Custom Resources list.
- Click New Folder to organize custom resources in folders.