UserEnvManager Manual

1. What it is

This software enables administrators to maintain shortcuts, network drive mappings and network printers on Windows computers (client computers or terminal servers) of a centrally managed network without writing complicated login scripts. One of the goals thereby is to separate any program logic from the descriptive configuration data which is included in XLM files. This makes it possible to generate or maintain those files by the included frontend or any other corporate applications.

Items can be assigned depending on the following properties:

- user name
- user groups the user belongs to
- the organizational unit (Active Directory) the user belongs to
- the computer's name
- the computer's IP address

For all these properties wildcards (? and *) are also supported.

The application is also terminal server and Citrix XenApp aware, meaning if it is running in the context of a terminal session the real client's computer name and IP address are evaluated and not the one of the terminal server. For this functionality you have to register the ActiveX control called SystemInformation.dll which is part of the package.

Shortcut assignments the software makes are state aware, meaning if you have assigned a shortcut to a group of users and one user is no longer member of that group that shortcut is removed from the user's desktop or start menu.

2. Prerequisites

Both the core component and the frontend are written in Jscript and therefore require an operating system with Windows Scripting Host installed. Windows 2000, Windows XP, Windows Server 2003, Windows Vista and Windows Server 2008 meet this requirement by default.

3. How it works

For getting the software up and running you have to:

- Copy the UserEnvManager.js file to each client computer / terminal server or put it at a central file share.

- Copy the SystemInformation.dll to each client computer / terminal server and register it with regsvr32.exe if you want to use IP address evaluation or client machine name evaluation which is also terminal server aware.

- If you want to use the web frontend for modifying the configuration files, simply extract the folder *frontend* anywhere you like and make the <*dataPath*> tag within the *settings.xml* file point to the root folder where you want to place the UserEnvManager configuration files.

- Change the element <*settings*><*general*><*localeId*> within the *settings.xml* file if you want to use a different language for the frontend. Currently *en* (English) and *de* (German) are supported. If you are eager you can create your own localization by adding an xml file which contains translations to the locale subfolder.

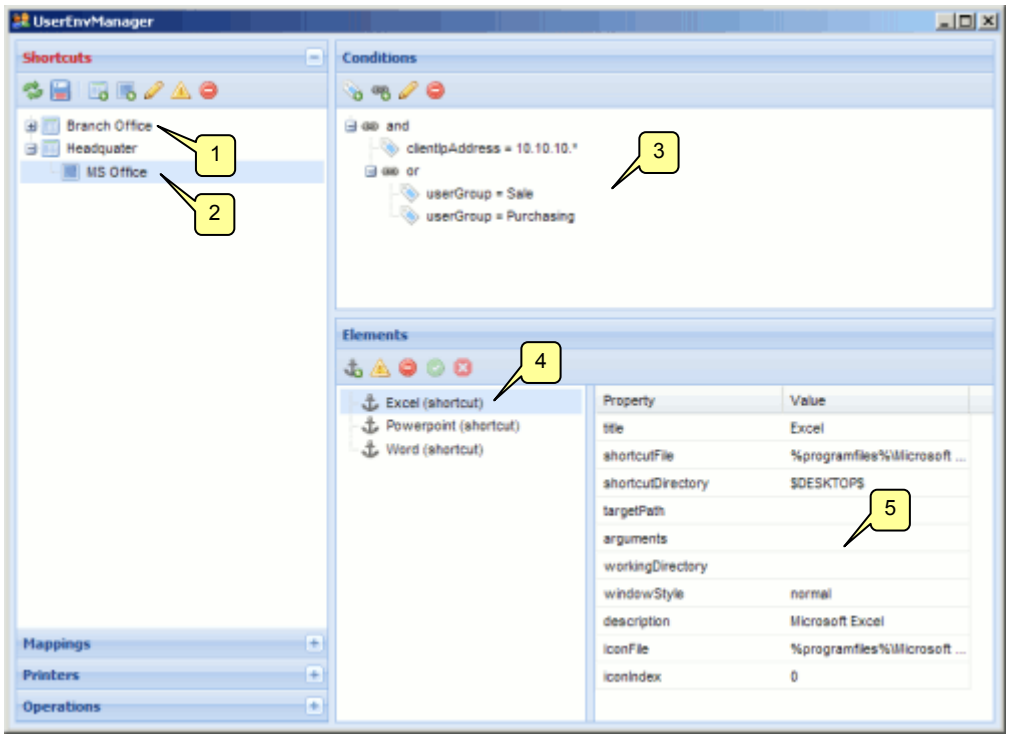
- Make the desired assignments within the UserEnvManager configuration files *mappings.xml*, *printers.xml*, *shortcuts.xml*, *operations.xml* directly with an editor or by the included frontend and put them on a central file share. Alternatively you could spread your assignments across multiple xml files and put them in subfolders named *mappings*, *printers*, *shortcuts* and *operations.*

- Choose your preferred method of assigning a login script to your users and put the following command there. Replace the word <*path*> by the path of the folder where you put the configuration files:

  UserEnvManager.js <*path*>

4. The Frontend

The frontend for maintaining the configuration files is a client-side web frontend implemented as an HTA (Hyper Text Application) application. The advantage is that you can put it anywhere on your file system and run it by simply launching the *index.hta* file without the need of an additional web server. The downside is that it therefore requires Internet Explorer to run as it directly modifies files on the local files system, which only an HTA application is capable of, because it runs in a special security context.



1. Entry representing a configuration file (*Branch Office.xml* within folder *shortcuts* in this example). If you want to create the default configuration file for the section then leave the name empty.

2. Entry representing an <*item*> tag within the configuration file with the title attribute “MS Office”).

3. This panel shows the conditions of the selected <*item*> tag.

4. Entry representing a <*shortcut*> tag within the selected <*item*> tag with the title attribute “excel”.

5. This panel shows the properties of the selected <*shortcut*> tag.

5. Configuration Files

Item assignments are done within different configuration files. Depending of what kind of item you intend to assign you put the assignments in a special named xml file or in multiple xml files located in a special named folder. The following configuration files/folders are supported:

- *mappings.xml* or *mappings* folder is used for assigning network drive mappings
- *printers.xml* or *printers* folder is used for assigning network printers
- *shortucts.xml* or *shortcuts* folder is used for assigning desktop and start menu shortcuts
- *operations.xml* or *operations* folder is used for assigning general operations like folder creation, file copying or execution of certain scripts or programs.

5.1 General Structure

All configuration files share the following common structure:

- Every configuration file may contain multiple <*item*> tags. An item tag is used to group together one or more assignments which will be processed when a certain set of conditions are met.
- Each item tag may contain one <*conditions*> tag which defines a set of conditions under which the <*item*> tag will be processed.

The following attribtues can be specified for each <*item*> tag:

- <*title*> specified the title which is shown inside the frontend for this item.
- <*disabled*> is a boolean attribute which can be set to *true* to avoid this item from being processed.

The following conditional tags are possible inside the <*conditions*> tag. If multiple tags are specified only one condition has to be met for the item to be processed (logical OR). Within the specified expression also wildcard characters like * and ? may be used. The same tag can be specified multiple times:

- <*userName*> specifies the user's name.
- <*userGroup*> specifies a group the user is member of.
- <*userOU*> specifies the full distinguished name within Active Directory where the user object is located.
- <*computerName*> specifies the computer name userEnvManager is running on.
- <*clientComputerName*> specifies the client's computer name. If the process runs inside the context of a terminal session the name of the computer running the RDP or ICA client is evaluated and not the one of the terminal server.
- <*ipAddress*> specifies the IP address of the computer userEnvManager is running on.
- <*clientIpAddress*> specifies the client's IP address. If the process runs inside the context of a terminal session the IP address of the computer running the RDP or ICA client is evaluated an not the one of the terminal server.

- <*and*> can be used to group multiple tags together which are combined by a logical AND operator. This tag can also be nested within other <*and*> and <*or*> tags.

- <*or*> can be used to group multiple tags together which are combined by a logical OR operator. This tag can also be nested within other <*and*> and <*or*> tags.

On all the above tags the attribute *not* can be stated to negate the result of the operation. The value which is specified to the *not* attribute is irrelevant as it only has to be stated for xml compliance reasons.

The following is an example of the general structure which is valid for all configuration files:

```
<item>
  <conditions>
    <and>
      <or>
        <userGroup>finance</userGroup>
        <userGroup>sales</userGroup>
      </or>

      <clientIpAddress not="">10.10.10.*</clientIpAddress>
    </and>
  </conditions>
</item>
```

5.2 Network Drive Mappings

Network drive mappings are specified within the *mappings.xml* configuration file. The document tag <*mappings*> can contain multiple <*item*> tags as described in the "General Structure" section.

An <*item*> tag can contain multiple <*mapping*> tags. Each <*mapping*> tag represents one network drive mapping.

For the <*mapping*> tag the following attributes are supported:

- <*title*> specified the title which is shown inside the frontend for this tag.

- <*disabled*> is a boolean attribute which can be set to *true* to avoid this tag from being processed.

Within that tag the following tags can be used to specify further mapping details:

- <*localName*> specifies the local drive the network resource is mapped to. The trailing colon has to be included.

- <*remoteName*> specifies the network resource in UNC notation. If this parameter is being omitted, the network drive mapping specified by <*localName*> is being removed.

- <*persistent*> specifies if the mapping is retained after a logoff or reboot. Possible values are *true* or *false* (default).

An <*item*> tag can contain one <*options*> tags. Options specified here are not related to a single mapping but do apply to mappings as a whole. Within that tag the following tags can be used to specify general mapping options:

- *<removeInitialMappings>* can be *true* or *false* and specifies if mappings which already existed on the client side before the program was started are being removed.

The following is an example of a *mappings.xml* configuration file:

```
<?xml version="1.0" encoding="UTF-8"?>

<mappings>
  <item>
    <conditions>
      <userGroup>finance</userGroup>
    </conditions>

    <mapping>
      <localName>H:</localName>
      <remoteName>\\server\share1</remoteName>
      <persistent>true</persistent>
    </mapping>

    <mapping>
      <localName>K:</localName>
      <remoteName>\\server\share2</remoteName>
      <persistent>false</persistent>
    </mapping>

    <options>
      <removeInitialMappings>true</removeInitialMappings>
    </options>
  </item>
</mappings>
```

5.3 Network Printers

Network printer assignments are specified within the *printers.xml* configuration file. The document tag *<printers>* can contain multiple *<item>* tags as described in the “General Structure” section.

An *<item>* tag can contain multiple *<printer>* tags. Each *<printer>* tag represents one network printer assignment.

The following attribtues can be specified for each *<printer>* tag:

- *<title>* specified the title which is shown inside the frontend for this item.
- *<disabled>* is a boolean attribute which can be set to *true* to avoid this item from being processed.

Within that tag the following tags can be used to specify further printer assignment details:

- *<remoteName>* specifies the printer resource in UNC notation.

An <*item*> tag can contain one <*options*> tags. Options specified here are not related to a single printer assignment but do apply to printer assignments as a whole. Within that tag the following tags can be used to specify general printer assignment options:

- <*removeInitialConnections*> can be *true* or *false* and specifies if connections which already existed on the client side before the program was started are being removed.

- <*defaultPrinter*> specifies a printer resource in UNC notation which you want to be assigned as the default printer.

The following is an example of a *printers.xml* configuration file:

```
<?xml version="1.0" encoding="UTF-8"?>

<printers>
  <item>
    <conditions>
      <userGroup>finance</userGroup>
    </conditions>

    <printer>
      <remoteName>\\server\printer1</remoteName>
    </printer>

    <printer>
      <remoteName>\\server\printer2</remoteName>
    </printer>

    <options>
      <removeInitialConnections>true</removeInitialConnections>
      <defaultPrinter>\\server\printer1</defaultPrinter>
    </options>
  </item>
</printers>
```

5.4 Shortcuts

Shortcut assignments the software makes are state aware, meaning if you have assigned a shortcut to a group of users and one user is no longer member of that group that shortcut is removed from the user's desktop or start menu. If a shortcut is removed which is the last one inside a subfolder the empty subfolder is also removed automatically.

Desktop and start menu shortcuts are specified within the *shortucts.xml* configuration file. The document tag <*shortcuts*> can contain multiple <*item*> tags as described in the "General Structure" section.

An <*item*> tag can contain multiple <*shortcut*> tags. Each <*shortcut*> tag represents one shortcut which is placed on the user's desktop, start menu or quicklaunch area.

The following attribtues can be specified for each <*shortcut*> tag:

- <*title*> specified the title which is shown inside the frontend for this item.

- <*disabled*> is a boolean attribute which can be set to *true* to avoid this item from being processed.

Within that tag the following tags can be used to specify further shortcut details. For url type shortcuts only the tags *shortcutFile*, *shortcutDirectory*, *targetPath*, *iconFile* and *iconIndex* are supported:

- <*shortcutFile*> specifies the filename which represents the shortcut including the *.lnk* or *.url* extension.

- <*shortcutDirectory*> specifies the directory where the shortcut should be placed. It always has to begin with the literal *$DESKTOP$*, *$STARTMENU$* or *$QUICKLAUNCH$* depending on weather you want to create a desktop, start menu, or quicklaunch shortcut. If the path specified contains folders which do not exist, they are created automatically.

- <*targetPath*> specifies the path to the executable/document which should be started/opened when the shortcut is selected.

- <*arguments*> specifies arguments which should be passed to the called executable. This tag is optional.

- <*workingDirectory*> specifies the working directory of the called executable. This tag is optional.

- <*windowStyle*> specifies the window style of the called executable. This tag is optional and can be one of the following values: *normal*, *minimized*, *maximized*, *hidden*. If omitted the default value is *normal*.

- <*description*> specifies the description which is displayed when the mouse is moved over the shortcut. This tag is optional.

- <*iconFile*> specifies the file which contains the icon for the shortcut. This tag is optional. If omitted the first icon contained in the executable is selected.

- <*iconIndex*> specifies the index of the icon if the icon file specified contains multiple icons. This tag is optional. If omitted the first icon of the file is selected.

The following is an example of a *shortcuts.xml* configuration file:

```
<?xml version="1.0" encoding="UTF-8"?>

<shortcuts>
  <item>
    <conditions>
      <userGroup>finance</userGroup>
    </conditions>

    <shortcut>
      <shortcutFile>Internet Explorer.lnk</shortcutFile>
      <shortcutDirectory>$STARTMENU$\Communication</shortcutDirectory>
      <targetPath>%programfiles%\Internet Explorer\IExplore.exe</targetPath>
      <windowStyle>normal</windowStyle>
      <description>Browse the web with Internet Explorer</description>
      <iconFile>%programfiles%\Internet Explorer\IExplore.exe</iconFile>
      <iconIndex>6</iconIndex>
    </shortcut>
  </item>
</shortcuts>
```

5.5 General Operations

General operations are specified within the *operations.xml* configuration file. The document tag <*operations*> can contain multiple <*item*> tags as described in the “General Structure” section.

An <*item*> tag can contain multiple <*operation*> tags. Each <*operation*> tag represents one operation to perform.

The following attributes can be specified for each <*operation*> tag:

- <*title*> specifies the title which is shown inside the frontend for this item.
- <*disabled*> is a boolean attribute which can be set to *true* to avoid this item from being processed.
- <*type*> specifies the type of operation to perform. Further details about the operation are defined by additional sub tags of the <*operation*> tag. Which sub tags are valid here depends on the type of operation.
- <*delayed*> is a boolean attribute which indicates if the operation is performed after all other sections like shortcuts, network or printer mappings have been processed. Normally operations are performed first before all other sections.

The following operations are supported:

- *type=”copyFile”* can be used to copy files or folders. The following sub tags are valid to define further details of the operation:
  - <*source*> specifies the source of operation. It can be a file or folder. Wildcard characters * or ? can also be used. If you want environment variables to be expanded in the source string, then you can specify the attribute *expandEnvVar=”true”*.
  - <*destination*> specifies the target of operation. It can be a file or folder. Wildcards must not be used. If you want environment variables to be expanded in the destination string, then you can specify the attribute *expandEnvVar=”true”*.
  - <*subfolders*> is a boolean value which specifies if subfolders should be processed or not. Valid values are *true* or *false*. This parameter is optional. If omitted the default value is *false*.
  - <*emptyfolders*> is a boolean value which specifies if empty sub folders should be created on the target side or not. Valid values are *true* or *false*. This parameter is optional. If omitted the default value is *false*.
  - <*overwrite*> is a boolean value which specifies if already existing files on the target side are overwritten. Valid values are *true* or *false*. This parameter is optional. If omitted the default value is *false*.

- *type="createFolder"* can be used to create a folder. The following sub tags are valid to define further details of the operation:
  - <*foldername*> specifies the name of the folder to be created. If you want environment variables to be expanded in the foldername, then you can specify the attribute *expandEnvVar="true"*.
  - <*recursively*> is a boolean value which specifies if all folders within the given path which do not exist are created automatically. Valid values are *true* or *false*. This parameter is optional. If omitted the default value is *false*.

- *type="execute"* can be used to execute a program. The following sub tags are valid to define further details of the operation:
  - <*command*> specifies the command line to be executed.
  - <*windowStyle*> specifies the window style of the created process. This tag is optional and can be one of the following values: *normal*, *minimized*, *maximized*, *hidden*. If omitted the default value is *normal*.
  - <*wait*> is a boolean value which specifies if UserEnvManager pauses it's execution until the executed command has finished. Valid values are *true* or *false*. This parameter is optional. If omitted the default value is *false*.

The following is an example of an *operations.xml* configuration file:

```
<?xml version="1.0" encoding="UTF-8"?>

<operations>
  <item>
    <conditions>
      <userGroup>finance</userGroup>
    </conditions>

    <operation type="copyFile" delayed="true">
      <source expandEnvVar="false">\\server\deploy\settings\*.*</source>
      <destination expandEnvVar="false">C:\settings</destination>
      <subfolders>true</subfolders>
      <emptyfolders>true</emptyfolders>
      <overwrite>false</overwrite>
    </operation>
  </item>

</operations>
```

6. Disclaimer Copyright

The software is copyright protected by Frank Gruber (gig mbh berlin) and is delivered free of charge for non-commercial and commercial use. It is not allowed to sublicense, rent, sell, lease or repackage it as part of another solution without the explicit agreement of the author. We assume no liability for damages, direct or consequential, which may result from its use. Using this software indicates your acceptance of this agreement. If you do not agree to the terms and conditions, remove the software from all your systems.

7. Contact

The most recent version can always be downloaded from www.gig-mbh.de. If you have any questions, comments, suggestions or enhancement requests do no hesitate to write a mail to support@gig-mbh.de.

8. Credits

- The web frontend is build with the javascript framework extjs (www.extjs.com).
- Most icons within the web frontend are based on the “Silk icon set” of Mark James (www.famfamfam.com/lab/icons/silk).