Titan Backup Plugin File Specification

Version 1.0
Created 04/04/2008
http://www.titanbackup.com

Introduction

Titan Backup has an engine that allows the execution of plug-ins.
A plugin, is a file that describes a set of consistent backup operations (files, folders and registry keys). The plugins are meant to be used to describe what needs to be saved for certain applications.

Titan Backup comes with a set of predefined plugins for popular applications, such as mail clients (Outlook Express, Microsoft Outlook, Thunderbird, Opera, Eudora, etc) and other program settings (Firefox, Internet Explorer, Winamp, etc.). You can find the default plug-ins in "Drive:\Program Files\Titan Backup\Plugins".

The plugin file is in standard XML format, and it is open to anyone who wants to write their own plugins.
Before writing your own plugin we suggest to study the default plugins that come with Titan Backup (built by our own programmers).

If you wrote a plugin and want to share it with others please post it on the Titan Backup community page or send it to support@titanbackup.com
We may even include it in one of our future releases.

Plugin File Format

The Plugin file is a standard XML file, with specific tags described below:

<plugin>

The root tag, required for all plugins.

<id>

Unique identifier for each plugin. It is a GUID, that you can generate with any GUID generation tool (http://www.fileformat.info/tool/guid.htm).

Example:<id>{D1A4D46B-0204-4357-9F69-75E1E4C62D34}</id>

<version>

Plugin format version. Must be 1.0 for now.

Example: <version>1.0</version>

<name>

The name of the plugin. This will appear in Titan Backup in the plugins window when creating a task.

Example: <name>Winamp</name>

<hint>

The hint that appears in the plugins window in Titan Backup when the user moves the mouse cursor over the plugin.

Example: <hint>Saves Winamp Settings</hint>

<category>

The plugin category of this plugin. Check the Titan Backup plugin windows to see the list of plugin categories.

Example: <category>Graphics and Web-Design</category>

<group>

Certain plugins must have sub-options, and the ability to select each one individually. Outlook Express is such an example, there are Mails, Rules, Accounts.
You will need to create a plugin file for each sub-option separately. All these plugins are grouped under one node, named by the <group> tag. It usually is the name of the program.

Example 1 (for simple plugins with no sub-nodes): <group/>
Example 2 (for plugins with sub-nodes): <group>Outlook Express</group>

<priority>

Determines the order of the plugin in the Category or Group. Large values will push the plugin to the bottom of the list, and lower values to the top of the list.

Example: <priority>80</priority>

<tree>

This tag can have only 2 values: *User* or *Mail*. The User value will make this plugin appear under User Settings, the Mail value will make this plugin appear in the Email window.

Example: <tree>User</tree>

<path> and <resources>

These two tags are the most important parts of a plugin and before reading on you should know that there are a few environment variables you can use when defining paths. The environment variables enclosed with "%" are system environment variables (i.e. %PROGRAMFILES%) and the ones enclosed with "$" are custom variables (i.e. APP_PATH). Below is a list of all the environment variables you can use:

Custom variables:
$USER_NAME$ - name of the currently logged on user
SYS_DRIVE - letter of the drive where the operating system is installed (i.e. "C:")
SYS_DIR - the directory where the operating system is installed (i.e. "C:\Windows")
$MYDOCUMENTS$ - path to the currently logged on user's My Documents folder (i.e.
"C:\My Documents")

System variables:
%APPDATA% - i.e. "C:\Documents and Settings\All Users"
%PROGRAMFILES% - i.e. "C:\Program Files"
%SYSTEMDRIVE% - i.e. "C:"
%SYSTEMROOT% - i.e. "C:\Windows"
%TEMP% - i.e. "C:\Documents and Settings\[username]\Local Settings\Temp"

%TMP% - same as %TEMP%
%USERNAME% - name of the currently logged on user
%USERPROFILE% - i.e. "C:\Documents and Settings\[username]"
%WINDIR% - i.e. "C:\Windows"
%PUBLIC% - i.e. "C:\Users\Public"
%ALLUSERSPROFILE% - i.e. "C:\Documents and Settings\All Users"
%PROGRAMFILES(X86)% - i.e. "C:\Program Files (X86)"

<path>

This tag contains several sub tags that are used to identify the path to the application for which your plugin is designed. If your plugin is not designed for an application, leave this tag empty.
It is necessary to know the path to the respective application so that Titan Backup can determine if it is installed and if so, extract the icon from the executable. The extracted icon will be shown in the plugins list. The path can be either taken from the registry (the sub tags **<key>**, **<string>**, **<exe_name>** are used for this) or it can be provided by you (in the **<exe_path1>**, **<exe_path2>** tags) if the application doesn't keep the path to its installation folder in the registry and always installs in the same directory.

<key>

This value represents a registry key where the application stored its path (the application's installation folder).

Example: <key>HKEY_CURRENT_USER\Software\Neobyte Solutions\Invisible Secrets 4\Options</key>

<string>

The name of the registry value that holds the path to the application's installation folder. Leave this empty if the path is kept in the registry value named **(Default)**.

Example: <string>ProgramPath</string>

<exe_name>

The name of the application's executable file.

Example: <exe_name>invsecr.exe</exe_name>

<exe_path1> and <exe_path2>

As stated above, these are used if the application doesn't keep the path to its installation folder in the registry, but always installs in the same directory. In this case you must

provide the full path to the application's executable file. Note that it is not required to use both <exe_path1> and <exe_path2> except if necessary. You can use variables to define the path.

Example: <exe_path1>%PROGRAMFILES%\Google\Google Talk</exe_path1>
*   <exe_path2>%PROGRAMFILES(X86)%\Google\Google Talk</exe_path2>*

Note: If the path will be taken from the registry, there is one small trick you can use. For some applications (i.e. Internet Explorer) the path to their installation folder can be found here: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths. In this case you don't have to specify the entire registry path in the **<key>** tag. It is enough to enter just the last part of the path and Titan Backup will do the rest. Here's how the **<key>**, **<string>** and **<exe_name>** values would look for Internet Explorer:

Example 1 (normal version):
<key>HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths\IEXPLORE.EXE</key>
*   <string></string>*
*   <exe_name>IEXPLORE.EXE</exe_name>*

Example 2 (short version):   <key>IEXPLORE.EXE</key>
*   <string></string>*
*   <exe_name>IEXPLORE.EXE</exe_name>*

<resources>

This is the most important part of a plugin because it contains the files, folders, and registry keys that will be saved by the plugin you create. There are three main resource categories: folder, files, registry. Each resource type is kept in its own sub tag**.**
Note: For each plugin that is processed, Titan Backup creates a folder in the destination with the following format: ...\Plug\[Plugin_Name]. So if you have a backup task with the archive name "MyBackup" that will be created in the destination folder "D:\Backups" and you selected the Internet Explorer Settings plugin, the destination path for all this plugin's resources will be: *"D:\Backups\MyBackup\Plug\Internet Explorer-Settings"*. To simplify explanations, from here on this path will be called the ***plugin's destination path***.

<file>

Path to a file that will be backed up. There is one parameter you can use within this tag:

sub (i.e. sub="Test"). This will instruct Titan Backup to create a subfolder with the name you provided for the *sub* parameter in the plugin's destination path. The file will be transferred in this folder.

*Example 1 (without the **sub** parameter): <file>APP_PATH\i_view32.ini</file>*
*Example 2 (with the **sub** parameter): <file sub="AppPath">APP_PATH\i_view32.ini</file>*

<folder>

Path to a folder that will be backed up. There are several parameters you can use:

sub - same as the <file> tag above
inclmask (i.ie. inclmask="*.xml") - only files having the specified extension will be transferred from that folder. You can use several masks but you need to separate them with a semicolon (i.e. inclmask="*.xml;*.ini")
exclmask (i.ie. exclmask="*.xml") - files having the specified extension will NOT be transferred from that folder. Several masks can be used here also (see the *inclmask* parameter above)

You cannot use at the same time the **inclmask** and **exclmask** parameters inside a <folder> tag, but you can use **sub** with any of them.

Example 1 (no parameters):
<folder>%USERPROFILE%\AppData\Roaming\Winamp</folder>
*Example 2 (with the **sub** parameter): <folder*
sub="Vista">%USERPROFILE%\AppData\Roaming\Winamp</folder>
*Example 3 (with **sub** and **inclmask**): <folder sub="Vista"*
inclmask=".xml">%USERPROFILE%\AppData\Roaming\Winamp</folder>*

<reg>

A registry key that will be saved in a .REG file in the plugin's destination path. Only one parameter can be used within this tag:
sub - see the <file> tag above

*Example 1 (without the **sub** parameter)*:
<reg>HKEY_CURRENT_USER\Software\e-merge\WinAce</reg>
*Example 2 (with the **sub** parameter): <reg*
sub="Current">HKEY_CURRENT_USER\Software\e-merge\WinAce</reg>

<inipath>

This tag can be used to extract the path to a folder or file from an .INI file, and save what resides at that location.
The value contained inside the tag is the path to the .INI file. You can use the parameters described below:

sub - see the <file> tag above
file (i.e. file="") - specifies that the path to be extracted is a path to a file.\
folder (i.e. folder="")- specifies that the path to be extracted is a path to a folder
inclmask - see the <folder> tag above. This parameter can be used only together with the **folder** parameter
exclmask - see the <folder> tag above. This parameter can be used only together with the **folder** parameter
section (i.e. section="Test") - the section from the .INI file where the path can be found. This parameter is critical in order to successfully extract the path from the .INI file
key (i.e. key="Settings") - the name of the value that represents the path to the file or folder to be backed up. This parameter is critical in order to successfully extract the path from the .INI file and it must be a value found in the section given in the **section** parameter

Three of the above parameters are absolutely required inside a **<inipath>** tag in order to successfully extract the path to a file or folder from an .INI file: **file** or **folder**,

section,and **key**.

Example (this saves a folder, refferenced in an INI file, section "Loggin", key "Directory"):
<inipath folder="" section="Logging" key="Directory">APP_PATH\users\default\talk.ini</inipath>

<regpath>

This tag can be used to extract the path to a folder or file from a registry value. It is similar to the <inipath> tag, but instead of a INI file it extracts the path from a registry key. The value contained inside the tag is the registry path to a value specified in the **key** parameter. You can use the parameters described below:

sub - see the <file> tag above
file - see the **<inipath>** tag above
folder - see the **<inipath>** tag above
inclmask - similar to the **<inipath>** tag
exclmask - similar to the **<inipath>** tag
key - the name of the registry value which contains the path to the folder or file to be backed up. This parameter is absolutely required in order to obtain a path. If the path is kept in the registry value named **(Default)**, leave this string empty (i.e. key="")

Example: <regpath folder=""
key="Path">HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\App
Paths\wmplayer.exe</regpath>

<action>

If the plugin reffers to an application, this tag describes the application.
It has 2 sub-tags: **<name>** and **<process>**

<name>
The name of the application.

<process>
The process name of the application as it appears in the Task Manager.

If the task is not designed for an application the Action tag should be empty.

Example:
<action>
<name>ACDSee 10</name>
<process>ACDSee10.exe</process>
</action>

<icons>

If the plugin reffers to an application, Titan Backup will show the icon of that application. If this is not related to an application, but some other specific scripting (like Windows

Settings), you can insert your own custom icon.
The icon must be an .ICO file, encoded in Base64. You can set the icon for the plugin or for the group. If the plugin is not part of a group, place the encoded icon in the plugin tag.

Example:
<icons>
 <group/>
  <plugin>AAABAAEAEBAAAAEAIABoBAAAFgAAACgAAAAQAAAAIAAAAAEAIAAAAAAAAAAAAAAAAAAAA
bFUv//7+Fdv8VS///MxsO5TQaDtQ3HxWtPSohekAyLkhDMzMsAAAAAUBAQAhrpeaTFUv//xVL//8VS///NoT//+
nv+/hXb/xLC1/
zaE//9VSkz4NR0S4jkjGcA7KSKPOyoocQAAgAJ1pcktN4T//9u+qf//3K///eC4//7syP/969X/
7LCA///huf/48er/e6f//2BvjPw1HRLrNyAWzTgkILozMzMFYqHsjUCB8P//37n/
+t26//3uz//+9eb/
7MKo//jJl///4sH///fv//////97pv//f7P//zopIew2JB3cMzNNCkKc+f9CnPn//+zT///u2///+/
b///79/92fdP/+26///+zX///48f///fz//////02c//9KaIz4NiMb3Gmf0D8/kvr/+ODI///y4v//+/
b///7+//LczP/
1yJn//+nM///79////v7//////+31//8vkf//P0RS7TgkHclsr+qtQpz5///48v//+vX///n0//359v/
rwqP//tyv///z4////fz///////////+Mw///Wqz//zYhFtM9KSGnQpz5/+HJtf//8+f///Po///8+f/
z6uX/7 sil///oyf//9Oj///r1////////5/
P//OZ3//3iJnvY4Ixm2PiwleUKc+f//7t3/////////M3f//nbr//7K82f/
47d////79///////////pdL//1Gt/
v84IhjTPCohkUMzLU81lf//fsX//5mw1LWbsdWzRDIvR0Y4MzdJYZhAx93+/
8Pe/////////1Kw//+ForzpPCkipz0wKWREODgsVZ7pOEZGURZAQEAcQDU1GEc5ORJVQEAMSUlJB1UrKway
AMAAAACAAAAAgABVAwAAORIAAPfoAAD//wAAoP8AAI7/AACl/wAArv8AALT/AAC7/
wAAwf8BAMj//wHO/w</plugin>
</icons>

Technical Support

Feel free to contact us with any questions you might have while creating your own plugin, or if you want to publish your plugin on our website.

Web: http://www.titanbackup.com/support.html

Email: support@titanbackup.com