**Introduction**

Follow the procedures in this work instruction to add Google Analytics tracking code and user feedback forms to your WebHelp systems.

**Download Ruby**

Download the rubyinstaller-1.8.7-p352.exe file from the [RubyInstaller](http://rubyinstaller.org) site to your hard disk and run it to install Ruby on your system.

**Download the script**

1. Download the [hpp.zip](https://docs.google.com/a/mdsol.com/viewer?a=v&pid=explorer&chrome=true&srcid=0By7Er1_HcptyMTAwZGJjOTItMjJlNC00NTJmLWI5MGItNmI3OGYzYTc4NThh&hl=en_US) file and unzip it to a folder on your hard disk, for example C:\hpp.

This work instruction assumes that the script is installed in C:\hpp.

**Verify the script version**

The current version is **1.0.0**.

The version displays each time you run the script on a help system. You can also view the version without processing a help system. To do so:

1. Launch the Ruby command window by clicking **Ruby1.8.7-p352 > Start Command Prompt with Ruby**on the main Windows menu.
2. Type cd c:\hpp\ruby in the command window to navigate to the folder containing the script.
3. Type ruby hpp.rb -v or ruby hpp.rb --version to display the script version.

**Prepare WebHelp system for processing**

1. Follow the standard workflow described in the [Online Help Development Lifecycle](https://sites.google.com/a/mdsol.com/km-team/km-process-home-page/tcs-processes/online-help-development-lifecycle) to prepare your WebHelp system for publication.
2. Make backup copies of your WebHelp systems in case you need to re-run the script.

If you need to re-run the script, do so on a backup copy of your WebHelp system. Do not try to run the script twice on the same system.

**Changing the Windows locale**

This procedure applies only if you are processing Japanese files and the names of the HTML files contains Japanese characters. If not, go straight to [Configu](https://sites.google.com/a/mdsol.com/glynn/#_Move_content_files_1)re the script.

1. Select **Control Panel > Region and Language** to open the **Region and Language** dialog box.
2. Select the Administrative tab.
3. Click **Change system locale**.
4. Select “Japanese (Japan)” from the list of locales.

**Configure the script**

1. If you are using the script for the first time, rename the ...\ruby\hpp-example.yml settings file to hpp.yml or to <name>.yml, where you specify <name>.

By default the script uses the hpp.yml file. If you specify a settings file when you run the script, the script uses that file. This allows you to run the script on multiple help systems.

1. Update the product key with the name of the help system you are processing.

For example, product: Balance.

The settings file contains all the valid values for the product key. Copy and paste one of these values into the key.

1. Update the webhelp key with the full path and name of the root file of the WebHelp system.

For example, webhelp: C:/WebHelp/Balance.htm processes the WebHelp system in the C:\WebHelp folder whose root file is Balance.htm. The root file is the HTML file used to launch the help system.

**Note**: always use forward slashes in paths in the sncettings file, even though Windows displays paths with backward slashes.

1. If the folder containing your WebHelp content files is not the same as the name of the root file without the .htm extension, update the webhelp\_content\_folder key with the fully-qualified path of the content folder.

For example, if your root file is C:/WebHelp/Balance.htm and your content folder is C:/WebHelp/BalanceHelpText, specify the following keys:

webhelp:  C:/WebHelp/Balance.htm

webhelp\_content\_folder:  C:/WebHelp/BalanceHelpText

You can include <LANG> placeholders in the path.  See the next step for an explanation of <LANG> placeholders.

1. If the folder containing your WebHelp content files is the same as the name of the root file without the .htm extension, specify "default" for the webhelp\_content\_folder key.

For example, if your root file is C:\WebHelp\Balance and your content folder is C:\WebHelp\Balance, specify the following keys:

webhelp:  C:/WebHelp/Balance.htm

webhelp\_content\_folder:  default

1. Update the language key with the standard three-letter code for the language of the WebHelp system you are processing.

For example, language: ENG specifies an English WebHelp system.

If you store your WebHelp system in a path that includes a language folder, you can include <LANG> placeholders in the WebHelp path to tell the script to replace <LANG> with the language specified in the language key.  For example, the settings

webhelp: C:/WebHelp/<LANG>/Balance.htm

language: ENG

tell the script to process the C:\WebHelp\ENG\Balance.htm WebHelp system.

This feature is mainly useful for processing multiple language versions of a WebHelp system at the same time. To do so, specify multiple languages separated by commas in the language key. For example, the settings

webhelp: C:/WebHelp/<LANG>/Balance.htm

language: ENG,JPN

tell the script to process the C:\WebHelp\JPN\Balance.htm and C:\WebHelp\ENG\Balance.htm WebHelp systems.

**Note**: You must always specify a value for the language key even if you do not use <LANG> in your paths.

1. If you want to tag showme HTML wrappers with Google Analytics code, see Tag showme wrappers.

**Tag showme wrapper files**

You can tag all the HTML files in a folder as showme files so that they appear as showmes in Google Analytics statistics when they are accessed.

You would typically use this feature once only, when you first tag your showme wrappers. When you subsequently add showmes, will probably be easier to copy and rename existing wrappers.

1. Create a folder beneath the contents folder of your Webhelp system and copy your showme wrappers into it.
2. Uncomment the showme\_wrappers\_folder key and update it with the location of this folder.

For example, the setting showme\_wrappers\_folder: C:\Balance\Showmes tells the script to tag all the files in the C:\Balance\Showmes folder as showme wrappers.

**Note**: this setting has no effect if you are processing a legacy WebHelp system. If the settings file contains the setting webhelp\_content\_folder: legacy, the script does not process showme wrappers even if the showme\_wrappers\_folder key contains a value.

**Run the script**

1. Launch the Ruby command window by clicking **Ruby1.8.7-p352 > Start Command Prompt with Ruby** on the main Windows menu.
2. Type cd c:\hpp\ruby in the command window to navigate to the folder containing the script.
3. Type ruby hpp.rb to launch the script and use the default hpp.yml settings file, or ruby hpp.rb <settings-file> to launch the script and use the <settings-file>.yml settings file.

For example, type ruby hpp.rb balance to launch the script and use the balance.yml settings file.

If the script runs without errors, the screen displays the following text:

File: <webhelp-file-name>  
Working……Done!