-=-=-=-=-=-=-=-=-=-=-=-=-

Janice {bulleted list}

Sysgem File SafeGuard, or SFiS, is a module in the Sysgem Enterprise Manager suite of products. It manages the monitoring, distribution, and update of configuration files on remote multi-platform servers.

At the heart of the solution is a set of central ‘Source Files’ that define the content of text files held on distributed servers and how those files vary between servers.

SFiS distributes new files to servers, modifies the content of existing files and monitors remote files by checking for changes.

Alarms are raised by automated central procedures when unexpected text is detected.

A full audit trail is maintained giving:

- details of changes to the central source files
- details of updates to target servers
- who made the changes

-=-=-=-=-=-=-=-=-=-=-=-=-

Mike

Let’s have a look at SFiS in operation.

Start the SFiS Source File Manager.

Source files can be categorized under different ‘projects’ which are held in separate ‘Base Directory-Drawers’.

Select the required directory-drawer.

You will notice that folders are subdivided into… UNIX… VMS… and Windows. Keep the source files in the appropriate folders for the different platforms.

There is also a ‘Wastebasket’ folder for deleted source files

… and an ‘Archive’ folder which contains all previous generations of source files after they have been edited in the source file manager window.

-=-=-=-=-=-=-=-=-=-=-=-=-
Janice

We will now make a short demonstration with source files that are destined for Linux servers.

We select the source file named “Demo4”. This defines the content of a file (also called demo4) that is held on remote Linux Servers

Select the menu option: “Synchronize UNIX Target Files”…

The four Linux servers in the Agent list are still selected from the last time we ran this menu option on this source file …

The SFiS software has now run on the four servers…

You can see that the display is showing that all four servers conform to the definition held in the central file.

-=-=-=-=-=-=-=-=-=-=-=-=-

Mike

We will now use the Sysgem File and Directory Explorer to look at the files on the target servers.

We will modify one of the remote files and then re-run the synchronize option to detect the change we made.

We drill down into the DemoLinux2 system.

A typical configuration file on a Linux system may look like this... This is a copy of a “sudoers” file.

However, for this demonstration we will use a simpler text file in which we can more easily recognize changes.

The file has now been modified and we will refresh the synchronize window to detect the change.

Now the display shows that there are differences on DemoLinux2

Using the menu option “Show Differences” we can see what was found in the target file in comparison to what was expected.

We can get a full listing of the file on the remote server…

… and we can get a full listing of the central source file...

Now we will update the file so that it conforms…   and the display shows the status “Updated”.

Refresh the window and it finds that all four files on the target servers are again the “Same” as the central source file.

As one final test, we will use the File and Directory Explorer again and have a look at the file on the target server.

The file has been reset to conform to the source file.

-=-=-=-=-=-=-=-=-=-=-=-=-

Janice {bulleted list}

That is the end of the interactive demonstration, but let’s outline some of the other features in File SafeGuard.

Multiple files can be included in a single synchronization by having a ‘list’ of source files in a “Staging Template”. Then, simply use the same synchronize option as we have just seen with the staging template file.

If a file does not exist on a target server the “Update Target File” menu option will create it. This can be very useful for when a new (empty) server is introduced on a network and a whole set of new files need to be created to configure the server.

The synchronize option can be run automatically, in the background, on a central windows system to monitor the content of files on remote, multi-platform, servers. Alarms will trigger an email message when changes are detected.

The monitoring task can automatically update files when changes, within a set threshold, are detected.

‘Pattern matching’ options provide a simpler report showing not only which servers have had changes, but which servers have the ‘same’ changes. This is particularly useful when large numbers of servers are being monitored.

Pre- and post- processing scripts can be run on the target servers immediately before and / or immediately after a ‘file update’.

Exclusion clauses can be set to prevent an update on particular servers when a certain condition applies.

A simple meta-language is used in the source files to define how the text should look on the remote servers and how it differs on different servers.

A menu option can be used to assist in the initial creation of a source file by picking an example file from a remote server. This results in new source file containing a well-structured meta-language definition of the selected file.