229P-2013

Here Comes Your File!
File-Watcher Tool with Automated SAS® Program Trigger

Rajbir Chadha, Cognizant Technology Solutions

Abstract

In a typical warehousing environment, users are dependent on files being made available by operations team. **Users have to wait** till the file is available and run their SAS processes. There was a need for an **automated tool** to **trigger** the **user processes** as soon as input files become available. This display introduces a UNIX based **File-Watcher** tool that **searches for input files** and checks to see when they were **last updated**.

The tool uses **UNIX Shell Scripting** and is scheduled using a ‘**CRON**’ scheduler. The parameters to the file-watcher tool are supplied using a ‘**wrapper**’ script. This can be **modified by the user** as and when required. The tool can be programmed to look for the file within a specified time period. Once the updated version is found, the SAS program execution begins.

The tool incorporates **error handling** and **e-mail notifications**. The file-watcher tool **reduces average wait time** and manual effort for users by **automating** most of the process, allowing them to focus on other pressing tasks.

File-Watcher

A program or a script designed to look for specific files on the file system is typically called a **File-Watcher**. Such scripts **monitor one or more folders** for new or updated files at regular intervals. File-watcher can be **kept running at all times** with inbuilt sleep cycles, or scheduled in a CRON like scheduler.

File-Watcher script **can send out e-mails** to users and **trigger custom actions** like kicking off a downstream process. Such scripts can be used for **pre-processing files** as soon as they arrive, **manage incoming FTP files**, **load files in databases** etc.

File-Watcher uses **five** parameters to control the search operation (**Figure 1**) and are typically managed by the Wrapper script (*See section on Wrapper Script*).

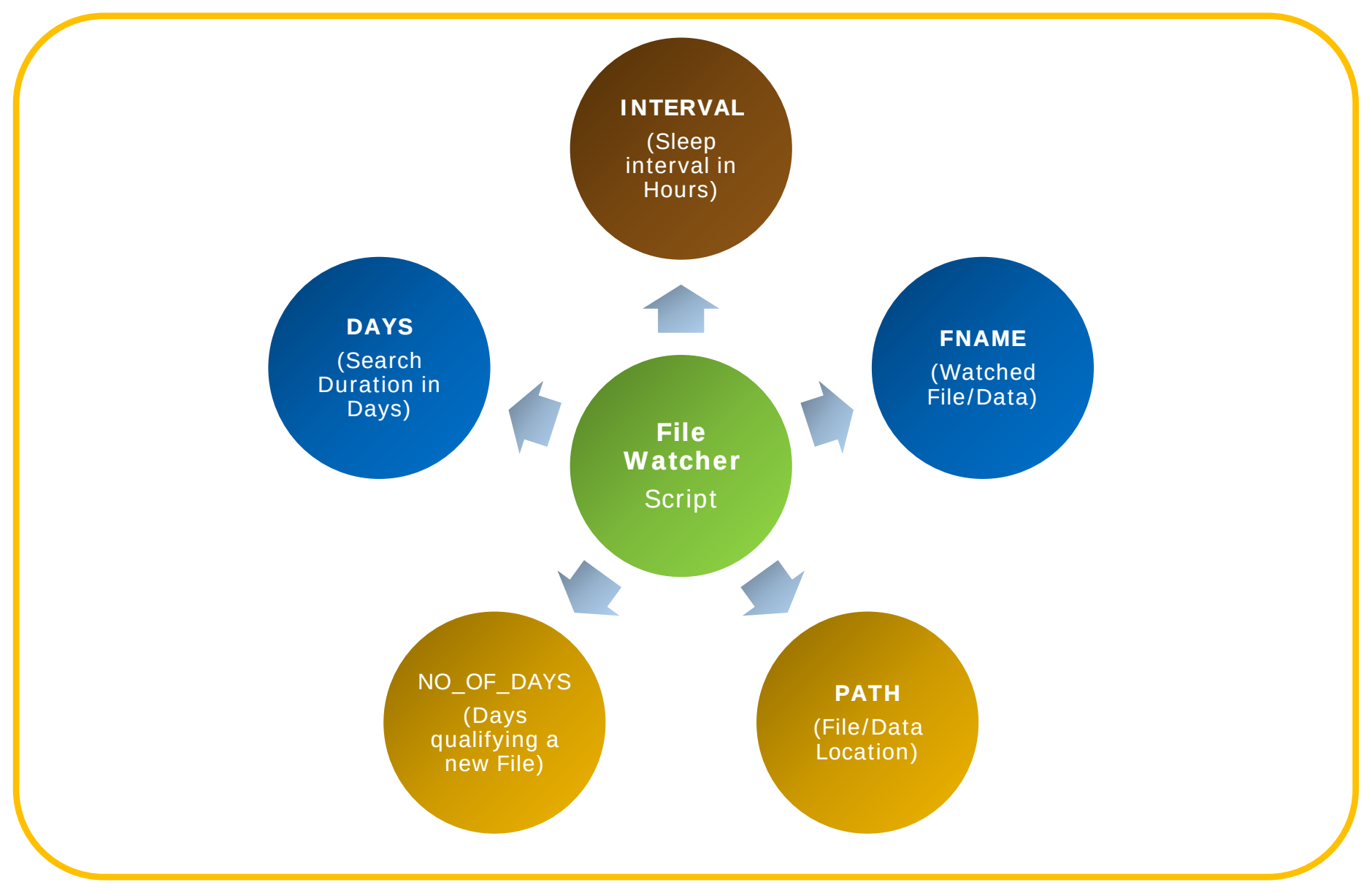


Figure 1. File-Watcher Parameters

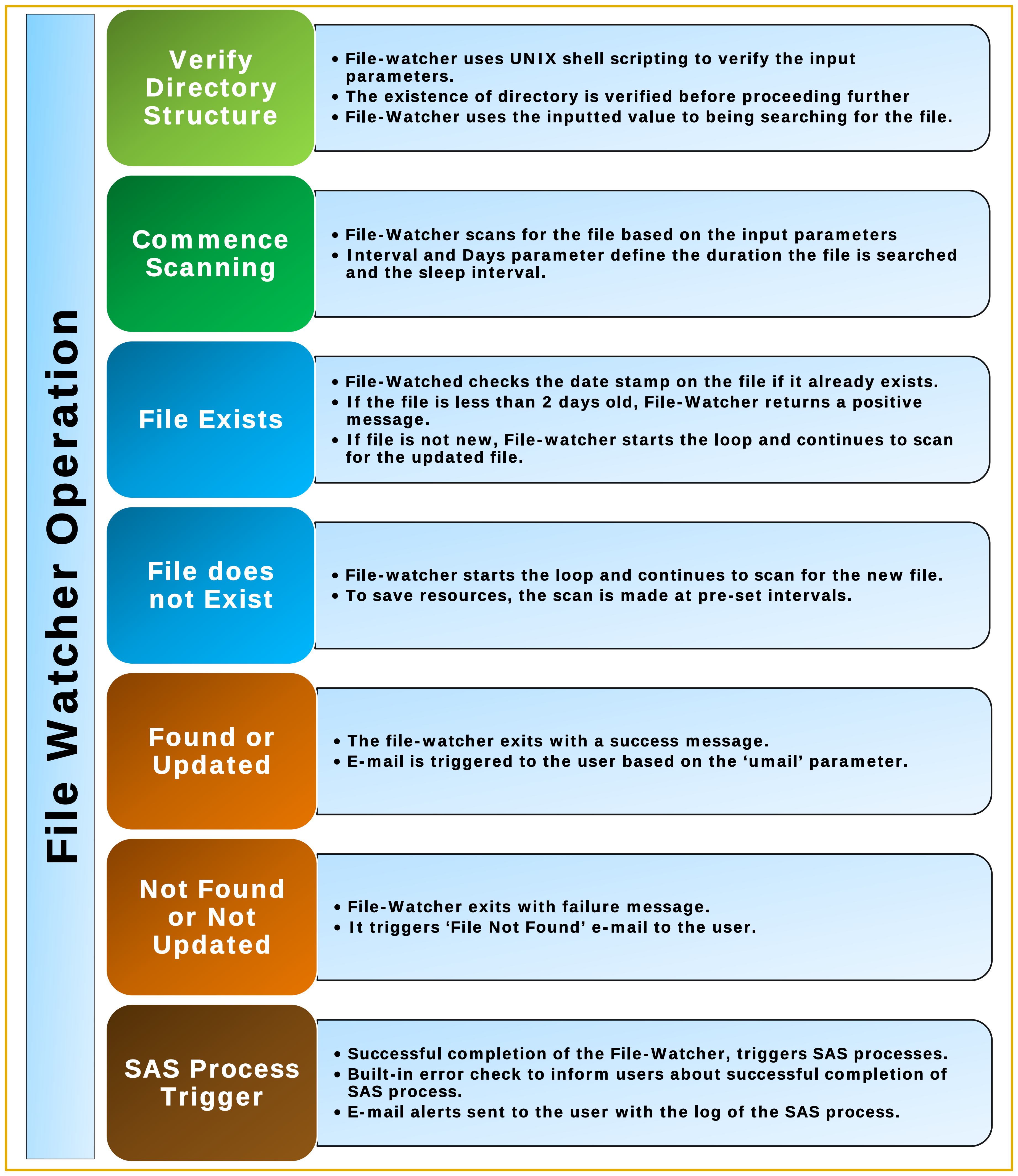
File-Watcher Operation

The ‘**umail**’ parameter contains the list of e-mail address where the notification will be sent. This parameter is set in the wrapper script that is discussed later in the Wrapper Section.

The figure alongside (**Figure 2**) provides an overview of the File-Watcher. In order to **prevent false triggering** of the SAS process, the File-Watcher **analyzes the time-stamp** on the file if it already exists. If the time-stamp is more than **2** days old (this value can be modified), it does not trigger the file-watcher. The script **continues to monitor** the file, scanning it based on the **specified parameters**.

If the file gets updated during that time, the script triggers the ‘**File Found**’ e-mail and subsequently the SAS process. If the file is not updated, the file-watcher throws an exception and triggers the ‘**File Not Found**’ e-mail.

The figure alongside (**Figure 3**) illustrates how the File-Watcher triggers SAS programs when the file is found.



File Watcher Operation

Verify Directory Structure
- File-watcher uses UNIX shell scripting to verify the input parameters.
- The existence of directory is verified before proceeding further
- File-Watcher uses the inputted value to being searching for the file.

Commence Scanning
- File-Watcher scans for the file based on the input parameters
- Interval and Days parameter define the duration the file is searched and the sleep interval.

File Exists
- File-Watched checks the date stamp on the file if it already exists.
- If the file is less than 2 days old, File-Watcher returns a positive message.
- If file is not new, File-watcher starts the loop and continues to scan for the updated file.

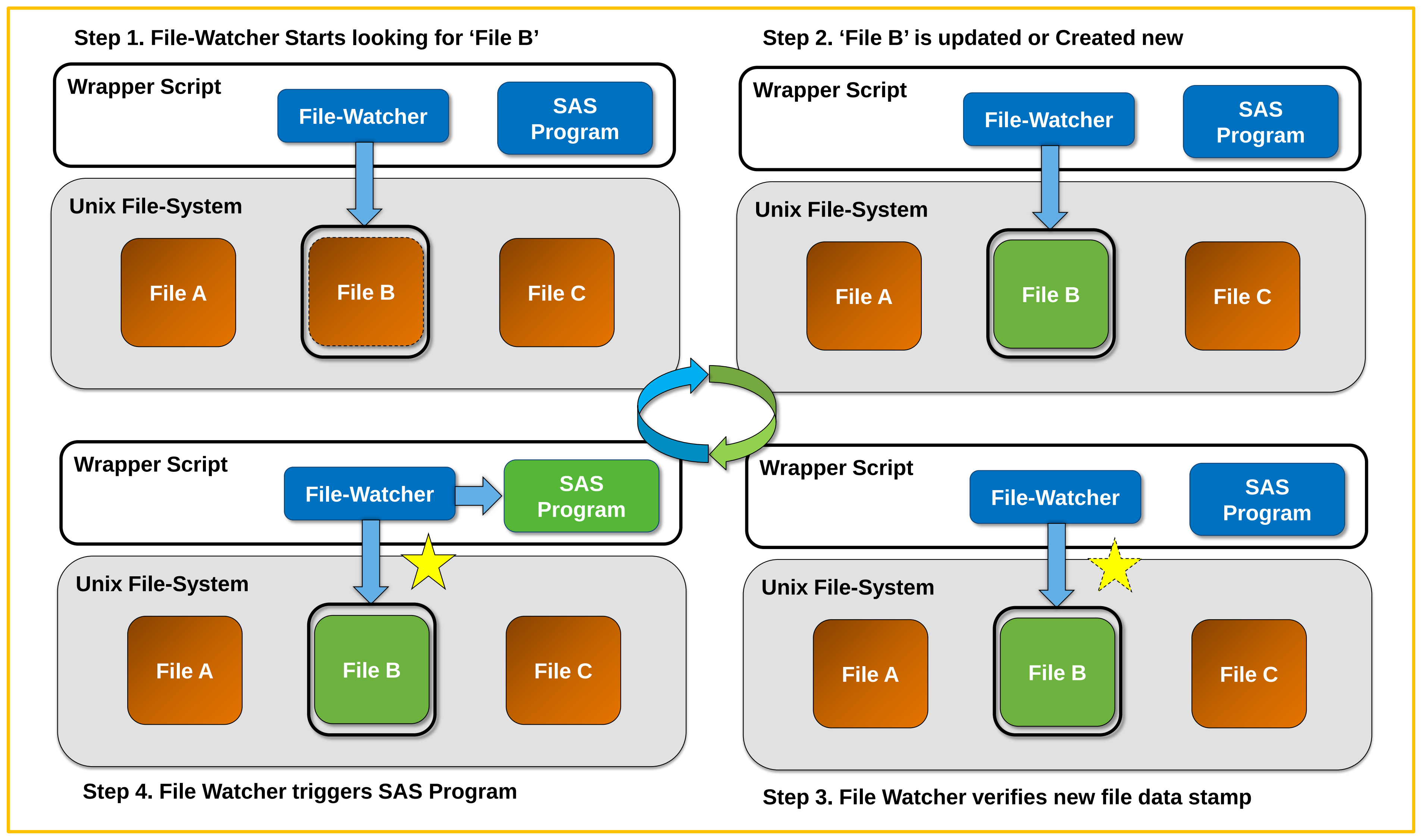
File does not Exist
- File-watcher starts the loop and continues to scan for the new file.
- To save resources, the scan is made at pre-set intervals.

Found or Updated
- The file-watcher exits with a success message.
- E-mail is triggered to the user based on the ‘umail’ parameter.

Not Found or Not Updated
- File-Watcher exits with failure message.
- It triggers ‘File Not Found’ e-mail to the user.

SAS Process Trigger
- Successful completion of the File-Watcher, triggers SAS processes.
- Built-in error check to inform users about successful completion of SAS process.
- E-mail alerts sent to the user with the log of the SAS process.

Figure 2. File-Watcher Operation Overview



Step 1. File-Watcher Starts looking for ‘File B’

Step 2. ‘File B’ is updated or Created new

Step 3. File Watcher verifies new file data stamp

Step 4. File Watcher triggers SAS Program

Figure 3. File-Watcher Operation Steps

Wrapper

The Wrapper script, as the name suggests, simply “**wraps**” around an existing script, in this case, the file-watcher. It is used to shield the users from a more complicated script and to prevent tampering of the inner script.

The wrapper contains user defined parameters, required by the file-watcher. It is scheduled using the **CRON** scheduler in UNIX. Users can modify the wrapper to search for different files and to execute various SAS programs once those files are found.
The Wrapper Script Parameters are shown here (**Figure 4**).

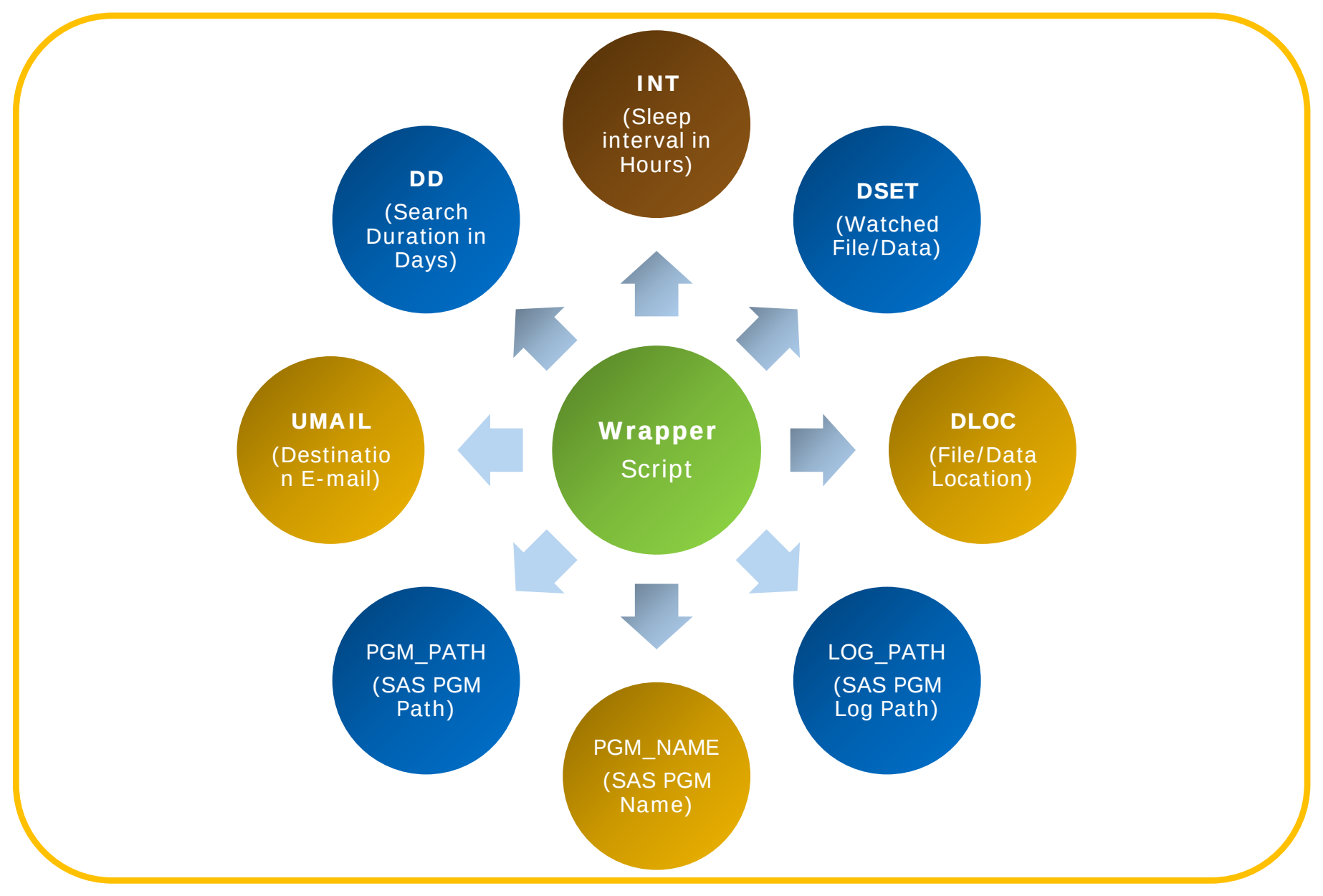


Figure 4. Wrapper Parameters

The wrapper has a **40** minute delay sequence built in to allow **FTP** or **QC** processes to complete before executing user programs. This is entirely optional and may be removed depending upon your requirements. Wrapper can trigger **multiple SAS programs** which can be run in **parallel or in sequence**.

Error Handling

The wrapper script checks SAS program for errors based on return code (**RC**). E-mails alerts may be sent out to users informing them about the status of the SAS program.

```
RC=$?
if [[ $RC -ge 2 ]] ; then
   print "FATAL ERROR JOB ${pgm_name}.sas Failed, check log file
${pgm_name}.log " `date`
   print "Return Code: ${RC}"
   exit 1;
else
   print "JOB ${pgm_name}.sas Completed Successfully " `date`
fi
```

Figure 5. Error Check in Wrapper

UNIX return code can be obtained by ‘**$?**’ command in ‘**Korn**’ shell. Return codes range from **0-256**, where **0** indicates a successful operation. SAS typically returns **0** for success, **1** for warnings and **2** for errors.

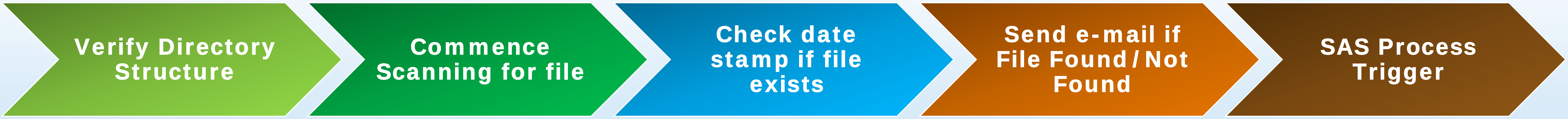
E-mail Notifications

The figure below illustrates the code to add e-mail notification for the SAS program (**Figure 6**). The ‘**mailx**’ command is used to send e-mail messages to the specified recipients.

The ‘**uuencode**’ command converts **binary** file to **ASCII** data which can be included in the email. ‘**uuencode**’ takes the file name on the standard input and produces an encoded file at the standard output. The encoding uses printable ASCII characters.

```
mailx -s "SAS Program Completed: ${pgm_name}.sas " -r "SAS-
FileWatcher@XYZ.com" $umail << EOT
****This is an automated mail from the SAS UNIX server****
EOT


uuencode /sas/apps/sas_prod/user/demo_user/demographics.xls
demographics.xls > attachment
uuencode ${log_path}/${pgm_name}.lst ${pgm_name}.lst >> attachment

mailx -s "SAS Program Completed: ${pgm_name}.sas " -r "SAS-
FileWatcher@XYZ.com" $umail< attachment
```

Figure 6. E-mail Notification code



Verify Directory Structure

Commence Scanning for file

Check date stamp if file exists

Send e-mail if File Found / Not Found

SAS Process Trigger

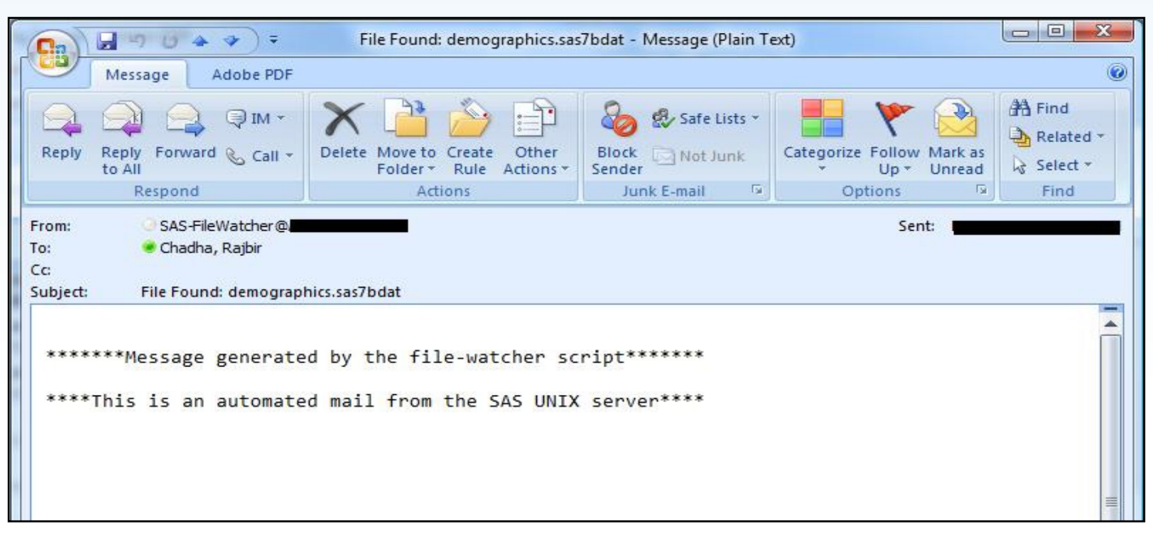
Sample E-mails & Logs

The figures alongside (**Figures 7, 8**) illustrate the e-mail message and the file-watcher log when the file is found. (**Figures 9, 10** depict the condition when file is not found) The log also reports the number of cycles that the file-watcher iterated before triggering the subsequent SAS process.

This example uses a delay of **7** seconds and runs for a maximum of **30** cycles.

The figure alongside (**Figure 11, 12**) illustrates a sample e-mail that contains and excel attachment. '**Uuencode**' is used to encode the excel file '**demographics.xls**'.

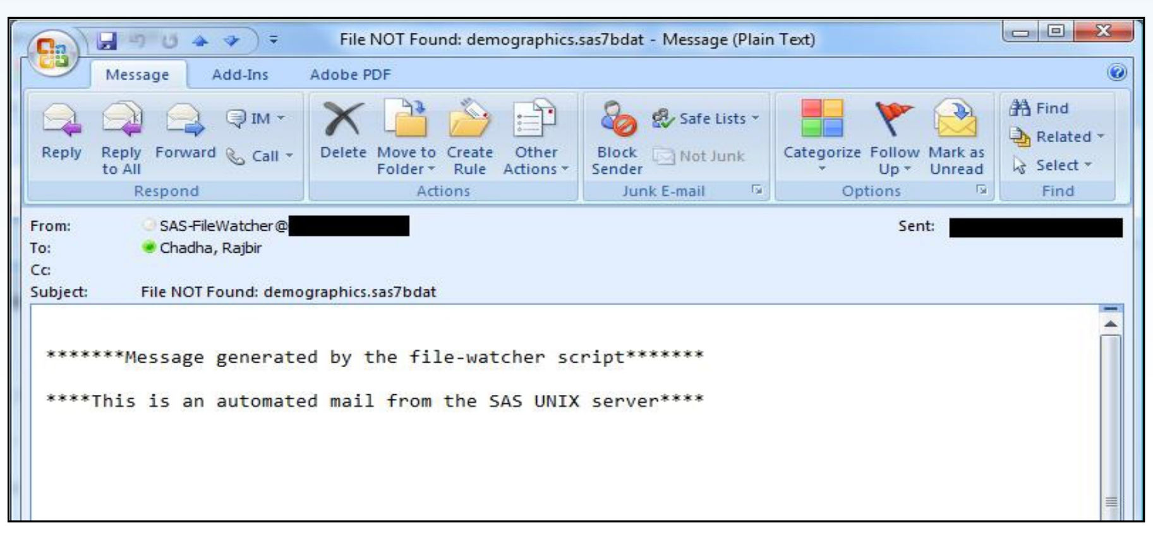
The encoded file is directed to an attachment. The '**listing**' output generated by the SAS program is also encoded using uuencode and included in the same attachment file. Attachments can be opened directly from the e-mail.


File Found: demographics.sas7bdat - Message (Plain Text)

From: SAS-FileWatcher@[redacted] Sent: [redacted]
To: Chadha, Rajbir
Cc:
Subject: File Found: demographics.sas7bdat

```
*******Message generated by the file-watcher script*******

****This is an automated mail from the SAS UNIX server****
```

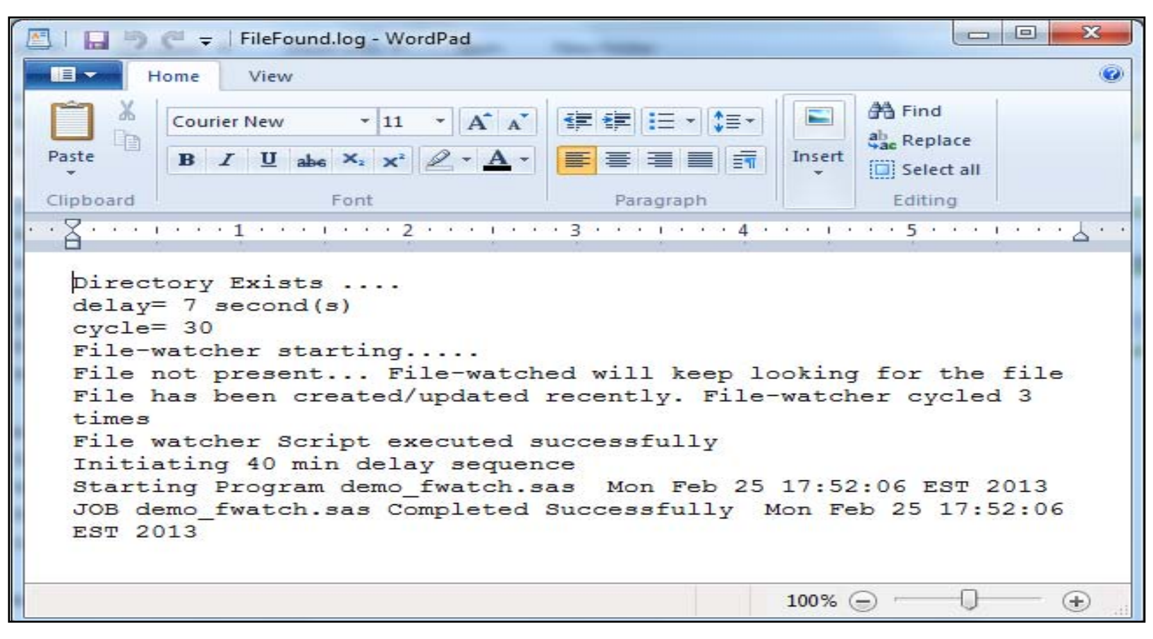
Figure 7. File-Watcher: File Found E-mail


File NOT Found: demographics.sas7bdat - Message (Plain Text)

From: SAS-FileWatcher@[redacted] Sent: [redacted]
To: Chadha, Rajbir
Cc:
Subject: File NOT Found: demographics.sas7bdat

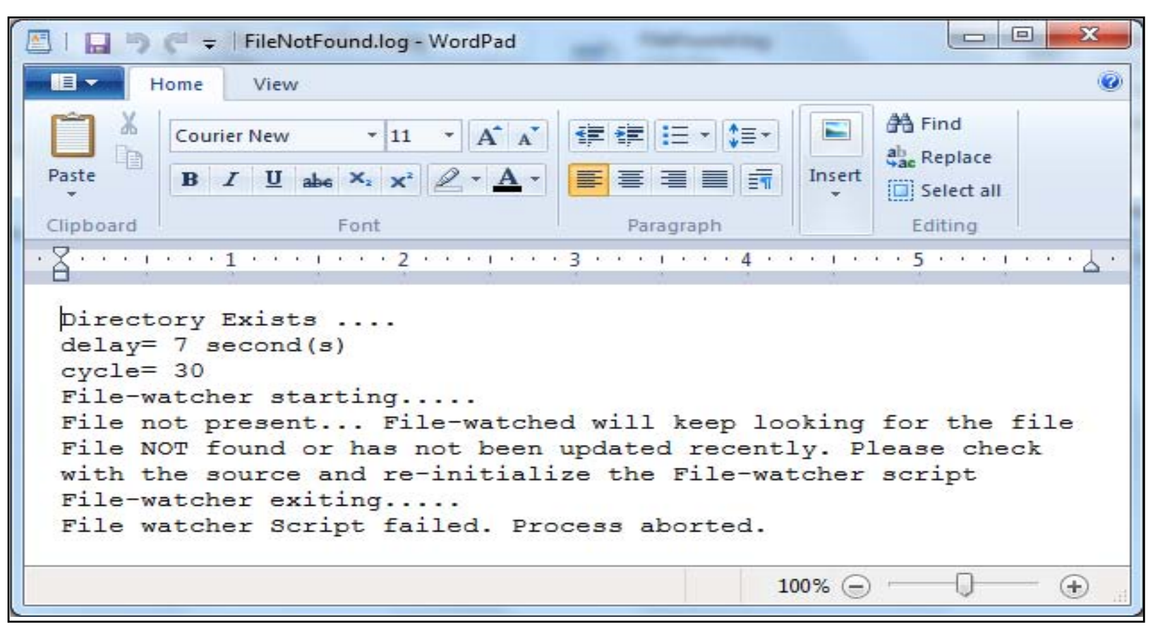
```
*******Message generated by the file-watcher script*******

****This is an automated mail from the SAS UNIX server****
```

Figure 9. File-Watcher: File Not Found E-mail


FileFound.log - WordPad

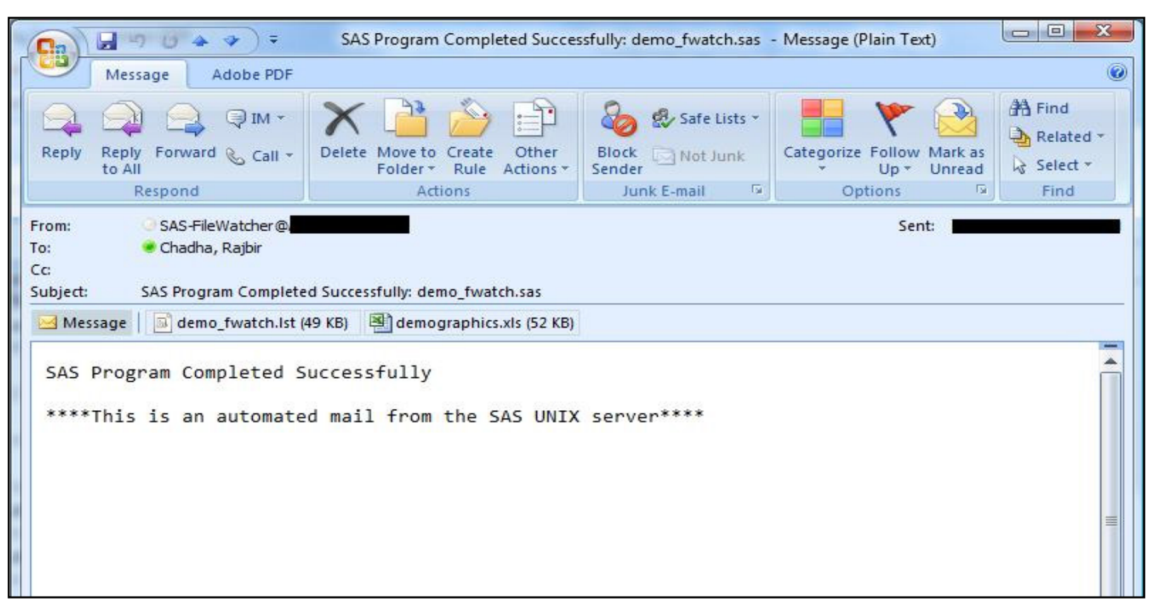
```
Directory Exists ....
delay= 7 second(s)
cycle= 30
File-watcher starting.....
File not present... File-watched will keep looking for the file
File has been created/updated recently. File-watcher cycled 3
times
File watcher Script executed successfully
Initiating 40 min delay sequence
Starting Program demo_fwatch.sas  Mon Feb 25 17:52:06 EST 2013
JOB demo_fwatch.sas Completed Successfully  Mon Feb 25 17:52:06
EST 2013
```

Figure 8. File-Watcher: File Found Log


FileNotFound.log - WordPad

```
Directory Exists ....
delay= 7 second(s)
cycle= 30
File-watcher starting.....
File not present... File-watched will keep looking for the file
File NOT found or has not been updated recently. Please check
with the source and re-initialize the File-watcher script
File-watcher exiting.....
File watcher Script failed. Process aborted.
```

Figure 10. File-Watcher: File Not Found Log

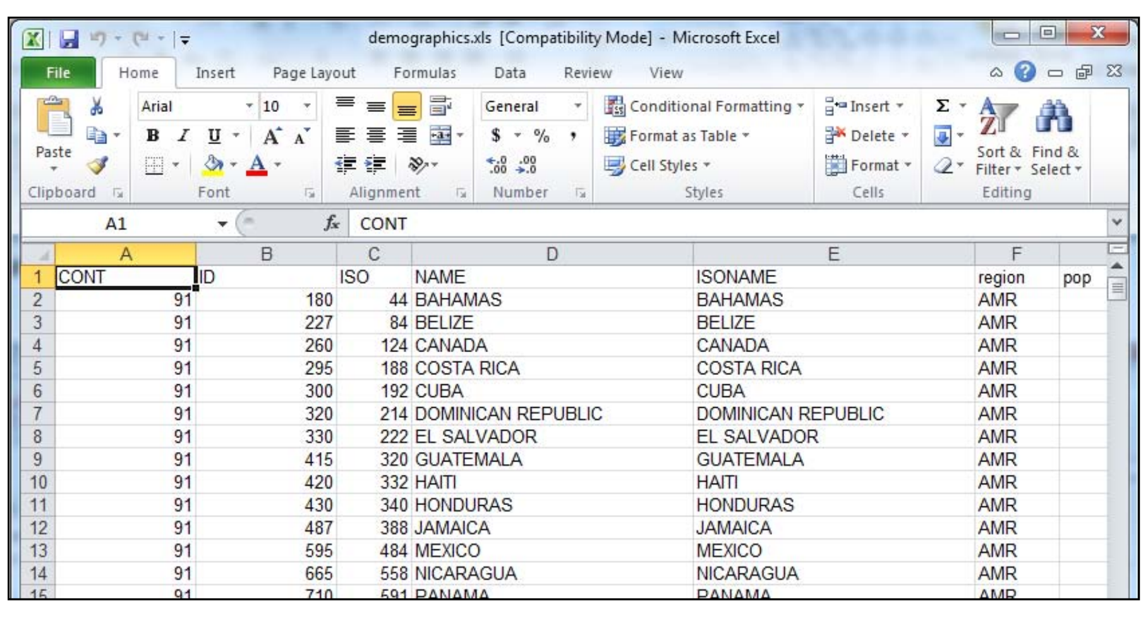

SAS Program Completed Successfully: demo_fwatch.sas - Message (Plain Text)

From: SAS-FileWatcher@[redacted] Sent: [redacted]
To: Chadha, Rajbir
Cc:
Subject: SAS Program Completed Successfully: demo_fwatch.sas

Message | demo_fwatch.lst (49 KB) | demographics.xls (52 KB)

```
SAS Program Completed Successfully

****This is an automated mail from the SAS UNIX server****
```

Figure 11. Notification E-Mail with Attachments


demographics.xls [Compatibility Mode] - Microsoft Excel

A1 fx CONT

	A	B	C	D	E	F	
1	CONT	ID	ISO	NAME	ISONAME	region	pop
2	91	180	44	BAHAMAS	BAHAMAS	AMR	
3	91	227	84	BELIZE	BELIZE	AMR	
4	91	260	124	CANADA	CANADA	AMR	
5	91	295	188	COSTA RICA	COSTA RICA	AMR	
6	91	300	192	CUBA	CUBA	AMR	
7	91	320	214	DOMINICAN REPUBLIC	DOMINICAN REPUBLIC	AMR	
8	91	330	222	EL SALVADOR	EL SALVADOR	AMR	
9	91	415	320	GUATEMALA	GUATEMALA	AMR	
10	91	420	332	HAITI	HAITI	AMR	
11	91	430	340	HONDURAS	HONDURAS	AMR	
12	91	487	388	JAMAICA	JAMAICA	AMR	
13	91	595	484	MEXICO	MEXICO	AMR	
14	91	665	558	NICARAGUA	NICARAGUA	AMR	
15	91	710	591	PANAMA	PANAMA	AMR	

Figure 12. Attachment opened from the E-mail

CRONTAB Utility

CRON is a **UNIX** based scheduler that enables users to schedule processes and run them at a pre-set time and date. **CRONTAB** is a file used to maintain jobs in CRON. Each line in the CRONTAB file represents a CRON job.

CRONTAB can be accessed from the UNIX command line using some basic commands. It typically uses vi-editor commands to edit/modify the scheduled jobs. CRONTAB file has five fields for specifying day , date and time followed by the command to be run at that interval.

The permissions to create a CRON file or schedule a CRON job are handled by a set of files in UNIX – '**/etc/cron.allow'** and '**/etc/cron.deny'**.

Conclusion

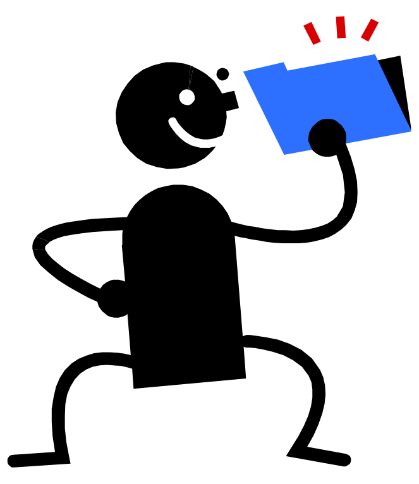
File-Watcher tool **reduces wait times and effort** on part of the users by **automating** the **program triggers**, allowing them to focus on other pressing tasks. The code is **very flexible** and can be modified to fit any **UNIX SAS environment**. E-mail triggers can be added to the script to **notify users** once their programs have executed and even include the **log file or list report in the e-mail**. Multiple programs can be **triggered either simultaneously or in a sequence**, without any manual intervention, thus **saving time** and **other valuable resources**.

References

- **"CRONTAB Command"**. © Copyright IBM Corporation 2010, 2012. Available at http://pic.dhe.ibm.com/infocenter/aix/v7r1/index.jsp?topic=%2Fcom.ibm.aix.cmds%2Fdoc%2Faixcmds1%2Fcrontab.htm
- **"CRON"**. Wikipedia®, the free encyclopedia. Wikimedia Foundation, Inc. (February, 2013). Available at http://en.wikipedia.org/wiki/Cron
- Paida, Vinodh and Pandya, Niraj J. 2011. Proceedings of the PharmaSUG 2011 Conference, Nashville, TN (May, 2011). **"Let the system do the work! Automate your SAS code execution on UNIX and Windows platforms"**. Available at http://www.pharmasug.org/proceedings/2011/AD/PharmaSUG-2011-AD11.pdf

Contact Information

Your comments and questions are valued and encouraged. Contact the author at:



Rajbir Chadha
Cognizant Technology Solutions
Wilmington, DE
Phone: 1 (302) 290-6130
E-mail: Rajbir.Chadha@Cognizant.com / Rajbir.Chadha@Gmail.com

Cognizant™