

```

1 #!/bin/bash
2 # -----
3 #
4 # (C) Copyright European Space Agency, 2018
5 #             European Space Operations Centre
6 #             Darmstadt, Germany
7 #
8 # -----
9 #
10 # Sub-System    : REALS
11 #
12 # File Name     : reals_cronjob.sh
13 #
14 # Author        : Lorenzo Maria Gagliardini
15 #
16 # Description   : This script is executed from the Linux cron system as a
17 #                   cronjob. It will process all the generated events codified
18 #                   as files.
19 #
20 # -----
21 #
22 #                         NOTA BENE
23 #
24 #-----
25 #####!!!!!!#####!!!!!!#####!!!!!!#####!!!!!!#####!!!!!!#####!!!!!!#####
26 #####
27 #TENTATIVO reals_cronjob.sh DA TRASFERIRE IN      home/se2ops/REALS/scripts !!!!!!!
28 #####
29 #####!!!!!!#####!!!!!!#####!!!!!!#####!!!!!!#####!!!!!!#####
30 #####
31 #
32 #
33 # Detecting REALS variables defining file locations (MISSION SPECIFIC)
34 reals_config_file="reals_config.cfg"
35 reals_script_path=`dirname $0`..
36 reals_config_path="/home/se2ops/REALS"
37 Latest_Whitelists_folder_path="$reals_config_path/Latest_Whitelists"
38 whitelist_prefix="whitelist_"
39
40 if [ -f "$reals_config_path/$reals_config_file" ]; then
41     reals_config_path="$reals_config_path/$reals_config_file"
42 elif [ -f "$reals_script_path/config/$reals_config_file" ]; then
43     reals_config_path="$reals_script_path/config/$reals_config_file"
44 else
45     echo "Cannot find '$reals_config_file'... Exiting..."
46     exit 1
47 fi
48
49 # Including REALS common variables file
50 echo $reals_config_path
51 . $reals_config_path
52
53
54 # Checking correct user
55 check_reals_user
56
57 debug_message "-- INIT EXECUTION --"
58
59 # Check cronjob running
60 cronjob_status=$(check_cronjob_status)
61 if [ "$cronjob_status" == "1" ]; then
62     debug_message "-- Another cronjob task is already running. Exiting !"
63     exit 1
64 fi
65
66 set_cronjob_running
67
68
69 debug_message "Checking if REALS service is running..."
70
71 # Check if the REALS service is actually running
72 reals_running=$(check_reals_status)
73 if [ "$reals_running" != "1" ]; then

```

```

74     debug_message "REALS service is not running. Aborting script."
75     set_cronjob_not_running
76     exit 1
77 fi
78 debug_message "REALS service is running"
79
80
81 # Starting execution
82 debug_message "Using python command: '$PYTHON_COMMAND'"
83
84 cd $REALS_HOME
85
86 debug_message "Using REALS logfile '$REALS_LOGFILE'"
87 debug_message "Folder to read event files: '$REALS_EVENTS'"
88
89 pattern_search=$(getEventFilenameReady $EVENT_FILE_PATTERN)
90 echo "Pattern search: '$pattern_search' "
91
92 debug_message "Will search for files starting with: '$pattern_search'"
93
94 if [ "$SEND_SMS" == "1" ]; then
95     debug_message "SMS sending option is activated and will use '$SMS_CLIENT_NAME' as
client"
96 else
97     debug_message "SMS sending option is deactivated"
98 fi
99
100 debug_message "Time to sleep between sending of event: $TIME_SLEEP seconds"
101 debug_message "Attribute to read regular expressions from config file:
'$EXPRESSION_ATTRIBUTE'"
102 debug_message "Delete event file after parsing is set to: $DELETE_AFTER_PARSING"
103
104
105
106
107
108
109
110 list_files=`ls -rt $REALS_EVENTS/$pattern_search* 2> /dev/null`
111
112 if [ "$list_files" == "" ]; then
113     debug_message "No event files found to be processed in configured folder !"
114     for file in "Latest_Whitelists_folder_path/*"; do
115         if [[ $file == *"/$whitelist_prefix"* ]]; then
116             var=`echo $file | rev | cut -d "/" -f1 | rev`
117             old_dir="$reals_config_path/scripts/$var"
118             cp -f $file $old_dir
119         fi
120     done
121     set_cronjob_not_running
122     exit 1
123 fi
124
125
126
127
128
129 # Parsing files
130 for file in $list_files; do
131     echo "File: '$file' "
132
133     debug_message "Parsing file: '$file'"
134     domain_file=`echo $file | cut -d '_' -f2`
135     debug_message "-- File is for domain: $domain_file"
136
137
138     # Checking if this domain must be avoided
139     avoid="0"
140     for domainAvoid in $AVOID_DOMAINS
141     do
142         if [ "$domainAvoid" == "$domain_file" ]; then
143             avoid="1"
144             debug_message "-- Found domain $domainAvoid inside the list of domains to be

```

```

145     avoided"
146     break
147   fi
148   done
149
150   if [ "$avoid" == "1" ]; then
151     debug_message "-- Domain $domainAvoid is indicated to be avoided. Skipping
152     file..."
153     continue
154   fi
155
156   # If this domain must be notified we continue
157   eventMessage=`cat $file`
158   debug_message "-- Message inside: '$eventMessage'"
159
160   #
161   #-----#
162   # Obtaining port and config file to use for this domain (domain_file contains
163   # the domain number 0/1/2..etc)
164   #
165   #-----#
166
167   portDomain=$(getPortFromDomain $domain_file)
168   configDirDomain=$(getConfigDirFromDomain $domain_file)
169   configFileDomain=$(getConfigFileFromDomain $domain_file)
170
171   debug_message "-- Obtained port to send event is '$portDomain'"
172   debug_message "-- Obtained config directory for checking event expressions is
173   '$configDirDomain'"
174   debug_message "-- Obtained config file for checking event expressions is
175   '$configFileDomain'"
176
177   #
178   #-----#
179   #full_path_config points exactly the testmission.properties file for the
180   #relative domain (e.g.
181   /home/se2ops/REALS/generated-events/0/Testmission_0_Action.properties)
182
183   #
184   #-----#
185   full_path_config="$configDirDomain""$configFileDomain"
186   echo "Domain file: '$domain_file' "
187   echo "Full path config: '$full_path_config' "
188
189   # Checking available regular expressions in action config file
190   debug_message "-- Reading regular expressions from config file..."
191
192   #
193   #-----#
194   #grep $EXPRESSION_ATTRIBUTE $full_path_config searches for "\.match" in the
195   #Testmission_0_Action.properties, once found..
196   #grep -v "#" searches for all the lines content in
197   #Testmission_0_Action.properties that are NOT commented, once found..
198   #cut -d='-' -f2 cuts the selected line in two parts, divided by "=" and takes the
199   #second (e. g. (\d{4})\.\d{3}\.\d{2}\.\d{2}\.\d{3}) (\w+)
200   #Current state: (NOMINAL) value: (\d+) .+
201   #sed 's/ /#/g' substitutes (s) all the spaces (/ /) with a replacement (#) for
202   #ALL (g) the spaces found and puts all into "expression".
203
204   #
205   #-----#
206
207   expressions=`grep $EXPRESSION_ATTRIBUTE $full_path_config | grep -v "#" | cut
208   -d='-' -f2 | sed 's/ /#/g'`
209
210   if [ "$expressions" == "" ]; then
211     debug_message "-- No regular expressions found in config file with
212     configured attribute !"
213     set_cronjob_not_running
214     exit 1
215   fi

```

```

194
195
196
197     # Checking if generated message matches any of the regular expressions
198     found="0"
199
200     for exp in $expressions; do
201         expConverted=`echo $exp | sed 's/#/ /g'`
202
203         # Checking regular expression
204         result=`$PYTHON_COMMAND -c 'import re; expConverted("'"$expConverted"'"; eventMessage='"$eventMessage"'; print re.match(expConverted, eventMessage))'` 
205
206         debug_message "---- Checking expression '$expConverted' against event message
207         '$eventMessage' ..."
208         if [ "$result" != "None" ]; then
209             # Expression matches compared Event message
210             debug_message "----- MATCH"
211
212             if [ "$SEND_SMS" == "1" ]; then
213
214                 debug_message "----- Sending message to '$SMS_CLIENT_NAME'"
215                 debug_message "----- Arguments: '$portDomain', '$configFileDomain' and
216                 '$eventMessage'"
217
218                 if [ "$DEBUG" == "1" ] ; then
219                     $PYTHON_COMMAND $SMS_CLIENT_NAME "$portDomain" "$configFileDomain"
220                     "DEBUG $eventMessage" 1>> $DEBUG_LOGFILE 2>&1
221                 else
222                     $PYTHON_COMMAND $SMS_CLIENT_NAME "$portDomain" "$configFileDomain"
223                     "$eventMessage"
224
225                     fi
226
227                     else
228                         debug_message "----- Wont send SMS message as option is deactivated"
229                     fi
230
231                     log_message "Sent '$eventMessage' to REALS service in domain '$domain_file'
232                     on port '$portDomain'"
233
234                     found="1"
235                     break
236
237             else
238                 # Expression does not matches compared Event message
239                 debug_message "----- NO MATCH"
240
241             fi
242
243             done
244
245             if [ "$found" != "1" ]; then
246                 log_message "Not sent '$eventMessage' as no regular expression matched"
247                 debug_message "Not sent '$eventMessage' as no regular expression matched"
248                 # report that no expression matched
249                 fi
250
251             # Deleting processed file
252             if [ "$DELETE_AFTER_PARSING" == "1" ]; then
253                 debug_message "Deleting file: '$file'"
254                 rm -f $file
255
256             fi
257
258             $PYTHON_COMMAND -c "import time; time.sleep('$TIME_SLEEP')"
259         done
260
261         -----
262         #Ripristino il contenuto delle whitelists
263
264         for file in "Latest_Whitelists_folder_path/*"; do
265             if [[ $file = *"/$whitelist_prefix"* ]]; then
266                 var=`echo $file | rev | cut -d "/" -f1 | rev`
267                 old_dir="$reals_config_path/scripts/$var"
268                 cp -f $file $old_dir
269             fi
270         done

```

```
260 #-----  
261-----  
261 debug_message "-- END EXECUTION --"  
262 set_cronjob_not_running  
263-----  
264 exit 0  
265
```