

```

1 #!/bin/bash
2 #
3 #
4 #
5 # (C) Copyright European Space Agency, 2018
6 # European Space Operations Centre
7 # Darmstadt, Germany
8 #
9 # -----
10 #
11 # Sub-System : REALS
12 #
13 # File Name : reals_action_1_ool.sh
14 #
15 # Author : Lorenzo Maria Gagliardini
16 #
17 # Description : This script is executed from the SCOS-2000 ACTIONS module
18 # associated to the different events (see documentation).
19 #
20 # -----
21 #
22 # Detecting REALS common variables file location
23 reals_config_file="reals_config.cfg"
24 reals_script_path=`dirname $0`
25 reals_config_path="/home/se2ops/REALS"
26 domainName=`basename $0 | cut -d '_' -f3`           #extracting the domain number from
the script name (number 1).
27 handling=`basename $0 | cut -d '_' -f4 | cut -d '.' -f1`    #extracting the object
to handle (ool or obe) from the scripts name.
28 reals_whitelist_name="whitelist_"${domainName}"_"${handling}".txt"
29 reals_script_name=`basename $0`'
30 reals_file_path=`echo ${reals_script_path}/${reals_script_name}`'
31 reals_whitelist_path=`echo ${reals_script_path}/${reals_whitelist_name}`'
32
33 if [ -f "${reals_config_path}/${reals_config_file}" ]; then
34     reals_config_path="${reals_config_path}/${reals_config_file}"
35 elif [ -f "${reals_script_path}/config/${reals_config_file}" ]; then
36     reals_config_path="${reals_script_path}/config/${reals_config_file}"
37 else
38     echo "Cannot find '${reals_config_file}'... Exiting..."
39     exit 1
40 fi
41
42 # Including REALS common variables file
43 . ${reals_config_path}
44
45 #Filtering against a Whitelist (whitelist_ool_s2a.txt)
46
47 MSG_param_name=`echo $@ | xargs | cut -d' ' -f2`'
48 go_ahead=`grep -c -i -m 1 "$MSG_param_name" ${reals_whitelist_path}`'
49
50 if [ "$go_ahead" == "1" ]; then
51
52     WL_entire_line=`grep -A 0 -m 1 "$MSG_param_name" ${reals_whitelist_path}`'
53     WL_counter=`echo ${WL_entire_line} | rev | cut -d ' ' -f3 | rev`'
54     WL_reference=`echo ${WL_entire_line} | rev | cut -d ' ' -f4 | rev`'
55     WL_row_section=`echo ${WL_entire_line} | rev | cut -d ' ' -f5- | rev | sed "s/ /`'
/g"```#The content of ${WL_row_section} is changed to be divided by TAB, not by
SPACE.
56     requires_call=`echo ${WL_entire_line} | rev | cut -d ' ' -f2 |`'
rev`'                                #since it will be written into the Whielist (having
TAB separated format)
57     WL_counter_update=$[1+$WL_counter]
58     sed -i
59         "s/${WL_row_section}\t${WL_reference}\t${WL_counter}/${WL_row_section}\t${WL_reference}\t${WL_counter}_update/g" ${reals_whitelist_path}
60     #echo ${WL_row_section}
61     #echo ${WL_counter}
62     #echo ${WL_counter_update}
63
64     if [ "${WL_counter}" == "${WL_reference}" ]; then
65
66         # Including REALS common variables file

```

```

66     #. $reals_config_path
67     MSG_time_stamp=`echo $@ | cut -d ' ' -f1`
68     MSG_current_value=`echo echo $@ | rev | cut -d ' ' -f1 | rev`
69     WL_SubSystem=`echo $WL_entire_line | rev | cut -d ' ' -f5 | rev`
70     WL_param_descr=`echo $WL_entire_line | cut -d ' ' -f2`
71     WL_num_words=`echo "$WL_entire_line" | wc -w`
72
73     if [ "$WL_num_words" == "9" ]; then #If the WL line contains only 9
74         characters (the OOL is non NUMERICAL but a STATUS) missing HIGH VALUE and
75         UNIT.
76
77         WL_exp_state=`echo $WL_entire_line | cut -d ' ' -f4`
78         full_event="$MSG_time_stamp $MSG_param_name|$WL_param_descr| Expected
79         State: $WL_exp_state| Current State: $MSG_current_value| SubSystem:
80         $WL_SubSystem"
81
82     elif [ "$WL_num_words" == "10" ]; then #If the WL line contains 10
83         characters (the OOL is NUMERICAL but without unit) missing UNIT.
84
85         WL_exp_state=`echo $WL_entire_line | cut -d ' ' -f4,5`
86         full_event="$MSG_time_stamp $MSG_param_name|$WL_param_descr| Hard
87         Limits: <$WL_exp_state> | Current Value: $MSG_current_value| SubSystem:
88         $WL_SubSystem"
89
90     else
91
92         full_event=`echo $*`      #If the row pattern is unknown the message
93         content ($*) is simply copied into $full_event to become the SMS content.
94
95     fi
96
97     #now=`$PYTHON_COMMAND -c 'import time; print "".join((%.20f" %
98     time.time()).split("."))'`  

99     now=`date +"DOY%j_h%H_m%M_s%S_mi%3N"`
100
101     temp_filename="$EVENT_FILE_PATTERN""$domainName""_""$handling""_""$now""_""$RA
102     NDOM"".txt"
103     final_filename=$(getEventFilenameReady $temp_filename)
104
105     if [ "$requires_call" == "Y" ]; then
106
107         temp_filename="$EVENT_FILE_PATTERN""$domainName""_""$handling""_""$now""_"
108         "$RANDOM"".txt"
109         final_filename=$(getEventFilenameReady $temp_filename)
110         cd $REALS_EVENTS
111         echo "Sentinel requires call" > $temp_filename #This text (Sentinel
112         requires call) has to match the regex in the Testmission file.
113         chmod 777 $temp_filename
114         mv $temp_filename $final_filename
115     fi
116     exit 0
117
118     echo '
119     echo "Reference $WL_reference != Counter $WL_counter"
120     exit 0
121 fi

```

```
122
123 else
124     echo ''
125     echo 'Missing match/Missing whitelist'
126     exit 0
127
128
129 fi
130
```