Scripting_Report_Templates_Example

Introduction

This script may be used to graph signals in a report template. Instructions to use this script may be found here.

Sample Script

```
! ========================================================
! Set Parameters
! ========================================================

Prompt_For_Multiple_Pipeline_Parameter_Values
/PIPELINE_PARAMETER_NAME=PAGE_COUNT+COLUMN+ROW+EVENT_1+EVENT_2
 +SIGNAL_TYPE+SIGNAL_FOLDER+COLOR_R+COLOR_L
 +GRAPH_TRIAL_STYLE+GRAPH_MEAN+GRAPH_STDDEV
/DATATYPE=s+s+s+s+s+s+s+s+s+s+s+s+s
/DEFAULT_VALUE=1+1+1+HS+HS+LINK_MODEL_BASED+ORIGINAL+990000+109900+NULL+TRUE
+TRUE
! PAREMETER LIST EXPLAINED:
! PAGE_COUNT
!  Starting page
! COLUMN
!  Starting column
! ROW
!  Starting row
! EVENT
!  If graphing kinematics use "HS", if graphing kinetics use "ON"
!  There is a loop later in the script (::SIDE) which will loop through L &
R
! The colors are specified in HEX however only numbers (no letters) can be
specified in the Expression
!  To pick colors: http://www.w3schools.com/tags/ref_colorpicker.asp
!  The first value in the color MUST be 1 or greater (it cannot be 0)
!  Color information
!  990000 = Red
!  109900 = Green
!   100099 = Blue/Purple
!  If not using the "Set Pipeline Parameter From Expression" command to set
!   the line color, any value can be typed in the "LINE_COLOR" parameter of
the
!   "Make Line Graph" command
;
```

```
 Set_Pipeline_Parameter
! List of components that will be graphed
/PARAMETER_NAME=COMPONENT_LIST
/PARAMETER_VALUE=X+Y+Z
;

! ========================================================
! Indicate the signals being graphed using prompt
!  (1) kinematic (2) kinetic or (3) all
! ========================================================

Set_Pipeline_Parameter
! List of signals that will be graphed
!  The SIDE parameter will specify whether the signal is L or R
/PARAMETER_NAME=KINEMATIC_SEGMENTS
/PARAMETER_VALUE=AnkleAngle+KneeAngle+HipAngle
;

Set_Pipeline_Parameter
! List of signals that will be graphed
!  The SIDE parameter will specify whether the signal is L or R
/PARAMETER_NAME=KINETIC_SEGMENTS
/PARAMETER_VALUE=AnkleMoment + KneeMoment + HipMoment + AnklePower +
KneePower + HipPower
;

Set_Pipeline_Parameter
! List of signals that will be graphed
!  The SIDE parameter will specify whether the signal is L or R
/PARAMETER_NAME=ALL_SEGMENTS
/PARAMETER_VALUE=::KINEMATIC_SEGMENTS& + &::KINETIC_SEGMENTS&
;

Prompt_For_Pipeline_Parameter_Value
/PIPELINE_PARAMETER_NAME=SEGMENT_TYPE
/PROMPT=Select data type:
/DATA_TYPE=STRING+STRING+STRING
/DEFAULT_VALUE=KINEMATIC_SEGMENTS + KINETIC_SEGMENTS + ALL_SEGMENTS
;

Set_Pipeline_Parameter
! List of signals that will be graphed
!  The SIDE parameter will specify whether the signal is L or R
/PARAMETER_NAME=DATA_NAME_LIST
/PARAMETER_VALUE=&:&:&::SEGMENT_TYPE
/MULTI_PASS=TRUE
;

! ========================================================
! Set Counters
```

```
! ========================================================

Set_Pipeline_Parameter
! Set the original column number (this will be incremented at the end of
each SIDE loop)
! At the end of each SIDE loop, the column will be set back to this value
/PARAMETER_NAME=COLUMN_INITIAL
/PARAMETER_VALUE=::COLUMN
;

Set_Pipeline_Parameter
! Set the original row number (this will be incremented at the end of each
COMPONENT loop)
! At the end of each COMPONENT loop, the row will be set back to this value
/PARAMETER_NAME=ROW_INITIAL
/PARAMETER_VALUE=::ROW
;

! ========================================================
! Begin Graphing
! ========================================================

For_Each
! A new page will be created for each signal in the DATA_NAME_LIST
/ITERATION_PARAMETER_NAME=DATA_NAME
/ITEMS=::DATA_NAME_LIST
;

Set_Report_Page_Title
! The title of the current page will be set to the current signal name
/PAGE_NUMBER=::PAGE_COUNT
/PAGE_TITLE=::DATA_NAME
;

For_Each
! For each side (left/right) a column of graphs - the column counter will be
incremented at the end of this loop
/ITERATION_PARAMETER_NAME=SIDE
/ITEMS=L+R
;

Set_Pipeline_Parameter_From_Expression
! This command is comparing the current side (::SIDE) to determine the color
of the graph
!  If the items in the SIDE list are changed the comparisons will need to be
changed as well ("R" and "L")
! The colors are specified in HEX however only numbers (no letters) can be
specified in the Expression
!  To pick colors: http://www.w3schools.com/tags/ref_colorpicker.asp
/PARAMETER_NAME=COLOR
/EXPRESSION=( "&::SIDE&" == "R" ) * &::COLOR_R& + ( "&::SIDE&" == "L" ) *
```

```
&::COLOR_L&
/AS_INTEGER=TRUE
;

For_Each
! Each component (X,Y,Z) will be added to a new graph since the row number
will be incremented at the end of each loop
/ITERATION_PARAMETER_NAME=COMPONENT
/ITEMS=::COMPONENT_LIST
;

Make_Line_Graph
/PAGE_NUMBER=::PAGE_COUNT
/COLUMN_NUMBER=::COLUMN
/ROW_NUMBER=::ROW
! /COLUMN_SPAN=1
! /ROW_SPAN=1
/GRAPH_TITLE=::SIDE&::DATA_NAME& - &::COMPONENT
! /FILES_TO_GRAPH=ALL_FILES
! /DISPLAY_LEGEND=FALSE
! /LEGEND_POSITION=UR
! /LEGEND_TEXT=
! /DISPLAY_LEGEND_TAG=FALSE
! /DISPLAY_LEGEND_FOLDER=FALSE
! /DISPLAY_LEGEND_SIGNAL=FALSE
/LINE_STYLE=::GRAPH_TRIAL_STYLE
/LINE_COLOR=#&::COLOR
! /LINE_BOLD=FALSE
! /SHOW_POINTS=FALSE
/GRAPH_MEAN=::GRAPH_MEAN
/GRAPH_STDDEV=::GRAPH_STDDEV
/MEAN_STDDEV_LINE_STYLE=Solid
/MEAN_STDDEV_LINE_COLOR=#&::COLOR
! /MEAN_STDDEV_LINE_BOLD=FALSE
/USE_EVENT_RANGE=TRUE
/START_EVENT_LABEL=::SIDE&::EVENT_1
/END_EVENT_LABEL=::SIDE&::EVENT_2
! /INTERMEDIATE_EVENTS=
! /EXCLUDE_EVENTS=
/X_AXIS_LABEL=% Gait Cycle
/X_DATA_TYPE=FRAME_NUMBERS
/X_DATA_NAME=FRAMES
/X_DATA_COMPONENT=0
/X_DATA_PROCESSED=ORIGINAL
! /X_AXIS_SCALE=Normalize 0% to 100%
! /X_AXIS_MIN=
! /X_AXIS_MAX=
! /Y_AXIS_LABEL=
/Y_DATA_TYPE=::SIGNAL_TYPE
/Y_DATA_NAME=::SIDE&::DATA_NAME
```

```
/Y_DATA_COMPONENT=::COMPONENT
/Y_DATA_PROCESSED=::SIGNAL_FOLDER
! /Y_AXIS_SCALE=Global Min to Max
! /Y_AXIS_MIN=
! /Y_AXIS_MAX=
! /Y_USE_SCIENTIFIC_NOTATION=FALSE
! /USE_P2D_STDDEV=FALSE
! /STDDEV_COMPONENT=3
! /USE_P2D_BARGRAPH=FALSE
! /P2D_BARGRAPH_POSITION=
! /P2D_BARGRAPH_WIDTH=
;

Set_Pipeline_Parameter_From_Expression
! At the end of each COMPONENT loop, a new row is created
! Allows each COMPONENT to have it's own row
/PARAMETER_NAME=ROW
/EXPRESSION=&::ROW& + 1
! /AS_INTEGER=TRUE
;

End_For_Each
! End COMPONENT loop
/ITERATION_PARAMETER_NAME=COMPONENT
;

Set_Pipeline_Parameter
! At the end of each SIDE loop, row counter is set back to original value
! Allows each side to start with a graph at the original row specified
/PARAMETER_NAME=ROW
/PARAMETER_VALUE=::ROW_INITIAL
;

Set_Pipeline_Parameter_From_Expression
! At the end of each SIDE loop, a new column is created
! Allows each side to have it's own column
/PARAMETER_NAME=COLUMN
/EXPRESSION=&::COLUMN& + 1
! /AS_INTEGER=TRUE
;

End_For_Each
! End SIDE loop
/ITERATION_PARAMETER_NAME=SIDE
;

Set_Pipeline_Parameter
! At the end of each DATA_NAME loop, column counter is set back to original
value
! Allows each page to start with a graph at the original column specified
/PARAMETER_NAME=COLUMN
```

```
/PARAMETER_VALUE=::COLUMN_INITIAL
;

Set_Pipeline_Parameter_From_Expression
! At the end of each DATA_NAME loop, a new page is created
! Allows each signal to have it's own page
/PARAMETER_NAME=PAGE_COUNT
/EXPRESSION=&::PAGE_COUNT& + 1
! /AS_INTEGER=TRUE
;

End_For_Each
! End DATA_NAME loop
/ITERATION_PARAMETER_NAME=DATA_NAME
;
```

From:
https://has-motion.com/wiki/ - **Wiki Documentation Site**

Permanent link:
https://has-motion.com/wiki/doku.php?id=visual3d:documentation:reports:scripting_report_templates_example&rev=1721149242

Last update: **2024/07/16 17:00**