

Automatically Start ADAMS/View With Your Own Macros

Jesper Slattergren
Mechanical Dynamics, Sweden
e-mail: Jesper_Slattergren@online.idg.se

Abstract

A general way to automatically start A/View with your own macros. It also shows a way to read in system parameters into A/View variables. Just locate the executable script somewhere in your search-path, and execute it. It will then start A/View and create your macros, panels and menus.

File: whatever (executable)

```
#!/bin/csh
setenv MDI_MACRO /own_macro_path
ln -s /own_macro_path/own_macro_collection.cmd ./aview.cmd
mdi -c aview ru-s i e
rm aview.cmd
```

File: own_macro_collection.cmd (reads files that reads macros, panels and menus
located in /own_macro_path)

```
!
!-----Read system variable for installation directory-----
!
system command=("echo variable create variable=own_macro_path string="\`$MDI_MACRO\`" >
tmp.tmp") &
send=off
file command read file="tmp.tmp"
system command="/bin/rm tmp.tmp" send=off
!
!-----READ MACROS-----
file command read file=(eval(own_macro_path)//"/macros.cmd")
!-----READ MENUS-----
file command read file=(eval(own_macro_path)//"/menus.cmd")
!-----READ PANELS-----
file command read file=(eval(own_macro_path)//"/panels.cmd")
```

An ADAMS/View Macro To Apply Initial Velocities To Parts in an ADAMS Model

Jesper Slattergren
Mechanical Dynamics, Sweden
e-mail: Jesper_Slattengren@online.idg.se

Abstract

A short macro that applies the same initial velocities to all parts in the model except ground. The macro might be modified not to set all initial velocities.

File icall.cmd (Reads macro, creates button and panel):

```
macro read macro_name=icall file_name="icall.mac" &
wrap_in_undo=yes create_panel=yes

menu create menu_name=.gui.pre_par_mod_rig &
label_string="INITIAL VELOCITY ALL PARTS"&
command_string="panel display panel_name=macro_icall"
```

File icall.mac (the macro itself):

```
!USER_ENTERED_COMMAND part modify rigid_body icallparts
!
! $nvx:t=real:u=0.0
! $nvy:t=real:u=0.0
! $nvz:t=real:u=0.0
! $nwx:t=real:u=0.0
! $nwy:t=real:u=0.0
! $nwz:t=real:u=0.0
!
!
FOR VARIABLE_NAME=loopvar OBJECT_NAMES="[^{ground}]*" &
TYPE=part

part modify rigid_body initial_velocity &
part_name=(eval(loopvar)) &
vx=$nvx vy=$nvy vz=$nvz &
wx=$nwx wy=$nwy wz=$nwz
END
```