here you will find a listing of the visual3d command categories, each with links to a full list of commands.

file commands

a list of all commands are found here: **file commands**

file commands are used on any files within the workspace or referenced by the pipeline. file commands are organized based on three categories:

- file_management: commands that act on files
- file open/import:commands for opening data files and command for importing signals into existing data files.
- file save/export: commands for saving the visual3d workspace and commands for exporting c3d files and signals.

event commands

a list of all commands are found here: **event commands**

event commands are used in the visual3d pipeline to identify and manage signal events in your data. event commands are separated into two categories:

- event creation: commands for creating events.
- event management: commands for managing events, such as renaming, copying, deleting, etc.

force commands

a list of all commands can be found here: **force commands**

force commands are used in the visual3d pipeline to process force signals from c3d files. force commands are separated into four categories:

- force platforms: commands for creating/modifying force platforms
- force structures: commands for creating/modifying force structures
- force assignment: commands for creating/modifying force assignments
- miscellaneous: additional force commands

metric commands

a list of all commands can be found here: **metric commands**

metric commands are used in the visual3d pipeline to calculate, compute, and find discrete quantitative values, known as metrics. these commands are separated into four categories:

- metric computation: advanced computation commands.
- metric math: mathematical commands.
- metric events: metric values about events, or found at or between events.
- miscellanous: other metric commands.

signal commands

a list of all commands can be found here: **signal commands**

signal commands can be used in the visual3d pipeline on all signal types (eg., forces, linke_model_based, etc.), when specified in the command. signal commands are separated into four main categories:

- signal math: mathematical commands for any signal (time-based or not).
- signal data/management: organizing signal folders, names, and data.
- signal filter: filtering signals.
- signal process: other signal processing commands.

model commands

a list of all commands can be found here: **model commands**

model commands are used to build and modify models applied to c3d motion files. model commands are separated into three main categories:

- model management
- model building
- model based data computation

real time commands

a list of all commands can be found here: **real time commands**

report commands

a list of all commands can be found here: **report commands**

report commands are used to control the 'reports' page in the visual3d workspace. report commands are separated into three main categories:

- report format: modify the template of your report.
- report graph format: modify the data and format in reported graphs.
- export data: exporting data, graphs, templates and reports.

pipeline control commands

a list of all commands can be found here: **pipeline control commands**

pipeline control commands mainly use parameters to store multiple text strings or numerical values, making it easy to loop through different signals or files for a series of commands. these control commands set, prompt, and read parameters for easy pipeline control.

model based data computation

the command compute_model_based_data will compute a signal using a model's segment, kinetic_kinematic data. details and examples of this command can be found here: **compute_model_based_data**

other commands

a full list of miscellaneous commands can be found here: **other commands**

meta commands

a meta-command is a visual3d pipeline into which parameters can be passed. the meta-command should be stored in a folder labeled meta-commands, which is located in the visual3d plugins folder. the header in the meta-command defines the data that should be passed. for more details and examples, see here: **meta commands**.

plugins

visual3d has a modest scripting language that allows automated processing of data. using scripts, custom data processing (that is not available through the graphical user interface) has been implemented for selected customers. these are placed in the plugins folder in visual3d's program files. more details about plugins are found here: **pipeline commands:plugins**

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

Permanent link:
https://has-motion.com/wiki/doku.php?id=visual3d:documentation:pipeline:general_information:pipeline_commands_reference&rev=1718804254

Last update: **2024/06/19 13:37**