Language: **English** • français • italiano • português • español ****

===== Contents =====

Objectives

The objective of this Tutorial is to demonstrate the Command Processing Pipeline. Tutorial #3 has two main objectives:

1. Teach you how to use the command processing pipeline
2. Introduce you to working with Visual3D pipeline commands.

Overview

The Command Processing Pipeline provides access to the core of Visual3D’s functionality by providing a command line interface into all of Visual3D's functions. The Pipeline is typically used to automate processing steps, which is useful for multiple, repeated tasks. The Pipeline is a set of Visual3D commands that are processed in sequence. The Pipeline has the ability to manage files, define events, execute signal processing computations, create and edit models, and, create and modify reports. The pipeline processor can be launched from the Pipeline menu option or from the Visual3D toolbar. It is not a scripting language nor a programming language, so no special scripting or programming experience is required to use the Pipeline. It is simply a series of commands that are run sequentially.

Any pipeline command that can be run interactively through the pipeline processor may be saved to a text file. This file may be reloaded later, or combined with other command files to build a more complex pipeline. The Pipeline dialog (or any text processor) can be used to create the pipeline.

Prerequisites

This tutorial assumes that Visual3D has been installed and that a model has been created and movement data has been applied to the model. It is suggested that you first complete the previous Tutorial: Visualizing Data; the file Tutorial2.cmo contains the results from that previous tutorial and can be used as the starting material here.

Preparation

1. In the File menu, select Open and select Tutorial2.cmo. This is the file resulting from Tutorial #2, which you either created through Tutorial #2 or downloaded complete from the C-Motion website.
2. Click on Signal and Event Processing to visualize the animation of the model based on the movement data and the model that was applied to it. If the animation does not appear in the 3D Animation viewer, check the active file combo box on the toolbar. It should read 'Walking Trial 1.c3d' rather than ALL_FILES
3. If the animation is not playing, click on the PLAY button of the VCR controls at the bottom of the screen.
4. There are many viewing options for the Animation viewer available under the View Menu item or by clicking with the Right Mouse Button in the Animation Viewer itself. You should play around with these options to see the effects they have; most of the effects are intuitively obvious.

Overview of Command Menu

To open the command menu, click on the "Pipeline" button on the toolbar or select "workshop" from the Pipeline Menu.

The pipeline workshop will open. To orient you, commands are located in the left column. The commands are organized into folders for:

- File Import/Export and Management
- Event/Creation and Management
- Metric Creation
- Model Building
- Computing Model Based Data
- Signal Processing and Math
- Reports

Commands can be moved over into the main pipeline (middle column) by using the **ADD»** (double arrow) buttons and can be reordered in any sequence. The right most column will show a preview of the command when the command is highlighted. Any command editing can be done here. Open, Save and other execution buttons are on the top. The tutorial will walk you through the operation of the Pipeline Workshop.

Editing a Command

The pipeline commands can be edited to customize processing. A quick overview for the command syntax follows.

Pipeline Command Syntax:

- Each Pipeline command consists of two parts - the command and its parameters.

Command_Name /Parameter1= something **;**

- The generalized command above passes one parameter to the command **Command_Name**.
- Each parameter begins with a slash followed by the parameter name, followed by an equals sign, then the parameter.
- The command is terminated with a semicolon.
- Comments start with “!”. It is good practice to document command purpose and parameter settings with comment lines
- A command parameter prefixed with an “!” is assigned it's default value.
- White spaces are ignored
- To pass more than one set of parameters to the command, list the values of each parameter, separated by “+” signs (e.g. Parameter 2 below).

/Parameter2= this_value + another_value

- Optional parameters are commented out - thus they start with “!”.
- If the Parameter is not listed in the command or if the “!” character is at the beginning of a line, the parameter value is ignored, and the default values for the parameter are used in the processing.

Command_Name /Parameter1= something **!/Parameter2**= **;**

- When a command is added to the pipeline, all parameters are listed, so the user doesn’t have to continually check the documentation for the syntax or the default values. The script text is provided so that these optional parameters may be un-commented and additional parameter values added.
- When parameters **Signal_Name**, **Signal_Type**, and **Signal_Folder** are included in the command, signals that are checked in the data tree can be imported into the command at the push of the **Import Selected Signals** button.

Load 動画

Executing the Pipeline

The following examples are a brief demonstration of constructing and executing a simple pipeline. Clicking the **Execute Pipeline** button will cause all of the commands in the middle list box of the Pipeline window to be executed in sequence. When the pipeline has finished a status screen will pop-up listing the commands that were processed, and any warnings or errors that were generated.

Note: The pipeline will continue to process even if there were errors in any of the commands, so it is important that you check the status screen at the end of the processing for any errors.

If you would like the Pipeline processing to halt after the first error. Select the **Halt on First Error** checkbox located beside the **Execute Pipeline** button. If you would like to execute the pipeline commands one command at a time, then click the **Step** button. This is useful when debugging. If you would like to execute a series of commands, select the group of commands, then click the **Step** button.

Example 1 - Execute File_Open Command

Select the File_Open command from the list of commands within the **File Open/Import** folder from the command tree on the left.

Click **ADD»** The following command will be added to the pipeline. **File_Open** The following will be added to the parameters frame.

```
File_Open
!/FILE_NAME=
;
```

Note: The default value for the *FILE_NAME* parameter is a blank and in addition the parameter is commented out by default with the leading exclamation point which means it will be ignored anyway. Either way the result is that the user will be prompted to select a file when the pipeline is executing.

Click **Execute Pipeline**

The *Open Movement Files* dialog will appear. Browse and select a c3d file of your choice. The c3d file will be opened in the workspace and the processing results window will show you that the pipeline

was processed with out error.

Verify that the **Workspace Status** tab shows that the file you selected was indeed opened.

Example 2 - Execute File_New and File_Open Commands

Select the File_New command from the list of commands within the **File Management** Folder. **Note:** The **File_New** command clears the Visual3D Workspace.

Click **ADD»**. The *File_New* command is added to the pipeline but it is placed below the File_Open command so it will need to be promoted or moved up.

Click the promote command button (shown in red below) on the right edge of the pipeline to move the *File_New* command up one level above the *File_Open* command. Your pipeline should look like the following:

Click **Execute Pipeline.**

The *Open Movement Files* dialog will appear again. Choose a file that is different from the one previously opened.

Verify that the **Workspace Status** tab shows that the file you selected was indeed opened and that the other file was removed from the **Workspace Status** tab.

Pipeline Commands and the Active_Files

File selection is important in Visual3D because many files can be open simultaneously. As mentioned in earlier tutorials there is a file selection combo box at the top of the Visual3D user interface (upper right corner). Pipeline commands usually process the “Active Files”; e.g. those files selected in the combo box on the Visual3D toolbar or designated in a command parameter.

"Active Files" Selection in the Combo Box

- If only one file is selected in the file selection box, the pipeline, when executed, will perform actions only on the data from that file. (i.e. walk1.c3d)
- If **ALL_FILES** is selected, the script will perform actions on every file individually, as if the script had been run sequentially for each file.
- If a **TAG** (see next section) is selected, the script will perform actions on every file with that TAG individually, as if the script had been run sequentially for each file. (i.e. backward)

"Active Files" Selection by Command Parameter

- Commands that include the parameter **FILE_NAME** explicitly act on that file rather the Active Files.

File_Open /FILE_NAME= C:\MyData\walk 1.c3d **;**

- The user can control the Active Files in the Pipeline by adding the command **Select_Active_File** to the Pipeline. Example 3 will illustrate.

Example 3 - Select_Active_File Command

Add the Select_Active_File command to the previous pipeline.

Select the Select_Active_File command from the list of commands within the **File Management** folder in the command tree on the left.

Click **ADD»**.

Click **Execute Pipeline**.

Select a motion file when prompted to do so. When the Pipeline is finished executing the status dialog should look like the following: **Note:** The error message after the *Select_Active_File* command indicates that the command was not executed properly.

To edit the *Select_Active_File* command. Select the *Select_Active_File* command in the main pipeline and click on the **Edit** button below the parameter frame.

In the Edit window, remove the leading exclaimation point from the */FILE_NAME* parameter.

To the */FILE_NAME* parameter add a specific file name with the path as shown below. For example *C:Documents and SettingsC-MotionMy DocumentsTutorialsTutorial 2Walking Trial 1.c3d not just Walking Trial 1.c3d.*

For this tutorial, add C:\demo files\tutorials\Walking Trial 1.c3d to the */FILE_NAME* parameter

Click **OK.**

Click **Execute Pipeline.**

This time select *Tutorial2.cmo* when prompted to select the motion file.

If you get a file selection error or it appears that the file was not selected as requested, see the explanation below.

Challenges with identifying files

This is a potentially frustrating issue related to the Select_Active_File command (and many other commands), which is worth discussing before moving on because it confuses many users. The Filename must be the complete Filename as seen in the Visual3D Workspace. In Visual3D the complete Filename includes the path to the file. Visual3D uses the path to the file to determine the uniqueness of a file that is loaded in the Workspace. This allows users to use the same filenames for every data collection session (e.g. static, trial1, trial2, etc), distinguishing the files by the folder in which the files are stored on disk.

To cope with this difficulty add a path (show in italics) before the file name as shown below:

Select_Active_File /FILE_NAME= *C:\demo files tutorials*\Walking Trial 1.c3d **;** Many commands, however, allow the use of a wildcard that can circumvent the problem with the complete specification of the path. For example, the following change to the previous command will find all files whose name ends with Trial 1.c3d

Select_Active_File /FILE_NAME= *Trial 1.c3d **;**

Workspace TAGS

A tag is simply a user defined file attribute. Tags are provided as a method of classifying (or categorizing) files.

- Files can be associated with other files opened in the Visual3D workspace by assigning a tag to the file.
- For the Pipeline, if the tag is selected as the Active File, all Files of the tag are selected; Pipeline Commands will act on all of these files.
- The file tagging mechanism allows you to define specific subsets of your C3D files, to facilitate processing these as a group.
- A file can have multiple tags associated with it.
- A tag can have multiple files associated with it.

In the illustration below all files associated with trials in which the subjects walked barefoot have a tag labeled '**Barefoot**' assigned to them. And those that wore '**shoes**' have a tag labeled Shoes.

- The user has defined tags to distinguish movement trials in which the subject walked barefoot from those in which the subject wore shoes.
- The first two columns of the table, headed **Models/Calibration Files** and **Motion Files** are always present. The other two columns were added when the user defined two file tags: '**Barefoot**' and '**Shoes**'.
- The checkboxes identify which files are tagged and which are not.
- The relationship between movement trials and models is indicated graphically by the coloring of

the rows in the table.

You can also have more than one type of attribute. For example, all files associated with trials in which the subjects walked barefoot can have a tag labeled Barefoot assigned to them. And those trials that are are only women subjects could have another tag labeled Female. So you could select only the subjects that are women and who did not wear shoes.

As was mentioned before, if you make only these files active then they will be affected by the Pipeline. This greatly increases the processing power of Visual3D as you are allowed to selectively eliminate and choose various trials to process based on various attributes.

To assign a tag to a file via Workspace:

1. Open the **Workspace Status** Tab
2. Click the **Add New File Tag** button.
3. Type a tag name of your choice into the Edit Box that appears. This tag will then be added to the grid.
4. Check the box beside the files that you want assigned to that tag.

Example 4 - Assign_Tags_To_Files Command

Add the **Assign_Tags_To_Files** command to the previous pipeline.

- Select the **Assign_Tags_To_Files** command from the list of commands within the **File Management** folder in the command tree on the left.
- Click **ADD»**.
- To edit the *Assign_Tags_To_Files* command. Select the *Assign_Tags_To_Files* in the main pipeline and click on the **Edit** button below the parameter frame.
- In the Edit window, remove "!" in front of */MOTION_FILE_NAMES* parameter and enter "C:\demo files\tutorials\Walking Trial 1.c3d"
- Remove "!" in front of */TAGS*, enter "Walk"

```
Assign Tags To File
/MOTION_FILE_NAMES=C:\demo files\tutorials\Walking Trial 1.c3d
! /QUERY=
/TAGS=Walk
;
```

- Click **OK.**
- Click **Execute Pipeline.**
- This time select *Tutorial2.cmo* when prompted to select the motion file. The end result will be that the file Walking Trial 1.c3d will be tagged *Walk*

Deleting Tags

The mechanism for deleting file tags is not always intuitive: if no files are checked in a tag's column, the tag will be deleted as soon as you switch to a different page. If you accidentally delete a tag in this way, you can just create it again using the **Add New File Tag** button on the Workspace Status

page.

Pipeline PARAMETERS

An important feature of the pipeline is the ability to create and use global parameters. A global parameter is a way to store a text string for use in Pipeline commands. In one sense it is similar to specifying a global variable in a scripting language, such as body weight, that could be used in computations. It is actually much more flexible than that in the pipeline. The Visual3D pipeline commands permit multiple entries on a single line, and since the entire line can be represented as a string, a single global parameter can represent multiple entries. The use of global parameters will be described by example.

Example 5 - Set_Pipeline_Parameter_to_Folder_Path Command

The following commands will create a global parameter that will contain the Folder, and the Open_File command will use this parameter.

Select the **File_New** command from the list of commands within the **File Management** Folder.
Note: The **File_New** command clears the Visual3D Workspace.

Click **ADD».**

Select the **Set_Pipeline_Parameter_To_Folder_Path** command from the list of commands within the **Pipeline Control** Folder.

Click **ADD».**

Click the **Edit** button

After **/PARAMETER_NAME=** add *FOLDER*

Remove the leading exclamation point from **/PARAMETER_VALUE=**

After **/PARAMETER_VALUE=** add the path to your tutorials folder as shown below. For example *C:\Documents and Settings\C-Motion\My Documents\Tutorials*. **Note:** Be sure to include the trailing slash (\) after the path statement or the script will not execute properly.

Select the **File_Open** command from the list of commands within the **File Open/Import** Folder.

Click **ADD».**

Click the **Edit** Button.

Remove the leading exclamation point from **/FILE_NAME=**.

After **/FILE_NAME=** add *::FOLDER&*.c3d* **Note:** To use a global parameter in another command, the PARAMETER_NAME should be preceded by two colons (::).

Select the **Switch_to_Event_Processing_Mode** command from the list of commands within the **Other** Folder.

Click **ADD»**.

Click **Execute Pipeline.**

This example will produce similar results to previous pipelines but now multiple files will be opened and Visual3D should switch to the **Signal and Event Processing** tab.

Syntax for using a Pipeline Parameter in a another Pipeline Command:

The ampersand (&) indicates that the strings ”::FOLDER” and ”*.c3d” should be concatenated to produce

'**C:\Documents and Settings\C-Motion\My Documents\Tutorials*.c3d**'

Note: The command in the above example uses a wildcard to load all of the c3d files in a folder. Executing this pipeline will cause Visual3D to open the two c3d files that we used in Tutorial 1 and Tutorial 2, provided that the path name makes sense on your disk. To make the pipeline more general it is convenient to separate the FOLDER containing your data files from the command because in many cases it is only the name of the FOLDER that changes from subject to subject.

A folder is really a special case of a global parameter because it is used so often. If the PARAMETER_VALUE is left blank, on execution a browse dialog will appear to allow you to browse and select the folder. This eliminates the need to type in the FOLDER parameter for every subject. Through judicious use of parameter and wildcards very general pipelines can be developed that can be used as batch processing scripts for every day processing of data.

Pipeline SIGNAL PROCESSING

Signal and event processing will be covered in much greater detail in the next tutorial but we begin looking at it now because much Signal and Event Processing in Visual3D is done using the pipeline. For example a very common process in treating Motion Capture data is to Interpolate missing frames of data, then smooth the resulting signal. The next example will demonstrate this using the Pipeline.

Example 6 - Interpolate and Low Pass Filter Commands

For this exercise, you will need to graph the original X signal component of LAS before and after the example so that you can compare the original data to the revised data.

In the Data Tree, open on the **TARGET** folder

Open the **ORIGINAL** subfolder.

Right-click *LAS*, select **Graph X** and **New Graph**. This will create a *LAS X* graph on the right side of the screen.

On the main toolbar, click on the Pipeline button to Pipeline Processing Dialog.

Select Interpolate from the **Signal Process** command folder.

Click **Add»**

Select Lowpass_Filter from the **Signal Filter** command folder.

Click **Add»**

Highlight both the *Interpolate* and *Lowpass_Filter* commands in the pipeline.

In the Data Tree on the **Signal and Event Processing** tab, click the check box of the **ORIGINAL** folder to highlight all target names in that folder. All items should automatically be checked with a checkmark. This will select which signals to interpolate—in this case, all the targets. **Note:** Visual3D allows you to do this even with the Pipeline window open.

Click **Import Checked Signals from Tree** at the bottom of the Pipeline parameters box. This will automatically import the checked signals.

Click **Execute Pipeline** to interpolate and filter these signals.

A processing results box will appear, listing both the Interpolate and Lowpass Filter results. Verify that the number of errors is zero.

Click **Ok** to close the Pipeline.

Verify that a new folder labeled, **PROCESSED** is present in the Data Tree. **Note:** For subsequent signal processing, you must use signals from this processed list. Otherwise, you will simply overwrite the data you have already processed.

Open the **PROCESSED** folder.

Right-click *LAS*, select **Graph X** and **New Graph**. This will create a second *LAS X* graph on the right side of the screen.

Compare the 2 graphs.

Summary

Proceed to next Tutorial: Signal Processing

Retrieved from “”