

c3d format]] is a publicly defined format and is currently implemented by most motion capture vendors. data file stored in formats other than the c3d format must be translated into the c3d format. c-motion provides several plugin for translating data files. there are a wide variety of ascii formatted data files. visual3d will load many of these files into visual3d; translating them into c3d format in the process. it is quite a challenge to read all varieties of ascii formatted files, but we do our best to accommodate data that our users collect. if visual3d does not import your formatted files, contact support@c-motion.com and request that we implement your format. we recognize that the first task to analyze your data is to be able to load the files, so we try hard to make sure that users have their data. ascii files that can be imported. users that create their own custom file formats may be able to adapt their own file format fairly simply. if a user has an ascii file format that visual3d does not currently import, please contact support@c-motion.com and we will either create a file importer for you, or provide you with an sdk to create your own visual3d plugin. a universal ascii file converter may be downloaded from the [[https://www.c-motion.com/products/free-downloads/|c-motion free downloads]] page. documentation for the ascii file converter may be found on the [[index.php?title=file_converter&action=edit&redlink=1|file converter]] wiki page. ===== p2d_format ===== the p2d file format was part of the origin move3d software written by tom kepple in dr. steven stanhope's laboratory at the united states national institutes of health (nih) (<http://pdb.cc.nih.gov>). visual3d was the result of the technology transfer of the move3d software. ===== visual3d_ascii_format ===== visual3d ascii file format for importing and exporting signals visual3d exports data into several file formats. the simplest and default file format for data export to ascii files is the following: this data file can be exported from visual3d using the pipeline command [[visual3d:documentation:pipeline:file_commands:export_data_to_ascii_file|export_data_to_ascii_file]] this data file can be imported to visual3d using the pipeline command [[visual3d:documentation:pipeline:file_commands:import_data_from_ascii_file|import_data_from_ascii_file]] report data can be exported from the report tab [[#export_data_from_report_dialog|as described here]]. ===== fsv_format ===== the ****import_fsv_files**** visual3d plugin is a fairly general ascii file format translator to c3d, that may be suitable for a variety of ascii files. the fsv/tsv file formats were originally developed by manfred berger from innovision systems and are currently used by qualisys and innovision. ===== json_format ===== signal data can be exported to json format (javascript object notation) using the [[index.php?title=export_data_to_ascii&action=edit&redlink=1|export_data_to_ascii]] pipeline command. data from a report graph can be exported in json format using the [[#export_data_from_report_dialog|export data from report dialog]] option in the report tab. see also [[visual3d:documentation:visual_3d_formatting:json_formats|json formats]]. ===== kinemetrix ===== the **import_kinemetrix_files** visual3d plugin is an ascii file format translator to c3d. these files are produced by kinemetrics. this is a preliminary importer because we had only one example file to work with. ;-) ===== opening a text file ===== a visual3d ascii file can be opened in the visual3d workspace directly using the file open menu option or the file_open command. in order for visual3d to interpret this standalone file correctly, it is important that a signal be included specifying the time. |- |c:\exampledata\examplewalk.c3d|c:\exampledata\examplewalk.c3d|filename from which the data came.\\ e.g. filename withing the visual3d workspace| |- |analogtime |emg9 |signal name | |- |frame_numbers |analog |signal type | |- |original |original |signal folder (e.g. original, processed, etc.) | |item|0 |x |component. (analog signals have channel 0 or channel x | |1 |0.01 |23.0 |- | |2 |0.02 |22.5 |- | |3 |0.03 |23.0 |- | |4 |0.04 |22.0 |- | |5 |0.05 |21.0 |- | ===== importing data from an ascii file ===== visual3d currently imports signals from an ascii file. [[visual3d:documentation:pipeline:file_commands:import_data_from_ascii_file|import_data_from_ascii_file]] ****/file_name=c:\exampledata\test command input signals\testinputdata.txt\ **;** visual3d allows the importing of data to a c3d file from an ascii file. see the description of the default ascii file format for visual3d. one of the constraints on the file import is that the first line of the signal column must match an open c3d file. this allows visual3d to recognize which file to add the data to. for analog

and target (point) data the following rules apply: if there are more frames of data in the ascii file than there are in the c3d file, the ascii signal is truncated. if there are fewer frames of data in the ascii file than there are in the c3d file, the signal is padded with frames of data containing data_not_found. note: for derived signals all rows are added to the data file within visual3d without truncation or padding. this is partly because derived signals cannot be exported to the c3d file at this point, so there is no conflict with the c3d file format. ===== import ascii signals adds the signals to the derived folder in the data tree ===== i would like these signals added to the analog folder because i would like to export the c3d file with all of the data combined.===== there are two flavours of importing signals from an ascii file. the first uses the import ascii signals command in the pipeline. this is a legacy command that is used to import signals that have been processed by an external executable (e.g. matlab). these data are always imported as derived signals. the second uses custom import commands to merge the analog signals with the motion capture signals. there are many varieties of file formats that can be imported, but ascii files are pretty fragile (meaning they can cause system crashes quite easily). if you have signals that you need merged, please contact support@c-motion.com to confirm that your files can be imported. if not, we usually write new filters for importing data at no cost to our customers. ===== export signals to an ascii file using data tree ===== datatreeexportascii.gif\ select one of the signals in the data tree using a click of the left mouse button. a dialog appears that contains two tabs. one tab is a graph of the signal, the other tab is a list of the data values. selecting the export ascii button causes a browse window to appear to save the data to a visual3d ascii formatted data file. === pipeline command for exporting signals to an ascii file === the visual3d pipeline command exports ascii data to one of 2 formatted data files (visual3d default and p2d). for example:

```
[**[[visual3d:documentation:pipeline:file_commands:export_data_to_ascii_file|export_data_to_ascii_file]]**] | /signal_types=analog |signal type | /! /signal_folder=original |signal folder | /signal_names=emg8 |signal names | /file_name=c:\demo files\example1\test.txt |the filename (including path) for the exported data | /! /signal_components= |signal component. this is a legacy parameter, if using version 5 or later, please use component_sequence. | /component_sequence=x |signal component | /! /signal_precision= |indicates the number of decimal places | /! /start_label= |event label indicating the first frame of data to be exported. this is a legacy parameter, if using version 4 or later, please use event_sequence. | /! /end_label= |event label indicating the last frame of data to be exported. this is a legacy parameter, if using version 4 or later, please use event_sequence. | /! /event_sequence= |event sequence the signals should be exported between (ex. lhs+lhs,rhs+rhs) | /! /exclude_events= |if an event sequence contains one of the events listed in this parameter (ex. bad,bad) the event sequence will not be exported. | /! /use_point_rate=false |analog signals can be exported at either the analog rate or the point rate | /! /normalize_data=false |option to time normalize the data | /! /normalize_points=101 |number of output points from the time normalization | /! /export_mean_and_std_dev=false |option to export the mean and standard deviation | /! /use_p2d_format=false |see the description of p2d files in the documentation. | /! /use_xml_format=false | /! /use_json_format=false | /! /use_short_filename=false |the file path for each c3d file will not be exported when set to true | /! /export_empty_signals=false | /! /export_without_header=false |the headers (filename, signal type, signal folder, signal name, signal component) will not be exported when set to true. | **; ** | the following dialog appears when the export_data_to_ascii_file command has been added to the pipeline and the user double-clicks with the left mouse button on the command. exportascii.png \ ===== export data from report dialog ===== select one of the graphs in the edit report dialog. a button will appear below the graph list box titled "export selected". selecting this button will pop up a dialog for exporting the data from the graph to an ascii file; either a visual3d ascii formatted file or a p2d formatted file. the options exist to save the raw data (e.g. frame by frame) or to save time normalized data (e.g. a specified number of frames) using the properties of the graph template (e.g. start frame and end frame). the picture below shows a highlighted graph and selecting a
```

```

[[visual3d:documentation:visual_3d_formatting:json_formats|json output format]]. exportreport.png
===== export 6 degree of freedom data ===== it is possible to export 6 degree of freedom data to
an ascii format as either an origin and quaternion or as an origin and 3x3 rotation matrix. exporting
the position and orientation of segment rft as an origin and quaternion \\
[[visual3d:documentation:pipeline:file_commands:export_data_to_ascii_file|export_data_to_ascii_file]]
! /file_name= /signal_types=frame_numbers+link_model_based+link_model_based
/signal_names=time+origin+quaternion /signal_folder=original+rft+rft ! /start_label= ! /end_label= !
/use_point_rate=false ! /normalize_data=false ! /normalize_points=101 !
/export_mean_and_std_dev=false ! /use_p2d_format=false ; \\ exporting the position and orientation
of segment rft as an origin and rotation matrix. visual3d exports the following rotation matrix
[[visual3d:documentation:kinematics_and_kinetics:six_degrees_of_freedom#6_dof_tracking|[ xyz local
] = [r2] [xyz global]]] \\
[[visual3d:documentation:pipeline:file_commands:export_data_to_ascii_file|export_data_to_ascii_file]]
! /file_name= /signal_types=frame_numbers+link_model_based+link_model_based
/signal_names=time+origin+rotation /signal_folder=original+rft+rft ! /start_label= ! /end_label= !
/use_point_rate=false ! /normalize_data=false ! /normalize_points=101 !
/export_mean_and_std_dev=false ! /use_p2d_format=false ; ===== 6 dof ascii format ===== this
example file contains the origin and rotation matrix for the
[[visual3d:documentation:modeling:segments:segment_overview#default_segment_names|rsk
segment]] at the first 5 frames of data. note that the elements are tab delimited. |
c:\data\mvc002.c3d|c:\data\mvc002.c3d|c:\data\mvc002.c3d|c:\data\mvc002.c3d|c:\data\mvc002.c3d
c:\data\mvc002.c3d|c:\data\mvc002.c3d|c:\data\mvc002.c3d|c:\data\mvc002.c3d|c:\data\mvc002.c3d
c:\data\mvc002.c3d|c:\data\mvc002.c3d| | |origin |origin |origin |rotation |rotation |rotation |rotation
|rotation |rotation |rotation |rotation |rotation | | |kinetic_kinematic |kinetic_kinematic
|kinetic_kinematic |kinetic_kinematic |kinetic_kinematic |kinetic_kinematic |kinetic_kinematic
|kinetic_kinematic |kinetic_kinematic |kinetic_kinematic |kinetic_kinematic |kinetic_kinematic | | |lsk
|lsk |lsk |lsk |lsk |lsk |lsk |lsk |lsk |lsk |lsk | |item|x |y |z |0 |1 |2 |3 |4 |5 |6 |7 |8 | |1 |0.37046
|0.00863 |0.40545 |0.96567 |0.21640 |-0.14375 |-0.24416 |0.56693 |-0.78675 |-0.08876 |0.79483
|0.60030 | |2 |0.35441 |0.05135 |0.41706 |0.97954 |0.19208 |-0.05999 |-0.16101 |0.56933 |-0.80619
|-0.12070 |0.79935 |0.58861 | |3 |0.35060 |0.08595 |0.43204 |0.99376 |0.11142 |0.00542 |-0.06290
|0.59975 |-0.79771 |-0.09213 |0.79239 |0.60301 | |4 |0.35067 |0.11824 |0.44677 |0.99773 |0.06720
|0.00507 |-0.03870 |0.63296 |-0.77322 |-0.05516 |0.77126 |0.63412 | |5 |0.35359 |0.14253 |0.46174
|0.99978 |0.01001 |0.01839 |0.00611 |0.70059 |-0.71353 |-0.02002 |0.71349 |0.70038 | or in the
following format > |
c:\data\mvc002.c3d|c:\data\mvc002.c3d|c:\data\mvc002.c3d|c:\data\mvc002.c3d|c:\data\mvc002.c3d
c:\data\mvc002.c3d|c:\data\mvc002.c3d|c:\data\mvc002.c3d|c:\data\mvc002.c3d|c:\data\mvc002.c3d
c:\data\mvc002.c3d|c:\data\mvc002.c3d| | |origin |origin |origin |rotation |rotation |rotation |rotation
|rotation |rotation |rotation |rotation |rotation | | |link_model_based |link_model_based
|link_model_based |link_model_based |link_model_based |link_model_based |link_model_based
|link_model_based |link_model_based | | | | |lsk |lsk |lsk |lsk |lsk |lsk |lsk |lsk |lsk |lsk |lsk | |item|x
|y |z |0 |1 |2 |3 |4 |5 |6 |7 |8 | |1 |0.37046 |0.00863 |0.40545 |0.96567 |0.21640 |-0.14375 |-0.24416
|0.56693 |-0.78675 |-0.08876 |0.79483 |0.60030 | |2 |0.35441 |0.05135 |0.41706 |0.97954 |0.19208
|-0.05999 |-0.16101 |0.56933 |-0.80619 |-0.12070 |0.79935 |0.58861 | |3 |0.35060 |0.08595 |0.43204
|0.99376 |0.11142 |0.00542 |-0.06290 |0.59975 |-0.79771 |-0.09213 |0.79239 |0.60301 | |4 |0.35067
|0.11824 |0.44677 |0.99773 |0.06720 |0.00507 |-0.03870 |0.63296 |-0.77322 |-0.05516 |0.77126
|0.63412 | |5 |0.35359 |0.14253 |0.46174 |0.99978 |0.01001 |0.01839 |0.00611 |0.70059 |-0.71353
|-0.02002 |0.71349 |0.70038 | ===== other ascii formats ===== the following ascii commands read
slightly different flavors of ascii signals. the names are derived from the customers that have
requested the importer. if c-motion provided you with information on the command, you can make
the call using the command name you were givin; even though you may not see the command listed
in the pipeline. \\ **import_frappier_analog_signals**\\ **import_ntnu_analogsignals**\\

```

****import_cresswell_analogs**** \ ****import_natick_analogs**** \ this command was written for frappier acceleration products but has been modified to accomodate many different flavors of ascii files from our customers. the command has several aliases as you can see from the list. as of visual3d version 3.20, all of these commands are hidden in the pipeline, until we have a convenient way in our licensing scheme to make it visible only to a specific user with a specific name. each command is slightly different, but the differences are recognized automatically. this may not always be the case, so we have given each lab their own command name, that would allow us to modify the processing if necessary. the name of the ascii file containing the analog signals is expected to have the same name as the c3d file; eg. trial1.c3d and trial1.anc the basic format of the file is: \ |file_type: |analog | | | | |board_type:|pci-mio-64e| | | | |trial_name:|test |#channels:|6 | | | |name |fx1 |fy1 |fz1 |mx1 |my1 |mz1 | |rate |1000 |1000 |1000|1000|1000|1000| |0.000 |1003 |997 |1009|1009|1006|1023| |0.001 |1003 |997 |1009|1009|1006|1023| [[===== example ===== specifying a root folder and a list of subject folders. each subject folder contains a set of c3d files. for example, subject1 trial16.c3d subject2 trial16.c3d subject3 trial16.c3d and so on ... open each c3d file and export a signal to a filename whose prefix is the same as the c3d file. } } } }

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

Permanent link:
https://has-motion.com/wiki/doku.php?id=visual3d:documentation:definitions:file_formats:ascii_format&rev=1718801248

Last update: **2024/06/19 12:47**

