

Event Processing

Ojectives

The objective of this tutorial is **to provide an overview of event processing in Visual3D**.

Most event processing functionality is available through the **Command Pipeline**, so it is advisable to start with that tutorial.

Movement data is often collected for more time that is required for the analysis. This is often done to minimize the effects of filtering at the endpoints. Sometimes it is because more than one cycle can be collected at a time or to ensure that the entire cycle is collected. Sometimes it is because there are long periods of meaningless activity between epochs. Regardless of the reasons for collecting extra data, it should not be necessary to analyze all of the data. The interval of interest must be extracted from the trial. The user must be able to define a time frame of interest. Most often the region of interest is defined by specific events, such as heel strike or toe off. This can be accomplished by editing the data file or by entering the start and end frame. This process is extremely tedious and in many circumstances can be automated.

An **Event** is simply an occurrence of interest that correlates to a motion capture frame, i.e. to a particular time point within a movement. The key events for gait studies (e.g., heel strike, toe-off) are now so well standardized that Visual3D can detect and mark them automatically. For other applications, you can define your own events, either manually (using the mouse to double-click on significant points on any graph) or computationally.

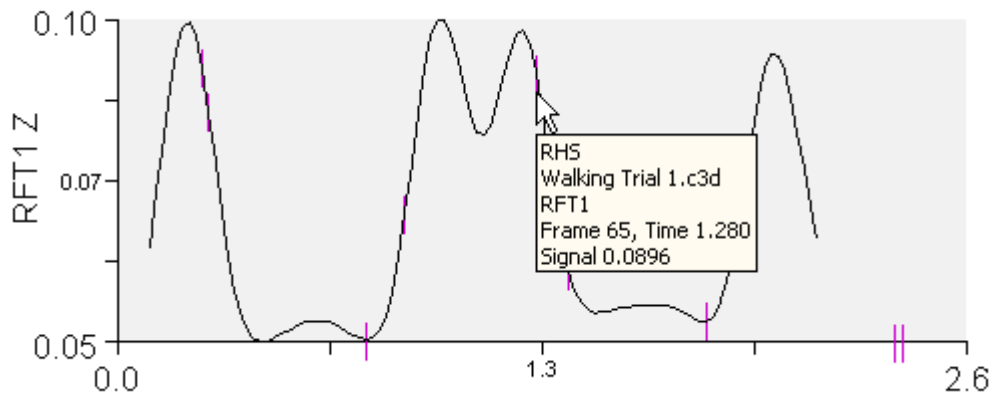
If you would like to follow this tutorial in the form of a walk-through Youtube video, please follow this link: [Visual3D Tutorial Video 6: Event Processing](#)

Event Syntax

Every event has a name, called the event label, so you can refer to it in other contexts, e.g., you can conduct analyses and create reports which are constrained to the interval between two chosen events.

In Visual3D, it is perfectly OK to have multiple occurrences of a given event (e.g., two separate occurrences of right heel strike and toe-off, detected by separate force platforms) in a single movement trial.

Events appear on all signal graphs as short vertical-line markers. Positioning the mouse pointer over an event marker (without clicking) displays the event label and other details, as shown in the following illustration.



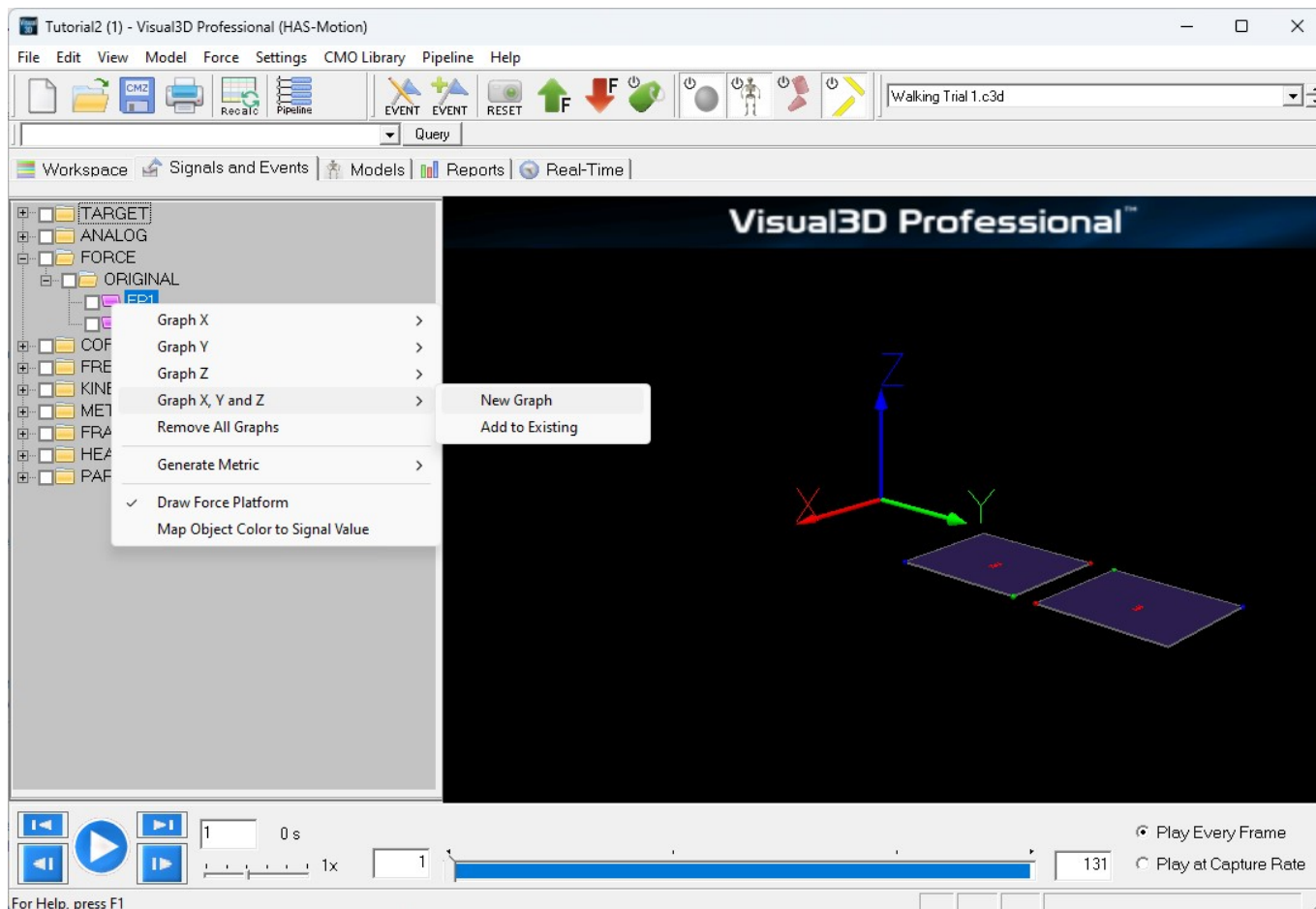
Multiple Occurrences of an Event

An event is a time of interest that corresponds to a motion capture frame.

Visual3D analyzes multiple movement cycles within and across trials. Visual3D automatically determines common events related to gait analysis such as: heel strike, mid-stance, and foot-off during gait. Events can also be entered manually or calculated from any signals, including those generated by the program, such as inverse dynamics. Additionally, the plug-in architecture supports the custom determination of events.

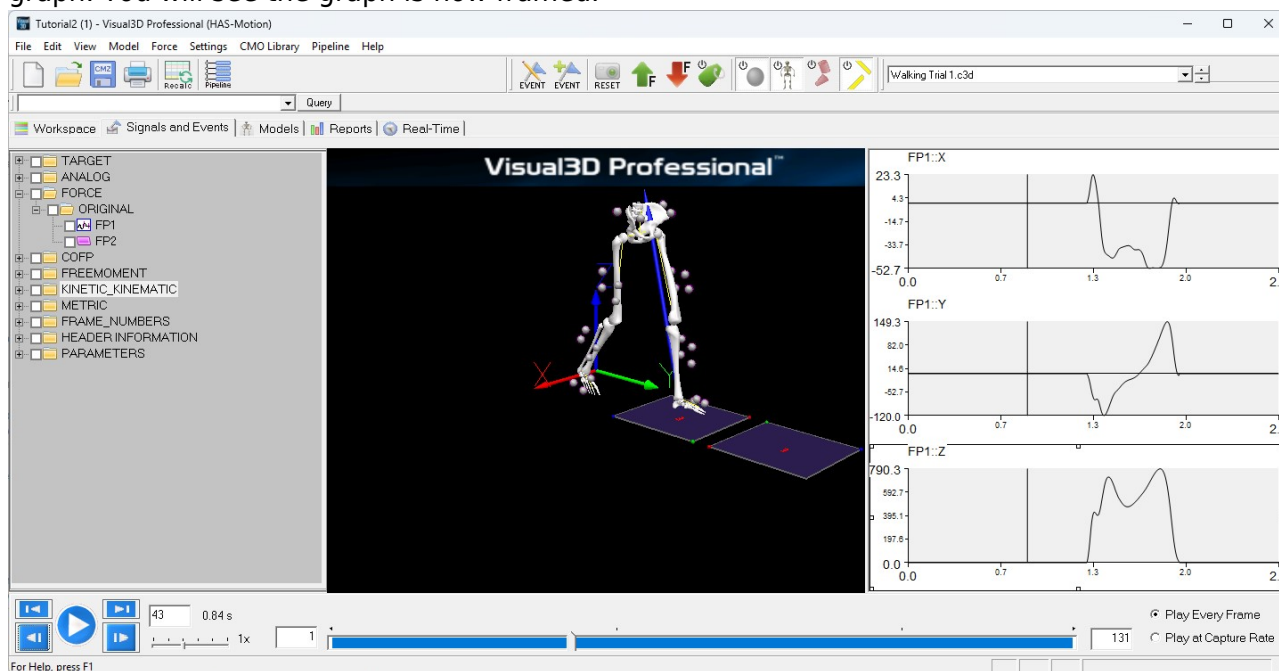
Preparation

1. Download the following ZIP file: [Visual3D Tutorial 6 Sample Data](#). This contains:
 - **Tutorial6_EventProcessing_Start.cmz**: Starting point for this tutorial, .cmz file which is the same as the start of the [Visualizing Data](#) tutorial. Within the CMZ, the *Lower Body Static Trial.c3d* calibration file is loaded as well as the *Walking Trial 1.c3d* motion file.
 - **Tutorial6_EventAdded_End**: Final result of tutorial, with events added to the CMZ file.
1. Click on the **Signal and Event Processing** Tab to visualize the animation of the model based on the movement data and the model that was applied to it. If the animation doesn't 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.
2. Expand the **FORCE** folder and **ORIGINAL** subfolder. Right-click the **FP1** signal and select **Graph X,Y,and Z** and **New Graph** to graph all 3 components of one of the Ground Reaction force signals in the data tree.



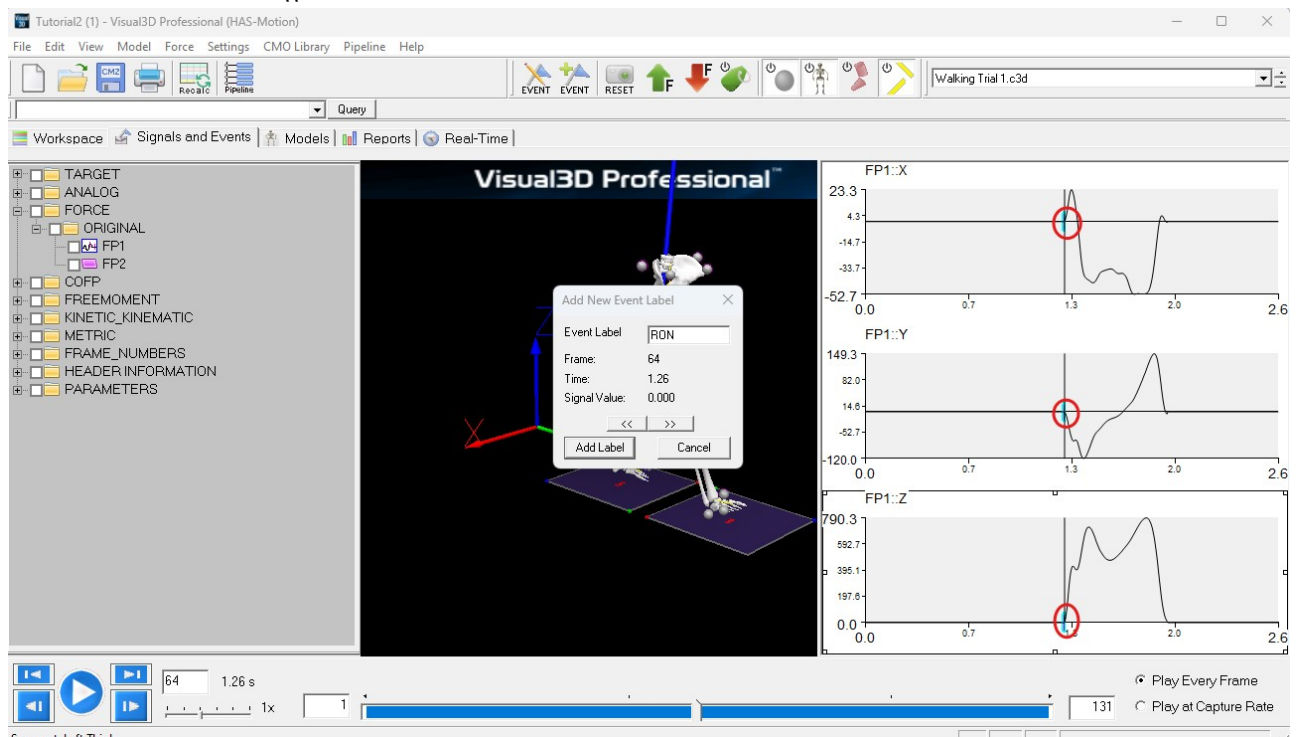
Creating an Event Manually

1. Select the graph of the Z-component of the Ground Reaction Force signal by Left clicking on the graph. You will see the graph is now framed.



2. To create the event, double click on the Fz force signal at the frame that the signal rises above zero.
3. Enter the name **RON** into the popup dialog box. **RON** is a typical name used in Visual3D to

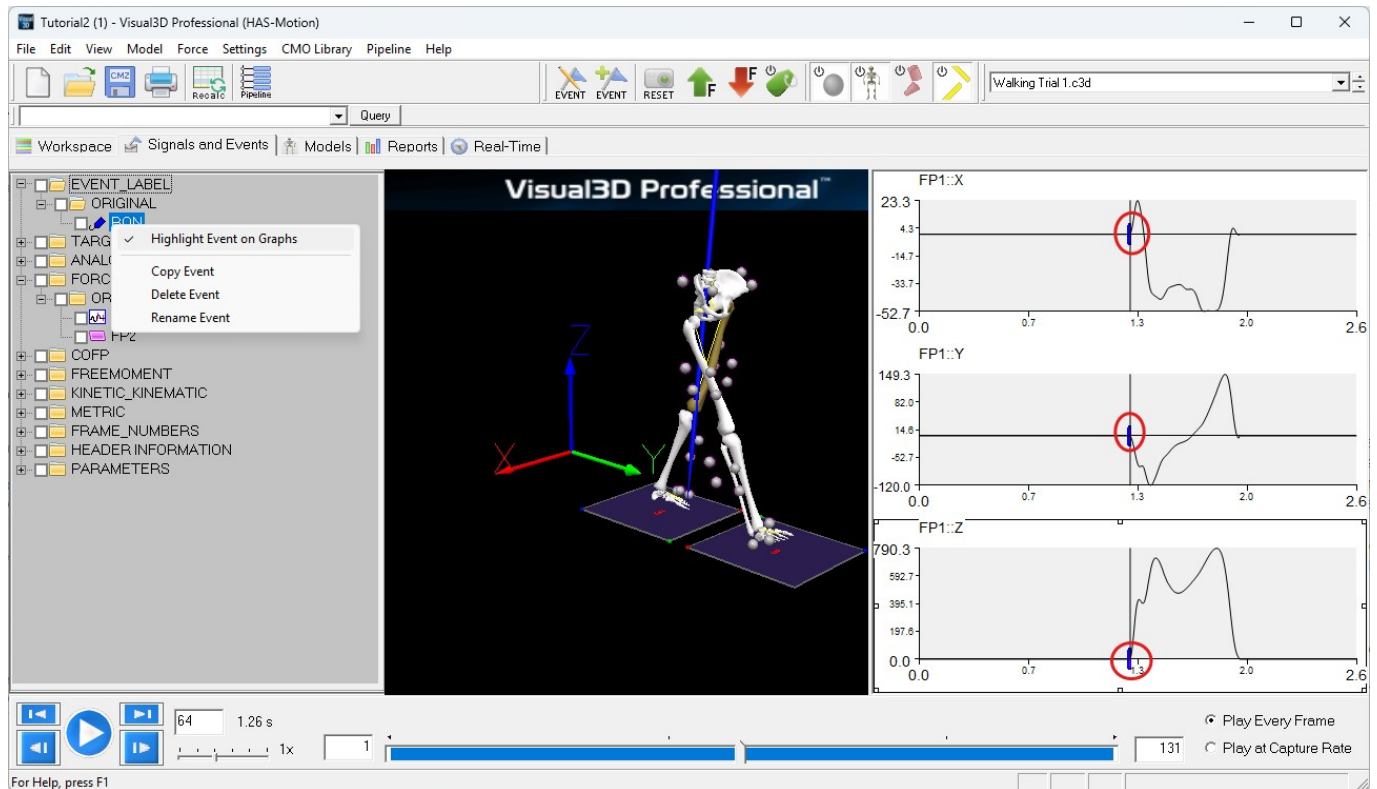
indicate that the Right has contacted a force platform. **In principle** the event can be assigned any name, but when Visual3D computes Heel Strike and Toe Off automatically, the events are labeled ON and OFF.\\



4. An event labeled **RON** has been added to the Event_Label data tree.

Highlight Event Label

1. To highlight this event (actually all occurrences of this event), select the RON signal within the **EVENT_LABEL** subfolder with a Right Mouse Button click and select **Highlight Event on Graphs**. The Frames corresponding to RON will be highlighted in blue on all signals in each graph.

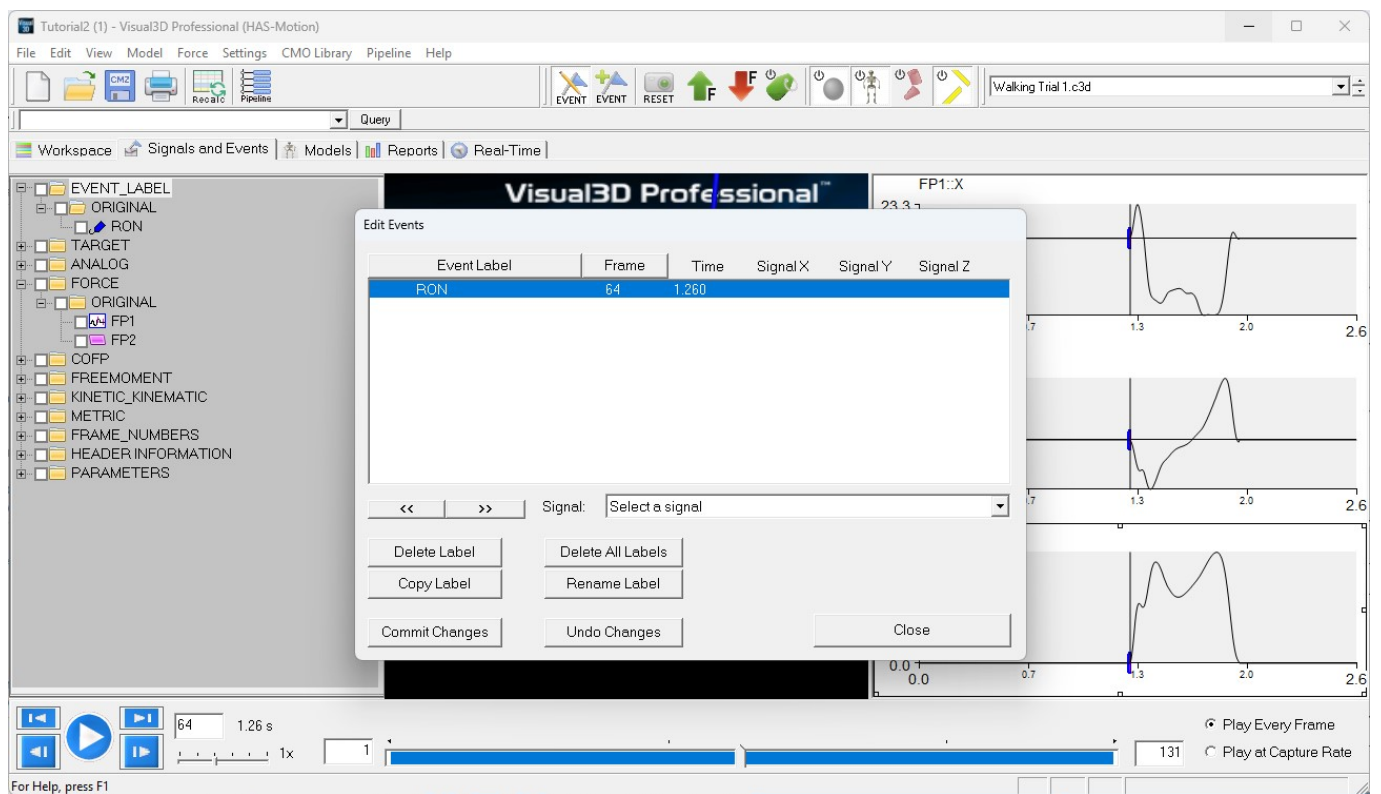


Editing an Event Label

Events can be edited:

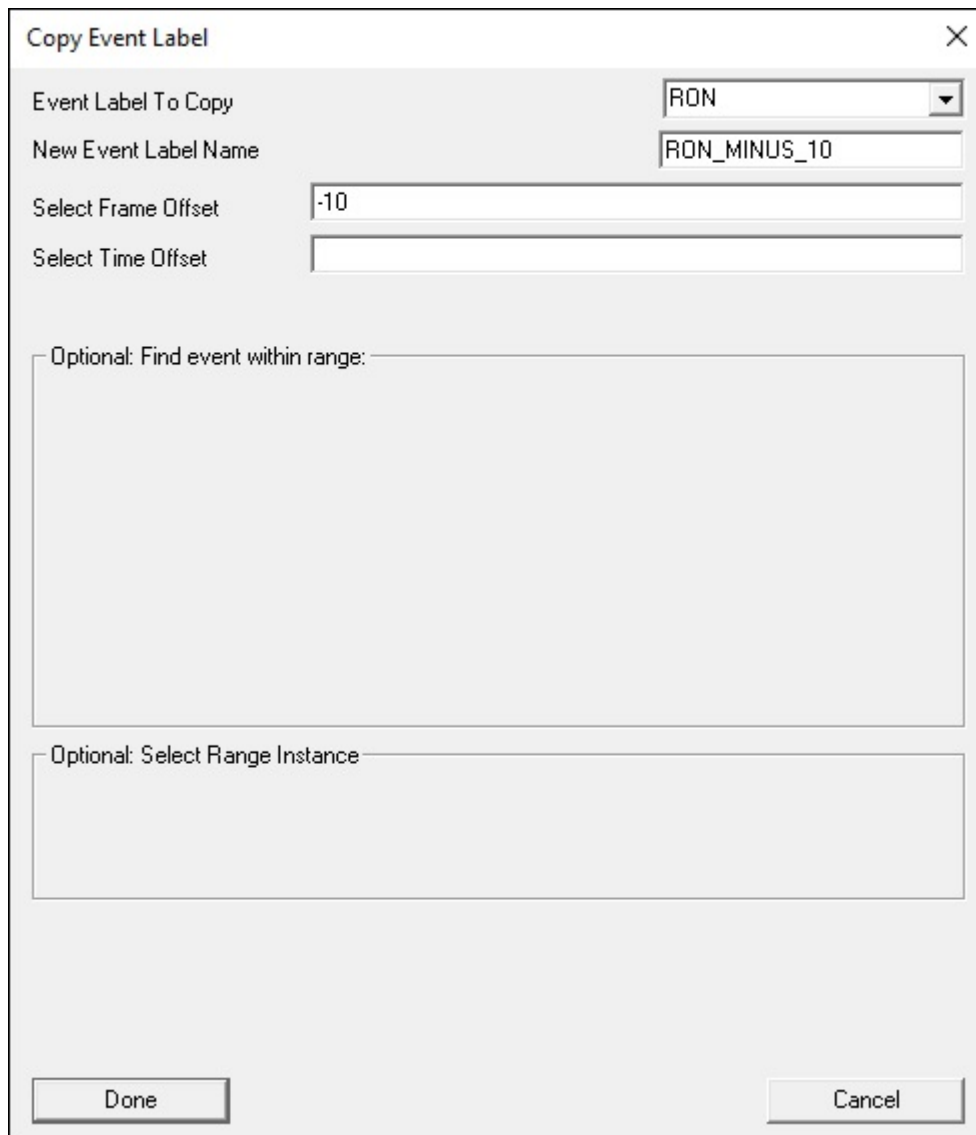
1. You can modify the frame number for the event
2. Copy the event
3. Rename the event

To edit the event labels select the **Edit Events** icon  on the Visual3D toolbar. The following dialog should appear.



For example, to create an event 10 frames before the **RON** event.

1. Highlight the RON Event in the list box
2. Select **Copy Label** in the dialog
3. Fill in the edit boxes as below



Copy Event Label

Event Label To Copy: RON

New Event Label Name: RON_MINUS_10

Select Frame Offset: -10

Select Time Offset:

Optional: Find event within range:

Optional: Select Range Instance:

Done Cancel

4. A new event will appear in the dialog that is 10 frames before the RON event

Saving the Event Labels

Events don't explicitly get saved. When a file is saved as either a CMZ file or a C3D file, the Events simply go with the file.

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

Permanent link:
https://has-motion.com/wiki/doku.php?id=visual3d:tutorials:events:event_processing

Last update: 2026/02/03 18:49

