

Event Define Event Sequence

Overview

This command in Visual3D's pipeline allows users to define a sequence of events based on an ordered list of existing events, useful for identifying movement phases within a dataset. Essentially defining an [Event Sequence](#) as a single command parameter, acting as an extension to defining a Start and Stop Event that indicate which frames should be processed.

Pipeline Command

```
Event_Define_Event_Sequence
/EVENT_SEQUENCE_NAME=
! /EVENT_SEQUENCE=
! /EXCLUDE_EVENTS=
! /INSIDE_OF_SEQUENCE=
! /OFFSET_FROM_START=
! /OFFSET_FROM_END=
! /OFFSET_BY=PERCENT
! /EVENT_SEQUENCE_INSTANCE=
;
```

Command Parameters

The following table shows the command parameters and descriptions:

Parameter	Description
/EVENT_SEQUENCE_NAME=	Specifies the name of the event sequence being defined.
/EVENT_SEQUENCE=	Defines the ordered list of existing events that form the sequence.
/EXCLUDE_EVENTS=	Lists events to be excluded from the sequence, preventing them from being part of the defined sequence.
/INSIDE_OF_SEQUENCE=	Restricts the event sequence to only include events that occur inside a specified sequence.
/OFFSET_FROM_START=	Applies a time or frame offset from the start of the sequence.
/OFFSET_FROM_END=	Applies a time or frame from the end of the sequence.
/OFFSET_BY=PERCENT	Offsets the event within the sequence by a specified percentage of the sequence duration.
/EVENT_SEQUENCE_INSTANCE=	Specifies which instance of the event sequence to use when multiple sequences exist.

Dialog

After adding the **Event_Define_Event_Sequence** command into the pipeline on the Visual3D application, the user can double-click with the left mouse button in order to open the following dialog:

- **Defined Events** and **Event Sequence**, lists of available events in the dataset, and list of selected events that define the sequence, respectively.
- **Event Sequence Instance**: Defines which instance of the event sequence should be used.
- **Range Start and End Offsets**: Allows adjustments to where the sequence starts and ends.
- **Define Sequence Inside of Outer Sequence**: Allows for nested event sequences, meaning the defined sequence occurs within a broader sequence.

Examples

The following examples will go through the use of the **Event_Define_Event_Sequence** command in the Visual3D application.

Example 1

A common example of an event sequence is the gait cycle, representing the repetitive sequence of events that occur from one heel strike of a foot to the next heel strike of the same foot.

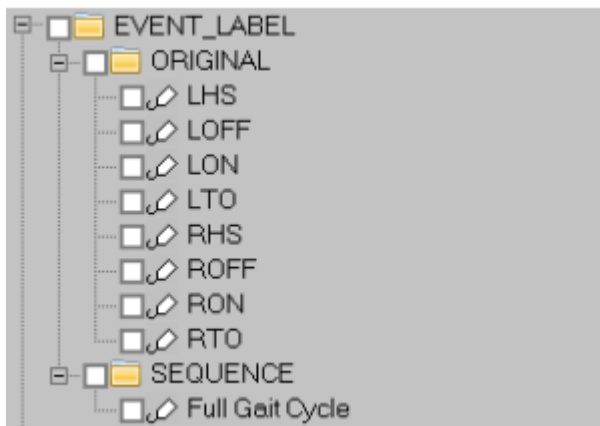
The **Event_Define_Event_Sequence** command can be used to name this event sequence “Full Gait Cycle” and include the events detected using the [Automatic_Gait_Events](#) command. The full gait cycle

is considered LHS-LHS, the instance will be set to 0 to consider all instances of the event sequence.

```
Automatic_Gait_Events
! /FRAME_WINDOW=8
! /USE_TPR=TRUE
! /TPR_EVENT_INSTANCE=1
;

Event_Define_Event_Sequence
/EVENT_SEQUENCE_NAME=Full Gait Cycle
/EVENT_SEQUENCE=LHS+RTO+RHS+LTO+LHS
! /EXCLUDE_EVENTS=
! /INSIDE_OF_SEQUENCE=
! /OFFSET_FROM_START=
! /OFFSET_FROM_END=
! /OFFSET_BY=PERCENT
/EVENT_SEQUENCE_INSTANCE=0
;
```

Once this pipeline is executed, you should notice a new folder labelled **SEQUENCE** within the Event_Label folder in the data tree as seen below:



Example 2

As an extension to the previous example, we can add the **Event_Define_Event_Sequence** command twice again to the pipeline in order to define the Swing and Stance phases, both embedded inside of the Full Gait Cycle event sequence.

```
Event_Define_Event_Sequence
/EVENT_SEQUENCE_NAME=R_Stance_Phase
/EVENT_SEQUENCE=RHS+RTO
! /EXCLUDE_EVENTS=
/INSIDE_OF_SEQUENCE=Full Gait Cycle
! /OFFSET_FROM_START=
! /OFFSET_FROM_END=
! /OFFSET_BY=PERCENT
/EVENT_SEQUENCE_INSTANCE=0
;
```

```
Event_Define_Event_Sequence
/EVENT_SEQUENCE_NAME=R_Swing_Phase
/EVENT_SEQUENCE=RTO+RHS
! /EXCLUDE_EVENTS=
/INSIDE_OF_SEQUENCE=Full Gait Cycle
! /OFFSET_FROM_START=
! /OFFSET_FROM_END=
! /OFFSET_BY=PERCENT
/EVENT_SEQUENCE_INSTANCE=0
;
```

Notes

Equivalent Pipeline Commands

The following will show a different method to create an event sequence.

In this case, the event sequence will start one frame before the sequence RON+ROFF. The resulting signal can be used wherever an Event_Sequence is used.

For example, the use case in the Metric_Mean command. Instead of the following syntax:

```
Metric_Mean
/RESULT_METRIC_NAME=_MEAN
/APPLY_AS_SUFFIX_TO_SIGNAL_NAME=TRUE
! /RESULT_METRIC_FOLDER=PROCESSED
/SIGNAL_TYPES=LINK_MODEL_BASED
/SIGNAL_NAMES=RKNEE_ANGLE
! /SIGNAL_FOLDER=ORIGINAL
! /SIGNAL_COMPONENTS=ALL_COMPONENTS
/EVENT_SEQUENCE=RHS+RTO
! /EXCLUDE_EVENTS=
! /GENERATE_MEAN_AND_STDDEV=TRUE
! /APPEND_TO_EXISTING_VALUES=FALSE
;
```

You could use the following line in replacement to achieve the same.

```
/EVENT_SEQUENCE=R_Stance_Phase
```

Visual3D Versions supporting Subject Prefixes

NOTE: When using events in a command, the commands will iterate across subjects contained in the current workspace. As such, events being created should list the generic event label name, and as the event command iterates through subjects, it will create the subject specific events that are prefixed with the specific subject prefix.

In commands using event labels and event sequences to process data, the event labels and sequences listed to process between should be generic (without a subject prefix), and as each subject is processed, the event range/sequence specified will automatically use the specific events prefixed for each subject as they are processed.

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

Permanent link:
https://has-motion.com/wiki/doku.php?id=visual3d:documentation:pipeline:event_commands:event_define_event_sequence&rev=1745859561

Last update: **2025/04/28 16:59**

