

Automating Work Flow With Directory Watchers in The GUI

In this tutorial we will walk you through the steps of setting up Sift to automatically process your data when it is loaded. You will learn how to create a directory watcher and point it to a folder your data will be dropped in, then set it to execute all your Visual3D and Sift processing automatically.

The Task

Before automating a workflow, we need a clear research question. In this example, we want to look at the right and left ankle movement from a treadmill-running trial for atypical stride cycles.

The Workflow

1. Visual3D to calculate the right and left ankle joint angles
2. Right heel strike to right heel strike for right stride cycles
3. Left heel strike to left heel strike for left stride cycles
4. Sift PCA to summarize the major differences in curve shape
5. Group-level Local Outlier Factor (LOF) to flag ankle cycles that are different from the other cycles on the same side

To begin, [download and unzip this folder](#)

'Directory_Watcher_Tutorial.'

The C3D file remains outside the watched **Data** directory until you are ready to start the workflow.

Setting Up

Visual3D Pipeline

Before creating the watchers, open `V3D_Pipeline.v3s` in a text editor.

The script performs four steps:

1. Opens the treadmill-running C3D
2. Calculates right and left ankle joint angles
3. Creates the heel-strike events used to define stride cycles
4. Saves a CMZ into the CMZs folder

For a lower-body model, an ankle joint angle is calculated using the foot as the segment and the shank as the reference segment. The X component represents flexion and extension for the segment coordinate systems used in this example. See [Model Based Computations](#) for more information about joint-angle definitions.

Before running, replace every occurrence of [REPLACE](#) with the full path to the unzipped `Directory_Watcher_Tutorial` folder.

For example:

```
C:\Users\maya\Desktop\Directory_Watcher_Tutorial\Data
```

Only the File_Open and File_Save_As commands require this path update.

```
File_Open
/FILE_NAME=DE3_1_pose_0.c3d
/FILE_PATH=[[REPLACE]]\Data
! /SEARCH_SUBFOLDERS=FALSE
! /SUFFIX=
! /SET_PROMPT=File_Open
! /ON_FILE_NOT_FOUND=PROMPT
! /FILE_TYPES_ON_PROMPT=
;

Compute_Model_Based_Data
/RESULT_NAME=RAnkleAngle
/SUBJECT_TAG=ALL_SUBJECTS
/FUNCTION=JOINT_ANGLE
/SEGMENT=RFT
/REFERENCE_SEGMENT=RSK
/RESOLUTION_COORDINATE_SYSTEM=
! /USE_CARDAN_SEQUENCE=FALSE
! /NORMALIZATION=FALSE
! /NORMALIZATION_METHOD=
! /NORMALIZATION_METRIC=
/NEGATEX=FALSE
/NEGATEY=FALSE
/NEGATEZ=FALSE
! /AXIS1=X
! /AXIS2=Y
! /AXIS3=Z
! /TREADMILL_DATA=FALSE
! /TREADMILL_DIRECTION=UNIT_VECTOR(0,1,0)
! /TREADMILL_SPEED=0.0
;

Compute_Model_Based_Data
/RESULT_NAME=LAnkleAngle
/SUBJECT_TAG=ALL_SUBJECTS
/FUNCTION=JOINT_ANGLE
/SEGMENT=LFT
/REFERENCE_SEGMENT=LSK
/RESOLUTION_COORDINATE_SYSTEM=
! /USE_CARDAN_SEQUENCE=FALSE
! /NORMALIZATION=FALSE
! /NORMALIZATION_METHOD=
! /NORMALIZATION_METRIC=
```

```
! /NEGATEX=FALSE
! /NEGATEY=FALSE
! /NEGATEZ=FALSE
! /AXIS1=X
! /AXIS2=Y
! /AXIS3=Z
! /TREADMILL_DATA=FALSE
! /TREADMILL_DIRECTION=UNIT_VECTOR(0,1,0)
! /TREADMILL_SPEED=0.0
;
```

```
Event_Explicit
/EVENT_NAME=RHS
/FRAME=50
! /TIME=
;
```

```
Event_Explicit
/EVENT_NAME=RHS
/FRAME=350
! /TIME=
;
```

```
Event_Explicit
/EVENT_NAME=RHS
/FRAME=505
! /TIME=
;
```

```
Event_Explicit
/EVENT_NAME=RHS
/FRAME=650
! /TIME=
;
```

```
Event_Explicit
/EVENT_NAME=RHS
/FRAME=810
! /TIME=
;
```

```
Event_Explicit
/EVENT_NAME=RHS
/FRAME=950
! /TIME=
;
```

```
Event_Explicit
/EVENT_NAME=RHS
/FRAME=1110
! /TIME=
```

```
;

Event_Explicit
/EVENT_NAME=LHS
/FRAME=120
! /TIME=
;

Event_Explicit
/EVENT_NAME=LHS
/FRAME=260
! /TIME=
;

Event_Explicit
/EVENT_NAME=LHS
/FRAME=425
! /TIME=
;

Event_Explicit
/EVENT_NAME=LHS
/FRAME=580
! /TIME=
;

Event_Explicit
/EVENT_NAME=LHS
/FRAME=730
! /TIME=
;

Event_Explicit
/EVENT_NAME=LHS
/FRAME=880
! /TIME=
;

Event_Explicit
/EVENT_NAME=LHS
/FRAME=1030
! /TIME=
;

File_Save_As
/FILE_NAME=DE3_Ankle_Angles.cmz
/FOLDER=[[REPLACE]]\CMZs
! /SET_PROMPT=Save CMZ file as
! /SAVE_EMBEDDED_GRAPHICS=FALSE
! /CREATE_FOLDER_PATH=FALSE
```

;

Sift Query

The Sift Query q3d file contains two signal groups:

- RAnkleAngleX, normalized from right heel strike to the next right heel strike
- LAnkleAngleX, normalized from left heel strike to the next left heel strike

```
<queries>
  <query name="RAnkleAngleX">
    <Definition condition="Right Ankle" query="RAnkleAngleX"
global="FALSE" subject_tag="NO_SUBJECT" type="LINK_MODEL_BASED"
folder="ORIGINAL" name="RAnkleAngle" component="X" max_elements="0"
negate_result="FALSE" events="RHS,RHS" exclude_events="" has_tag_ref="No"
tag_logic="OR" tag="" workspace_name_req="" normalize_info="101,0"
query_type="0" data_type="1"/>
  </query>
  <query name="LAnkleAngleX">
    <Definition condition="Left Ankle" query="LAnkleAngleX"
global="FALSE" subject_tag="NO_SUBJECT" type="LINK_MODEL_BASED"
folder="ORIGINAL" name="LAnkleAngle" component="X" max_elements="0"
negate_result="FALSE" events="LHS,LHS" exclude_events="" has_tag_ref="No"
tag_logic="OR" tag="" workspace_name_req="" normalize_info="101,0"
query_type="0" data_type="1"/>
  </query>
</queries>
```

Sift Batch Script

The BATCH script does not need to be updated, the script gets a reference to its location itself (the %~dp0 you see in the script).

There are a couple things to note about the script, the first line is ::SIFT, a .bat file will only be executed if this line is present, this prevents a unintended script from being executed by accident, you will also notice that the entire sift command is on one line. This is because they all need to be passed at once via a command line call and new line characters will execute everything preceding it and treat everything remaining as a brand new command to make this easier to manage you can enable wordwrap on your text editor of choice.

```
::SIFT
"%ProgramFiles%\Sift\Sift.exe" -LoadLib "%~dp0\CMZs" -LoadQuery
"%~dp0\Sift_Query.q3d" -SelectSignals -RunPCA "AnkleStridePCA" -RunLOF
grouping "group" autoExclude -ExportPCA file "%~dp0\Exports\Ankle Stride -
LOF Outliers.txt" lof lofThreshold exportFormat "transposed" -SaveProject
"%~dp0\Exports\Ankle Stride Screening.i3d" -Exit
```

The script performs the following:

1. Loads the CMZ library
2. Loads and calculates the ankle-angle queries
3. Selects both ankle signal groups
4. Creates a PCA named AnkleStridePCA
5. Runs LOF separately within each signal group
6. Automatically excludes the cycles identified as outliers
7. Exports the LOF values and threshold
8. Saves the completed Sift project

Opening the Directory Watchers Dialog

Launch Sift normally.

From the main page, select the Directory Watcher icon .

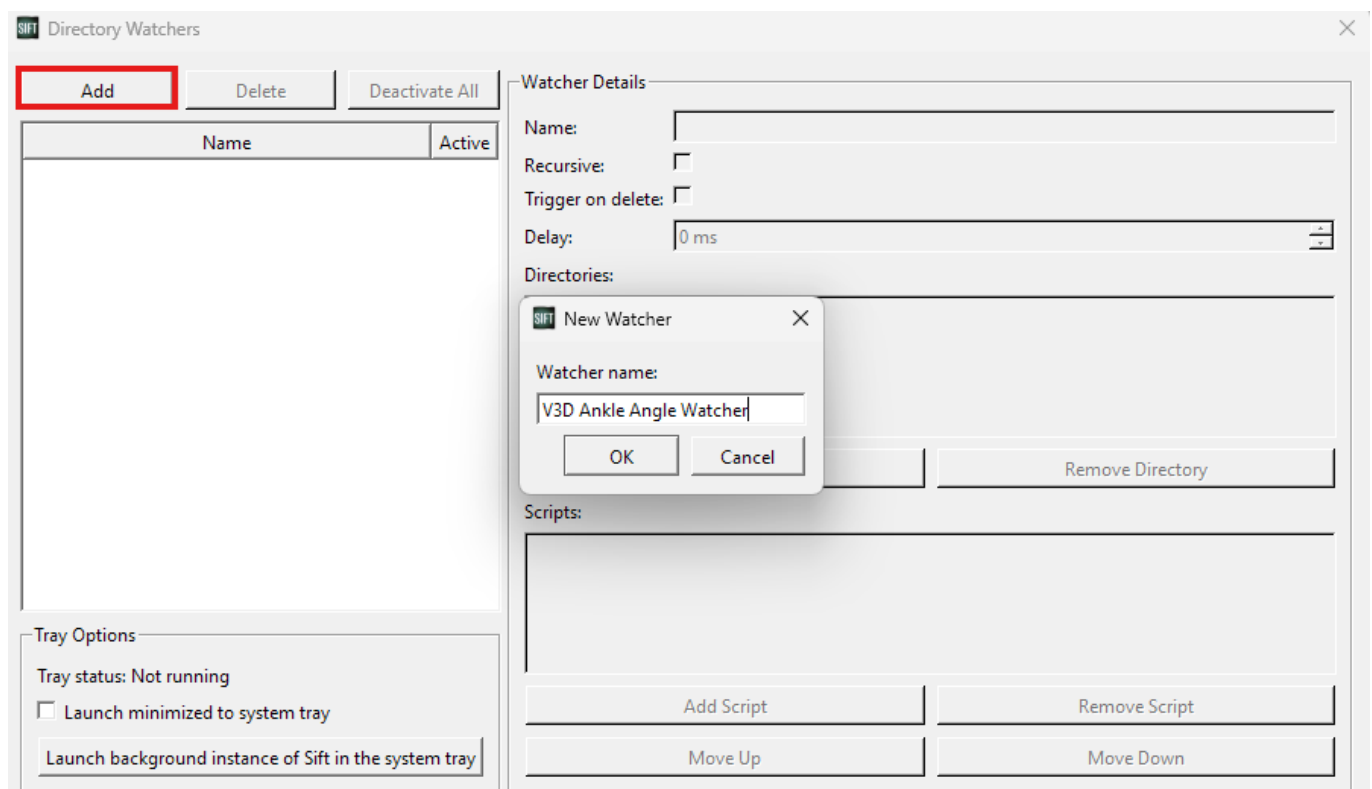
The table on the left lists all configured watchers and whether each watcher is active. The right side displays the settings for the selected watcher.

Creating the Visual3D Watcher

Select Add above the watcher table.

In the name dialog, enter:

V3D Ankle Angle Watcher

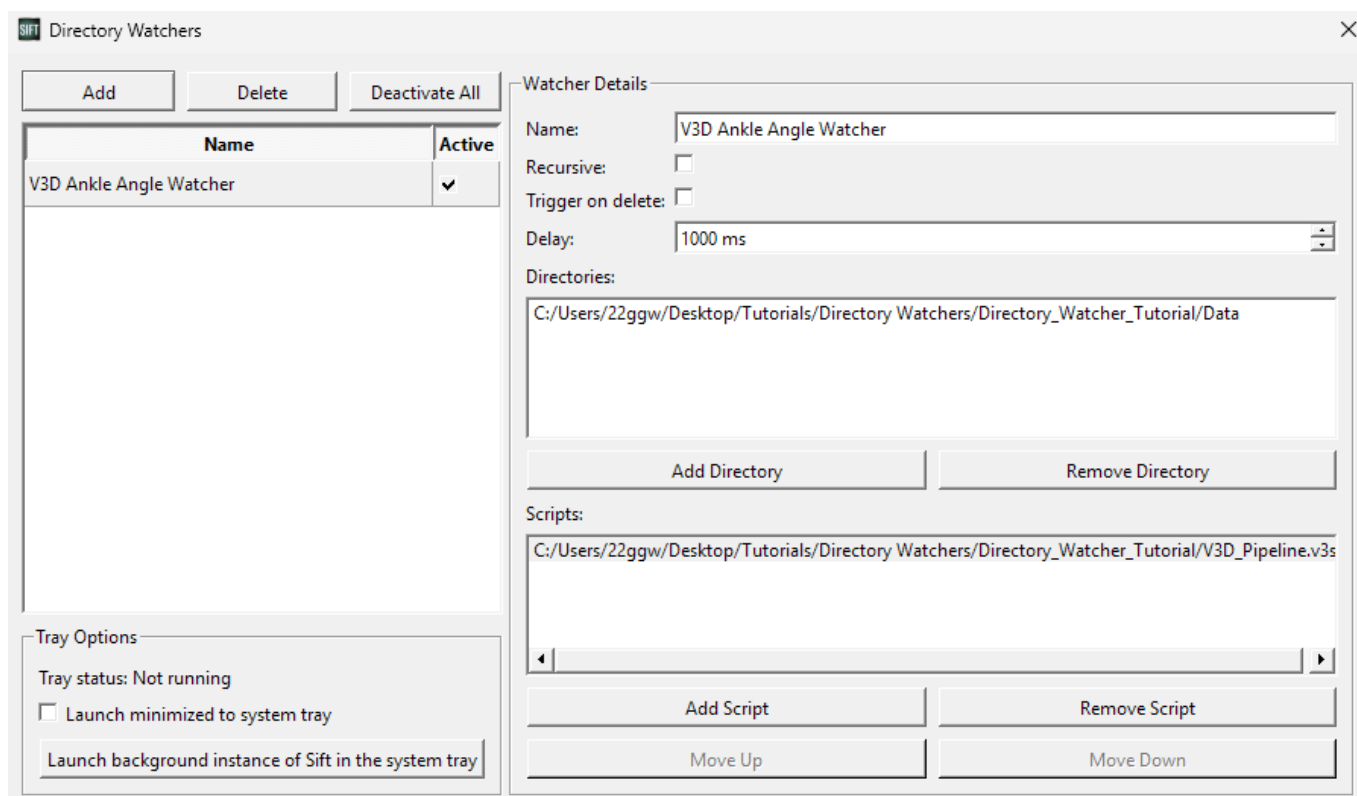


When prompted to add the directory, browse to `Directory_WatcherTutorial\Data` and select the folder. When prompted for the script, browse to `Directory_WatcherTutorial\V3D_Pipeline.v3s` and select the pipeline. With the new watcher created, configure the following settings on the right side of the dialog.

Setting	Value
Name	V3D Ankle Angle Watcher
Active	Enabled
Recursive	Disabled
Trigger on delete	Disabled
Delay	1000 ms

The delay gives Windows time to finish copying the C3D before Visual3D attempts to open it. A longer delay may be needed for larger files or network directories.

The first watcher should look like this.



When the C3D is copied into **Data**, the watcher will launch the Visual3D pipeline.

Creating the Sift Watcher

Select Add again.

Name the second watcher:

Sift Ankle Outlier Watcher

When prompted to add the directory browse to `Directory_WatcherTutorial\CMZs` and select the

folder. When prompted for the script, browse to `Directory_WatcherTutorial\Sift_Script.bat` and select. With the new watcher created, configure the following settings on the right side of the dialog.

Setting	Value
Name	Sift Ankle Outlier Watcher
Active	Enabled
Recursive	Disabled
Trigger on delete	Disabled
Delay	1000 ms

The screenshot shows the 'SIFT Directory Watchers' application window. On the left, there is a list of watchers with columns 'Name' and 'Active'. Two watchers are listed: 'Sift Ankle Outlier Watcher' (Active) and 'V3D Ankle Angle Watcher' (Active). Above the list are buttons for 'Add', 'Delete', and 'Deactivate All'. Below the list is the 'Tray Options' section, which includes 'Tray status: Not running', a checkbox for 'Launch minimized to system tray' (unchecked), and a checkbox for 'Launch background instance of Sift in the system tray' (checked). On the right, the 'Watcher Details' section shows the configuration for the selected watcher: 'Name: Sift Ankle Outlier Watcher', 'Recursive: [unchecked]', 'Trigger on delete: [unchecked]', 'Delay: 1000 ms', 'Directories: C:/Users/22ggw/Desktop/Tutorials/Directory Watchers/Directory_Watcher_Tutorial/CMZs', and 'Scripts: C:/Users/22ggw/Desktop/Tutorials/Directory Watchers/Directory_Watcher_Tutorial/Sift_Script.bat'. There are buttons for 'Add Directory', 'Remove Directory', 'Add Script', 'Remove Script', 'Move Up', and 'Move Down'.

The completed watcher should monitor CMZs and contain the Sift batch script.

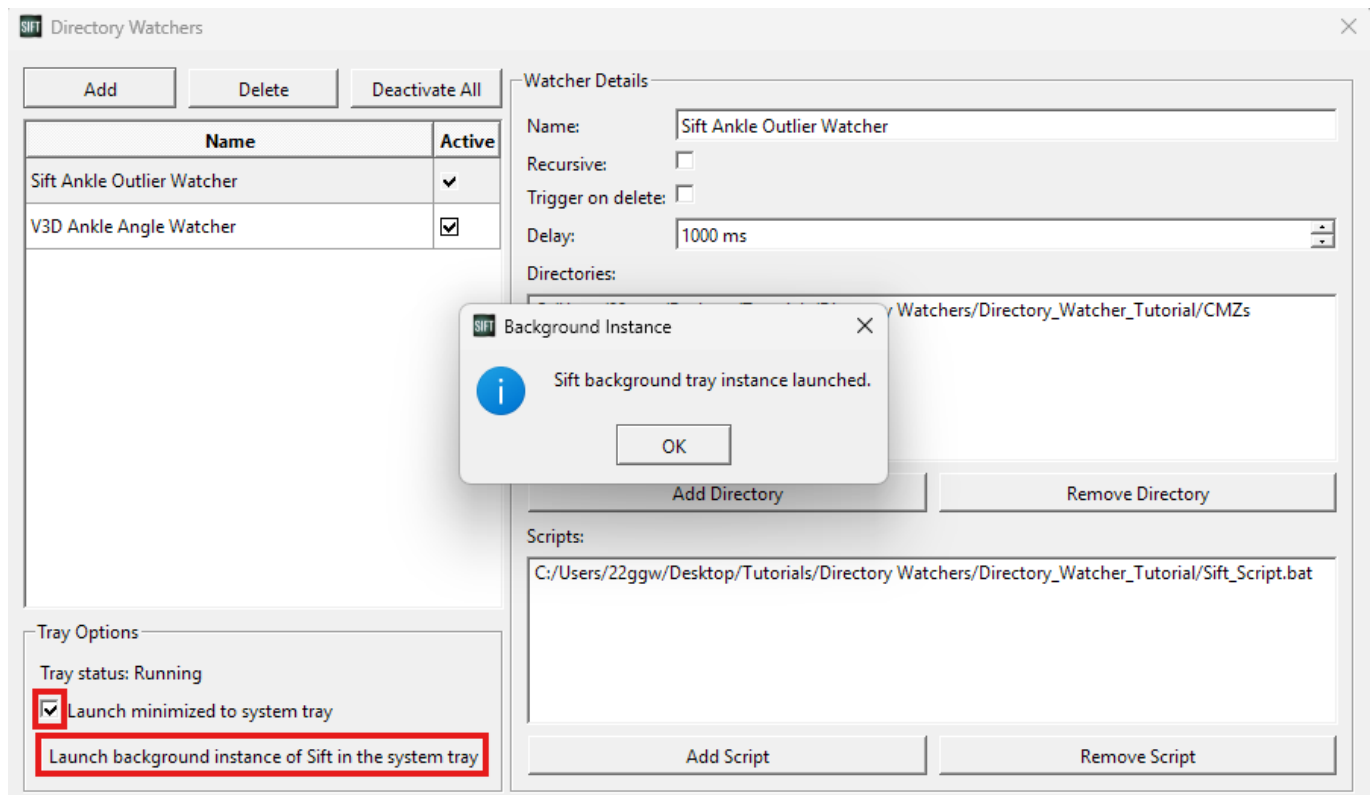
Now both watchers should be activated.

A watcher can be deactivated without deleting its settings if needed. Deactivate All stops every watcher at once.

Launching the Background Instance

Directory watchers respond while the Sift background instance is running.

In the Tray Options section, select Launch background instance of Sift in the system tray and check Launch minimized to system tray. This will ensure Sift starts in the tray by default.



A Sift icon should appear in the Windows system tray.

The background instance **MUST** continue running while the directory watchers are in use, or else they will not work.

Before continuing, close the normal Sift window if it is still open. The Sift batch script needs to launch its own processing instance, and only one normal Sift instance should be open at a time.

Using the Watchers

Open the following folders in separate File Explorer windows:

- Data
- CMZs
- Exports

The sample C3D, DE3_1_pose_0.c3d, should still be in the root Directory_Watcher_Tutorial folder.

Copy or drag the C3D into:

Directory_Watcher_Tutorial\Data

After the watcher delay, the V3D Ankle Angle Watcher will launch V3D_Pipeline.v3s.

The pipeline will calculate:

- RAnkleAngle
- LAnkleAngle

- right heel-strike events
- left heel-strike events

It will then save:

CMZs\DE3_Ankle_Angles.cmz

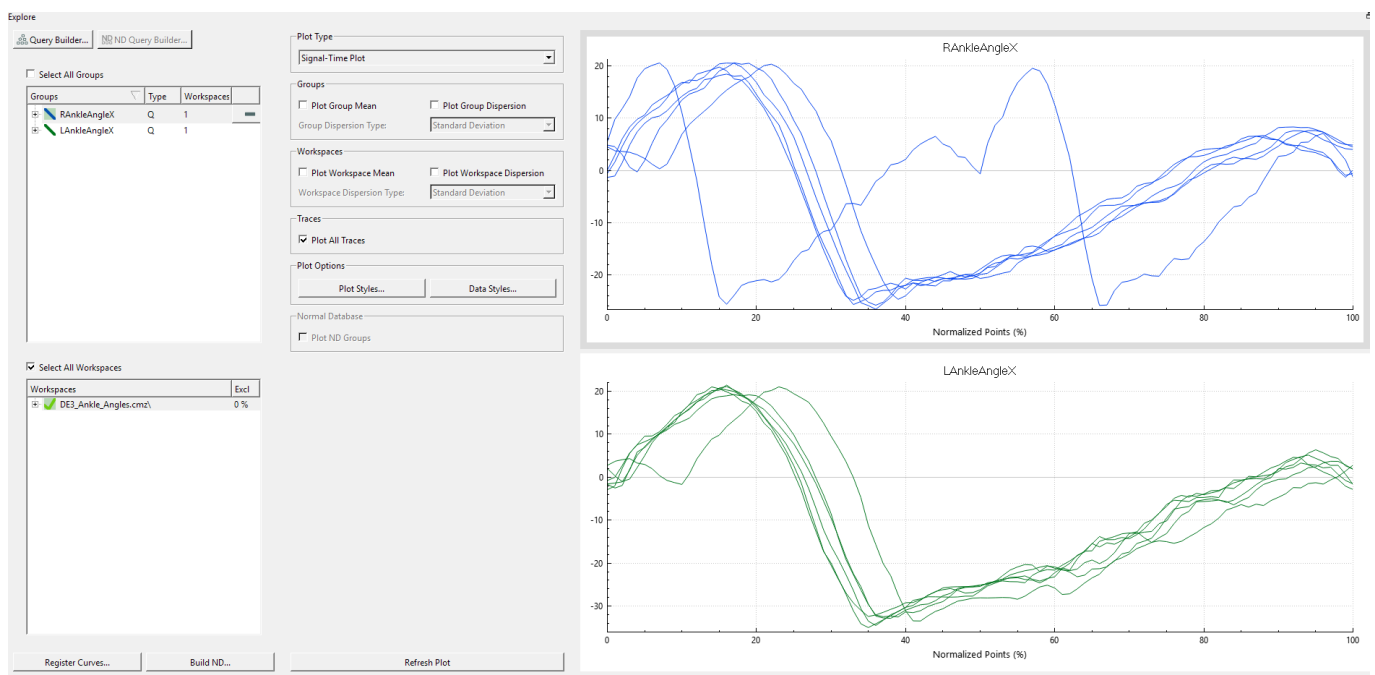
The new CMZ will trigger the Sift Script and sift will load the ankle strides, run PCA and LOF analysis and create two more files.

Exports\Ankle Stride - LOF Outliers.txt
Exports\Ankle Stride Screening.i3d

Reviewing the Results

Open Ankle Stride - LOF Outliers.txt to review the LOF score for each right and left ankle stride and the threshold used to classify outliers.

Load Ankle Stride Screening.i3d in Sift to view the analysis visually. You can view the graphs on the **Explore** page.



On the **Analyse** page, look at the workspace scores and LOF results.

Editing an Existing Watcher

To edit an existing watcher open the dialog  in Sift again and select a watcher in the table to edit its configuration.

You can:

1. Change its name
2. Activate or deactivate it
3. Enable or disable recursive directory monitoring
4. Enable or disable delete events
5. Change the delay
6. Add or remove directories
7. Add or remove scripts

Any changes made to the selected watcher are used the next time a relevant directory event occurs.

Removing a Watcher

Select the watcher and use Delete to remove it permanently.

Deleting a watcher does not delete its directories, scripts, CMZs, or exports.

Conclusion

The Directory Watchers dialog allows a complete automated workflow to be created and managed without entering watcher commands manually.

In this example, adding one C3D to the Data folder automatically:

1. Calculated ankle joint angles in Visual3D
2. Divided the signals into right and left stride cycles
3. Ran PCA in Sift
4. Looked into each side for different ankle cycles using LOF
5. Exported the outlier results
6. Saved a Sift project

The same approach can be used to automate other workflows.

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

Permanent link:
https://has-motion.com/wiki/doku.php?id=sift:tutorials:using_directory_watchers_in_the_gui&rev=1784316464

Last update: **2026/07/17 19:27**

