

BodyBuilder session 2

Marker files, Parameter Files,
Static Trials

Session Objectives

1. Creating an .MKR file
2. Creating an .MP file
3. Associate the files with the subject
4. Using a static trial
 - Why use a static trial
 - How to designate a static trial
 - The IF condition – designating code as static trial only
 - PARAM - Exporting parameters to the .MP file
5. Non-Assigning Statements

Review of BodyBuilder file types

.MKR File – Marker File

- A list of marker labels used in the trial and/or model
- The labels listed in the .mkr file can belong to both real markers or virtual points created as model outputs
- In legacy software this file was also used for autolabeling. Nexus now uses the .vst/.vsk file approach for autolabeling

.MP File – Model Parameters

- Constants used in calculations (Anthropometric measurements)
- These can be entered manually by the user or directly from the BodyLanguage routine (see *PARAM* statement)

.MOD File – Model Code

- This file contains the BodyLanguage routine code the user writes

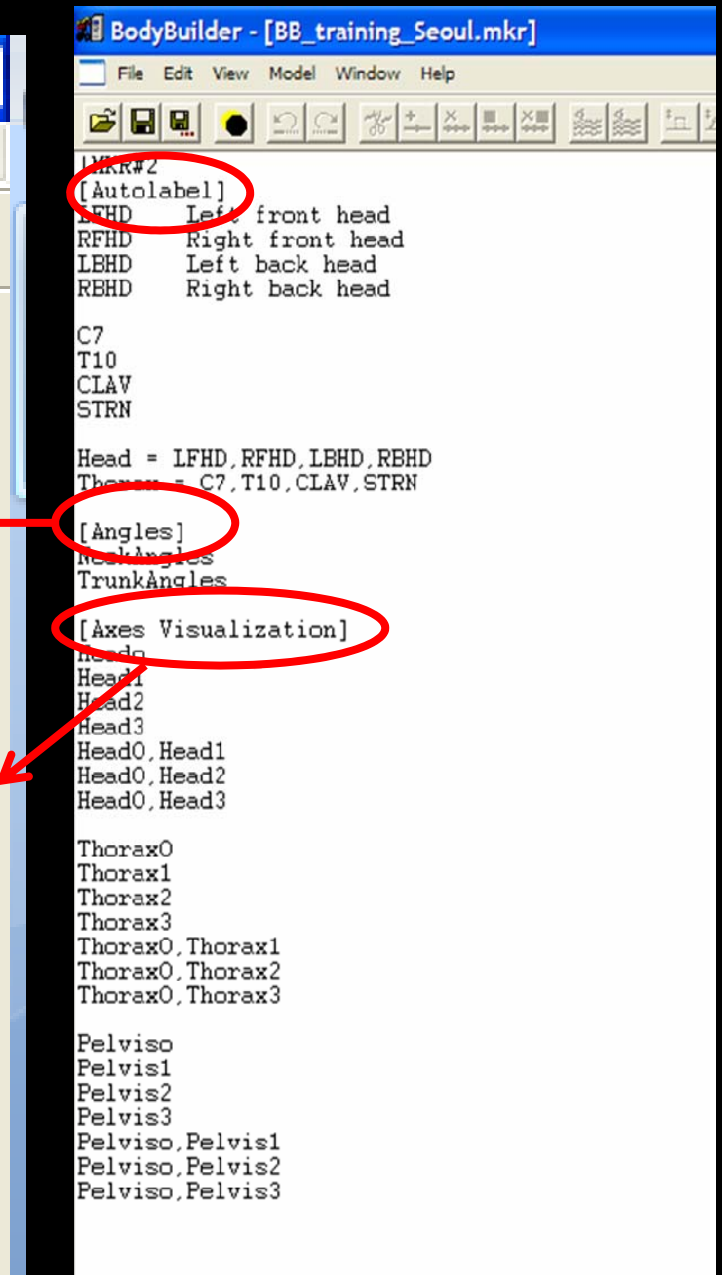
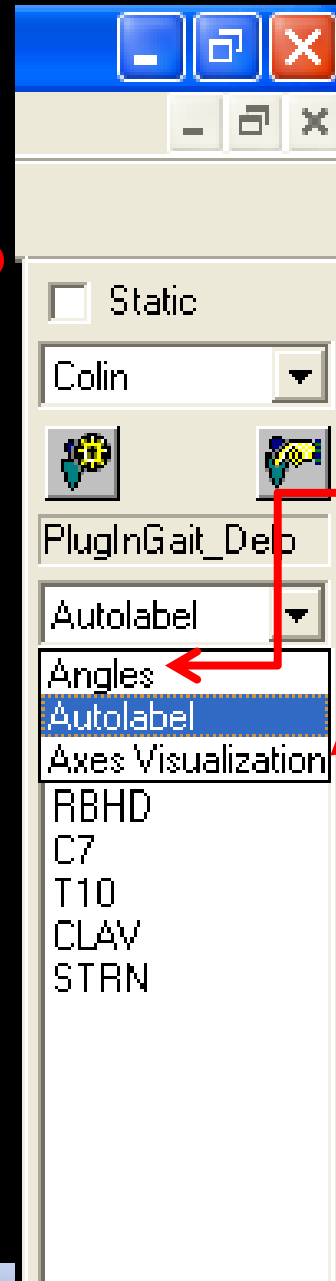
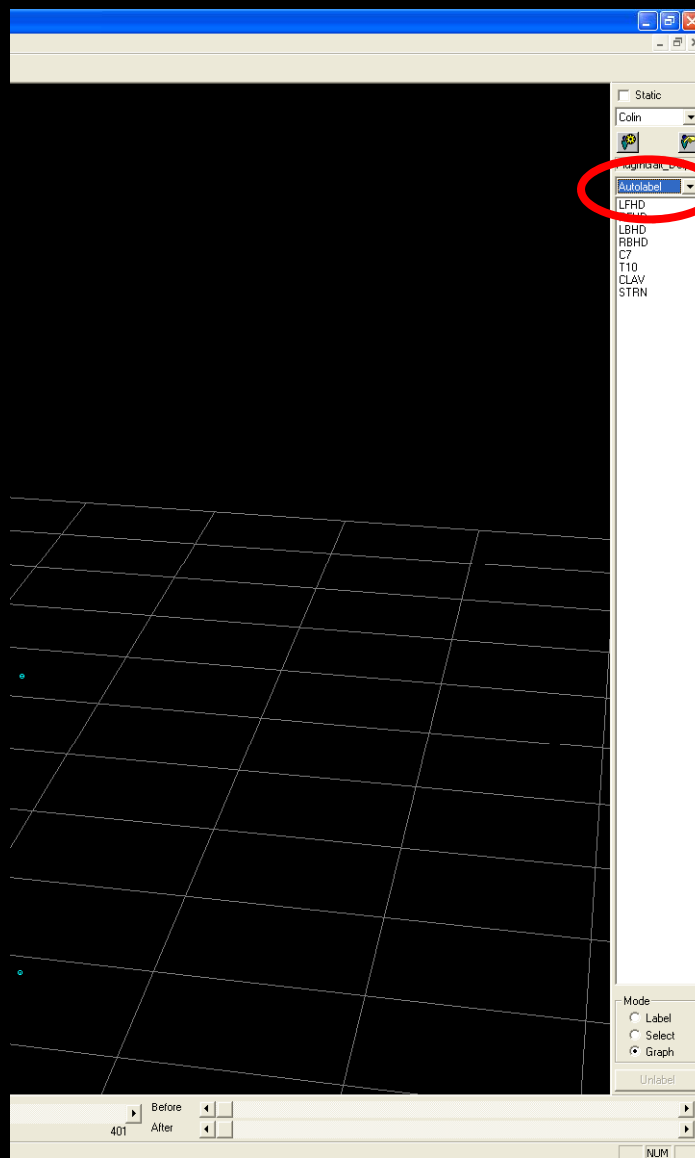
.C3D File – Processed Data

- This file is created by Nexus to hold processed marker data and analogue data
- When a BodyLanguage routine is executed on the data, the results of the calculations can also be saved to the .c3d file.

Creating an .MKR File

Creating an .MKR File

1. The .MKR file can be created using any text editor
2. Header – The first line of an .MKR file must always be as follows
 - !MKR#2
3. Utilizing Different Sections
 - Use a section header to break the file contents into different sections used in BodyBuilder
 - [autolabel], [display], [angles], [kinetics]
 - Each section can be selected using the marker section drop-down menu in BodyBuilder



Creating an .MKR File

1. The .MKR file can be created using any text editor
2. Header – The first line of an .MKR file must always be as follows
 - !MKR#2
3. Utilizing Different Sections
 - Use a section header to break the file contents into different sections used in BodyBuilder
 - [autolabel], [display], [angles], [kinetics]
 - Each section can be selected using a drop-down menu in BodyBuilder
4. Syntax
 - One label per line
 - A description of the marker can be added, separated from the marker label by a tab
 - LASI Left Anterior Superior Iliac Spine
 - LHipAngles Left Hip Angles
 - Marker labels case-insensitive
5. Creating linkages between markers **in the BodyBuilder 3D workspace**
 - Define the markers to be joined in the 3D workspace
 - List marker labels separated by a comma
 - LASI,LPSI – draws a green line from the LASI to the LPSI markers
 - LASI,LPSI,RASI,LASI – draws lines between every marker in the list

NOTE: The connection lines created in the .MKR file are NOT visible in the Nexus workspace. Nexus uses the .VST file to create linkages between the markers in the 3D workspace

.MKR file example

File Header
Section Header

Marker Label
Marker Description

Section Header

Section Header

Link between the
markers in the workspace

NOTE: the markers to be linked
need to be first declared in the SAME
section where the link has been
defined

NOTE2: A marker can be defined in more
than one section

```
BodyBuilder - [BB_training_Seoul.mkr]
File Edit View Model Window Help

!MKR#2
[Autolabel]
LFHD    Left front head
RFHD    Right front head
LBHD    Left back head
RBHD    Right back head

C7
T10
CLAV
STRN

[Angles]
NeckAngles
TrunkAngles

[Axes Visualization]
Heado
Head1
Head2
Head3
Head0,Head1
Head0,Head2
Head0,Head3

Thorax0
Thorax1
Thorax2
Thorax3
Thorax0,Thorax1
Thorax0,Thorax2
Thorax0,Thorax3

Pelviso
Pelvis1
Pelvis2
Pelvis3
Pelviso,Pelvis1
Pelviso,Pelvis2
Pelviso,Pelvis3
```

Autolabel section

Angles Section

Axes Visualization
Section

BodyBuilder Practice

Lets go to BodyBuilder and practice creating an .MKR file

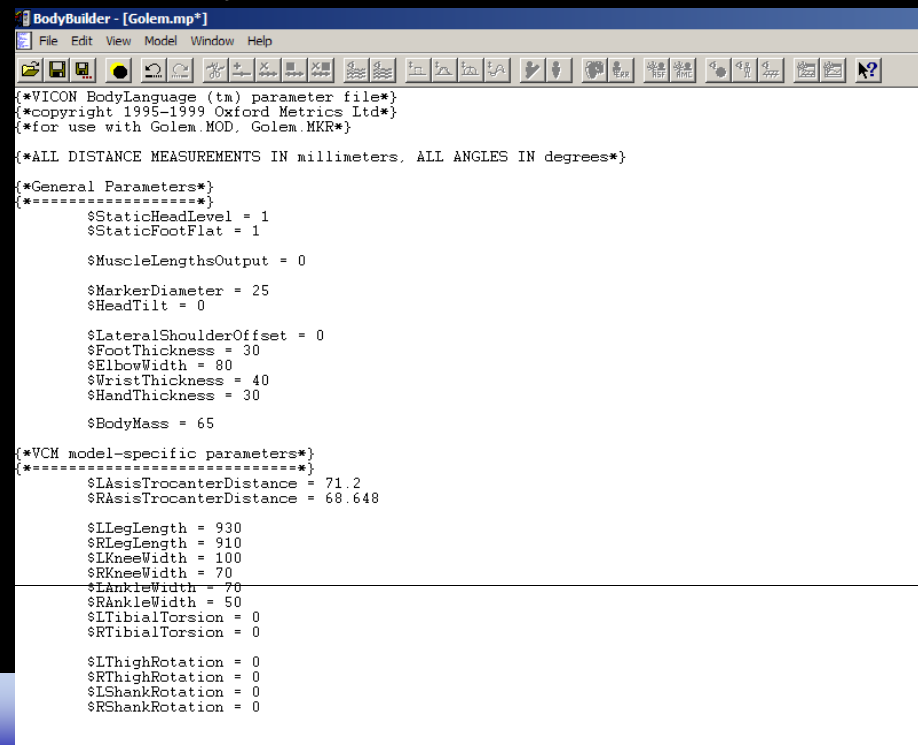
1. Create a blank text file using a text editor and save it with the .MKR extension
2. Open the newly created .MKR file in BodyBuilder
3. Enter the header
4. Create the Autolabel section
5. Add markers in the Autolabel section
6. Create sticks between markers
7. Add a section for the model outputs

Creating an .MP file

Creating an .MP file

- The .MP file is a text file that can be created, viewed or edited in BodyBuilder or using any generic text editor
- The .MP file contains all subject-specific parameters required by the BodyLanguage routine to be used
 - Anthropometric Measurements
 - Static Offsets
- For each subject a different .MP file is needed
- The .MP file gets merged with the .MOD file when executing the BodyLanguage routine
- Each parameter name must begin with the '\$' symbol
 - \$Height = 1800
- Comments can be added for clarity
 - *{* this is a comment *}*

NOTE: If the BodyLanguage routine to be used does not require subject-specific parameters, an empty .MP file must be created and associated to the subject anyway.

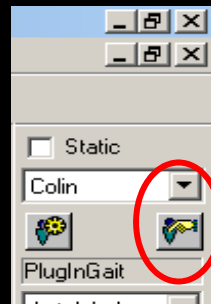


```
BodyBuilder - [Golem.mp*]  
File Edit View Model Window Help  
[*VICON BodyLanguage (tm) parameter file*]  
[*copyright 1995-1999 Oxford Metrics Ltd*]  
[*for use with Golem.MOD, Golem.MKR*]  
  
[*ALL DISTANCE MEASUREMENTS IN millimeters, ALL ANGLES IN degrees*]  
  
{*General Parameters*}  
{******}  
$StaticHeadLevel = 1  
$StaticFootFlat = 1  
  
$MuscleLengthsOutput = 0  
  
$MarkerDiameter = 25  
$HeadTilt = 0  
  
$LateralShoulderOffset = 0  
$FootThickness = 30  
$ElbowWidth = 80  
$WristThickness = 40  
$HandThickness = 30  
  
$BodyMass = 65  
  
{*VCM model-specific parameters*}  
{******}  
$LAsisTrocanterDistance = 71.2  
$RAsisTrocanterDistance = 68.648  
  
$LLegLength = 930  
$RLegLength = 910  
$LKneeWidth = 100  
$RKneeWidth = 70  
$LAnkleWidth = 70  
$RAnkleWidth = 50  
$LTibialTorsion = 0  
$RTibialTorsion = 0  
  
$LThighRotation = 0  
$RThighRotation = 0  
$LShankRotation = 0  
$RShankRotation = 0
```

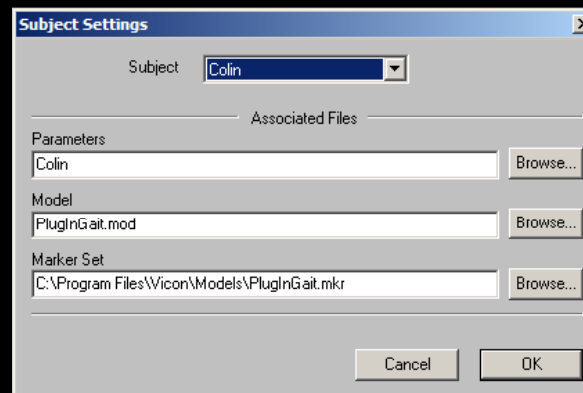
Associate the files with the subject

Once the .MKR, .MP and .MOD files are created, they need to be associated with the subject

1. Open the trial in BodyBuilder
2. Click on the 'Subject Settings' button



3. Browse for the .MKR, .MP and .MOD files to be associated with the subject



4. Click OK

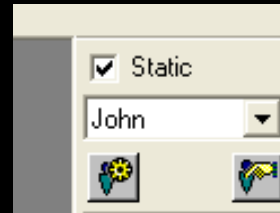
Using a Static Trial

Why Use a Static Trial

- Modelling in BodyLanguage is often broken up into “Static” and “Dynamic” processing.
- The Static processing is performed on a static trial where the subject is standing still in a neutral position.
- A static trial can be used to calculate static parameters from your subject that will be used later to execute your BodyLanguage routine on dynamic trials
 - Offset Angles
 - Anthropometric measurements calculated by the BodyLanguage routine using markers on the subject
- The static parameters are usually written to the .MP file.
- The static trial can also be used to capture markers that are removed in the dynamic trials
 - medial knee and ankle markers used in static trial to calculate joint width

How to Designate a Trial as Static

BodyBuilder – Use the Static check box on the right side of the interface



Nexus – It is not possible to designate a trial as static in Nexus. It is possible to run the static part of the BodyLanguage routine from the Nexus pipeline by selecting the 'Perform Static BodyLanguage modelling'.

See Plug-in Modeller for further details.

The IF condition – create code to run on the Static Trial only

To designate code that will run only on a static trial you use a conditional statement in your BodyLanguage routine

```
IF $Static == 1 THEN  
    <code that runs only on a static trial>  
ENDIF
```

If you would like to designate code that runs only on dynamic trials you can use the code below. In that case, the block of instructions included in the IF condition is run only if the \$Static variable equals to zero, meaning that the trial analyzed is a dynamic trial

```
IF $Static == 0 THEN  
    <code that runs only on a dynamic trial>  
ENDIF
```


About .MP files and static trials

When modelling includes the use of a static trial to store static parameters in the .MP file, there is a slightly different procedure in place for the creation of the .MP file in order to make it clearer to read and interpret by the users.

In fact, a generic .MP file (usually named <model name>.MP) can be created alongside with the .MOD file (<model name>.MOD). The generic .MP file can include comments, or parameters that are not dependent on the subjects (i.e. local coordinates of a muscle insertion).

This generic .MP file gets associated to the subject prior to executing the BodyLanguage routine on the static trial.

Usually the static trial is opened in BodyBuilder first. A generic MP file (usually <model name>.MP) is associated with the subject in the static trial, along with the model and marker files. When the model is run on the static trial the parameters that are calculated and need to be stored for use with the dynamic trials are stored in the MP file. This subject-specific MP file is then saved with a new name, for example <subject name>.MP. Then when a dynamic trial is opened the subject-specific MP file is associated with the trial along with the model and marker files. This allows the BodyLanguage routine to use the subject-specific parameters when modelling on the dynamic trials

Non Assigning Statements

What are Non Assigning Statements?

Non-assigning statements are used in BodyLanguage for three functions:

Function	BodyLanguage Instruction
Write to the model parameter (.MP) file	PARAM
Show newly created variables in the 3D workspace and .C3D file	OUTPUT
Flag an object as being optional	OptionalXXX

Depending on what kind of object has to be declared as optional, one of the following has to be used:

- OptionalNumbers() – scalars
- OptionalPoints() – markers (real and/or virtual)
- OptionalForces() – forces
- OptionalMoments() – moments
- OptionalReactions() – reactions

Non Assigning statement - PARAM

PARAM(*variable1, ..., variableN*)

- The PARAM command automatically takes the average of a variable over the whole trial and writes it to the .MP file.
- A typical situation when the PARAM command needs to be used is when some static parameters get calculated and have to be written to the subject's .MP file

Example: determine the “Interasisdistance” width as the distance between The RASI and the LASI.

```
$InterASISDist=DIST(LASI,RASI)
```

```
{* calculates the average of KNEwidth and writes it to  
.MP file *}
```

the

Add the line:

```
PARAM($InterASISDist)
```

Will put the value in the MP file to be remembered during the dynamic trials

The PARAM command – writing static parameters to the .MP file

Once a static parameter has been calculated in the static section of the BodyLanguage model, we may want to export that value to the .MP file using the PARAM function

Example:

```
IF $Static == 1 THEN  
    <code that runs only on a static trial>  
    PARAM(Variable)  
ENDIF
```

Non Assigning statement - OUTPUT

OUTPUT(*variable1*, ..., *variableN*)

- The OUTPUT command makes a newly created variable visible in the 3D workspace.
- When the save button is pressed in BodyBuilder, the variable gets also written to the .C3D file

Example – Lets say you calculate the Sacrum as the midpoint between the PSI markers. You may want to output that location to the C3D file inorder to see the “virtual point” you just created.

$SACR = (LPSI + RPSI) / 2$
OUTPUT(SACR)

Non Assigning statement – Optional Objects

OptionalPoints(Point1, ..., PointN)

Any Scalar, Marker, Force, Moment, Power, Reaction that may or may not be present in the trial has to be declared as optional, or the line that references them when they are not present in the trial will produce an error when the BodyLanguage routine gets executed

A typical example when the *OptionalPoints()* command comes in useful is when the BodyLanguage routine has to execute different sets of instructions according to which marker is attached to the subject

Correct Code	Erroneous Code
OptionalPoints(LASI) IF EXIST(LASI) THEN instructions ELSE differentInstructions ENDIF	IF EXIST(LASI) THEN instructions ELSE differentInstructions ENDIF

The Erroneous Code will run successfully as long as the LASI marker is present in the trial. If the LASI marker is not present in the trial, then the IF EXIST(LASI) line will be flagged as erroneous and the execution will stop.

Assignment 2 – part 1

1. Open the golemPIM.mkr file in BodyBuilder, delete all test except for the header and save this file as training.mkr.
2. Create an AutoLabel section
3. Enter in the marker names for the Pelvis markers
4. Create green sticks between all pelvis markers
 1. Attach this MKR to your example trial (Walking static.c3d) and ensure you see only the pelvis labeled and green sticks between all markers.
5. Now create a new section in your MKR file called “test” and create sticks here that outline the pelvis markers (no criss-cross markers in the middle of the segment).
 1. Attach the MKR to your example trial and change between the Autolabel and test sections to see the sticks change.
6. Go back to the Autolabel section and create a segment called Pelvis that includes all Pelvis markers.
 1. Attach the MKR to your example trial and see how this changes the sticks.

Assignment 2 – part 1

7. Now open the GolemPIM.mp file, delete all of the text and save this file as training.mp.
 1. Ensure this training.mp file is saved to the same folder as training.mkr
8. Now open GolemPIM.mod, delete all of the text and save this file as training.mod
 1. Ensure this training.mod is in the same folder as training.mp and training.mkr
10. Enter in the code in the MOD file to find the Sacrum point.
 1. Add the SACR marker to your MKR file
 2. Output the SACR marker from the MOD file.
 3. Run the mod file and ensure that you see the SACR marker appear.
11. Enter in the necessary code for a Static only section.
12. Now enter code to calculate a static parameter called InterAsisDistance as described earlier in this lecture.
 1. Be sure to send this parameter to the .MP file and check to make sure this happens when you run the model.