Vicon Skeleton Template (.vst) Files B

The Vicon Skeleton Template (*.vst*) file is used by Vicon software to describe relationships between Vicon markers and a subject's segments and joints. The format of this file enables you to describe any sort of kinematic model from a single, rigid object to a complex advanced model of a human being. A *.vst* file for a basic kinematic model of a human being describes the generic hierarchy and geometry of a human skeleton. The *.vst* file for a musculoskeletal model can be extended to include normalized attachment data for muscles, ligaments, and wrap objects as well as models for muscle and ligament force.

A *.vst* file for use with the RealTime Engine consists of the following three main sections:

- **Parameters:** Contains the parameters used to describe variations in the subject proportions and marker placement
- **Skeleton:** Describes the skeletal hierarchy and geometry
- **MarkerSet:** Describes the markers that form part of the template and their position with respect to assigned segments. Segments are defined by their markers, not the other way around.

These three main sections must be present in every *.vst* file for use with the RealTime Engine, but the content of each section can vary according to specific requirements of the model you are creating. The *.vst* files for use with other Vicon application software may contain different sections. For details, see the documentation for that Vicon software.

VST File Format

Vicon Skeleton Template (*.vst*) files are written in eXtensible Markup Language (*.xml*) file format. These *.xml* files are used to store structured information and to enclose or encapsulate information so that it can be passed between different computing systems. As in any *.xml* file, *.vst* file sections are delimited with start and end section tags, and subsections are nested beneath main sections to reflect their hierarchical relationship, for example:

```
<Parameters>
   <Parameter NAME="HalfHipWidth" VALUE="90"/>
   <Parameter NAME="BackHeight" VALUE="250"/>
   <Parameter NAME="ClavicleHeight" VALUE="250"/>
   <Parameter NAME="NeckHeight" VALUE="250"/>
</Parameters>
```

Important

The *.vst* file can be viewed in most Web browsers or standard text editors like any other *.xml* file. You may find it helpful to view one of the sample *.vst* files provided under the Models folder for your Vicon application software (by default, *C:\Program Files\Vicon\Models*) while you read this appendix describing the general format and content of a *.vst* file for the human body. This appendix assumes you have a basic understanding of the use and presentation of *.xml* files.

VST File Structure

The full structure of a *.vst* file for use with the RealTime Engine is outlined below. Details for the contents of each of the main sections of the file are provided in the following sections of this appendix.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<KinematicModelTemplate VERSION="1.0">
   <Parameters>
      <Parameter NAME="" ATTRIBUTE="value"/>
      ...
   </Parameters>

   <Skeleton>
      <Segment NAME="" ATTRIBUTE="value">
```

```
            <JointType ATTRIBUTE="value"/>
               <Segment ATTRIBUTE="value">
               ...
               </Segment>
         </Segment>
         ...
      </Skeleton>

      <MarkerSet NAME="">
         <Markers>
            <Marker NAME="" ATTRIBUTE="value"/>
            ...
         </Markers>

         <Sticks>
            <Stick ATTRIBUTE="value"/>
            ...
         </Sticks>
      </MarkerSet>
      <ScalingGroups>
         <Group/>
         ...
      </ScalingGroups>
</KinematicModelTemplate>
```

XML Header

The header `<?xml version="1.0" encoding="UTF-8" standalone="no"?>` is essential for the file to be recognized as *.xml* by Vicon applications and other *.xml* viewing packages. Do not edit this section.

VST File Attribute Syntax

Each option in a given section of a VST file consists of a keyword and one or more attribute="value" pairs. This is written in the following way in a *.xml* file:

```
<Keyword ATTRIBUTE="value"/>
```

The `Keyword` identifies the element with which the Vicon application is to be concerned, the `ATTRIBUTE` is a characteristic of the specified keyword, and the `value` is the actual data that the Vicon application is to use. For most options, you must specify the value required for your specific implementation of the *.vst* file. To help you understand how to construct these definitions in your *.vst* file, when giving the syntax for a section, this appendix provides the `Keyword` and `ATTRIBUTE` words, followed by a description of the type of data you must supply for the value within the quotation marks for each `ATTRIBUTE`.

For example:

```
<Joint NAME="" POSITION=""/>
```

where:

`NAME`	is a name that uniquely identifies the joint.
`POSITION`	is the joint's position in the local coordinate system (specified as x y z).

A number of attributes take values based on the item's position in the local coordinate system. These are specified in the following format "`x y z`" where the `x`, `y` and `z` values are in local segment coordinates.

Where a keyword can take multiple `ATTRIBUTE="value"` pairs, the order in which you specify each pair is not significant.

Parameters Section

The `<Parameters>` section of a *.vst* file for a kinematic model contains a list of the parameters that are used to describe the geometry of the skeleton.

The start and end section tags for this section are:

```
<Parameters>
   ...
</ Parameters>
```

Within this section, one or more individual <Parameter> definitions are included in the following format:

```
<Parameter NAME="" VALUE=""/>
```

where:

NAME	is the unique name of the parameter.
VALUE	is the value of the parameter. Length units are specified in millimeters, and rotational units are specified in radians.

Parameters are provided to enable the real geometry of a model to be scaled to a specific subject. For example, a Thigh parameter might be used to represent the length of the femur and to scale the ltibia and rtibia segments. A LeftFemurLength parameter might be used to represent the length of the femur, to scale the LeftFemur segment, and to locate the LkneeAngles joint type in the <Skeleton> section of the *.vst* file. For specific examples, see the sample *.vst* files supplied with your Vicon application software.

Skeleton Section

The <Skeleton> section of a *.vst* file for a basic kinematic model contains information about the hierarchical connections between a set of segments. Each segment has a joint that connects it to its parent.

The start and end section tags for this section are:

```
<Skeleton>
   ...
</Skeleton>
```

Within this section, one or more individual definitions are included in the following format:

```
<Segment NAME="" BOUNDS="" ORIENTATION="" POSITION="">
   <JointType/>
</Segment>
```

where:

NAME	is the name of the skeleton segment. If you are streaming RealTime data through Workstation into Polygon, the names of segments in the *.vst* file must exactly match the names of mesh segments in the mesh (*.obj*) file. For example, a trial in which Plug-In Gait has been run contains the segments `pelvis`, `lfemur`, `ltibia`, etc. Look in the *RT_PlugInGait_FullBody_COM.vst* example file to confirm that the `NAME` attributes specified for the corresponding `<Segment>` parameters match the name of the mesh segments in the *RT_PlugInGaitBones.obj* file.
BOUNDS	specifies the 3D bounding box for the segment to be displayed in the 3D Workspace View as a pair of vectors, each in `x y z` format. The values for the first vector `b1-b3` identify one corner and `b4-b6` the opposite corner with respect to the segment's local coordinate system.
ORIENTATION	specifies the orientation of the joint attaching the segment to its parent. This is expressed in the reference coordinate system of the parent segment. The orientation is specified as a helical vector. The direction of this vector gives the direction of the axis. The magnitude of this vector gives the amount of rotation around that axis in radians.
POSITION	specifies the position of the segment's joint attaching it to its parent in the reference coordinate system of the parent segment.

`JointType` identifies the type of joint used between segments, one of:

`JointDummy`	A fixed joint with 0 (zero) Degrees of Freedom (DOF), for example a joint that has been surgically pinned. This joint type has no translational or rotational freedom.
`JointFree`	A 6 DOF free joint. This joint type has full translational as well a rotational freedom. It is typically used for the root segment to allow the root to move freely with respect to the global origin.
`JointBall`	A 3 DOF ball-and-socket joint. This joint type has full rotational (but not translational) freedom. The position of the child segment is defined from the position of the parent and joint, but its orientation can vary freely.
`JointHardySpicer`	A 2 DOF "Hardy-spicer" (or universal) joint. This joint type has two rotational degrees of freedom around two orthogonal axes. You must specify two vectors defining the directions of the two axes in the coordinate system of the parent segment around which the joint can rotate in the following format: `AXIS-PAIR="x y z x y z"`

`JointHinge`	A 1 DOF hinge joint This joint type has a rotational degree of freedom around a single axis. You must specify a vector defining the axis of the hinge in the format: `AXIS="x y z"`

Important

If you are using the Plug-in Gait option for Workstation, the *RT_PlugInGait_FullBody_COM.vst* sample file assumes that skeleton kinematics have been calculated by Plug-In Gait, where the majority of joints connecting segments are considered to be 3 DOF revolute joints, i.e. of type `JointBall`.

Each `JointType` has additional common attributes that specify its behavior:

`MEAN`	The mean joint angle. This is calculated automatically by the subject calibration procedure, which returns a vector of the same length as the number of DOF for the joint: `"x y z a b c"` - 6 DOF free joint `"x,y,z"` - 3 DOF ball-and-socket joint `"x y"` - 2 DOF "Hardy-spicer" (or universal) joint `"x"` - 1 DOF hinge joint

COVARIANCE	The joint state covariance. This value is taken from a square matrix representing the amount of movement in a particular joint. The number of rows and columns in the matrix correspond to the number of DOF for the joint (as specified in MEAN).
T	The joint state T-pose. This represents constraints that can be assumed for this joint in the specified T-pose. State variables that are not a known value in the T-pose can be replaced with an asterisk (*) wildcard character.

MarkerSet Section

The <MarkerSet> section of a *.vst* file for a kinematic model specifies all the markers that form part of the template and assigns each of the markers to a segment. Each marker's position with respect to that segment also is specified.

The start and end section tags for this section are:

```
<MarkerSet>
   ...
</MarkerSet>
```

Within this section, include one or more of the following individual components:

<Markers>	The name of the Vicon marker.
<Sticks>	The lines connecting two points displayed in the 3D workspace.

The format of each of these <MarkerSet> section components is described in the following sections.

Markers

The <Markers> component of the <MarkerSet> section defines the Vicon markers for a human kinematic model.

A segment is defined by the markers, so the template must constrain the local coordinate systems of the segments with respect to the markers being tracked.

The format of this component is:

```
<Markers>
   <Marker NAME="" POSITION="" SEGMENT="" COVARIANCE=""/>
   ...
</Markers>
```

where:

NAME	is the name of the Vicon marker, for example, LELB. If you are using Workstation, this must match the name of the marker in the corresponding *.mkr* file.
POSITION	is the position of the marker in the coordinate frame of the segment to which it is attached in x y z format. The position can be specified as a mixture of numerical values (in millimeters) and NAME values from a <parameter> specified in the <Parameters> section of the *.vst* file. For example, the position of the RANK marker could be specified as: POSITION="RANKx -50 -Shin"
SEGMENT	is the segment to which the marker is attached.
COVARIANCE	is the movement covariance for the marker. This value is taken from a 3 x 3 square matrix representing the accuracy with which the marker is located in the segment's local coordinate system.

Sticks

The <Sticks> component of the <MarkerSet> section defines two markers between which a line is to be drawn in the 3D Workspace. Sticks are a visual aid to illustrate the connection between reconstructed marker positions or between virtual points created by a kinematic model. Sticks have no significance other than enabling the user to visualize the relationship between two specified points. The user can turn off the display of sticks if desired.

The format of this component is:

```
<Sticks>
   <Stick MARKER1="" MARKER2="" RGB=""/>
   ...
</Sticks>
```

where:

MARKER1	is the first marker from which to start drawing the stick.
MARKER2	is the second marker to which to connect the end of the stick.
RGB	is the color of the stick, specified in RGB (Red Green Blue) format, for example 255 0 0 is the color red.