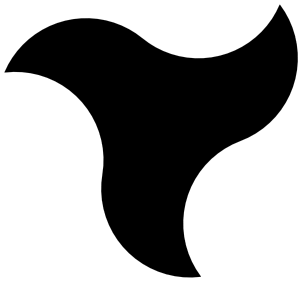
YEI Skeleton API



YEI Skeleton API

Virtual Skeleton Interaction System

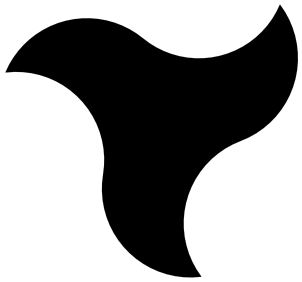
User's Manual

YEI Technology
630 Second Street
Portsmouth, Ohio 45662

www.YeiTechnology.com
www.PrioVR.com
www.3SpaceSensor.com

This page intentionally left blank

This page intentionally left blank



YEI Skeleton API

Virtual Skeleton Interaction System

User's Manual

YEI Technology
630 Second Street
Portsmouth, Ohio 45662

Toll-Free: 888-395-9029
Phone: 740-355-9029

www.YeiTechnology.com
www.PrioVR.com
www.3SpaceSensor.com

Table of Contents

This page intentionally left blank

This page intentionally left blank

1. Usage/Safety Considerations

1.1. Usage Conditions

- Please refer to the Usage/Safety Considerations section of the hardware being used with the YEI Skeleton API.

1.2. Technical Support

YEI provides technical and user support via our toll-free number (888-395-9029), via email (support@yeitech.com), and via community forum (forum.yeitechnology.com).

2. Overview of the YEI Skeleton API

2.1. Introduction

The YEI Skeleton API is an open source application program interface for creating skeletons that use external devices to manipulate the skeleton and the bones of the skeleton. The API is a collection of convenience functions for creating and using skeletons with YEI Prio and 3-Space Sensor devices or any other kind of device for use in a program written in C/C++ or any language that can import a compiled library (.dll, .so, etc). YEI Technology offers a 32-bit and 64-bit version of the API.

2.2. System Requirements

Operating System:

- Windows 7 or higher (32-Bit/64-Bit)
- Linux (Coming Soon)
- Mac OS (Coming Soon)

2.3. License

The YEI Skeleton API is released under the YEI 3-Space Open Source License, which allows for both non-commercial use and commercial use with certain restrictions.

- For Non-Commercial Use, your use of Covered Works is governed by the GNU GPL v.3, subject to the YEI 3-Space Open Source Licensing Overview and Definitions.
- For Commercial Use, a YEI Commercial/Redistribution License is required, pursuant to the YEI 3-Space Open Source

Licensing Overview and Definitions. Commercial Use, for the purposes of this License, means the use, reproduction and/or Distribution, either directly or indirectly, of the Covered Works or any portion thereof, or a Compilation, Improvement, or Modification, for Pecuniary Gain. A YEI Commercial/Redistribution License may or may not require payment, depending upon the intended use.

Full details of the YEI 3-Space Open Source License can be found online at
http://www.yeitechnology.com/yei-3-space-open-source-license

3. Getting Started with the YEI Skeleton API

There are a few things to know first that will help you to understand how the system works and what the data you are getting means.

3.1. API Terminology

There are a few terms you should be familiar with when using the API.

Processor – A processor is an instance that runs a process on a skeleton instance.

Skeleton ID – The skeleton ID is an ID that the API uses to communicate with skeleton and processor instances. This ID is needed for many functions of the API.

Standard – This term denotes that the method or attribute is the default for the API.

T-Pose – A pose for a skeleton where the arms are in line with each other creating a “T” shape.

Clasped Pose – A pose for a skeleton where the arms are straight in front with the hands together creating a “7” shape.

Neutral Pose – A pose for a skeleton where the arms are straight down the side with the thumbs going in creating an “I” shape.

Calibrate – Calibrating is the process of initializing the origin orientation for a device(s) being used by the skeleton or process.

3.2. Setting Up the YEI Skeleton API

Before using the YEI Skeleton API it is best to be sure to have the latest version of the API from the website (www.yeitechnology.com) and to have the latest software or hardware version of the devices to be used. This will give you the best results from the API and the devices.

For instructions for setting up to use the YEI Skeleton API in Visual Studio please refer to the documentation *YEI Skeleton API Quick Start Guide: Setup Environment for Visual Studio*.

4. Using the YEI Skeleton API

The YEI Skeleton API makes it easy for users to use devices to manipulate skeletons.

For an overview on the different ways of creating API instances, please refer to the *YEI Skeleton API Quick Start Guide* or the examples *creating_instances.c* and *using_processors.c* files.

4.1. Creating and Destroying Instances with the YEI Skeleton API

The first step the API must do is create an instance of a skeleton. The API provides 3 methods to create skeleton instances. For simplicity the following snippet example is only going to show the *yeiskel_createStandardSkeletonWithAge* function.

```
yei_skeleton_id skel_id;
uint8_t is_male = 1;
uint32_t age = 26;
skel_id = yeiskel_createStandardSkeletonWithAge(is_male, age);
if (skel_id == YEI_SKELETON_INVALID_ID)
{
    printf("Failed to create a skeleton\n");
    return 1;
}
```

The function *yeiskel_createStandardSkeletonWithAge* creates a standard skeleton using the parameters on the user's sex and age, and returns a skeleton ID. The skeleton ID is used to directly interact with the skeleton. It is also good practice when using the creating skeleton instances to check if the skeleton ID is a valid one. The API provides an enum of skeleton ID masks that can be used to check with.

Once a skeleton instance is created, a pose for the skeleton need to be set. The API provides 3 methods to set the pose for a skeleton instance. For simplicity the following snippet example is only going to show the *yeiskel_setStandardTPose* function. This function sets the skeleton to be in a “T” shape pose.

```
YEI_SKELETON_ERROR error;

error = yeiskel_setSkeletonToStandardTPose(skel_id);
if (error != YEI_SKELETON_NO_ERROR)
{
    printf("ERROR: %s\n", yei_skeleton_error_string[error]);
}
```

The function takes a *yei_skeleton_id* type as a parameter, and returns a *YEI_SKELETON_ERROR* enum type that denotes whether an error happened or not. Almost every API function has this parameter and returns this type. It is always best to check if the command succeeded, as it is very helpful to debug what went wrong.

After this initial setup the skeleton instance is ready to be used in your application. Please note that when you are finished with the skeleton instance calling *yeiskel_destroySkeleton* will destroy the instance. Also once you are done with the API, calling *yeiskel_resetSkeletonApi* will clean up anything remaining in the library and destroy loose instances.

```
yeiskel_destroySkeleton(skel_id);

error = yeiskel_resetSkeletonApi();
if (error != YEI_SKELETON_NO_ERROR)
{
    printf("ERROR: %s\n", yei_skeleton_error_string[error]);
}
```

5. Plug-ins and Wrappers

The YEI Skeleton API is capable of being integrated into many project types. This section will provide overviews on how to incorporate the API into game engines and other languages. The below examples assume some level of experience with each system.

5.1. Using the YEI Skeleton API with Unity & C#

This section is to help those that would like to create their own Unity plug-in for the YEI Skeleton API. For users that want to use the YEI Skeleton API Unity Plug-in provided by YEI Technology, please refer to the *YEI Skeleton API Unity Plug-in Quick Start Guide*.

5.1.1. Setting up the Plug-in

To use the API in any version of Unity, the library file should be placed in the **Unity\Editor** folder and in the same directory as the build project executable. To use the API with the Unity Pro version, follow the instructions on Unity's website.

To make calls to the API functions a C# script can be used to wrap the methods needed. The following is a snippet example from the *yei_skeleton_api.cs* file used for the YEI Skeleton API Unity Plug-in. Please refer to this file for more examples.

```
namespace yei_skeleton_api
{
    public static class yeiskel
    {
        [DllImport("Skeleton_API_32",
                   CallingConvention = CallingConvention.Cdecl,
                   EntryPoint= "yeiskel_createStandardSkeletonWithAge")]

        public static extern uint createStandardSkeletonWithAge(byte is_male,
                                                                 uint age);
    }
}
```

5.1.2. Using the Plug-in

When finished importing the functionality needed, simply importing the file into any Unity script will grant access to all the newly wrapped API methods and attributes.

```
void Start()
{
    skel_id = yeiskel.createStandardSkeletonWithAge(1, 26);
}
```

For more information on Unity plug-ins please refer to the Unity documentation.

5.2. Using the YEI Skeleton API with Python

This section is to help those that would like to create their own Python wrapper for the YEI Skeleton API. For users that want to use the YEI Skeleton API Python wrapper provided by YEI Technology, please refer to the *YEI Skeleton API Python Wrapper Quick Start Guide*.

The following is a snippet example from the *yei_skeleton_api.py* file used for the YEI Skeleton API Python wrapper. Please refer to this file for more examples.

```
from ctypes import cdll

YEI_SKELETON_API = cdll.LoadLibrary('./Skeleton_API_32.dll')

def createStandardSkeletonWithAge(is_male, age):
	return YEI_SKELETON_API.yeiskel_createStandardSkeletonWithAge(is_male, age)
```

For more information on developing library wrappers please refer to the Python documentation.

6. Reference Guide

6.1. YEI Skeleton API Specific Methods

The YEI Skeleton API specific methods are convenience functions that allow for creating instances of skeletons and processors.

Name	Description
yeiskel_getVersionString	Fill the given buffer with the version string.
yeiskel_resetSkeletalApi	Reset the Skeletal API.
yeiskel_getStandardBoneName	Gets the standard name of the bone.
yeiskel_createStandardSkeletonWithAge	Create a standard skeleton using user's age.
yeiskel_createStandardSkeletonWithHeight	Create a standard skeleton using user's height.
yeiskel_createSkeletonFromFile	Create a skeleton from file.
yeiskel_destroySkeleton	Destroys the skeleton.
yeiskel_loadBoneHierarchy	Sets up the skeleton with the contents of the given hierarchy file.
yeiskel_setSkeletonToStandardTPose	Set the skeleton to a standard T pose.
yeiskel_setSkeletonToStandardClaspedPose	Set the skeleton to a clasped pose with arms forward.
yeiskel_setSkeletonToStandardNeutralPose	Set the skeleton to a neutral pose with arms straight down.
yeiskel_addBoneAlias	Set an alias for a bone of the skeleton.
yeiskel_addProcessorToSkeleton	Add a processor to the skeleton at the given index.
yeiskel_removeProcessorFromSkeleton	Remove a processor to the skeleton at the given index.
yeiskel_getProcessorListFromSkeleton	Get the list of processors of the skeleton.
yeiskel_getProcessorNameListFromSkeleton	Get the names of processors of the skeleton.
yeiskel_update	Update the given skeleton.
yeiskel_getRootBoneName	Get the name of the root bone of the skeleton.
yeiskel_getBoneNameList	Get the names of bones of the skeleton.
yeiskel_getBoneOrientation	Get the orientation of the bone.
yeiskel_getBonePosition	Get the position of the bone.
yeiskel_getBoneVelocity	Get the velocity of the bone.
yeiskel_extractSkeletonHierarchy	Read out the skeleton hierarchy as an XML string.
yeiskel_isCalibrated	Tells if the skeleton has been calibrated yet or not.
yeiskel_getSkeletonName	Get the name of the skeleton.
yeiskel_setSkeletonName	Set the name of the skeleton.
yeiskel_getSkeletonUnit	Get the unit length measurement of the skeleton.
yeiskel_setSkeletonUnit	Set the unit length measurement of the skeleton.
yeiskel_utilityAddBone	Add a bone to the skeleton.
yeiskel_utilitySetBonePoseOrientation	Set the pose orientation of a bone.
yeiskel_utilitySetBoneLength	Set the length of a bone.
yeiskel_utilitySetBoneMass	Set the mass of a bone.

Name	Description
yeiskel_utilityLoadDeviceXmlMap	Maps the skeleton with the device of the given device xml file.

Serial Number:

_____

Notes:

YEI Technology

630 Second Street
Portsmouth, Ohio 45662

Toll-Free: 888-395-9029
Phone: 740-355-9029

www.YeiTechnology.com
www.PrioVR.com
www.3SpaceSensor.com