



Private Public Partnership Project (PPP)

Large-scale Integrated Project (IP)



## Virtual Characters Installation and Administration Guide

**Project full title:** Future Internet Core

**Project acronym:** FI-Core

**Contract No.:** 632893

# Contents

<b>Virtual Characters - Installation and Administration Guide</b>	<b>2</b>
<b>Introduction</b>	<b>2</b>
<b>System Requirements</b>	<b>2</b>
Hardware Requirements . . . . .	2
Operating System Support . . . . .	2
Software Requirements . . . . .	2
<b>Software Installation and Configuration</b>	<b>2</b>
<b>Sanity check procedures</b>	<b>3</b>
End to End testing . . . . .	3
List of Running Processes . . . . .	3
Network interfaces Up & Open . . . . .	3
Databases . . . . .	3
<b>Diagnosis Procedures</b>	<b>3</b>
Resource availability . . . . .	3
Remote Service Access . . . . .	3
Resource consumption . . . . .	3
I/O flows . . . . .	3

# Virtual Characters - Installation and Administration Guide

## Introduction

This document describes the installation & administration requirements for the Virtual Characters GE.

## System Requirements

### Hardware Requirements

A GPU that supports WebGL vertex and pixel shaders is required.

### Operating System Support

This GE is operating system independent, as it's JavaScript code running in a browser.

### Software Requirements

A web browser that supports WebGL rendering is required, such as

- Firefox 4.0+
- Chrome 8.0+
- Opera 15.0+
- Opera Mobile 12.0+
- Chrome for Android 31.0+
- Firefox for Android 25.0+

For serving the client HTML and JavaScript code, an HTTP server such as Apache can be used. For testing the browser can also be pointed to local files.

## Software Installation and Configuration

The Virtual Characters GE is delivered as part of the WebTundra JavaScript libraries, which also implement the scene model, rendering and network synchronization. It depends on the scene model and rendering code but only optionally of the networking code.

There are two options for getting it:

- Download a packaged release from <https://forge.fi-ware.org/frs/download.php/1151/MIWI-VirtualCharacters-3.3.zip/>
- Clone the git repository at <https://github.com/realXtend/WebTundra/>

The JavaScript code can be run from local files for testing, ie. it does not need to be served from an actual HTTP server. Note that several examples in the WebTundra codebase require a Tundra server to be running for network synchronization; these belong to the 3D-UI and Synchronization GE's and not to the Virtual Characters GE, which itself does not require client-server networking.

## Sanity check procedures

### End to End testing

To test that the necessary functionality is implemented by the browser, run the following HTML file from the WebTundra directory structure: `examples/Avatar/index.html`. You should see a robot character with attached clothes and swords playing a running animation. On a browser with lacking WebGL support (such as Internet Explorer 10) you will see a blank page.

### List of Running Processes

As the Virtual Character GE is client-only, typically the only process that is running is the web browser, for example Firefox or Chrome.

### Network interfaces Up & Open

N/A

### Databases

N/A

## Diagnosis Procedures

The most typical failure is that on browsers that do not support WebGL or do not have WebGL enabled a 3D web application or library such as the Virtual Characters GE will fail to run at all. Therefore check WebGL support first.

### Resource availability

Exact amounts of resources required depend on the scene that is being run, but rough estimation is that 1 GB of system RAM and 1 GB of hard disk space should be available. Considerably less is required for simple scenes. For example, the `examples/Avatar/index.html` test scene consumes 160 MB memory when run on a Firefox browser on Windows, and 5-10% CPU usage on an eight-core, 3 GHz Intel Core-i7 system. There should be no disk I/O after the scene has loaded.

### Remote Service Access

N/A

### Resource consumption

Because resource consumption (memory and CPU) depends on the scene being run, absolute abnormal values are not possible to give. Note the amount of memory use by the browser process once the scene has started/loaded, and see whether it starts to grow rapidly, for example doubles in a few seconds. That case would indicate a likely memory leak within the GE or the scene/application using it.

### I/O flows

N/A