SRP Engine Server

The SRP Engine Server allows you to setup an asynchronous engine farm. It was built with speed in mind, so it spins up as many engines as you request at startup and keeps them alive as long the server is up and running. This allows you to very efficiently hand off many self contained tasks in a fire-and-forget model. For example, you could write an MFS that sends a command to the engine farm to synchronize a record to another database at the moment it's being written without delaying the client. You can use it for a data processing center in which thousands of tasks arrive in a minute and are distributed to several waiting engines.

SRP Engine Server 1.3 is only compatible with SRP Utilities 1.5.5 or later.



Product Information

Version History

Contact Us for Download

Overview

The SRP Engine Server involves two pieces: client and server. The server consists of SRPEngineServer.exe and SRPEngineServer.ini. The client consists of SRP_TcpClient, which comes with SRP Utilities.

Installation

SRP will supply you with SRPEngineServer.exe and SRPEngineServer.ini. Place both of these into the root directory of OpenInsight. That is all you need to do for the server side.

For clients, you need to run our SRP Utilities installer to make sure the copy of OI that will act as the client has the SRP_TcpClient routine.

Server Setup

Setting up the server involves modifying the INI file. Here is a what the complete INI looks like:

```
[General]
Title = SRP Engine Server

[Directories]
Location = C:\Revsoft\OInsight
WorkingDir = C:\Revsoft\OInsight

[EnginePool]
Port = 7777
EngineCount = 4
ShowEngines = True
BasePipeName = EN

[OInsight]
Database = SYSPROG
Username = SYSPROG
Password =

[Autocommand]
Command =
Interval =
InitCommand =

[Network]
SocketTimeout = 10

[Support]
EnableLogging = False
LogDirectory = SRPEngineServerLogs
```

Title: The text that appears in the title bar of the SRP Engine Server window.

Location: The location option tells the engine where OpenInsight can be found.

WorkingDir: The working directory is the "active" directory used when the engine server is running. If you placed SRPEngineServer.exe into the root directory of OpenInsight, then set both working directory and location to the root directory of OpenInsight.

Port: The port upon which the engine server will be listening.

EngineCount: The number of the engines to spawn. All engines are started immediately when the engine server starts. This maximizes performance by not spinning up engines on an as-needed basis. Engines are always active and ready.

ShowEngines: When this is set to true, then the engines will be visible in the Windows taskbar. When false, the engines run invisibly and can only be seen in the task manager.

BasePipeName: The prefix used for the pipe names of all the child engines. For example, setting this to "PIPE" will cause the first engine to use "PIPE1", the second to use "PIPE2", and so on. The default is "EN".

Database: The OpenInsight application that each engine will log into.

Username: A valid username of the application.

Password: The username's password.

Command: If you want the engine to run a command regularly, then set this to a valid command string. A valid command is any command that can be successfully executed from OpenInsight's System Monitor

Interval: If you specified a command, then the interval determines how often it runs. Set this value to the number of milliseconds the engine should wait between calls. If you set this to 1000, for example, then the above command will execute every 1 second on the next available engine.

InitCommand: If you want each engine to run a command as soon as it starts, then set this to a valid command string. A valid command is any command that can be successfully executed from OpenInsight's System Monitor. This command will one exactly once when the engine starts.

SocketTimeout: Determines how long, in whole seconds, a connection is idle before being forcibly closed. This option is for maintenance purposes. Sometimes network problems happen and client connections get lost without the engine knowing it. Therefore, it needs to check every so often for orphaned connections. For example, you can set this to 10. If a connection has had no activity for 10 seconds, it will be disconnected automatically.

EnableLogging: If you set this to True, then the engine server will create logs in a subfolder of the current working directory called SRPEngineServerLogs. You can review these logs to locate possible errors. Logging slows things down, so be sure to turn this off when it is not needed.

LogDirectory: The directory used for storing log files. By default, the logs are stored in a subdirectory within the current working directory called "SRPEngineServerLogs." You can customize this setting, which could be useful if you are running multiple SRP Engine Servers. You can specify a full path or a relative one. Relative paths will use the current working directory as the parent folder.

Running Server

To run the server, just double-click SRPEngineServer.exe from Windows Explorer. It will use the INI file to determine its behavior. You will see a window displaying any commands that are in queue. Clicking either the Hide button or the X button in the top right corner will hide this window. To close the engine server, you must right click the icon in the system tray and choose Exit. Alternatively, you can close the engine server when the main window is visible (and has focus) by pressing the Esc key.

Sending Commands

Once the engine server is up and running, it is ready to receive commands from remote clients. All commands are sent using the SRP_TcpClient routin e that comes with SRP Utilities. Here's an example:

```
If SRP_TcpClient(TcpClientHandle, "CONNECT", ServerAddress, ServerPort) then
   SRP_TcpClient(TcpClientHandle, "SEND_SES", Command)
   SRP_TcpClient(TcpClientHandle, "CLOSE_SES")
end
```

The above code attempts to connect to a given TCP/IP address and port. The port must match the port number you specified in the server's INI file. Note that the variable TcpClientHandle is passed in unassigned in the first call because the "CONNECT" service initializes it. Once connected, you used the "SEND_SES" service to send a command. A command, as usual, is any command that is supported by OpenInsight's system monitor. For example, you can call a routine like so:

```
"run Calculate_Fibonacci 10"
```

Note that you will not get a response. The purpose of the SRP Engine Server is to hand off self-contained routines efficiently without delaying the client. Once you no longer need to send commands, you close the connection using "CLOSE_SES".

Search this documentation