Remote Visualisation System

Server User Guide

Last Modified: 02 March 2005

Introduction

This document is a guide to running the RVS Server. It is assumed that the server has been installed correctly according to the latest installation notes. The RVS Server consists of multiple components, one of which is a C++ binary and the others are Java applications. This documents contains information about the different components and how they can be stopped and started.

RVS Server Components

There are five components that make up the RVS Server. These components execute as independent processes and may be distributed over a LAN. These components are as follows.

Web Application (Command Centre)

This component executes as a Java web application, inside a web server such as Apache Tomcat. It has been given the name Command Centre because it is the user entry-point into the system and is responsible for initiating operations over the various RVS components.

User Manager

This is a Java component and is responsible for maintaining user-specific information.

Security Service

This is a Java component and is responsible for creating and managing user and session keys, creating sessions, and ensuring the integrity of the system by monitoring the server environment for rogue data/processes.

Data Communications Manager

This is a Java component and is responsible for all remote data access.

Session Manager

The session manager is a binary component and is the main process engine of the system. It creates and manages all sessions in the system. Each image canvas in the is encapsulated inside a session.

Java Pixel Canvas (JPC) Server

The JPC Server provides all drawing facilities on a Java-based drawing canvas.

Starting the Server

Starting Tomcat

Firstly, you need to start Tomcat, if it isn't already running. There are two important configurations you should make before starting Tomcat:

- It is recommended that you increase the maximum heap size used by Tomcat to 512MB. This can be done by setting the JAVA_OPTS environment variable to -Xmx512m. Alternatively you may want to edit the Tomcat startup scripts to set this explicitly.
- The jacorb.properties file in the $HOME of the user that starts tomcat should be a link to (or a copy of) $RVS_HOME/config/jacorb.properties.

Configuring for RVS Server execution

Before starting the server you need to confirm that your environment is set up according to the “Environment Setup” section of the RVS Server Installation Notes. In particular you need to ensure that $RVS_HOME/config/jacorb.properties is copied or symlinked to the $HOME of the user that starts Tomcat.

The **$RVS_HOME/bin/rvs.sh** script is used to start the RVS Server components. On a live system, this script should be run as the root user (other uses may run it via sudo). For testing purposes, however, it is possible to run it as the RVS_USER (see below).

Following are some parameters in this script that need configuring before you execute it.

- RVS_HOME: The home location of the RVS installation
- JAVA_HOME: The home location of the Tomcat installation
- CATALINA_HOME: The home location of the Tomcat installation
- RVS_USER: The user that the RVS components will execute as. Only this user and 'root' may execute the rvs.sh script. Even when executed as root the components are run as this user.
- RVS_GROUP: The group the RVS user must be in. This is the group that will own the files created by RVS in this script.
- JAVA_SLEEP: The number of seconds the script should wait for each component to start. It's advised to leave this as the default value.
- LOG_DIR: Where you want to log files to be written to. If the script will not be run as root, then the log directory needs to be writable by the RVS_USER.
- PID_DIR: Where you want temporary process control files to be written to. If the script will not be run as root, then the pid directory needs to be writable by the RVS_USER.

Note: The user that started tomcat MUST either be the same as RVS_USER or be part of the RVS_GROUP.

Start all Components

You need to start all of the other RVS components listed earlier. You may do this on any host, as long as all configuration parameters are set correctly. The easiest way of starting the server is to start all the components on a single host, for example:

```
shell% $RVS_HOME/bin/rvs.sh start
```

This is highly recommended if you're usign the RVS server for the first time. The alternative method of starting is by distributing each component.

Distributing Server Components

It is possible to distribute all the components across different hosts on a network. The only requirement is for them to be on the same network. You may start each component individually as follows.

Start the Name Service first:

```
shell% $RVS_HOME/bin/rvs-ns.sh
```

Then you can start the following scripts in any order (in essence this is what the $RVS_HOME/bin/rvs.sh script does). They are all located in $RVS_HOME/bin.

- rvs-dcm.sh
- rvs-sec.sh
- rvs-umis.sh
- rvs-jpc.sh
- rvs-sis.sh

Multiple Instances

It is possible to start multiple instances of the JPC Server (rvs-jpc.sh) and the Session Manager (rvs-sis.sh). Multiple instances of these components may be started at any time after which time the load is shared between all of the components. This technique is particularly useful when instances are started on different hosts.

Stopping the RVS Server

The following command can be used to stop all components executed using the 'rvs.sh start' command.

```
shell% $RVS_HOME/bin/rvs.sh stop
```

NOTE: This command DOES NOT stop processes started using the rvs-*.sh commands. You will need to stop those manually.

Keeping the system clean

Every now and then there will be some files that are not cleaned up after a user has finished using the server. This usually occurs when clients don't exit gracefully and their sessions time out. To keep the server clean, there is a script; $RVS_HOME/bin/rvs-cleanup.sh. It is recommended that you schedule this script to run about once a day (on live systems). You can set the CLEANUP_AGE variable in the script to tell it how old a file/directory can get before it should be deleted (default is 24 hours).