



Red Hat Software Collections 1.x

1.1 Release Notes

Release Notes for Red Hat Software Collections

Red Hat Software Collections 1.x 1.1 Release Notes

Release Notes for Red Hat Software Collections

Lenka Špačková

Red Hat Engineering Content Services

lspackova@redhat.com

Eliška Slobodová

Red Hat Engineering Content Services

eslobodo@redhat.com

Jaromír Hradílek

Red Hat Engineering Content Services

jhradilek@redhat.com

Legal Notice

Copyright © 2014 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

The Red Hat Software Collections 1.1 Release Notes document the major features and contain important information about known problems in Red Hat Software Collections 1.1.

Table of Contents

CHAPTER 1. RED HAT SOFTWARE COLLECTIONS 1.1	3
1.1. MAIN FEATURES	3
1.2. CHANGES IN RED HAT SOFTWARE COLLECTIONS 1.1	5
1.2.1. New Components	5
1.2.2. New Component: Thermostat 1	6
1.2.3. New Component: Ruby on Rails 4.0	6
1.2.4. New Component: MongoDB 2.4.9	6
MongoDB 2.4.9 on Red Hat Enterprise Linux 6	7
MongoDB 2.4.9 on Red Hat Enterprise Linux 7	7
1.3. COMPATIBILITY INFORMATION	7
1.4. KNOWN ISSUES	7
Other Notes	11
CHAPTER 2. INSTALLATION AND USAGE	12
2.1. GETTING ACCESS TO RED HAT SOFTWARE COLLECTIONS	12
2.1.1. Using Red Hat Subscription Management	12
2.1.2. Using RHN Classic	13
2.2. INSTALLING RED HAT SOFTWARE COLLECTIONS	14
2.2.1. Installing Individual Software Collections	16
2.2.2. Installing Optional Packages	16
2.2.3. Installing Debugging Information	17
2.3. UNINSTALLING RED HAT SOFTWARE COLLECTIONS	17
2.4. REBUILDING RED HAT SOFTWARE COLLECTIONS	17
2.5. USING RED HAT SOFTWARE COLLECTIONS	17
2.5.1. Running an Executable from a Software Collection	17
2.5.2. Running a Shell Session with a Software Collection as Default	18
2.5.3. Running a System Service from a Software Collection	18
2.6. ACCESSING A MANUAL PAGE FROM A SOFTWARE COLLECTION	19
2.7. DEPLOYING APPLICATIONS THAT USE RED HAT SOFTWARE COLLECTIONS	19
2.8. MIGRATING FROM MYSQL 5.1 TO MYSQL 5.5	19
2.8.1. Notable Differences Between MySQL 5.1 and MySQL 5.5	19
2.8.2. Upgrading from MySQL 5.1 to MySQL 5.5	20
2.8.3. Using the mysql55-mysql-devel Package	22
2.8.3.1. Using Database Connectors for Dynamic Languages	22
2.8.3.2. Building Applications for MySQL 5.5 from Red Hat Software Collections	23
2.9. MIGRATING FROM POSTGRESQL 8.4 TO POSTGRESQL 9.2	23
2.9.1. Notable Differences Between PostgreSQL 8.4 and PostgreSQL 9.2	23
2.9.2. Upgrading from PostgreSQL 8.4 to PostgreSQL 9.2	24
2.10. REPORTING BUGS IN RED HAT SOFTWARE COLLECTIONS	26
CHAPTER 3. ADDITIONAL RESOURCES	28
3.1. RED HAT ENTERPRISE LINUX DEVELOPER PROGRAM GROUP	28
3.2. RED HAT PRODUCT DOCUMENTATION	28
APPENDIX A. REVISION HISTORY	29

CHAPTER 1. RED HAT SOFTWARE COLLECTIONS 1.1

For certain applications, more recent versions of some software components are often needed in order to use their latest new features. **Red Hat Software Collections** is a Red Hat offering that provides a set of dynamic programming languages, database servers, and various related packages that are either more recent than their equivalent versions included in the base Red Hat Enterprise Linux system, or are available for this system for the first time. For a complete list of components that are distributed as part of Red Hat Software Collections and a brief summary of their features, see [Section 1.1, “Main Features”](#).

Red Hat Software Collections does not replace the default system tools provided with Red Hat Enterprise Linux 6 or Red Hat Enterprise Linux 7. Instead, a parallel set of tools is installed in the `/opt` directory and can be optionally enabled per application by the user using the supplied `sc1` utility. The default versions of Perl or PostgreSQL, for example, remain those provided by the base Red Hat Enterprise Linux system.

With the notable exception of **Node.js** and **nginx**, all Red Hat Software Collections components are fully supported under Red Hat Enterprise Linux Subscription Level Agreements, are functionally complete, and are intended for production use. Important bug fixes and security errata are issued to Red Hat Software Collections subscribers in a similar manner to Red Hat Enterprise Linux for at least three years from the release of each major version. A new major version of Red Hat Software Collections is released approximately every 18 months, and in each major release stream, each version of a selected component remains backward compatible.

1.1. MAIN FEATURES

Red Hat Software Collections 1.1 provides recent stable versions of the tools listed in [Table 1.1, “Red Hat Software Collections 1.1 Components”](#).

Table 1.1. Red Hat Software Collections 1.1 Components

Component	Software Collection	Description
Perl 5.16.3	perl516	A release of Perl with a number of additional utilities, scripts, and <i>database connectors for MySQL and PostgreSQL</i> . This version provides a large number of new features and enhancements, including new debugging options, improved Unicode support, and better performance. Also, it adds <code>perl-DateTime</code> and <code>mod_perl</code> , which is supported only with the <code>httpd24</code> Software Collection package.
PHP 5.4.16	php54	A release of PHP with PEAR 1.9.4 and a number of additional extensions. PHP 5.4 provides a number of <i>language and interface improvements</i> . The APC , memcache , and Zend OPcache extensions are also included.
PHP 5.5.6	php55	A release of PHP with <i>enhanced language features</i> including better exception handling, generators, and Zend OPcache . The memcache and mongodb extensions are also included.

Component	Software Collection	Description
Python 2.7.5	python27	A release of Python 2.7 with a number of additional utilities. This Python version provides various new features and enhancements, including a new ordered dictionary type, faster I/O operations, and improved forward compatibility with Python 3. The python27 Software Collections contains the <i>Python 2.7.5 interpreter</i> , a set of extension libraries useful for programming web applications and <code>mod_wsgi</code> (only supported with the httpd24 Software Collection), MySQL and PostgreSQL database connectors, and <code>numpy</code> and <code>scipy</code> .
Python 3.3.2	python33	A release of Python 3 with a number of additional utilities. This Software Collection gives developers on Red Hat Enterprise Linux access to Python 3 and allows them to benefit from various advantages and new features of this version. The python33 Software Collection contains <i>Python 3.3.2 interpreter</i> , a set of extension libraries useful for programming web applications and <code>mod_wsgi</code> (only supported with the httpd24 Software Collection), PostgreSQL database connector, and <code>numpy</code> and <code>scipy</code> .
Ruby 1.9.3^[a]	ruby193	A release of Ruby 1.9.3 and Ruby on Rails 3.2.8 with a <i>large collection of Ruby gems</i> . This Software Collection gives developers on Red Hat Enterprise Linux access to Ruby 1.9, which provides a number of new features and enhancements, including improved Unicode support, enhanced threading, faster load times, and <code>mod_passenger</code> , which is supported only with the httpd24 Software Collection package.
Ruby 2.0.0	ruby200	A release of Ruby 2.0.0. This version provides substantial performance and reliability improvements and includes a number of new features and improved debugging capabilities, while maintaining source level backward compatibility with Ruby 1.9.3.
Ruby on Rails 4.0.2^[a]	ror40	A release of Ruby on Rails 4.0 , a web application development framework written in the Ruby language. This version provides a number of new features and improvements and adds live streaming for persistent connections. This Software Collection is supported together with the ruby200 collection.
MariaDB 5.5.37	mariadb55	A release of MariaDB, <i>an alternative to MySQL</i> for users of Red Hat Enterprise Linux. MySQL is binary compatible with MariaDB and can be replaced with it without any data conversions. This version adds the PAM authentication plugin to MariaDB.
MongoDB 2.4.9^[b]	mongodb24	A release of MongoDB, a cross-platform <i>document-oriented database system classified as a NoSQL database</i> . This Software Collection includes the mongo-java-driver package.

Component	Software Collection	Description
MySQL 5.5.37	mysql55	A release of MySQL, which provides a number of new features and enhancements, including improved performance.
PostgreSQL 9.2.8	postgresql92	A release of PostgreSQL, which provides a number of new features and enhancements, including <i>cascading replication</i> , <i>native JSON support</i> , improved scalability, and better performance.
Node.js 0.10 ^{[b][c]}	nodejs010	A release of Node.js with npm 1.3.24 . This Software Collection gives users of Red Hat Enterprise Linux access to this programming platform.
nginx 1.4.4 ^[d]	nginx14	A release of nginx, a web and proxy server with a focus on high concurrency, performance and low memory usage.
Apache httpd 2.4.6	httpd24	A release of the Apache HTTP Server (httpd), including a high performance <i>event-based processing model</i> , <i>enhanced SSL module</i> and <i>FastCGI support</i> . The <code>mod_auth_kerb</code> module is also included.
Thermostat 1	thermostat1	A release of Thermostat, a monitoring and instrumentation tool for the <i>OpenJDK HotSpot JVM</i> , with support for monitoring <i>multiple JVM instances</i> . This Software Collection depends on the <code>mongodb24</code> component.
<p>[a] A part of this Software Collection requires a JavaScript engine. The <code>v8314</code> Software Collection included in Red Hat Software Collections 1.1 provides the V8 JavaScript engine and is supported only as the Software Collection's dependency.</p> <p>[b] This Software Collection also requires <code>v8314</code>. The <code>v8314</code> Software Collection included in Red Hat Software Collections 1.1 provides the V8 JavaScript engine and is supported only as the Software Collection's dependency.</p> <p>[c] In Red Hat Software Collections 1.1, Node.js is included as a Technology Preview. For more information about Red Hat Technology Previews, see https://access.redhat.com/support/offerings/techpreview/.</p> <p>[d] In Red Hat Software Collections 1.1, nginx is included as a Technology Preview. For more information about Red Hat Technology Previews, see https://access.redhat.com/support/offerings/techpreview/.</p>		

1.2. CHANGES IN RED HAT SOFTWARE COLLECTIONS 1.1

1.2.1. New Components

Apart from upgrading the components from Red Hat Software Collections 1.0 to later versions, which provide a number of bug fixes and enhancements over the previous versions, Red Hat Software Collections 1.1 adds these new components:

- `php55`
- `ruby200`

- `ror40`
- `mongodb24`
- `nginx14`
- `httpd24`
- `thermostat1`
- `v8314` - the `v8314` Software Collection is supported only as a dependency for the `mongodb24`, `ruby193`, `ror40`, and `nodejs010` Software Collections.

1.2.2. New Component: Thermostat 1

The **Thermostat** Software Collection provides a monitoring and instrumentation tool for the OpenJDK HotSpot JVM, with support for monitoring multiple JVM instances. The system is made up of two components: an **Agent**, which collects data, and a **Client**, which allows users to visualize collected data. These components communicate via a storage layer: either directly via **MongoDB** or indirectly via a Web layer for increased security. A pluggable agent and GUI framework allows for collection and visualization of performance data beyond what is included out of the box.

To install the `thermostat1` collection, run the following command as `root`:

```
yum install thermostat1
```

To enable this collection, type the following command at a shell prompt:

```
scl enable thermostat1 bash
```

In order to deploy the `thermostat1-thermostat-webapp`, that is to start the web storage endpoint in Red Hat Software Collections, use the following command:

```
sudo service thermostat1-thermostat-tomcat start
```

For more information, please refer to the [Thermostat User Guide](#). In order to deploy Thermostat securely, see the [Configuration and Administration Guide](#).

1.2.3. New Component: Ruby on Rails 4.0

This Software Collection adds the `ruby200` package together with the `ror40` package. The **Ruby on Rails** collection can be enabled by the following command, which will automatically enable `ruby200`:

```
scl enable ror40 bash
```

These two collections are supported together.

1.2.4. New Component: MongoDB 2.4.9

To install the `mongodb24` collection, run the following command as `root`:

```
yum install mongodb24
```

To run the **MongoDB** shell utility, type the following command:

```
sc1 enable mongodb24 'mongo'
```

MongoDB 2.4.9 on Red Hat Enterprise Linux 6

If you are using Red Hat Enterprise Linux 6, the following instructions apply to your system.

To start the **MongoDB** daemon, type the following command:

```
sudo service mongodb24-mongodb start
```

To start the **MongoDB** daemon on boot, use this command:

```
sudo chkconfig mongodb24-mongodb on
```

To start the **MongoDB** sharding server, run this command:

```
sudo service mongodb24-mongodb-shard start
```

To start the **MongoDB** sharding server on boot, use the following command:

```
sudo chkconfig mongodb24-mongodb-shard on
```

MongoDB 2.4.9 on Red Hat Enterprise Linux 7

When using Red Hat Enterprise Linux 7, the following commands are applicable.

To start the **MongoDB** daemon, run the following command:

```
sudo systemctl start mongodb24-mongodb.service
```

To start the **MongoDB** daemon on boot, use this command:

```
sudo systemctl enable mongodb24-mongodb.service
```

To start the **MongoDB** sharding server, type the following command:

```
sudo systemctl start mongodb24-mongodb-shard.service
```

To start the **MongoDB** sharding server on boot, run this command:

```
sudo systemctl enable mongodb24-mongodb-shard.service
```

1.3. COMPATIBILITY INFORMATION

Red Hat Software Collections 1.1 is available for all supported releases of Red Hat Enterprise Linux 6 and Red Hat Enterprise Linux 7 on AMD64 and Intel 64 architectures.

1.4. KNOWN ISSUES

nodejs-hawk component

The `nodejs-hawk` package uses an implementation of the SHA-1 and SHA-256 algorithms adopted from the CryptoJS project. In this release, the client-side JavaScript is obfuscated. The future fix will involve using crypto features directly from the CryptoJS library.

postgresql component

The `postgresql92` package for Red Hat Enterprise Linux 6 does not provide the `sepgsql` module as this feature requires installation of `libselinux` version 2.0.99, which is not available in Red Hat Enterprise Linux 6.

python27 and python33 components

Due to a bug in the `redhat-rpm-config` package, it is not possible to rebuild the `python27` and `python33` packages on Red Hat Enterprise Linux 6.2 Extended Update Support.

coreutils component

Some utilities, for example, `su`, `login` or `screen` do not export environment settings in all cases, which can lead to unexpected results. It is therefore recommended to use `sudo` instead of `su` and set the `env_keep` environment variable in the `/etc/sudoers` file. Alternatively, you can run commands in a reverse order; for example:

```
su -l postgres -c "scl enable postgresql92 psql"
```

instead of

```
scl enable postgresql92 bash
su -l postgres -c psql
```

When using tools like `screen` or `login`, you can use the following command to preserve the environment settings: `source /opt/rh/<collection_name>/enable`.

httpd, mariadb, mongodb, mysql, nodejs, perl, php55, python27, python33, ruby193, ror40, ruby200, thermostat, and v8314 components

When uninstalling the `httpd24`, `mariadb55`, `mongodb24`, `mysql55`, `nodejs010`, `perl516`, `php55`, `python27`, `python33`, `ruby193`, `ror40`, `ruby200`, `thermostat1`, or `v8314` packages, the order of uninstalling can be relevant due to ownership of dependent packages. As a consequence, some directories and files might not be removed properly and might remain on the system.

mariadb, mysql, postgresql, mongodb components

Red Hat Software Collections contains the **MySQL 5.5**, **MariaDB 5.5**, **PostgreSQL 9.2** and **MongoDB 2.4** databases. The core Red Hat Enterprise Linux 6 provides earlier versions of the **MySQL** and **PostgreSQL** databases (client library and daemon). The core Red Hat Enterprise Linux 7 provides the same versions of the **MariaDB** and **PostgreSQL** databases (client library and daemon). Client libraries are also used in database connectors for dynamic languages, libraries, and so on.

The client library packaged in the Red Hat Software Collections database packages in the **PostgreSQL** component is not supposed to be used, as it is included only for purposes of server utilities and the daemon. Users are instead expected to use the system library and the database connectors provided with the core system.

A protocol, which is used between the client library and the daemon, is stable across database versions, so, for example, using the **PostgreSQL 8.4** client library with the **PostgreSQL 9.2** daemon works as expected.

The core Red Hat Enterprise Linux 6 and Red Hat Enterprise Linux 7 does not include the client library for **MongoDB**. In order to use this client library for your application, you should use the client library from Red Hat Software Collections and always use the `sc1 enable . . .` call every time you run an application linked against this **MongoDB** client library.

mariadb, mysql, mongodb components

MariaDB, MySQL, and MongoDB do not make use of the `/opt/<provider>/<collection>/root` prefix when creating log files. Note that log files are saved in the `/var/log/` directory, not `/opt/<provider>/<collection>/root/var/log/`.

mongodb, thermostat components

The Java driver from the **MongoDB** collection and most of the **Thermostat** collection cannot be rebuilt on Red Hat Enterprise Linux 6. The following steps are necessary before rebuilding the `mongodb24` or `thermostat1` components on Red Hat Enterprise Linux 6 from the provided source RPM (SRPM) packages:

1. Install the `java-1.7.0-openjdk` package from Red Hat Enterprise Linux 6.3 or later and all its dependencies.
2. Download, build, and install the maven package and all its dependencies from the SRPM packages for Red Hat Enterprise Linux 7 Beta or later. For architecture-independent packages it is possible to use the binary packages provided for Red Hat Enterprise Linux 7.
3. Download and build the main collection SRPM (`mongodb24` or `thermostat1`).
4. Download and build additional SRPM packages including their dependencies.

httpd component

Compiling external applications against the Apache Portable Runtime (APR) and APR-util libraries from the `httpd24` Software Collection is not supported. The `LD_LIBRARY_PATH` is not set in `httpd24` because it is not required by any application in this Software Collection.

httpd, ruby193 components

`httpd` is unable to execute the binary files in the `mod_passenger` module, namely `PassengerWatchdog`, `PassengerHelperAgent`, `PassengerLoggingAgent`, and `SpawnPreparer` in the `/opt/rh/ruby193/root/usr/lib64/gems/exts/passenger-4.0.18/agents/` directory. To work around this problem, disable SELinux by running the following command as `root`:

```
setenforce 0
```

nginx component

In Red Hat Enterprise Linux 6, no SELinux policy is applied for the `nginx` daemon.

mariadb component

The permissions for the `/var/log/mariadb55-mariadb/` directory, in which the log file is stored, are set incorrectly. Consequently, when the `/var/log/mariadb55-mariadb/mariadb.log` file is removed in Red Hat Enterprise Linux 7, the service fails to start because the `mysqld` daemon does not have permission to create log files. To work around this problem, either do not remove the log file or change the owner of the `/var/log/mariadb55-mariadb/` directory to `mysql:mysql`.

thermostat component

Thermostat applies configuration only from the `USER_THERMOSTAT_HOME/etc/ssl.properties` file while the `THERMOSTAT_HOME/etc/ssl.properties` file is ignored. Consequently, Transport Layer Security (TLS) handshakes can fail because the configuration appears to be missing. To work around this problem, specify the TLS configuration in the `USER_THERMOSTAT_HOME/etc/ssl.properties` file. You can copy a template from `THERMOSTAT_HOME/etc/ssl.properties`. Please refer to the [Thermostat User Guide](#) for differences between installation location and user data location.

thermostat component

Thermostat ignores the `CONFIG_LISTEN_ADDRESS` and `SAVE_ON_EXIT` properties in the system-wide `agent.properties` file. Consequently, it is impossible to configure a system-wide alternative command channel address or change **Thermostat's** system-wide behavior of purging its collected data on exit. To work around this problem, set the desired property in the user-specific agent properties file, `USER_THERMOSTAT_HOME/etc/agent.properties`.

thermostat component

The number of threads, which **Thermostat** uses to retrieve data from database back end, is configured incorrectly. As a consequence, the graphical client is significantly slow, especially when a large number of data is collected to display. The problem is particularly noticeable in the web back end due to network latency.

mysql, mariadb components

In Red Hat Enterprise Linux 7, the `mariadb55-mariadb` and `mysql55-mysqld` services run under the `mysql` user account by default. When the `/var/lib/mysql/mysql.sock` UNIX socket file is created by a different user, the services have insufficient permissions to check whether a process is listening on the socket but the services have sufficient permissions to delete the socket file. Consequently, the `mariadb55-mariadb` and `mysql55-mysqld` services can delete the socket file while a process is still using it.

perl component

In Red Hat Enterprise Linux 7, the perl516 Software Collection tapset conflicts with the core system tapset. As a consequence, the `systemtap` utility does not work correctly for perl516. To work around this problem:

- Either copy the perl516 tapset to a file renamed by adding the Software Collection's prefix - to do so, use the following command:

```
cp /opt/rh/perl516/root/usr/share/systemtap/tapset/libperl5.16.3-64.stp /opt/rh/perl516/root/usr/share/systemtap/tapset/perl516-libperl5.16.3-64.stp
```

- Or uninstall the core system perl-devel package.

python27 component

In Red Hat Enterprise Linux 7, when the user tries to install the `python27-python-debuginfo` package, the `/usr/src/debug/Python-2.7.5/Modules/socketmodule.c` file conflicts with the corresponding file from the `python-debuginfo` package installed on the core system. Consequently, installation of the `python27-python-debuginfo` fails. To work around this problem, uninstall the `python-debuginfo` package and then install the `python27-python-debuginfo` package.

Other Notes

php54 component

Note that **Alternative PHP Cache (APC)** in Red Hat Software Collections 1.1 is provided for user data cache only. For opcode cache, **Zend OPcache** is provided.

nodejs component

The `nodejs-tobi-cookie` package has been renamed to `nodejs-cookie-jar`.

ruby component

Previously, in Red Hat Software Collections 1.0, the V8 JavaScript engine was part of the `ruby193` Software Collection. In Red Hat Software Collections 1.1, the `v8` packages have been replaced by the `v8314` Software Collection, which is installed as a dependency. In order to use `therubyracer`, it is necessary to enable the `v8314` Software Collection as well.

nodejs component

Previously, in Red Hat Software Collections 1.0, the V8 JavaScript engine was part of the `nodejs010` Software Collection. In Red Hat Software Collections 1.1, the `v8` packages have been replaced by the `v8314` Software Collection, which is installed as a dependency.

python component

When the user tries to install both the `python27-scldevel` and `python33-scldevel` packages, a transaction check error message is returned. This is an expected behavior because the user can install only one set of the macro files provided by the packages (`%scl_python`, `%scl_prefix_python`).

php component

When the user tries to install both the `php54-scldevel` and `php55-scldevel` packages, a transaction check error message is returned. This is an expected behavior because the user can install only one set of the macro files provided by the packages (`%scl_php`, `%scl_prefix_php`).

ruby component

When the user tries to install both the `ruby193-scldevel` and `ruby200-scldevel` packages, a transaction check error message is returned. This is an expected behavior because the user can install only one set of the macro files provided by the packages (`%scl_ruby`, `%scl_prefix_ruby`).

libyaml component

A newer version of the `libyaml` package is provided with Red Hat Software Collections 1.1 for use by Red Hat Software Collections.

nodejs component

When installing the `nodejs010` Software Collection, `nodejs010` installs `GCC` in the base Red Hat Enterprise Linux system as a dependency, unless the `gcc` packages are already installed.

mariadb component

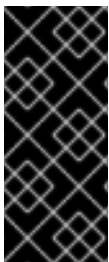
In Red Hat Software Collections 1.1 for Red Hat Enterprise Linux 7, the `mariadb55-mysqld.service` file has been renamed to `mariadb55-mariadb.service` and the `/var/log/mariadb55-mysqld.log` file has been moved to `/var/log/mariadb55-mariadb/mariadb55-mariadb.log`, for the sake of consistency with Red Hat Enterprise Linux 7.

CHAPTER 2. INSTALLATION AND USAGE

This chapter contains information related to the installation and usage of Red Hat Software Collections 1.1.

2.1. GETTING ACCESS TO RED HAT SOFTWARE COLLECTIONS

Depending on the subscription management service with which you registered your Red Hat Enterprise Linux system, you can either enable Red Hat Software Collections by using Red Hat Subscription Management, or by using RHN Classic. For detailed instructions on how to enable Red Hat Software Collections using RHN Classic or Red Hat Subscription Management, see the respective section below. For information on how to register your system with one of these subscription management services, see [Using and Configuring Red Hat Subscription Manager](#).



IMPORTANT

If you are running a version of Red Hat Enterprise Linux prior to 6.4, you will be unable to download Red Hat Software Collections through Red Hat Subscription Management. To obtain Red Hat Software Collections, either update to Red Hat Enterprise Linux 6.4, or register your system with RHN Classic. For more information, see <https://access.redhat.com/site/solutions/129003>.

2.1.1. Using Red Hat Subscription Management

If your system is registered with Red Hat Subscription Management, complete the following steps to attach the subscription that provides access to the repository for Red Hat Software Collections and enable the repository:

1. Display a list of all subscriptions that are available for your system and determine the pool ID of a subscription that provides Red Hat Software Collections. To do so, type the following at a shell prompt as **root**:

```
subscription-manager list --available
```

For each available subscription, this command displays its name, unique identifier, expiration date, and other details related to it. The pool ID is listed on a line beginning with **Pool Id**.

2. Attach the appropriate subscription to your system by running the following command as **root**:

```
subscription-manager subscribe --pool=pool_id
```

Replace *pool_id* with the pool ID you determined in the previous step. To verify the list of subscriptions your system has currently attached, run as **root**:

```
subscription-manager list --consumed
```

3. Display the list of available Yum list repositories to retrieve repository metadata and determine the exact name of the Red Hat Software Collections repositories. As **root**, type:

```
yum repolist all
```


The repository names depend on the specific version of Red Hat Enterprise Linux you are using and are in the following format:

```
rhel-variant-rhsc1-6-rpms
rhel-variant-rhsc1-6-debug-rpms
rhel-variant-rhsc1-6-source-rpms

rhel-server-rhsc1-6-eus-rpms
rhel-server-rhsc1-6-eus-source-rpms
rhel-server-rhsc1-6-eus-debug-rpms

rhel-variant-rhsc1-7-rpms
rhel-variant-rhsc1-7-debug-rpms
rhel-variant-rhsc1-7-source-rpms
```

Replace *variant* with the Red Hat Enterprise Linux system variant, that is, **server** or **workstation**. Note that Red Hat Software Collections is supported neither on the **Client** nor on the **ComputeNode** variant.

4. Enable the appropriate repository by running the following command as **root**:

```
yum-config-manager --enable repository
```

Once the subscription is attached to the system, you can install Red Hat Software Collections as described in [Section 2.2, “Installing Red Hat Software Collections”](#). For more information on how to register your system using Red Hat Subscription Management and associate it with subscriptions, see [Using and Configuring Red Hat Subscription Manager](#).

2.1.2. Using RHN Classic

If your system is registered with RHN Classic (applicable to Red Hat Enterprise Linux versions prior to 6.4), complete the following steps to subscribe to Red Hat Software Collections:

1. Display a list of all channels that are available to you and determine the exact name of the Red Hat Software Collections channel. To do so, type the following at a shell prompt as **root**:

```
rhn-channel --available-channels
```

The name of the channel depends on the specific version of Red Hat Enterprise Linux you are using and is in the following format, where *variant* is the Red Hat Enterprise Linux system variant (**server** or **workstation**):

```
rhel-x86_64-variant-6-rhsc1-1

rhel-x86_64-server-6.3.z-rhsc1-1
rhel-x86_64-server-6.4.z-rhsc1-1
rhel-x86_64-server-6.5.z-rhsc1-1

rhel-x86_64-variant-7-rhsc1-1
```

Note that Red Hat Enterprise Linux 7 channels are accessible only through Red Hat Satellite instances.

2. Subscribe the system to the Red Hat Software Collections channel by running the following command as **root**:

```
rhn-channel --add --channel=channel_name
```

Replace *channel_name* with the name you determined in the previous step.

3. Verify the list of channels you are subscribed to. As **root**, type:

```
rhn-channel --list
```

Once the system is subscribed, you can install Red Hat Software Collections as described in [Section 2.2, “Installing Red Hat Software Collections”](#). For more information on how to register your system with RHN Classic, see [Using and Configuring Red Hat Subscription Manager](#).

2.2. INSTALLING RED HAT SOFTWARE COLLECTIONS

Red Hat Software Collections is distributed as a collection of RPM packages that can be installed, updated, and uninstalled by using the standard package management tools included in Red Hat Enterprise Linux. Note that a valid subscription is required to install Red Hat Software Collections on your system. For detailed instructions on how to associate your system with an appropriate subscription and get access to Red Hat Software Collections, see [Section 2.1, “Getting Access to Red Hat Software Collections”](#).

IMPORTANT

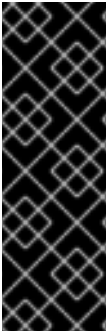
Some of the Red Hat Software Collections 1.1 packages require the **Optional** channel to be enabled in order to complete the full installation of these packages:

- The php54-php-ldap and php55-php-ldap packages require the libldap-client package from the Optional channel.
- The php54-php-recode and php55-php-recode packages require the recode package from the Optional channel.
- The perl516-perl-devel package requires the gdbm-devel package from the Optional channel.
- The mariadb55-mariadb-bench package requires the perl-GD package from the Optional channel.

In Red Hat Enterprise Linux 7, the following packages available only in the **Optional** channel are required:

- The **Node.js** Software Collection depends on the nodejs010-nodejs-devel package, which requires the c-ares-devel package from the Optional channel (applicable to minimal install).
- The httpd24-mod_ldap package requires the apr-util-ldap package from the Optional channel.
- The php54-php-aspell and php55-php-aspell packages require the aspell package from the Optional channel.
- The python27-python-debug package requires the tix package from the Optional channel.
- The thermostat1-thermostat package requires the apache-commons-beanutils, jansi, hawtjni, jansi-native, and objectweb-asm packages from the Optional channel (applicable to minimal install).
- The thermostat1-thermostat-webapp package requires the felix-framework package from the Optional channel (applicable to minimal install).
- The thermostat1-netty package requires the jzlib package from the Optional channel (applicable to minimal install).
- The apache-commons-logging package requires the xerces-j2 package from the Optional channel (applicable to minimal install).

For detailed instructions on how to subscribe your system to this channel, see the relevant [Knowledgebase article](#) on the [Customer Portal](#).



IMPORTANT

Use of Red Hat Software Collections 1.1 requires the removal of the pre-release versions. It is not possible to update to any Red Hat Software Collections 1.1 component from the Beta version. If you have previously installed any Red Hat Software Collections 1.1 component from the Beta version of Red Hat Software Collections, uninstall it from your system and install the new version as described in the [Section 2.3, “Uninstalling Red Hat Software Collections”](#) and [Section 2.2.1, “Installing Individual Software Collections”](#) sections.



IMPORTANT

The in-place upgrade from Red Hat Enterprise Linux 6 to Red Hat Enterprise Linux 7 is not supported by Red Hat Software Collections. As a consequence, the installed Software Collections might not work correctly after the upgrade. If you want to upgrade from Red Hat Enterprise Linux 6 to Red Hat Enterprise Linux 7, it is strongly recommended to remove all Red Hat Software Collections packages, perform the in-place upgrade, update the Red Hat Software Collections repository, and install the Software Collections packages again. It is advisable to back up all data before upgrading.

2.2.1. Installing Individual Software Collections

To install any of the Software Collections that are listed in [Table 1.1, “Red Hat Software Collections 1.1 Components”](#), install the corresponding meta package by typing the following at a shell prompt as **root**:

```
yum install software_collection...
```

Replace *software_collection* with a space-separated list of Software Collections you want to install. For example, to install php54 and mariadb55, type as **root**:

```
~]# yum install php54 mariadb55
```

This installs the main meta package for the selected Software Collection and a set of required packages as its dependencies. For information on how to install additional packages such as additional modules, see [Section 2.2.2, “Installing Optional Packages”](#).

2.2.2. Installing Optional Packages

Each component of Red Hat Software Collections is distributed with a number of optional packages that are not installed by default. To list all packages that are part of a certain Software Collection but are not installed on your system, type the following at a shell prompt:

```
yum list available software_collection-\*
```

To install any of these optional packages, run as **root**:

```
yum install package_name...
```

Replace *package_name* with a space-separated list of packages that you want to install. For example, to install the perl516-perl-CPAN and perl516-perl-Archive-Tar, type:

■

```
~]# yum install perl516-perl-CPAN perl516-perl-Archive-Tar
```

2.2.3. Installing Debugging Information

To install debugging information for any of the Red Hat Software Collections packages, make sure that the `yum-utils` package is installed and run the following command as `root`:

```
debuginfo-install package_name
```

For example, to install debugging information for the `ruby193-ruby` package, type:

```
~]# debuginfo-install ruby193-ruby
```

Note that in order to use this command, you need to have access to the repository with these packages. If your system is registered with Red Hat Subscription Management, enable the `rhel-variant-rhsc1-6-debug-rpms` repository as described in [Section 2.1.1, “Using Red Hat Subscription Management”](#). If your system is registered with RHN Classic, subscribe the system to the `rhel-x86_64-variant-6-rhsc1-1-debuginfo` channel as described in [Section 2.1.2, “Using RHN Classic”](#). For more information on how to get access to debuginfo packages, see <https://access.redhat.com/site/solutions/9907>.

2.3. UNINSTALLING RED HAT SOFTWARE COLLECTIONS

To uninstall any of the Software Collections components, type the following at a shell prompt as `root`:

```
yum remove software_collection*
```

Replace *software_collection* with the Software Collection component you want to uninstall.

Note that uninstallation of the packages provided by Red Hat Software Collections does not affect the Red Hat Enterprise Linux system versions of these tools.

2.4. REBUILDING RED HAT SOFTWARE COLLECTIONS

`<collection>-build` packages are not provided by default. If you wish to rebuild a collection and do not want or cannot use the `rpmbuild --define 'sc1 foo'` command, you first need to rebuild the metapackage, which provides the `<collection>-build` package.

Note that existing collections should not be rebuilt with different content. To add new packages into an existing collection, you need to create a new collection containing the new packages and make it dependent on packages from the original collection. The original collection has to be used without changes.

For detailed information on building Software Collections, refer to the [Red Hat Developer Toolset Software Collections Guide](#).

2.5. USING RED HAT SOFTWARE COLLECTIONS

2.5.1. Running an Executable from a Software Collection

To run an executable from a particular Software Collection, type the following command at a shell prompt:

```
scl enable software_collection... 'command...'
```

Or, alternatively, use the following command:

```
scl enable software_collection... -- command...
```

Replace *software_collection* with a space-separated list of Software Collections you want to use and *command* with the command you want to run. For example, to execute a Perl program stored in a file named `hello.pl` with the Perl interpreter from the `perl516` Software Collection, type:

```
~]$ scl enable perl516 'perl hello.pl'  
Hello, World!
```

You can execute any command using the `scl` utility, causing it to be run with the executables from a selected Software Collection in preference to their possible Red Hat Enterprise Linux system equivalents. For a complete list of Software Collections that are distributed with Red Hat Software Collections, see [Table 1.1, “Red Hat Software Collections 1.1 Components”](#).

2.5.2. Running a Shell Session with a Software Collection as Default

To start a new shell session with executables from a selected Software Collection in preference to their Red Hat Enterprise Linux equivalents, type the following at a shell prompt:

```
scl enable software_collection... bash
```

Replace *software_collection* with a space-separated list of Software Collections you want to use. For example, to start a new shell session with the `python27` and `postgresql92` Software Collections as default, type:

```
~]$ scl enable python27 postgresql92 bash
```

The list of Software Collections that are enabled in the current session is stored in the `$X_SCLS` environment variable, for instance:

```
~]$ echo $X_SCLS  
python27 postgresql92
```

For a complete list of Software Collections that are distributed with Red Hat Software Collections, see [Table 1.1, “Red Hat Software Collections 1.1 Components”](#).

2.5.3. Running a System Service from a Software Collection

Software Collections that include system services install corresponding init scripts in the `/etc/rc.d/init.d/` directory. To start such a service in the current session, type the following at a shell prompt as `root`:

```
service software_collection-service_name start
```

Replace *software_collection* with the name of the Software Collection and *service_name* with the name of the service you want to start. To configure this service to start automatically at boot time, run the following command as `root`:

```
chkconfig software_collection-service_name on
```

For example, to start the `postgresql` service from the `postgresql92` Software Collection and enable it in runlevels 2, 3, 4, and 5, type as `root`:

```
~]# service postgresql92-postgresql start
Starting postgresql92-postgresql service:           [ OK ]
~]# chkconfig postgresql92-postgresql on
```

For more information on how to manage system services in Red Hat Enterprise Linux 6, refer to the [Red Hat Enterprise Linux 6 Deployment Guide](#). For a complete list of Software Collections that are distributed with Red Hat Software Collections, see [Table 1.1, “Red Hat Software Collections 1.1 Components”](#).

2.6. ACCESSING A MANUAL PAGE FROM A SOFTWARE COLLECTION

Every Software Collection contains a general manual page that describes the content of this component. Each manual page has the same name as the component and it is located in the `/opt/rh` directory.

To read a manual page for a Software Collection, run the following command:

```
scl enable software_collection 'man software_collection'
```

Replace `software_collection` with the particular Red Hat Software Collections component. For example, to display the manual page for `mariadb55`, type:

```
~]$ scl enable mariadb55 "man mariadb55"
```

2.7. DEPLOYING APPLICATIONS THAT USE RED HAT SOFTWARE COLLECTIONS

In general, you can use one of the following two approaches to deploy an application that depends on a component from Red Hat Software Collections in production:

- Install all required Software Collections and packages manually and then deploy your application, or
- Create a new Software Collection for your application and specify all required Software Collections and other packages as dependencies.

For more information on how to manually install individual Red Hat Software Collections components, see [Section 2.2, “Installing Red Hat Software Collections”](#). For further details on how to use Red Hat Software Collections, see [Section 2.5, “Using Red Hat Software Collections”](#). For a detailed explanation of how to create a custom Software Collection or extend an existing one, read the [Red Hat Developer Toolset Software Collections Guide](#).

2.8. MIGRATING FROM MYSQL 5.1 TO MYSQL 5.5

2.8.1. Notable Differences Between MySQL 5.1 and MySQL 5.5

The following is a list of the most important changes between MySQL 5.1 and MySQL 5.5

- Starting with MySQL 5.5, the InnoDB storage engine (formerly known as InnoDB Plugin) is the default storage engine.
- InnoDB and some other plug-ins (for example, archive, blackhole and federated) were installable plug-ins in MySQL 5.1. Starting with MySQL 5.5, these plug-ins became compiled-in storage engines, that is, they cannot be installed or uninstalled by default.
- If you used InnoDB Plugin and it was loaded using the `plugin-load=innodb=ha_innodb_plugin.so` configuration option, you need to remove this configuration option as it does not work in MySQL 5.5.
- In MySQL 5.1, InnoDB Plugin included a configuration variable `innodb_file_io_threads`. However, this variable does not exist in MySQL 5.5; new variables, `innodb_read_io_threads` and `innodb_write_io_threads`, are used instead. To ensure proper functionality, either remove the former variable from the configuration file or replace it with the current variables.
- When upgrading from MySQL 5.1 to MySQL 5.5 using the in-place upgrading method, the `mysql.proxies_priv` table will not exist. To create the missing table, the `mysql_upgrade` utility has to be run as soon as the new daemon is started.
- MySQL 5.5 uses latin1 for the `stopword` file if the `character_set_server` variable is `ucs2`, `utf16` or `utf32`. Thus, if the table uses FULLTEXT indexes in these cases, users should repair the table using the `REPAIR TABLE table_name QUICK`.
- MySQL 5.1 used the `language` variable for specifying the directory which included the error message file. This option is now deprecated and has been replaced by the `lc_messages_dir` and `lc_messages` options. This also applies for configuration options. Also, error messages no longer contain mixed set of character sets and error messages are returned in the set following the `character_set_results` system variable instead. That is, some error messages can be different in MySQL 5.5.

Please note that the EXAMPLE plug-in is no longer distributed in Red Hat Software Collections packages.

For more information about MySQL 5.1 and MySQL 5.5, refer to the release notes available at <http://dev.mysql.com/doc/relnotes/mysql/5.1/en/> and <http://dev.mysql.com/doc/relnotes/mysql/5.5/en/>.



IMPORTANT

MariaDB is a community-developed drop-in replacement for MySQL. The differences between MySQL 5.1 and MySQL 5.5 are valid also for MySQL 5.1 and MariaDB 5.5.

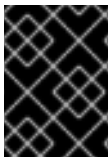
2.8.2. Upgrading from MySQL 5.1 to MySQL 5.5

Before migrating from MySQL 5.1 to MySQL 5.5, back up all your data, including any MySQL databases. Because the `mysql55` Software Collection does not conflict with the `mysql` packages from the core systems, it is possible to install the `mysql55` Software Collection together with the `mysql` packages. It is also possible to run both versions at the same time, however, the port number and the socket in the `my.cnf` files need to be changed to prevent these specific resources from conflicting.

Upgrading can be performed either by using the `mysqldump` and `mysqlimport` utilities or using in-place upgrade:

- In the first scenario, the whole dump of all databases from one database is generated, `mysql` is run with the dump file as an input, using `mysqlimport` or the `LOAD DATA INFILE SQL` command within the other database. At the same time, the appropriate daemons have to be running during both dumping and restoring. You can use the `--all-databases` option in the `mysqldump` call to include all databases in the dump. The `--routines`, `--triggers` and `--events` options can also be used if needed.
- During the in-place upgrade, the data files are copied from one database directory to another database directory. The daemons should not be running at the time of copying. Set the appropriate permissions and SELinux context for copied files.

After upgrading, start the server and run the `mysql_upgrade` command. Running `mysql_upgrade` is necessary to check and repair internal tables.



IMPORTANT

All scripts that work with a server from Software Collection, especially the `mysql_upgrade` script, should be run inside the `scl enable` environment.

In case the `root` user has a non-empty password defined (it should have it defined), it is necessary to call the `mysql_upgrade` utility with the `-p` option and specify the password.

The dump and restore upgrade method is recommended. The in-place upgrade method is usually faster, however, there are certain risks and known problems. For more information, refer to the [MySQL 5.5 Release Notes](#).

In addition, once the upgrade is complete, consider changing the appropriate settings in the `my.cnf` file to reflect the environment.

Example 2.1. Dump and Restore Upgrade

```
~]# service mysqld start
Starting mysqld: [ OK ]
~]# mysqldump --all-databases --routines --events > dump.sql
~]# service mysqld stop
Stopping mysqld: [ OK ]
~]# service mysql55-mysqld start
Starting mysql55-mysqld: [ OK ]
~]# scl enable mysql55 'mysql' < dump.sql
~]# scl enable mysql55 'mysql_upgrade -u root -p'
Enter password:
Looking for 'mysql' as: mysql
Looking for 'mysqlcheck' as: mysqlcheck
Running 'mysqlcheck with default connection arguments
Running 'mysqlcheck with default connection arguments
a.t1 OK
mysql.columns_priv OK
<skipped tables list>
mysql.user OK
Running 'mysql_fix_privilege_tables'...
OK
```

Example 2.2. In-place Upgrade

```

~]# service mysqld stop
Stopping mysqld: [ OK ]
~]# service mysql55-mysqld stop
Stopping mysql55-mysqld: [ OK ]
~]# rm -rf /opt/rh/mysql55/root/var/lib/mysql/
~]# cp -r /var/lib/mysql/ /opt/rh/mysql55/root/var/lib/
~]# chown -R mysql:mysql /opt/rh/mysql55/root/var/lib/mysql/
~]# restorecon -R /opt/rh/mysql55/root/var/lib/mysql/
~]# service mysql55-mysqld start
Starting mysql55-mysqld: [ OK ]
~]# scl enable mysql55 'mysql_upgrade -u root -p'
Enter password:
Looking for 'mysql' as: mysql
Looking for 'mysqlcheck' as: mysqlcheck
Running 'mysqlcheck with default connection arguments
Running 'mysqlcheck with default connection arguments
a.t1 OK
mysql.columns_priv OK
<skipped tables list>
mysql.user OK
Running 'mysql_fix_privilege_tables'...
OK

```

For more information about the upgrading process, refer to [MySQL 5.5 Reference Manual](#).

IMPORTANT

MariaDB is a community-developed drop-in replacement for **MySQL**. The steps for upgrading from **MySQL 5.1** to **MySQL 5.5** are valid also for upgrading from **MySQL 5.1** to **MariaDB 5.5**, with the exception of the following differences:

- The `mariadb55` component name should be used instead of the `mysql55`, so replace all occurrences of `mysql55` with `mariadb55`.
- The `systemd` unit name for **MariaDB 5.5** is `mariadb55-mariadb` in Red Hat Enterprise Linux 7, while the `SysV` unit script for **MariaDB 5.5** is called `mariadb55-mysqld` in Red Hat Enterprise Linux 6.

2.8.3. Using the `mysql55-mysql-devel` Package

Red Hat Software Collections contains the server part of MySQL 5.5 database. Red Hat Enterprise Linux 6 provides version 5.1 of this database (client library and server daemon). A protocol which is used between the client library and the daemon is stable across database versions, so using, for example, the MySQL 5.1 client library with the MySQL 5.5 daemon works as expected.

2.8.3.1. Using Database Connectors for Dynamic Languages



IMPORTANT

When a MariaDB or MySQL database contains old users created using old authentication schema, PHP using the `mysqlnd` driver will not be able to connect to the database. This is because the `old_password` setting in the `/etc/my.cnf` file is turned off by default on Red Hat Enterprise Linux 6 while it is enabled on Red Hat Enterprise Linux 5. To work around this problem, set `old_password` to 0, restart the MariaDB or MySQL service and set a new password for each user.

2.8.3.2. Building Applications for MySQL 5.5 from Red Hat Software Collections

MySQL 5.5 from Red Hat Software Collections does not include database connectors; client libraries packaged in the MySQL 5.5 Red Hat Software Collections database packages are not supposed to be used as they are included only for purposes of server utilities and the daemon. Users are instead expected to use the system libraries and database connectors provided with the core system.

It means that users who would like to link their application against the MySQL client library should compile it and link it to the core Red Hat Enterprise Linux 6 environment, not to the MySQL 5.5 Red Hat Software Collections environment.

The only exception to this are server-side plug-ins, which are expected to be built under the MySQL 5.5 Red Hat Software Collections environment. This means the build process should be run inside the `sc1 enable mysql55 '...'` call.

2.9. MIGRATING FROM POSTGRESQL 8.4 TO POSTGRESQL 9.2

Red Hat Software Collections 1.1 is distributed with PostgreSQL 9.2, which can be safely installed on the same machine in parallel with PostgreSQL 8.4 from Red Hat Enterprise Linux 6. It is also possible to run both versions of PostgreSQL on one machine at the same time, but you need to use different ports or IP addresses and adjust SELinux policy.

2.9.1. Notable Differences Between PostgreSQL 8.4 and PostgreSQL 9.2

The following is a list of the most important changes between PostgreSQL 8.4 and PostgreSQL 9.2:

- The following server configuration parameters have been removed and are no longer supported: `add_missing_from`, `regex_flavor`, `silent_mode`, `wal_sender_delay`, and `custom_variable_classes`. Do not use any of these parameters in the new configuration file.
- The `unix_socket_directory` parameter has been renamed to `unix_socket_directories` and can now be used to specify more than one UNIX socket to listen on. To do so, provide a list of comma-separated directories as the value of this option. The default value remains unchanged and is `/tmp`.
- New configuration parameters `ssl_ca_file`, `ssl_cert_file`, `ssl_cr1_file`, and `ssl_key_file` have been added. These configuration parameters can be used to specify the locations of server-side SSL files that were previously hard-coded as relative paths to the `root.crt`, `server.crt`, `root.cr1`, and `server.key` files in the data directory.

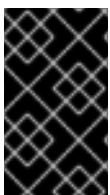
Note that the PostgreSQL server no longer reads the `root.crt` and `root.cr1` files by default. To load these files, change the corresponding parameters to non-default values.

- The `=>` operator has been removed and users are now advised to use the `hstore(text, text)` function.
- The default value of the `standard_conforming_strings` configuration parameter is now `on`. This configuration parameter controls if ordinary string literals (strings enclosed in single quotes) treat backslashes literally as specified in the SQL standard.
- A new configuration parameter, `backslash_quote`, has been added. This configuration parameter can be used to control whether a single quotation mark can be represented by `\'` in string literals. The default value is `safe_encoding`, which permits the use of `\'` only when the client encoding does not allow ASCII backslashes in multi-byte characters. As a consequence, `\'` can now be interpreted differently only in specific cases and only in string literals that do not conform to standards, including escape string syntax, `E' value '`.
- PostgreSQL 9.0 introduced access privileges for large objects. Consequently, a new configuration parameter, `lo_compat_privileges`, has been added to allow you to disable security checks related to the large objects affected by this change. To disable these security checks, change the value of this configuration parameter to `on`. The default value is `off`.

For a detailed list of known compatibility issues with earlier versions, see the official notes for [PostgreSQL 9.0](#), [PostgreSQL 9.1](#), and [PostgreSQL 9.2](#). For an in-depth list of changes in behavior, see the upstream [Release Notes](#).

2.9.2. Upgrading from PostgreSQL 8.4 to PostgreSQL 9.2

To migrate your data from PostgreSQL 8.4 that is distributed with Red Hat Enterprise Linux 6 to PostgreSQL 9.2 that is included in Red Hat Software Collections 1.1, you can either perform an in-place upgrade (recommended), or dump the database data into a text file with SQL commands and import it in the new database. Note that the second method is usually significantly slower and may require manual fixes; see the [official documentation](#) for more information about this upgrade method. If you need to migrate PostgreSQL databases to Red Hat Enterprise Linux 7, see <https://access.redhat.com/site/articles/541873> and <https://access.redhat.com/site/articles/694413>.



IMPORTANT

Before migrating your data from PostgreSQL 8.4 to PostgreSQL 9.2, make sure that you back up all your data, including the PostgreSQL database files that are by default located in the `/var/lib/pgsql/data/` directory.

Procedure 2.1. Performing In-place Upgrade

To perform an in-place upgrade of your PostgreSQL server, complete the following steps:

1. Stop the old PostgreSQL server to ensure that the data is not in an inconsistent state. To do so, type the following at a shell prompt as `root`:

```
service postgresql stop
```

To verify that the server is not running, type:

```
service postgresql status
```

2. Verify that the new data directory located in `/opt/rh/postgresql92/root/var/lib/pgsql/data/` does not exist:

```
file /opt/rh/postgresql92/root/var/lib/pgsql/data/
```

If you are running a fresh installation of PostgreSQL 9.2, this directory should not be present in your system. If it is, back it up by running the following command as root:

```
mv /opt/rh/postgresql92/root/var/lib/pgsql/data{, -scl-backup}
```

3. Copy the old database data to the new location by typing the following at a shell prompt as root:

```
cp -ra /var/lib/pgsql/data/ /opt/rh/postgresql92/root/var/lib/pgsql/
```

4. Open the `/opt/rh/postgresql92/root/var/lib/pgsql/data/pg_hba.conf` configuration file and verify that the `postgres` user is allowed to connect to the PostgreSQL server from `localhost` without a password. If not, you can edit this file and temporarily set the authentication method for the `postgres` user to `trust` or `ident`. For a detailed description of the `pg_hba.conf` file and a complete list of available configuration options, see the [official documentation](#).

5. Upgrade the database data for the new server by running the following command as root:

```
service postgresql92-postgresql upgrade
```

It is recommended that you read the resulting `/opt/rh/postgresql92/root/var/lib/pgsql/pgupgrade.log` log file to see if there were any problems with the upgrade.

6. Start the new server as root:

```
service postgresql92-postgresql start
```

It is also advised that you run the `analyze_new_cluster.sh` script as follows:

```
su - postgres -c 'scl enable postgresql92 ~/analyze_new_cluster.sh'
```

7. Optionally, you can configure the PostgreSQL 9.2 server to start automatically at boot time. To disable the old PostgreSQL 8.4 server, run the following command as root:

```
chkconfig postgresql off
```

To enable the PostgreSQL 9.2 server, type as root:

```
chkconfig postgresql92-postgresql on
```

Procedure 2.2. Performing a Dump and Restore Upgrade

To perform a dump and restore upgrade of your PostgreSQL server, complete the following steps:

1. Ensure that the old PostgreSQL server is running by typing the following at a shell prompt as root:

```
service postgresql start
```

2. Dump all data in the PostgreSQL database into a script file. As **root**, type:

```
su - postgres -c 'pg_dumpall > ~/pgdump_file.sql'
```

3. Stop the old server by running the following command as **root**:

```
service postgresql stop
```

4. Initialize the data directory for the new server as **root**:

```
service postgresql92-postgresql initdb
```

5. Start the new server as **root**:

```
service postgresql92-postgresql start
```

6. Import data from the previously created SQL file:

```
su - postgres -c 'scl enable postgresql92 "psql -f ~/pgdump_file.sql postgres"'
```

7. Optionally, you can configure the PostgreSQL 9.2 server to start automatically at boot time. To disable the old PostgreSQL 8.4 server, run the following command as **root**:

```
chkconfig postgresql off
```

To enable the PostgreSQL 9.2 server, type as **root**:

```
chkconfig postgresql92-postgresql on
```

8. If your configuration differs from the default one, make sure to update configuration files, especially the `/opt/rh/postgresql92/root/var/lib/pgsql/data/pg_hba.conf` configuration file. Otherwise only the `postgres` user will be allowed to access the database.

2.10. REPORTING BUGS IN RED HAT SOFTWARE COLLECTIONS

Bug reports are essential in making Red Hat Software Collections reliable. Reporting a bug can also help you bring a solution to your problem. Nevertheless, the main function of a bug report is to help the community by improving the next version of Red Hat Software Collections.

To report a bug in Red Hat Software Collections, complete the following steps:

1. Navigate to [Bugzilla](#), the Red Hat bug-tracking system. Note that you need to be logged in, or you will be prompted to do so.
2. On the top bar, click on **New** and on the next page, choose **Red Hat**. This brings you to a page with a list of Red Hat Products.
3. Click on **Red Hat Software Collections**, which brings you to this [form](#).

4. Fill in the relevant fields.
 - **Component** (a required field): a package name without a prefix.
 - **Version** (a required field): **RHSCL-1-*software_collection***
 - **Summary** (a required field).
 - **Description**: this field provides a simple template which can help you describe the bug. It is important to provide as much information about the problem as possible.
5. Optionally, you can add an attachment, which can be a screenshot of the problem, a patch, or similar.
6. Once you have completed your bug report, press the **Submit Bug** button. Your report will be assigned a Bugzilla number so you can get back to it later.

If you wish to use the **Automatic Bug Reporting Tool** (ABRT) utility, see the [Red Hat Enterprise Linux 6 Deployment Guide](#) or [Red Hat Enterprise Linux 7 System Administrator's Guide](#). For detailed information on bug reporting, refer to [Bugzilla's help](#).

CHAPTER 3. ADDITIONAL RESOURCES

For more information about Red Hat Software Collections 1.1 and Red Hat Enterprise Linux, refer to the resources listed below.

3.1. RED HAT ENTERPRISE LINUX DEVELOPER PROGRAM GROUP

Users of Red Hat Software Collections can access the Red Hat Enterprise Linux Developer Program Group in the Red Hat Customer Portal to get developer related information for the development tools available for Red Hat Enterprise Linux. In addition, users can find developer related papers and videos on topics that are of interest to developers, for example RPM building, threaded programming, performance tuning, debugging, and so on.

To visit the Red Hat Enterprise Linux Developer Program Group, log in to the [Customer Portal](#), click **Products** at the top of the page, choose **Services**, and then **Red Hat Enterprise Linux Developer Program** from the list.

3.2. RED HAT PRODUCT DOCUMENTATION

The following documents are directly or indirectly relevant to this book:

- [Red Hat Developer Toolset Software Collections Guide](#) – The *Software Collections Guide* for Red Hat Developer Toolset explains the concept of Software Collections, documents the `sc1` utility, and provides a detailed explanation of how to create a custom Software Collection or extend an existing one.
- [Using and Configuring Red Hat Subscription Manager](#) – The *Using and Configuring Red Hat Subscription Manager* book provides detailed information on how to register Red Hat Enterprise Linux systems, manage subscriptions, and view notifications for the registered systems.
- [Red Hat Enterprise Linux 6 Developer Guide](#) – The *Developer Guide* for Red Hat Enterprise Linux 6 provides more information for developers on the Red Hat Enterprise Linux platform.
- [Red Hat Enterprise Linux 7 Developer Guide](#) – The *Developer Guide* for Red Hat Enterprise Linux 7 provides an introduction to application development tools in Red Hat Enterprise Linux 7.
- [Red Hat Enterprise Linux 6 Deployment Guide](#) – The *Deployment Guide* for Red Hat Enterprise Linux 6 provides relevant information regarding the deployment, configuration, and administration of this system.
- [Red Hat Enterprise Linux 7 System Administrator's Guide](#) – The *System Administrator's Guide* for Red Hat Enterprise Linux 7 provides information on deployment, configuration, and administration of this system.

APPENDIX A. REVISION HISTORY

Revision 1.1-25	Thu 09 Jun 2016	Lenka Špačková
Fixed a path in the MySQL in-place upgrade example.		
Revision 1.1-14	Tue Jun 10 2014	Lenka Špačková
Updated Red Hat Software Collections 1.1 Release Notes with information on support for Red Hat Enterprise Linux 7.		
Revision 1.1-12	Wed Jun 04 2014	Lenka Špačková
Release of Red Hat Software Collections 1.1 Release Notes.		