



Revolution R Enterprise™ 7

Installation Guide for Linux

Systems

The correct bibliographic citation for this manual is as follows: Revolution Analytics, Inc. 2014. *Revolution R Enterprise 7 Installation Guide for Linux Systems*. Revolution Analytics, Inc., Mountain View, CA.

Revolution R Enterprise 7 Installation Guide for Linux Systems

Copyright © 2014 Revolution Analytics, Inc. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Revolution Analytics.

U.S. Government Restricted Rights Notice: Use, duplication, or disclosure of this software and related documentation by the Government is subject to restrictions as set forth in subdivision (c) (1) (ii) of The Rights in Technical Data and Computer Software clause at 52.227-7013.

Revolution R, Revolution R Enterprise, RPE, RevoScaleR, RevoDeployR, RevoTreeView, and Revolution Analytics are trademarks of Revolution Analytics.

Other product names mentioned herein are used for identification purposes only and may be trademarks of their respective owners.

Revolution Analytics
2570 West El Camino Real
Suite 222
Mountain View, CA 94040
U.S.A.

Revised on January 15, 2015

We want our documentation to be useful, and we want it to address your needs. If you have comments on this or any Revolution document, write to doc@revolutionanalytics.com.

Table of Contents

1	Quick Overview	5
2	Prerequisites and Dependencies	6
2.1	System Requirements	6
2.2	Package Dependencies.....	6
3	Managing Your Revolution R Enterprise Installation	9
3.1	File Ownership.....	9
3.2	Non-Root Installs.....	9
3.3	Unattended Installs	11
3.4	File Permissions.....	11
3.5	Installing to a Read-Only File System	11
4	Setting Up a Package Repository	12
4.1	Creating a Repository Mirror	12
4.2	Creating a Custom Repository.....	13
4.3	Configuring R to Use Your Local Repository.....	15
5	Removing Revolution R	15
6	Managing Multiple R Installations	16
6.1	Side-by-Side Revolution R Installation	16
6.2	Using Revolution R with Other R Installations	16
7	Using Revolution R Enterprise with Platform LSF	17
7.1	General Concepts	17
7.2	Downloading and Installing IBM Platform MPI.....	19
7.3	Downloading and Installing IBM Platform LSF	20
7.4	Optimizing for Speed (optional).....	21
7.5	Configuring the LSF Server Hosts	22
7.6	Configuring the Shared Data Directory	23
7.7	User Setup	24
7.8	Verifying the Installation	25

Table of Contents

8	Using Revolution R with a Third-Party IDE	27
8.1	Running Revolution R with RStudio	27
8.2	Running Revolution R with StatET.....	29

1 Quick Overview

Revolution R Enterprise has an automated install script, so for most users, installation consists of the following simple steps:

1. Log in as root or a user with sudo privileges. If the latter, precede commands requiring root privileges with sudo. (If you do not have root access or sudo privileges, you can install as a non-root user. See Section 3.2 for details.)
2. Download the Revolution R Enterprise distribution appropriate for your Linux distribution:

Revo-Ent-7.3.0-RHEL5.tar.gz

Revo-Ent-7.3.0-RHEL6.tar.gz

Revo-Ent-7.3.0-SLES11.tar.gz

3. Unpack the distribution using the tar command as follows:

```
tar -zxvf Revo-Ent-7.3.0-RHEL5.tar.gz
or
tar -zxvf Revo-Ent-7.3.0-RHEL6.tar.gz
or
tar -zxvf Revo-Ent-7.3.0-SLES11.tar.gz
```

4. Change directory to the RevolutionR_7.3.0 directory created in Step 2:

```
cd RevolutionR_7.3.0
```

5. Run the automated install script, install.py, as follows (requires root privileges):

```
./install.py
```

6. While the install script is running, answer the on-screen prompts as follows:
 - a. If you are not running as root, you are first asked if you want to continue with the non-root install. Either press y and then Enter to continue the non-root install, or press n and then Enter to stop the install. Log in as root and re-run the install script to proceed with the root install.
 - b. When asked if you want to install Revolution R Enterprise, press y and then Enter.
 - c. Press Enter to view the license agreement; press q to exit the pager.
 - d. When asked if you agree to the Revolution license, press y and then Enter.
 - e. When asked if you'd like to install the development tools for R, press y and then Enter if you plan to install additional R packages.
 - f. When asked if you'd like to load the rpart and lattice packages by default, press y and then Enter.

- g. When asked what prefix you'd like to use for the Revolution R Enterprise components, either type a path beginning with / or press Enter to accept the default prefix of /usr.
- h. If no R is found in the bin directory corresponding to your specified prefix, you are asked if you want to create a symbolic link between R and the Revolution R Enterprise script Revo64. Answer y to create the link—this is useful in allowing certain R utilities to function normally. (If R already exists on your system, you do not see this prompt.)

On Red Hat Enterprise Linux systems with Hadoop installed, the install.py script also tries to configure Revolution R Enterprise for use with Hadoop. On such systems, the install.py script runs another Python script that queries the Hadoop environment for certain environment variables and searches the Hadoop installation for certain files, writing a set of Hadoop environment variables required by Revolution R Enterprise to a file in the RRE installation directories. For complete details on Hadoop configuration, including troubleshooting when the automated configuration is incomplete or inaccurate, see the [Revolution R Enterprise 7Hadoop Configuration Guide](#).

If you receive messages about uninstalled dependencies, see Chapter 2 on Prerequisites and Dependencies.

2 Prerequisites and Dependencies

This chapter describes the minimum and recommended system requirements for Revolution R Enterprise, and also lists all known package dependencies.

2.1 System Requirements

Revolution R Enterprise for Linux has the following system requirements:

Processor: 64-bit processor with x86-compatible architecture (variously known as AMD64, Intel64, x86-64, IA-32e, EM64T, or x64 chips). Itanium-architecture chips (also known as IA-64) are not supported. Multiple-core chips are recommended.

Operating System: Red Hat Enterprise Linux 5.4, 5.5, 5.6, 5.7, 5.8, 5.9, 5.10, 5.11, 6.0, 6.1, 6.2, 6.3, 6.4, or 6.5, or SUSE Linux Enterprise Server 11. Only 64-bit operating systems are supported.

Memory: A minimum of 1GB of RAM is required; 4GB or more are recommended.

Disk Space: A minimum of 500MB of disk space is required.

2.2 Package Dependencies

Revolution R Enterprise, like most Linux applications, depends upon a number of Linux packages. A few of these, listed in Table 1 for RHEL systems and Table 2 for SLES 11 systems, are explicitly required by Revolution R Enterprise. The remainder are in turn required by these dependencies. These are automatically installed while the automated

script is running. These are listed in Table 3 for RHEL systems and Table 4 for SLES 11 systems. Finally, if you choose to install the development tools (required to build R packages from source), a few more packages are required. These are also installed automatically, and are listed in Table 5 for RHEL systems and Table 6 for SLES 11 systems.

Table 1. Packages Explicitly Required by Revolution R Enterprise on RHEL Systems

cairo-devel	python-devel
libgfortran	tk-devel
libcups	ed
pango	libjpeg*-devel
ghostscript-fonts	

Table 2 Packages Explicitly Required by Revolution R Enterprise on SLES 11 Systems

cairo-devel	python-devel
libgfortran43	tk-devel
ghostscript-fonts-std	ed
libjpeg*-devel	

Table 2. Secondary Dependencies Installed for Revolution R Enterprise on RHEL Systems

cairo	pkgconfig
libpng-devel	libX11-devel
pixman-devel	freetype-devel
fontconfig-devel	freetype
glibc	info
libX11	libgcc
libpng	libsepol
mesa-libGL	python
tk	bash
python-libs	gcc
glibc-devel	glibc-headers
gmp	kernel-headers
libXft-devel	libXft
libXdmp-devel	libXrender
libgomp	sed
mesa-libGL-devel	gib2
tcl-devel	fontconfig
libthai	xorg-x11-font-utils

8 Prerequisites and Dependencies

Table 3. Secondary Dependencies Installed for Revolution R Enterprise on SLES 11 Systems

cpp	freetype
glibc	glibc-common
libX11	libgcc
libpng	libsepol
mesa-libGL	python
tcl	fontconfig-devel
freetype-devel	gcc
glibc-devel	glibc-headers
gmp	kernel-headers
libX11-devel	libXau-devel
libXdmcp-devel	libXrender-devel
libgomp	libpng-devel
mesa-libGL-devel	python-libs
tcl-devel	

Table 4. Development Packages Required by R Development Tools on RHEL Systems

make	gcc-objc
gcc-c++	readline-devel
libtiff-devel	ncurses-devel
gcc-gfortran	libSM-devel
libcxx-devel	libXmu-devel
bzip2-devel	

Table 5 Development Packages Required by R Development Tools on SLES 11 Systems

make	gcc-objc
gcc-c++	readline-devel
libtiff-devel	ncurses-devel
gcc-fortran	xorg-x11-libSM-devel
xorg-x11-libXmu-devel	

3 Managing Your Revolution R Enterprise Installation

In this chapter, we discuss file management for your Revolution R Enterprise installation, including file ownership, file permissions, and so on.

3.1 File Ownership

If you use the Quick Install steps of Chapter 1, the installed files are all owned by root. For single-user workstations where the user has either sudo privileges or access to the root password, this is normally fine. In enterprise environments, however, it is often desirable to have third-party applications such as Revolution R Enterprise installed into an account owned by a non-root user; this can make maintenance easier and reduce security concerns. In such an environment, you may wish to create an “RUser” account, and change ownership of the files to that user. You can do that as follows:

1. Install Revolution R Enterprise as root, as usual.
2. Create the “RUser” account if it does not already exist. Assign this user to a suitable group, if desired.
3. Use the chown command to change ownership of the files (in the example below, we assume RUser has been made a member of the dev group; this command requires root privileges):

```
chown -R RUser:dev /usr/lib64/Revo-7.3
```

From Revolution R Enterprise version 6.1, you can also run the installer as a non-root user, even without sudo privileges. For details, see Section 3.2, Non-Root Installs.

3.2 Non-Root Installs

From Revolution R Enterprise version 6.1, you can run the installer as a non-root user without sudo privileges. You can run either a complete install or simply extract the files to a directory for subsequent installation using your own install scripts. For a complete install to succeed, however, the following conditions must be met:

- You must have all the prerequisite packages installed on your computer.
- You must have write permission to the specified base installation directory.

The installer will check that the prerequisite packages are installed and inform you of any that are missing. If you do not have sudo privileges, you will probably need to consult your system administrator in order to install them.

If all the prerequisites are installed, you complete the installation as follows:

1. Download the Revolution R Enterprise distribution, Revo-Ent-7.3.0-RHEL5.tar.gz, Revo-Ent-7.3.0-RHEL6.tar.gz, or Revo-Ent-7.3.0-SLES11.tar.gz
2. Unpack the distribution using the tar command as follows:

```
tar xzf Revo-Ent-7.3.0-RHEL5.tar.gz
```

or

```
tar xzf Revo-Ent-7.3.0-RHEL6.tar.gz
```

or

```
tar xzf Revo-Ent-7.3.0-SLES11.tar.gz
```

3. Change directory to the RevolutionR 7.3.0 directory created in Step 2:

```
cd RevolutionR_7.3.0
```

4. Run the automated install script, `install.py`, as follows:

```
./install.py
```

5. While the install script is running, answer the on-screen prompts as follows:
 - a. When asked if you want to install Revolution R Enterprise, press `y` and then Enter.
 - b. Press Enter to view the license agreement; press `q` to exit the pager.
 - c. When asked if you agree to the Revolution license, press `y` and then Enter.
 - d. When asked if you'd like to install the development tools for R, press `y` and then Enter if you plan to install additional R packages.
 - e. When asked if you'd like to load the `rpart` and `lattice` packages by default, press `y` and then Enter.
 - f. When asked what base directory you'd like to use for the Revolution R Enterprise components, either type a path beginning with `/` or press Enter to accept the default (`$HOME/local`). You must have write permission to this directory.
 - g. If no R is found in the `bin` directory corresponding to your specified base directory, you are asked if you want to create a symbolic link between R and the Revolution R Enterprise script `Revo64`. Answer `y` to create the link—this is useful in allowing certain R utilities to function normally. (If you have already installed R on this system, you do not see this prompt.)
6. (optional) Add the `bin` directory created under your base installation directory to your path to ensure that calls to `Revo64` will find the installed script.

The R installation is owned by the user who completed the installation; subsequent updates and uninstalls must be performed by the same user. In particular, multiple users can perform non-root installations—as long as the installations specify separate base installation directories, they will be completely independent. Non-root installations can be performed on systems that have had Revolution R installed as root, and vice versa.

If you simply want to extract the files to a directory for later installation, use the `-e` and `-p` flags in an unattended install, as described in Section 3.3.

3.3 Unattended Installs

You can bypass the interactive install steps by calling `install.py` with the `--no-ask` or `-n` flag. Additional flags can be used to specify which of the usual install options you want, as follows:

- `-c` or `--check-depends` check system dependencies. This can be useful in locked down environments when yum or zipper cannot be used to install dependencies automatically.
- `-d` or `--devel` install the development tools
- `-m` or `--no-md5` disable md5 checks on the rpm packages
- `-p` PATH or
`--prefix=PATH` base path of the desired installation directory
- `-s` or `--shortcut` disable links from R to Revo64 and Rscript to Revoscript
- `-a` or `--additionalpackages` load lattice and rpart on startup
- `-e` or `--extract-local` extract file tree to a local directory

For example, to extract the Revolution R Enterprise files to the local directory `/home/username/foo`, you could use `install.py` as follows:

```
./install.py -n -p /home/username/foo -e
```

For a standard unattended install, use the `-n`, `-d`, and `-a` flags, as follows:

```
./install.py -n -d -a
```

3.4 File Permissions

Normally, ordinary Revolution R Enterprise files are installed with read/write permission for owner and read-only permission for group and world. Directories are installed with execute permission as well, to permit them to be traversed. You can modify these permissions using the `chmod` command. (For files owned by root, this command requires root privileges.)

3.5 Installing to a Read-Only File System

In enterprise environments, it is common for enterprise utilities to be mounted as read-only file systems, so that ordinary operations cannot corrupt the tools. Obviously, new applications can be added to such systems only by first unmounting them, then re-mounting them for read/write, installing the software, and then re-mounting as read-only. This must be done by a system administrator.

4 Setting Up a Package Repository

One of the strengths of the R language is the thousands of third-party packages that have been made publicly available via CRAN, the Comprehensive R Archive Network. R includes a number of functions that make it easy to download and install these packages. However, in many enterprise environments, access to the Internet is limited or non-existent. In such environments, it is useful to create a local package repository that users can access from within the corporate firewall.

Your local repository may contain source packages, binary packages, or both. If some or all of your users will be working on Windows systems, you should include Windows binaries in your repository. Windows binaries are R-version-specific; if you are running R 3.0.3, you need Windows binaries built under R 3.0. These versioned binaries are available from CRAN and other public repositories. If some or all of your users will be working on Linux systems, you must include source packages in your repository. Because Revolution R is never the latest version of R, and because CRAN does not maintain source packages in a way that corresponds to R releases, your local repository should draw its source packages not from CRAN, but from the versioned source repository maintained by Revolution Analytics at <http://www.revolutionanalytics.com/repositories>.

There are two ways to create the package repository: either mirror an existing repository, or create a new repository and populate it with just those packages you want to be available to your users. However, both the CRAN repository and the Revolution Analytics repository are sizable, and if disk space is a concern you may want to restrict yourself to only a subset of the available packages. Maintaining a local mirror of an existing repository is typically easier and less error-prone, but managing your own repository gives you complete control over what is made available to your users.

4.1 Creating a Repository Mirror

Maintaining a repository mirror is simplest if you can use the `rsync` tool; this is available on all Linux systems and is available for Windows users as part of the Rtools collection (<http://cran.r-project.org/bin/windows/Rtools/>).

To mirror an existing repository:

1. Choose a location for your repository somewhere on your local network. If you have a corporate intranet, this is usually a good choice, provided URLs have the prefix `http://` and not `https://`. Any file system that is mounted for all users can be used; file-based URLs of the form `file://` are supported by the R functions. In this example, we suppose the file system `/local` is mounted on all systems and we will create our repository in the directory `/local/repos`.
2. To create a full mirror of the Revolution Analytics versioned source repository, use `rsync` as follows:

```
rsync -rtlzv --delete packages.revolutionanalytics.com::REVO-PKG  
/local/repos
```

(The `--delete` flag tells `rsync` to delete local files that are no longer on the Revolution repository.)

3. To create a full mirror of the CRAN repository, use `rsync` as follows:

```
rsync -rtlzv --delete cran.r-project.org::CRAN /dir/on/local/disc
```

(The `--delete` flag tells `rsync` to delete local files that are no longer on the CRAN repository.)

4. If you have created a CRAN mirror, be sure that your web server is configured to allow redirection. If you use the Apache web server, you can enable this feature by loading the `mod_rewrite` module.

4.2 Creating a Custom Repository

As mentioned above, a custom repository gives you complete control over which packages are available to your users. Here, too, you have two basic choices in terms of populating your repository: you can either `rsync` specific directories from an existing repository, or you can combine your own locally developed packages with packages from other sources. The latter option gives you the greatest control, but typically means you need to manage the contents using home-grown tools.

4.2.1 Creating a Partial Mirror

To create a partial mirror for source packages:

1. Choose a location for your repository somewhere on your local network. If you have a corporate intranet, this is usually a good choice, provided URLs have the prefix `http://` and not `https://`. Any file system that is mounted for all users can be used; file-based URLs of the form `file://` are supported by the R functions. In this example, we suppose the file system `/local` is mounted on all systems and we will create our repository in the directory `/local/repos`.
2. Create a directory `src` as follows:

```
cd /local/repos
mkdir -p src
```

If you need to support more than one version of Revolution R Enterprise in your organization, you can create version directories in the top-level `repos` directory, and then the `src/contrib` directories as subdirectories of that. For example:

```
cd /local/repos
mkdir -p 2.15/src
mkdir -p 3.0/src
mkdir -p 3.1/src
```

3. Populate your `src/contrib` directory with the contents of <http://packages.revolutionanalytics.com/cran/3.0/stable/src/contrib>. (If you are supporting earlier versions of Revolution R, you will also need the contents of the

14 Setting Up a Package Repository

corresponding 2.15, 3.0, etc., directories.) It is easiest to populate the directory using rsync as follows:

```
rsync -rtlzv --delete \  
packages.revolutionanalytics.com::REVO-PKG/3.1/stable/src/contrib \  
/local/repos/3.1/src
```

(If your corporate policies do not allow rsync usage, you can access the repository via ftp to ftp.revolutionanalytics.com using username anon and password anon.)

To create a partial mirror for Windows binary packages:

1. Choose a location for your repository somewhere on your local network. If you have a corporate intranet, this is usually a good choice, provided URLs have the prefix http:// and not https://. Any file system that is mounted for all users can be used; file-based URLs of the form file:// are supported by the R functions. In this example, we suppose the file system /local is mounted on all systems and we will create our repository in the directory /local/repos.
2. Create a directory bin/windows and a subdirectory contrib as follows:

```
cd /local/repos  
mkdir -p bin/windows/contrib
```

If you need to support more than one version of Revolution R Enterprise in your organization, you can create version directories in the top-level repos directory, and then the src/contrib directories as subdirectories of that. For example:

```
cd /local/repos  
mkdir -p bin/windows/contrib/2.15  
mkdir -p bin/windows/contrib/3.0  
mkdir -p bin/windows/contrib/3.1
```

3. Populate your bin/windows/contrib directory with the contents of <http://cran.r-project.org/bin/windows/contrib/3.1>. (If you are supporting earlier versions of Revolution R, you will also need the contents of the corresponding 2.15, 3.0, etc., directories.) It is easiest to populate the directory using rsync as follows:

```
rsync -rtlzv --delete \  
cran.r-project.org::CRAN/bin/windows/contrib/3.1 \  
/local/repos/bin/windows/contrib/3.1
```

(If your corporate policies do not allow rsync usage, you can access the repository via ftp to ftp.revolutionanalytics.com using username anon and password anon.)

4.2.2 Creating a Custom Package Directory

Your custom package directory can contain source packages, binary packages, or both. Windows binary packages should have a path of the form

```
/local/repos/bin/windows/contrib/3.1
```

while source packages should have a path of the form:

```
local/repos/src/contrib
```

Windows binary packages should be built with your current R version. Windows binary files will be zip files, package source files will be tar.gz files. Move the desired packages to the appropriate directories, then run the R function `write_PACKAGES` in the tools package to create the PACKAGES and PACKAGES.gz index files:

```
> tools::write_PACKAGES("/local/repos")
```

4.3 Configuring R to Use Your Local Repository

To make your local repository available to your R users:

1. Open the file `Rprofile.site` in the `/etc` directory of your installed R (if you installed to the default `/usr` prefix, the path is `/usr/lib64/Revo-7.3/R-3.1.1/lib64/R/etc/Rprofile.site`).
2. Find the following line in the `Rprofile.site` file:

```
r <- c(REVO=Revobase::getRevoRepos())
```

3. If you created a full mirror, add your local repository as follows:

```
r <- c(REVO=Revobase::getRevoRepos(),
      LOCAL="file:///local/repos/cran/3.0/stable")
```

4. If you created just one repository, add it as follows:

```
r <- c(REVO=Revobase::getRevoRepos(),
      LOCAL="file:///local/repos")
```

If you added more than one repository and used R version numbers, add multiple lines of the form `LOCAL_VER="file:///local/repos/VER"`, for example:

```
r <- c(REVO=Revobase::getRevoRepos(),
      LOCAL_3.1="file:///local/repos/3.1",
      LOCAL_3.0="file:///local/repos/3.0",
      LOCAL_2.15="file:///local/repos/2.15")
```

5. Save the file and your repository is ready for use.

5 Removing Revolution R

To remove Revolution R Enterprise from your computer, use the uninstall script created as part of the installation process. The file `uninstall.sh` is placed in the main Revolution R Enterprise directory, typically `/usr/lib64/Revo-7.3`. You can remove Revolution R Enterprise as follows (this requires root privileges):

```
/usr/lib64/Revo-7.3/uninstall.sh
```

You will see a confirmation screen asking if you would like to uninstall Revolution R, together with a warning that any data inside the installation will be lost if the uninstall continues. Press 'y' to complete the uninstall process.

6 Managing Multiple R Installations

You can install the latest Revolution R Enterprise side-by-side with an existing Revolution R Enterprise. In this chapter, we discuss maintaining multiple versions of Revolution R Enterprise, together with instructions for maintaining Revolution R Enterprise alongside versions of R obtained from CRAN or third-party repositories such as <http://dl.fedoraproject.org/pub/epel>. (Note, however, that bugfix releases such as Revolution R Enterprise 6.0.1 are intended to replace their associated main release, not live side by side with it. You must uninstall and reinstall such releases.)

6.1 Side-by-Side Revolution R Installation

If you have Revolution R Enterprise 7.2 or earlier installed and want to install Revolution R Enterprise 7.3, you can simply run the quick install as described in Chapter 1. The installer will detect the older version and rename the R, Revo64, and Revoscript scripts with the appropriate version number and a .old extension (for example, Revo64-7.2.old). You can continue to use these scripts to access your older version, while using Revo64 to launch the current version. If you later decide to uninstall Revolution R Enterprise 7.3, the older scripts will be restored to their original names, so that Revo64 will launch the earlier version of Revolution R Enterprise.

6.2 Using Revolution R with Other R Installations

If you have an installed R script in your /usr/bin directory and then install Revolution R Enterprise, the automated installer detects that R exists and does not prompt you to link R to Revo64. In this case, you can use both versions of R simply by calling them by their respective script names, "R" and "Revo64". To have Revolution R Enterprise become your default R, do the following:

1. Rename the R script by appending the R version, for example 3.1.1:

```
mv /usr/bin/R /usr/bin/R-3.1.1
```

2. Create a symbolic link from the Revo64 script to R:

```
ln -s /usr/bin/Revo64 /usr/bin/R
```

If you have installed Revolution R Enterprise and then install base R, again there should be no conflict; R and Revo64 point to their respective installations, as usual. You can use the above procedure to make Revolution R Enterprise your default version of R.

7 Using Revolution R Enterprise with Platform LSF

If you have access to a Platform LSF cluster equipped with Platform MPI, you can use the RevoScaleR package in Revolution R Enterprise to distribute computations on any available queue. In order to distribute computations, Revolution R Enterprise must be installed on all the nodes of the cluster (or at least all the nodes in a particular queue) and you must have configured your system with a shared directory to which all R users have read and write permissions, and which is NFS mounted with the same path on all nodes of the cluster. The cluster needs to be set up to allow passwordless-login between any two nodes, as through a single-sign-on system, and the cluster must be equipped with Platform MPI.

Client hosts (see the LSF Administrators Guide) are supported, but keep in mind that LSF captures the user's environment at login and all instances of a job are run in that environment. Because LSF is able to determine what the environment is at login, no runtime changes to the environment are applied to the remote environment in which the jobs run. In particular, if there are conflicting environment variables, the client environment variables will be used, and these may not have the proper settings for the cluster nodes to run their jobs. Thus, you may need to set up special user accounts on the client hosts that have a controlled environment that matches what is needed on the cluster.

If, in such a cluster, a given node is configured in such a way that both the R and Rscript commands work successfully and the bsub command works correctly, Revolution R Enterprise should be able to distribute computations on the cluster.

The shared directory is used to store distributed computing jobs from Revolution R Enterprise. In the default case, these jobs are cleaned up after their results are returned. However, users can configure their computations so that the jobs are left on disk until cleaned up by the user. If not cleaned up periodically, they can fill up the disk.

7.1 General Concepts

7.1.1 System Requirements

Revolution R Enterprise works with IBM Platform LSF 7, 8, and 9; specifically, it has been tested with LSF 7.0.6, LSF 8.3, and LSF 9.1.1. IBM Platform MPI is required.

All nodes to be configured with IBM Platform LSF need IBM Platform LSF licenses; all nodes that are to run Revolution R Enterprise must have Revolution R Enterprise licenses. All nodes must be running Red Hat Enterprise Linux 5 or 6 (or fully compatible operating system).

Minimum system configuration requirements are as follows:

Processor: 64-bit CPU with x86-compatible architecture (variously known as AMD64, Intel64, x86-64, IA-32e, EM64T, or x64 CPUs). Itanium-architecture CPUs (also known as IA-64) are not supported. Multiple-core CPUs are recommended.

Operating System: Red Hat Enterprise Linux 5.4, 5.5, 5.6, 5.7, 5.8, 5.9, 5.10, 5.11, 6.0, 6.1, 6.2, 6.3, 6.4, or 6.5. Only 64-bit operating systems are supported.

Memory: A minimum of 1GB of RAM is required; 4GB or more are recommended.

Disk Space: A minimum of 500MB of disk space is required.

7.1.2 Roles of Hosts in a Cluster

The following terms apply to computers within the cluster, and define the roles of hosts within the cluster.

Master Host: A host in the cluster that can schedule jobs. It may also be a server host, and is also a submission host, but is never a client host.

Server Host: A host in the cluster that can run jobs. It may also be a master host, and is also a submission host, but is never a client host.

Submission Host: Any host that can submit a job. Master hosts, server hosts and client hosts are submission hosts.

Client Host: A host that can submit jobs, but itself is not included in the list of hosts that will be used to run the job.

7.1.3 Configuring Network Architecture

It is often useful, though not strictly necessary, to set up your IBM Platform LSF cluster on a separate network subnet from your main network. In such a configuration, one master, or *primary*, host will have two IP addresses, one on your main network and one on the cluster subnet, and will serve as the gateway from clients on the main network to worker hosts on the cluster.

The main network address is configured using `/etc/sysconfig/ifcfg-eth0`, for example:

```
# Intel Corporation 82545EM Gigabit Ethernet Controller (Copper)
DEVICE=eth0
BOOTPROTO=dhcp
IPV6INIT=yes
IPV6_AUTOCONF=yes
ONBOOT=yes
DHCP_HOSTNAME=lsf-31
#need to set first DNS server to 127.0.0.1 for dnsmasq to resolve
#names later on
#these can be commented out until dnsmasq daemon is running
#correctly.
#reboot to make changes take effect
DNS1=127.0.0.1
DNS2=192.168.10.111
DNS3=192.168.10.99
```

The cluster subnet address is configured using `/etc/sysconfig/ifcfg-eth1`, for example:

```

#here we set a static address for eth1, the private network
# Intel Corporation 82545EM Gigabit Ethernet Controller (Copper)
DEVICE=eth1
#BOOTPROTO=dhcp
ONBOOT=yes
HWADDR=00:50:56:b7:6e:27
BOOTPROTO=static
BROADCAST=172.16.255.255
IPADDR=172.16.1.31
IPV6INIT=yes
IPV6_AUTOCONF=yes
NETMASK=255.255.0.0
NETWORK=172.16.1.0
ONBOOT=yes

```

In this configuration, it is important that IP packet forwarding is correctly configured. You do this by editing `/etc/sysctl.conf`, and editing ip forwarding line to read as follows:

```
net.ipv4.ip_forward = 1
```

You will also need to update `/etc/sysconfig/iptables` to disable the firewall, or explicitly open specific ports:

You will also want to edit `/etc/dnsmasq.conf` to specify the IP address range for your subnet cluster:

```

#dnsmasq is typically already installed on CentOS 5.
#below are the options that should be changed in the config file.
#use appropriate values for the DHCP range and the size of the
# cluster.here I have 8 addresses for a 5 node cluster
strict-order
interface=eth1
domain=revolution-computing.com
dhcp-range=172.16.1.32,172.16.1.40,255.255.0.0,7d
dhcp-host=00:50:56:b7:69:cf,172.16.1.35 #lsf-32
dhcp-host=00:50:56:b7:16:ab,172.16.1.38 #lsf-33
dhcp-host=00:50:56:b7:1b:c4,172.16.1.39 #lsf-34
dhcp-host=00:50:56:b7:6f:7d,172.16.1.32 #lsf-35

```

7.2 Downloading and Installing IBM Platform MPI

Revolution R Enterprise uses IBM Platform MPI to send jobs to the IBM Platform LSF scheduler; it must be included on all LSF hosts and client hosts.

The currently shipping and supported version of IBM Platform MPI is 9.1; download the file `platform_mpi-09.1.0.0r.x64.bin`.

To install the software, run the following as root, or via sudo:

```
./platform_mpi-09.1.0.0r.x64.bin
```

Accept the license when prompted.

If IBM Platform MPI is not installed or improperly licensed, high-performance analytics functions such as `rxSummary`, `rxLinMod`, and `rxGlm` will return errors when run in the `RxLsfCluster` compute context.

7.3 Downloading and Installing IBM Platform LSF

7.3.1 Downloading the Software

The currently shipping and supported version of IBM Platform LSF is 9.1.1. If you are purchasing and installing IBM Platform LSF for the first time, you will be downloading and installing this version. If you purchased and downloaded an earlier version of IBM Platform LSF, you may need to modify the following instructions.

7.3.2 Installing the Software

You need to follow the instructions in this section on each LSF server and client host in your network. (You will probably find it easiest to create your `install.config` file first on your master host, then copy it to the other hosts before completing the installation.)

Prior to installing LSF 9.1, you should have the following files collected in a single directory:

- `lsf9.1.1_documentation.tar.Z`
- `lsf9.1.1_linux2.6-glibc2.3-x86_64.tar.Z`
- `lsf9.1.1_lsfinstall_linux_x86_64.tar.Z`
- `platform_lsf_<ed>_entitlement.dat` (where `<ed>` is one of `std`, `exp`, or `adv`, depending on whether you are installing the standard, express, or advanced edition of LSF)

Place the entitlement file where it will eventually reside, typically in `/usr/share/lsf/lsf_distrib`:

```
mkdir -p /usr/share/lsf/lsf_distrib
cp platform_lsf_exp_entitlement.dat /usr/share/lsf/lsf_distrib
```

Extract the installer files using the following command:

```
zcat lsf9.1.1_lsfinstall_linux_x86_64.tar.Z | tar xvf -
```

Change directory to the `lsf9.1.1_lsfinstall` directory created:

```
cd lsf9.1.1_lsfinstall
```

Edit the file `install.config` to specify the following information:

- The location of the top-level LSF installation directory, typically `/usr/share/lsf`:

```
LSF_TOP=/usr/share/lsf
```

- The user names of all LSF administrators:

```
LSF_ADMINS="lsfadmin user1 user2"
```

- The name of the LSF cluster (be sure that this is different from the name of any host, user, or user group on your system):

```
LSF_CLUSTER_NAME="myLSFcluster"
```

- The names of the machines that will serve as your LSF server hosts; the first machine listed is the default master host:

```
LSF_MASTER_LIST="lsf-21 lsf-22 lsf-23 lsf-24 lsf-25"
```

- (Optional) The names of machines that will serve as LSF client hosts. LSF client hosts can submit jobs, but are not under control of the scheduler and perform no distributed tasks.

```
LSF_ADD_CLIENTS="client1 client2"
```

- The name and location of your LSF entitlement file:

```
LSF_ENTITLEMENT_FILE="/usr/share/lsf/lsf_distrib/platform_lsf_exp_entitlement.dat"
```

- Any additional settings specific to your LSF cluster; see the LSF documentation file *Installing IBM Platform LSF on UNIX and Linux* for more details.

As root, run the following command to complete the installation:

```
./lsfinstall -f install.config
```

(Remember, you must repeat the installation on each server and client host.)

- If you copied the entitlement file into the standard location, answer 'y' when warned about overwrite and asked if you wish to continue. The file will be preserved.
- When prompted, select the correct host type (which should be a glibc x86_64 type for 64 bit glibc Linux).

Before starting up the LSF software, we first will set up some other services and directories.

7.4 Optimizing for Speed (optional)

It is possible to optimize smaller clusters (less than 40-60 nodes) for increased speed. High use, high volume clusters should not perform these steps. However, performing this step will reduce scheduler overhead significantly.

On each host in the cluster, edit the file:

```
/usr/share/lsf/conf/lsbatch/CLUSTERNAME/configdir/lsb.params
```

And set the values of the following parameters as follows:

```
MBD_SLEEP_TIME = 2          #mbatchd scheduling interval (30 secs is
default)
SBD_SLEEP_TIME = 1          #sbatchd scheduling interval (15 secs is
default)
JOB_ACCEPT_INTERVAL = 1     #interval for any host to accept a job
JOB_SCHEDULING_INTERVAL=500 ms
```

If you are on a higher capacity cluster, you will want to consult the LSF System Administrator's guide for more information on tweaking these parameters.

7.5 Configuring the LSF Server Hosts

7.5.1 Disabling the Firewall

If you are running on an isolated network, disable the firewall with the following commands run as root:

```
service iptables stop
service ip6tables stop
chkconfig iptables stop
chkconfig ip6tables stop
```

Note that if your cluster is not on an isolated network, you will need to open the ports for nfs and lsf. Consult the lsf documentation for the specific ports that must be opened.

7.5.2 Ensuring Host Communication

On each LSF server host, run the `hostsetup` script as root to ensure the host is properly equipped to run LSF; the `boot` option configures the LSF daemons to start automatically on boot:

```
/usr/share/lsf/9.1/install/hostsetup --top="/usr/share/lsf" \  
--boot="y"
```

Then, edit the following files and ensure that they contain the appropriate host names:

- `/etc/sysconfig/network-scripts/ifcfg-eth0`
- `/etc/sysconfig/network`
- `/etc/hosts`

7.5.3 Disabling SELinux

To get an IBM Platform LSF cluster up and running quickly, it is usually easiest to disable SELinux, a Linux feature that provides enhanced security. If SELinux is properly configured, both IBM Platform LSF and Revolution R Enterprise can run together under it, but SELinux configuration is beyond the scope of this manual. However, disabling SELinux is simple:

1. Open the file `/etc/sysconfig/selinux`.
2. Change the line reading `SELINUX=enforcing` to `SELINUX=disabled`.
3. Reboot any host that has had its SELinux settings changed.

7.6 Configuring the Shared Data Directory

One host will serve as a “staging area” for data transfer between hosts. While it is a good idea to build redundancy into your cluster setup for this functionality, only one compute can be configured to host this staging area at any point in time.

7.6.1 Creating the Shared Directory

Revolution R Enterprise requires a shared directory that all users can write to, and this directory must be shared among all the nodes in the LSF cluster. We recommend creating a specific “allshare” directory that is world-writable, and recommend that individual user directories be created beneath this.

The node selected to host the shared directory must be visible on the network to all of the cluster servers; it is not required that this node be part of the cluster.

To create such a directory, on the node selected to host the shared directory, do the following:

```
mkdir /mnt/allshare
chmod -R 777 /mnt/allshare
```

Next, export the `/mnt/allshare` directory to your cluster’s IP network by editing the master’s `/etc/exports` file. For example, if your master node has an external facing IP address and a cluster-facing IP address, you would export the `/mnt/allshare` directory to both by adding the following lines to `/etc/exports`:

```
/mnt/allshare 192.168.10.0/24(rw, sync)
/mnt/allshare 172.16.1.0/16(rw, sync)
```

Start (or re-start) the NFS daemon for the export to take effect:

```
chkconfig nfs on
service nfs restart
```

7.6.2 Mounting the Shared Directory

On each computer in the cluster (except the one on which you created the shared directory), set up the file `/etc/fstab` to mount the share directory and data directories on all hosts. For example:

```
/dev/sdb1      /data  ext4    noatime,nodiratime,defaults 0 0
lsf-31:/mnt/allshare /mnt/allshare nfs     rw,intr,noatime,nodiratime 0 0
```

Then, as the root user, mount the directory:

```
mount -a
```

7.6.3 Configuring Client Hosts

On each LSF client host (if any), run the `hostsetup` script as root to ensure the host is properly equipped to run LSF; however, because LSF client hosts do not run the LSF daemons, set the boot option to “n”:

```
/usr/share/lsf/9.1/install/hostsetup --top="/usr/share/lsf" \  
--boot="n"
```

7.7 User Setup

7.7.1 Configuring the User Environment

Each user who will be submitting jobs via LSF must have their user environment modified to incorporate the LSF environment. Normally, this is done as part of LSF configuration, but if it was not, it must be done before submitting jobs from Revolution R Enterprise. This step must be performed on each submission host that the given user will be submitting jobs from (client, server or master).

Setting up the LSF environment is done by sourcing one of two LSF environment scripts: `cshrc.lsf` for `csh` and `tcsh` users, `profile.lsf` for `sh`, `ksh`, or `bash` users.

The following steps must be performed on each host that will be involved in LSF-based computations.

1. Ensure that the following four lines are part of a standard login profile file (`.bash_profile`, `/etc/profile`, or another file that is sourced by one of the standard profile files):

```
. /usr/share/lsf/conf/profile.lsf
export MPI_ROOT=/opt/ibm/platform_mpi
export MANPATH=$MANPATH:$MPI_ROOT/share/man
export PATH=$PATH:$MPI_ROOT/bin
```

(Modify the path to `profile.lsf` as necessary for your system; `/usr/share/lsf/conf` is the default, similarly, modify the path to `MPI_ROOT` as necessary.)

- Each user should log out and back in to pick up environment changes.

7.7.2 Passwordless authentication

In order for jobs to work, a user must have passwordless authentication between all the nodes of the cluster. By default, the ssh protocol is used by LSF.

If user logins are performed through a domain level authentication (e.g., Active Directory), the following should be added to the end of the `/usr/share/lsf/conf/lsf.conf` file:

```
LSF_RSH="ssh -n -c arcfour"
LSF_LSLOGIN_SSH="ssh -n -c arcfour"
#need to add AD domain to add support for hybrid Linux/Windows
#LSF clients.
LSF_USER_DOMAIN="MY_DOMAIN"
```

where `MY_DOMAIN` is the name of the domain into which users log in.

The second approach is to use key based logins. Use `ssh-keygen` to generate a key, and `ssh-copy-id` to copy the credentials to all the other hosts in the cluster.

7.7.3 A Note on X Forwarding

Many ssh clients allow for X Windows forwarding (tunneling over ssh), using the `-X` or `-Y` flags when using ssh. We highly recommend that X Windows forwarding be turned *off* (that is, omitting the `-X` and `-Y` flags) when using Revolution R Enterprise for distributed computing.

The reason for this is that LSF exports the user's local environment to each machine upon which a part of the overall job will run. This includes the X Windows forwarding, which adds considerable overhead to job execution. This can result in perceptible slowdowns of job and task execution.

7.8 Verifying the Installation

Beginning with any master hosts and then on all the LSF server hosts (but not on client hosts) in turn, do the following:

- Log on as root.
- Issue the following commands:

```
lsadmin limstartup
lsadmin resstartup
badmin hstartup
```

Wait a few minutes for the LSF server hosts to contact each other, select the master host, and exchange initialization information. Then issue the following commands to verify that your cluster is working properly:

```
lsadmin ckconfig -v
```

This should return something like the following:

```
Checking configuration files ...
```

```
Platform EGO 1.2.7.0 build 88888, Feb 12 2013
Copyright International Business Machines Corp, 1992-2013.
US Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corp.

binary type: linux2.6-glibc2.3-x86_64
Reading configuration from
/usr/share/lsf91/conf/ego/myLSF9cluster/kernel/ego.conf
Feb 13 10:31:32 2013 759460 6 1.2.7 Lim starting...
Feb 13 10:31:32 2013 759460 6 1.2.7 LIM is running in advanced workload
execution mode.
Feb 13 10:31:32 2013 759460 6 1.2.7 Master LIM is not running in
EGO_DISABLE_UNRESOLVABLE_HOST mode.
Feb 13 10:31:32 2013 759460 5 1.2.7 /usr/share/lsf91/9.1/linux2.6-
glibc2.3-x86_64/etc/lim -C
Feb 13 10:31:32 2013 759460 6 1.2.7 LIM is running as IBM Platform LSF
Standard Edition.
Feb 13 10:31:32 2013 759460 7 1.2.7 chuserId: setresuid/seteuid(0)
failed, Operation not permitted.
Feb 13 10:31:32 2013 759460 3 1.2.7 domanager():
/usr/share/lsf91/conf/lsf.cluster.myLSF9cluster(15): The cluster manager
is the invoker in debug mode
Feb 13 10:31:32 2013 759460 6 1.2.7 reCheckClass: numhosts 1 so reset
exchIntvl to 15.00
Feb 13 10:31:32 2013 759460 6 1.2.7 Checking Done.
-----
No errors found.
```

The `lsid` command tells you if your LSF environment is properly set up. If so, it will return information on the current version of LSF, the cluster name, and the master host name.

```
lsid
```

The `lshosts` command lists the LSF server and client hosts; the `bhosts` lists just the LSF server hosts:

```
lshosts
bhosts
```

At this point, your installation of Platform LSF is complete, and your cluster is operational. Moving forward, it is possible to shut down the cluster from any master host with the following command:

```
lsfshutdown
```

and start it up again with:

```
lsfstartup
```

Note that these commands should be run on one master host at a time.

7.8.1 A note on \$TMPDIR

Under some circumstances, LSF will attempt to overwrite the \$TMPDIR environment variable in the job environments. To determine if this issue exists on your cluster, run the following command:

```
bsub -I set | grep TMPDIR
```

If this job returns something of the form /tmp/12345.tmpdir/, you will probably want to stop the cluster, place the following line in /usr/share/lsf/conf/lsf.conf on each node, then restart the cluster:

```
LSB_SET_TMPDIR=n
```

8 Using Revolution R with a Third-Party IDE

Many Linux R users find that their productivity is greatly increased by the use of integrated development environments such as RStudio or Eclipse with the StatET plugin. These environments can be used with Revolution R Enterprise, although some care is needed to ensure that you are using compatible versions of the software. This chapter describes running RStudio and StatET with Revolution R Enterprise 7.

8.1 Running Revolution R with RStudio

RStudio (<http://www.rstudio.com/ide>) is an open-source, cross-platform integrated development environment for R. RStudio for Linux comes in both desktop and server versions. The server version, in turn, comes in both free and professional versions. For local computations (by which we mean computations on a single computer, as opposed to distributed computation across several computers), both the free and professional versions of RStudio Server work just fine. The free version can also be used as a client for launching distributed computing jobs on a Hadoop cluster. For distributed computing involving IBM Platform LSF, however, you must have the professional version of RStudio Server installed on one of the nodes of your LSF compute cluster. We recommend the free server version 0.97.551 for use on both RHEL5 and RHEL6 systems. (We have found the desktop version incompatible with RHEL5 as it requires newer development libraries than are available in the standard RHEL5 repositories.) The server version, however, will run “out-of-the-box” on both RHEL5 and RHEL6 systems.

8.1.1 Installing RStudio Server

To install and start the free version of RStudio Server, run the following commands at a shell prompt after installing Revolution R Enterprise:

```
wget http://download2.rstudio.org/rstudio-server-0.97.551-x86_64.rpm
```

28 Using Revolution R with a Third-Party IDE

```
sudo rpm -i rstudio-server-0.97.551-x86_64.rpm
```

(If you purchase the professional version, you will obtain your download link directly from RStudio. If you are installing RStudio Server to a cluster managed by Cloudera Manager, add the `--no-deps` flag to the rpm command above.)

You should see the following message when the installation completes:

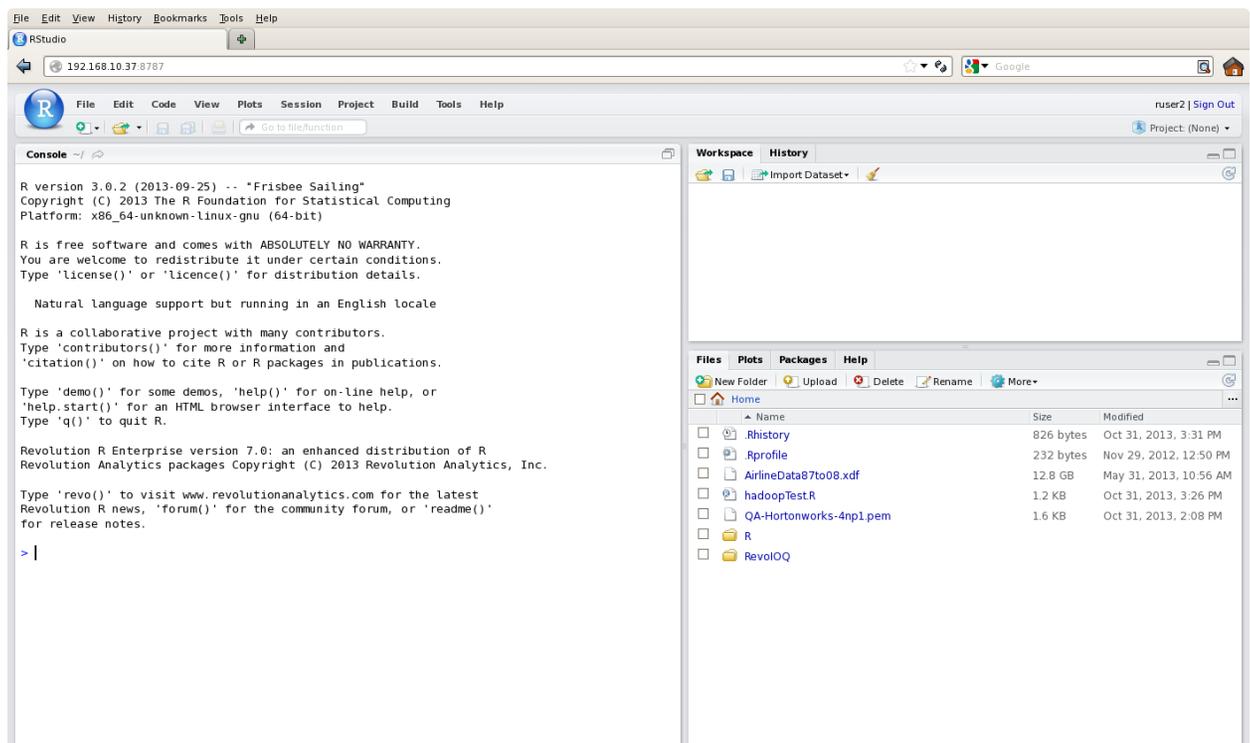
```
Starting rstudio-server:          [ OK ]
```

8.1.2 Accessing RStudio Server

To get started using RStudio Server, open a browser and navigate to the following URL:

<http://<IP-of-server>:8787>

You will be prompted for a username and password. RStudio uses the default authentication on your server to verify users, so it is possible you will need to add users via the standard Linux `useradd` command for successful authentication. Once you have successfully logged in to RStudio, you will see the IDE as shown below:



RStudio includes online documentation that discusses its features in detail.

8.2 Running Revolution R with StatET

Eclipse (<http://www.eclipse.org>) is a Java-based, open-source, cross-platform integrated development environment with base support for a number of programming languages, including Java, C, and C++.

StatET (<http://www.walware.de/?page=/it/statet/index.mframe>) is an Eclipse plug-in providing support for the R language.

To run Revolution R with StatET, you must install a Java JDK (1.6 or later), an Eclipse IDE (StatET's developers recommend the Eclipse IDE for Java Developers; these instructions were developed using the Eclipse IDE for C/C++ Developers), Revolution R Enterprise, and the two R packages `rj` and `rj.gd` available from the Walware web site. We will assume in what follows that you or your system administrator have installed the appropriate Java JDK on your system, and that you have already installed Revolution R.

8.2.1 Installing Eclipse

To install Eclipse, first download the desired 64-bit base IDE from <http://www.eclipse.org/downloads>. You should have a file with a name of the form

```
eclipse-<lang>-juno-SR1-linux-gtk-x86_64.tar.gz
```

where `<lang>` is one of `cpp` or `java`. The following commands should install and start Eclipse on your Linux system:

```
tar zxvf eclipse-<lang>-juno-SR1-linux-gtk-x86_64.tar.gz
cd eclipse
eclipse
```

8.2.2 Installing StatET

Before installing StatET, review the requirements at <http://www.walware.de/?page=/it/statet/installation.mframe>. Install the required packages `rj` and `rj.gd` by starting Revolution R and calling `install.packages` as follows:

```
install.packages(c("rj", "rj.gd"),
  repos="http://download.walware.de/rj-1.1",
  lib="/usr/lib64/Revo-7.1/R-3.0.3/lib64/R/library")
```

With some Java installations, it may be necessary to run R CMD `javareconf` before installing the `rj` package. If you see a message about being unable to load `libjvm.so` during the installation of `rj.gd`, you may need to create a symbolic link from your installed version of `libjvm.so` to the system library, such as the following:

```
ln -s /path/to/libjvm.so /usr/lib/libjvm.so
```

Once you have successfully installed the two prerequisite R packages, you install StatET as follows:

1. From the Eclipse IDE, select Install New Software from the Help. The Install dialog box appears.
2. To the right of the text box labeled Work with:, click Add. The Add Repository dialog box appears.
3. In the Name textbox, type StatET.
4. In the Location textbox, type or paste in the URL <http://www.walware.de/goto/statet>.
5. Click OK. The available WalWare downloads should appear.
6. Click Select All to select all available WalWare downloads.
7. Click Next.
8. Agree to the license terms and click Finish. The Eclipse installer will download and install the StatET plug-in.

Once StatET is installed, define an R environment for Revolution R Enterprise as follows:

1. From the Eclipse IDE, select Preferences from the Window menu. The Preferences dialog box appears.
2. In the left navigation panel, expand StatET, and then expand Run/Debug.
3. Click R Environments.
4. In the right pane, click Add. The Add R Environment Configuration dialog box appears.
5. In the text box labeled Location (R HOME), enter the full path to your Revolution R installation, typically `/usr/lib64/Revo-7.1/R-3.0.3/lib64/R`. If you do not know this path, you can obtain it from a shell prompt using the following command:

```
Revo64 RHOME
```

6. Click OK.

Next, set up a run configuration to use your R environment as follows:

1. From the Eclipse IDE, select Run Configurations. . . from the Run menu. The Run Configurations dialog box appears.
2. In the Name text box, enter a name for this configuration, such as "Revolution R Enterprise 6.2."
3. Click the R Config tab.
4. Select the configuration you created in the previous procedure.
5. Click Run. You should see an R console appear in your Eclipse window, as shown below.

