



Package Management for the TIBCO Spotfire® Environment

*Software Release 10.3 LTS
April 2019*

Important Information

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

ANY SOFTWARE ITEM IDENTIFIED AS THIRD PARTY LIBRARY IS AVAILABLE UNDER SEPARATE SOFTWARE LICENSE TERMS AND IS NOT PART OF A TIBCO PRODUCT. AS SUCH, THESE SOFTWARE ITEMS ARE NOT COVERED BY THE TERMS OF YOUR AGREEMENT WITH TIBCO, INCLUDING ANY TERMS CONCERNING SUPPORT, MAINTENANCE, WARRANTIES, AND INDEMNITIES. DOWNLOAD AND USE OF THESE ITEMS IS SOLELY AT YOUR OWN DISCRETION AND SUBJECT TO THE LICENSE TERMS APPLICABLE TO THEM. BY PROCEEDING TO DOWNLOAD, INSTALL OR USE ANY OF THESE ITEMS, YOU ACKNOWLEDGE THE FOREGOING DISTINCTIONS BETWEEN THESE ITEMS AND TIBCO PRODUCTS.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of TIBCO Software Inc.

TIBCO, Two-Second Advantage, TIBCO Spotfire, TIBCO Enterprise Runtime for R, TIBCO Spotfire Server, TIBCO Spotfire Web Player, TIBCO Spotfire Statistics Services, S-PLUS, and TIBCO Spotfire S+ are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Enterprise Java Beans (EJB), Java Platform Enterprise Edition (Java EE), Java 2 Platform Enterprise Edition (J2EE), and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle Corporation in the U.S. and other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. Please see the readme.txt file for the availability of this software version on a specific operating system platform.

THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

This and other products of TIBCO Software Inc. may be covered by registered patents. Please refer to TIBCO's Virtual Patent Marking document (<https://www.tibco.com/patents>) for details.

Copyright © 1999-2019 TIBCO Software Inc. All rights reserved.

Contents

Documentation and support services	6
Package Management	9
Package management orientation	9
Find help	9
R language primer	9
Recommendations for using R securely	10
Language options	11
Reviewing the list of known differences	11
Spotfire packages and R binary packages	12
Manage packages through roles	12
Developer role	12
Curator role	13
Administrator role	13
Package installation locations and recommendations for updating	13
Setting JAVA_HOME	16
Installing the rJava package	17
Manage your packages when you install a new version of TERR	18
Manage packages using Spotfire and TERR	19
Development tools for creating packages	20
Limitations and considerations	20
R package anatomy	21
Packages running in a local TIBCO Enterprise Runtime for R engine in Spotfire	22
Opening the TIBCO Enterprise Runtime for R console from Spotfire Analyst	22
Checking installed packages	23
Packages for use with Spotfire	23
Package repositories	24
Installation options for packages	24
Specifying an older package from TRAN	27
Testing packages locally	27
Getting package help	28
Removing a package from a Spotfire installation	29
The Spotfire SPK	30
Package workflow	30
Stage the packages	30
Distribute and use the packages	31
Obtaining the SpotfireSPK toolset	32
Creating the Spotfire SPK	33

Spotfire SPK versioning	34
Deploy the SPK to the Spotfire Server	35
Packages deployed for a small group	36
Spotfire package maintenance	37
Install packages on Spotfire Statistics Services	37
Uploading a package to Spotfire Statistics Services from a repository	38
Uploading a package to Spotfire Statistics Services from another computer	40
Uploading a package using TSSS Connector	41
Maintaining a package in TIBCO Spotfire Statistics Services	42
Validating the package upload	43
Manage packages between Spotfire and Spotfire Statistics Services	43
Changing the local engine option	44
Troubleshooting TERR and Spotfire packages	46

Documentation and support services

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the TIBCO Product Documentation website, mainly in HTML and PDF formats.

The TIBCO Product Documentation website is updated frequently and is more current than any other documentation included with the product. To access the latest documentation, visit <https://docs.tibco.com>.

TIBCO Spotfire Statistics Services documentation

You can find the following documents for TIBCO Spotfire® Statistics Services in the TIBCO Documentation Library.

- *Release Notes*
- *System Requirements*
- *Installation and Administration*
- *Package Management for the TIBCO Spotfire® Environment*
- *User's Guide*
- *Java API Reference*
- *C# API Reference*
- *URL API Reference*
- *SpotfireUtils Package Reference*
- *Release Notes*
- *License Agreement*

TIBCO Enterprise Runtime for R documentation

You can find the following documents for TIBCO Enterprise Runtime for R in the TIBCO Documentation Library.

- *TIBCO® Enterprise Runtime for R Technical Documentation*
- *Language Reference (HTML)*
- *Differences Between TIBCO® Enterprise Runtime for R and Open-Source R (HTML)*
- *Release Notes (PDF)*
- *License Agreement (PDF)*

You can also find links to CRAN package compatibility reports for this release on TIBCO Cloud™ Spotfire®.

TIBCO Spotfire Analyst documentation

You can find the following documents for TIBCO Spotfire® Analyst in the TIBCO Documentation Library.

- *TIBCO Spotfire® Analyst User's Guide*
- *TIBCO Spotfire® API Reference*
- *TIBCO Spotfire® Administration Manager User's Guide*

- *TIBCO Spotfire® Cobranding*
- *Deploying and Using a TIBCO Spotfire® Language Pack*
- *Running TIBCO Spotfire® on Citrix XenApp™*
- *Working with Cubes in TIBCO Spotfire®*
- *TIBCO Spotfire® Analyst Release Notes*
- *TIBCO Spotfire® License Agreement*
- *TIBCO Spotfire® Language Pack License Agreement*

TIBCO Spotfire Server documentation

The following documents for this product can be found in the TIBCO Spotfire® Server:

- *TIBCO Spotfire® Server Installation and Administration*
- *TIBCO Spotfire® Server Quick Start*
- *TIBCO Spotfire® Business Author and Consumer User's Guide*
- *TIBCO Spotfire® Cobranding*
- *TIBCO Spotfire® Server Web Services API*
- *TIBCO Spotfire® Server Platform API*
- *TIBCO Spotfire® Server Information Services API*
- *TIBCO Spotfire® Server REST API*
- *TIBCO Spotfire® Server Release Notes*
- *TIBCO Spotfire® Server License Agreement*

System Requirements for Spotfire Products

For information about the system requirements for Spotfire products, visit <http://spotfi.re/sr>.

Release Version Support

Some release versions of TIBCO Spotfire products are designated as long-term support (LTS) versions. LTS versions are typically supported for up to 36 months from release. Defect corrections will typically be delivered in a new release version and as hotfixes or service packs to one or more LTS versions. See also https://docs.tibco.com/pub/spotfire/general/LTS/spotfire_LTS_releases.htm.

How to Contact TIBCO Support

You can contact TIBCO Support in the following ways:

- For an overview of TIBCO Support, visit <http://www.tibco.com/services/support>.
- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the TIBCO Support portal at <https://support.tibco.com>.
- For creating a Support case, you must have a valid maintenance or support contract with TIBCO. You also need a user name and password to log in to <https://support.tibco.com>. If you do not have a user name, you can request one by clicking Register on the website.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can

submit and vote on feature requests from within the [TIBCO Ideas Portal](#). For a free registration, go to <https://community.tibco.com>.

For quick access to TIBCO Spotfire content, see <https://community.tibco.com/products/spotfire>.

Package Management

The Package Management guide provides information about working with two kinds of packages in the Spotfire ecosystem: the R package, which you use with TERR, and the Spotfire package, or SPK, which you deploy from the Spotfire Server to client users.


This guide provides an orientation for both kinds of packages, including creating them, deploying and installing them, managing them, and troubleshooting them.

Package management orientation

If you are an R package developer but have no experience with Spotfire, or if your organization is new to advanced data analysis using TERR with Spotfire Analyst, review the topics in this section.

Find help

Spotfire® includes many avenues to help with packages, whether they are R language packages to use with TERR™ or Spotfire packages (SPKs).

Task	Help resource
Building packages for open-source R.	R documentation  <p>Open-source R is available under separate open source software license terms and is not part of TERR. As such, open-source R is not within the scope of your license for TERR. Open-source R is not supported, maintained, or warranted in any way by TIBCO Software Inc. Download and use of open-source R is solely at your own discretion and subject to the free open source license terms applicable to open-source R.</p>
Building packages using TERR	<i>TIBCO® Enterprise Runtime for R (TERR™) Technical Documentation</i> , and Manage packages using TIBCO Spotfire and TERR
Creating a Spotfire SPK distribution.	<i>TIBCO Spotfire® Server Server and Environment Installation and Administration</i> at https://docs.tibco.com/products/tibco-spotfire-server
Learning about Spotfire Statistics Services architecture and server management.	<i>TIBCO Spotfire® Statistics Services Installation and Administration</i> at https://docs.tibco.com/products/tibco-spotfire-statistics-services

R language primer

The R language has been developed into the open-source R engine and the TERR engine (among others), all developed from the legacy S-PLUS language.

Some differences exist between the open-source R engine and the TERR engine; however, they are highly compatible. Most scripts and functions that you write in open-source R run in the TERR engine.



For more information, see *Differences Between TIBCO Enterprise Runtime for R and Open-Source R* at <https://docs.tibco.com/products/tibco-enterprise-runtime-for-r>.



Open-source R is available under separate open source software license terms and is not part of TERR. As such, open-source R is not within the scope of your license for TERR. Open-source R is not supported, maintained, or warranted in any way by TIBCO Software Inc. Download and use of open-source R is solely at your own discretion and subject to the free open source license terms applicable to open-source R.

This documentation is not intended to teach programmers how to write scripts and functions or delve too deeply into creating R language packages. Rather, we address techniques for skilled package developers who must test and share their packages across an organization that has deployed Spotfire (and optionally Spotfire® Statistics Services), and that uses the TERR engine to run the package code.

Recommendations for using R securely

The R Consortium, of which TIBCO is a proud member, has provided a summary of "Best Practices for Using R Securely."

We encourage anyone using open source R, whether with TIBCO products or not, to review those practices at the following site: <https://www.r-consortium.org/blog/2015/08/17/best-practices-for-using-r-securely>. This guidance essentially recommends that users who download R and R packages do so from a secure server using an encrypted HTTPS connection.

The following guidance provides information regarding how these recommendations do, or do not, apply to TERR.

Recommendation: If you download open-source R, always download it from a CRAN server using HTTPS

TERR is a commercial product, and you download it either from our secure TIBCO Product Download site (for customers who purchase TERR) or from the TIBCO Access Point (TAP) site (for members of the TERR Community who are using the free TERR Developer's Edition). Both of these sites use HTTPS.

Recommendation: If you download open-source R, check its MD5 checksums before you begin the installation

Customers downloading TERR from the TIBCO Product Download site should confirm the MD5 checksums following the same process as in detailed in the R Consortium blog post, cited in this topic.

Recommendation: If you have open-source R installed, configure it for secure file downloads

By default, TERR uses HTTPS for secure file download if a secure mirror is specified. There is no need to do any special configuration of TERR.

Recommendation: Always download CRAN packages from a secure mirror

We recommend TERR users follow this recommendation, and always download CRAN packages from a secure mirror. The [Best Practices post](#) includes a list of CRAN sites that use HTTPS.



Open-source R is available under separate open source software license terms and is not part of TERR. As such, open-source R is not within the scope of your license for TERR. Open-source R is not supported, maintained, or warranted in any way by TIBCO Software Inc. Download and use of open-source R is solely at your own discretion and subject to the free open source license terms applicable to open-source R.

Language options

Whether you use TERR or open-source R, you can use the resources and tools in the Spotfire predictive analytics platform.

TIBCO Enterprise Runtime for R Packages

- Test your functions using the TERR engine and use TERR to build your packages. (See [R package anatomy](#) for more information.)
- Use the SpotfireSPK package that is included in the TERR for Spotfire installation to create an SPK that you can put on a Spotfire® Server to distribute to other Spotfire Analyst users in your organization.

Open-source R packages

If you are an open-source R developer, you probably use either your own packages or those downloaded from the Comprehensive R Archive Network (CRAN). You can test either by running them in a local TERR engine.



TIBCO maintains a report of tests run in TERR of CRAN packages' help examples. The report details the success of every expression in every help example provided by the package developer. TIBCO is not responsible for developing, testing, or supporting packages published to CRAN.

For a list of the CRAN packages for which we have run these tests, see the report for your server platform at <https://docs.tibco.com/products/tibco-enterprise-runtime-for-r>.

- In rare cases, we have provided different versions of CRAN packages tested to work with TERR, or we have customized popular CRAN packages to work with TERR. These package versions are loaded from a special TIBCO repository by default when you install them by calling `install.packages()`.
- For more information about testing your open-source R packages, see [Testing packages locally](#).
- Check the [list of known differences](#) between the open-source R engine and the TERR engine by package.

Reviewing the list of known differences

In the installation TERR, you can find a list of known differences in function behavior between open-source R and TERR in the documentation.

You can access the list of known differences from Spotfire Analyst.

Procedure

1. From the menu, click **Tools > TERR Tools**.
2. Click **Open TERR Language Reference**.



The language reference is included in the installation. The links to the technical guides and readme files open these documents on the TIBCO documentation website <https://docs.tibco.com/products/tibco-enterprise-runtime-for-r>.

A web browser launches and displays the landing page with links to the documentation for TERR.

3. In the resulting browser window, click the link titled *Differences Between TERR and Open-Source R*.

Result

The resulting web page provides detailed information of known differences in function behavior between TERR and open-source R, sorted by their packages. (This list is compiled from like sections in the individual function help files.)

Spotfire packages and R binary packages

Spotfire has two different types of packages: the Spotfire package (SPK), and the R language binary package.

The SPK is specific to Spotfire add-in development and deployment. You can create a Spotfire SPK that contains TERR engine-compatible R binary packages, and then deploy the SPK to a Spotfire Server to be distributed to other users in your organization.

Spotfire Analyst includes a license option for the TERR local engine. Your R binary packages, deployed using the SPK mechanism, can be used to create Spotfire analyses that use data functions.

R binary packages can run faster in a local TERR engine than one running in the TERR engine installed remotely on a Spotfire Statistics Services installation. For more information see [Packages Running on a local TIBCO Enterprise Runtime for R engine in Spotfire](#).



Because you are creating or downloading a package to be distributed to Spotfire Analyst users, the package must be a Windows binary package. See [Limitations and considerations](#) for more information.

To create the SPK containing your R binary packages, you need the SpotfireSPK package. This package is provided with your TERR installation.

Manage packages through roles

Working with packages in a deployment that includes Spotfire, Spotfire Server, and (optionally) Spotfire Statistics Services can add layers of complexity to management policies.

The job of synchronizing package versions among your development computers, your testing computers, and your servers is an important package management concern for an organization. You can reduce the risk of confusion and streamline your processes by defining roles in your organization for dealing with packages. Ensure processes and rules are established to manage packages.

Additionally, you can develop and use an in-house package repository from which all users install the same package versions. See [Package repositories](#) for additional guidance.

Developer role

The developer is an R programmer or statistician who develops packages using TERR or open-source R.

The package developer accomplishes the following tasks using the Spotfire tools.

- Develop and test packages locally using the local TERR engine available from Spotfire Analyst.
- Create a Spotfire SPK containing the R code packages using the SpotfireSPK package, and then give them to the administrator who manages Spotfire distributions on the Spotfire Server.
- Upload packages for others to use in your organization's in-house repository. See [Package repositories](#) for more information.
- With granted administrative permissions, upload the packages to Spotfire Statistics Services for users accessing the advanced analytics on the Spotfire web client. (Alternatively, give the package to the Spotfire Statistics Services administrator to upload.)

Curator role

The curator maintains the standards and lists of officially-sanctioned packages. The curator keeps all of the package versions synchronized. The curator might be the same person who fills the developer role.

The approval process for adding a packaging is up to your organization, and might vary from minimal to extensive, depending on your usual practices. Designate a developer familiar with open-source R and TERR packages to be the package curator. The package curator works with package developers and server administrators to perform the following management tasks.

- Maintains the list of tested and sanctioned package versions (the gold standard), which would be the set of packages available for general use under Spotfire applications.
- Ensures that the SPK containing the gold standard package versions are placed on the Spotfire Server, and then distributed to Spotfire analysts who develop visualizations that use them.
- Creates and manages the organization's in-house repository from where all users can install packages.
- Ensures that the gold standard package versions are uploaded to Spotfire Statistics Services for use by Spotfire web client users.



Package versions used in these two services must be kept synchronized.

Administrator role

The Spotfire administrator manages packages on the Spotfire Server and on Spotfire Statistics Services. The responsibilities for the administrator role include the following.

- Deploy the SPK to be distributed to Spotfire Analyst users.
- Assign licenses for access to the Data Functions feature in Spotfire Analyst.
- Upload, maintain, and remove packages using the TERR console on Spotfire Statistics Services. (Might assign server permissions to the curator for this task.)

Package installation locations and recommendations for updating

You can use packages installed with TERR and with TERR in Spotfire Analyst, you can create and share your own packages, and you can download packages from a package repository such as CRAN. In all cases, you should know where they are installed and how they behave when you update your TERR or Spotfire installation.



Due to changes in open-source R version 3.5 and resulting compatibility changes in TERR 5.0, packages that are built with a version of TERR prior to 5.0 must be rebuilt.

- To install a binary package from a repository, always call `install.packages(pkgname)` from TERR. The `install.packages` function finds the correct binary version in the repository for your version of TERR. Manually downloading the binary package from CRAN can result in errors when you use it with TERR.
- To install a package from source, try installing it first with TERR (with `install.packages` in TERR or with `TERR CMD INSTALL` from a command line).
- To install a package from source that you cannot build with TERR, install the package with the version of open-source R tested with TERR.

This topic details where packages are installed by default with TERR, for those that you download and install from a package repository using the TERR console, those that you install using TERR Tools in Spotfire Analyst, and for those that you install from an SPK distributed by Spotfire Server.



From either TERR Developer Edition or from Spotfire Analyst TERR Tools, you can open the console and at the prompt, run the following commands to learn more about the location of installed packages.

- Run the function `.libPaths()` to discover where TERR finds packages. This is especially useful if you do not have write access to the Program Files directory on a Windows computer.
- Run the function `installed.packages()` to retrieve a list of all packages installed on the computer, and to discover other pertinent information, including where they are installed.



In Spotfire Analyst, you can use the TERR Tools Package Management tab to see a list of installed packages.

Installed by default with the TERR console application or Developer Edition

Default Installation location	Description	Updating to a new version of TERR
<i>TERR_HOME/library</i>	Do not remove or change these packages. Doing so can cause unexpected behavior.	When you install a new version of TERR, the old <code>library</code> directory is removed and the packages are deleted. Updated versions of the packages are installed with the new version of TERR.

Installed by default with TERR in Spotfire Analyst

Default Installation location	Description	Updating to a new version of TERR
<i>SPOTFIRE_HOME/Modules/TERR_version/engine/library</i>	Do not remove or change these packages. Doing so can cause unexpected behavior.	When you install a new version of Spotfire Analyst, the old TERR <code>library</code> directory is removed and the packages are deleted. Updated versions of the TERR packages are installed with the new version of Spotfire Analyst.

Installed from a package repository using the console application or Developer Edition

Default Installation location	Description	Updating to a new version of TERR
<i>TERR_HOME/site-library</i>	<p>Packages downloaded from a repository are placed in this directory. On Windows, you must have write access to <i>TERR_HOME</i> for them to be installed at this location.</p> <p>If you do not have write access, packages you download are installed in the user directory. On Windows, this directory is [My Documents]/TERR/x86_64-pc-windows-library/<i>version</i>.</p>	<p>When you install a new version of TERR, the path to the older installation <i>TERR_HOME/site-library</i> is retained. You can take one of the following two steps.</p> <ul style="list-style-type: none"> • Browse to the directory <i>site-library</i> for the older installation, and move the packages to the new installation directory <i>TERR_HOME/site-library</i>. (If you do not have write access, manage your packages by copying them from the older <i>version</i> to the new <i>version</i> in the user directory location.) • Download and reinstall the packages.

For more information, see [Manage your packages when you install a new version of TIBCO Enterprise Runtime for R](#).

Downloaded from CRAN and installed using TERR Tools in Spotfire Analyst

Default Installation location	Description	Updating to a new version of Spotfire
[My Documents]/TERR/x86_64-pc-windows-library/ <i>version</i>	In all cases, packages downloaded from the CRAN repository using TERR Tools are placed in this directory.	<p>When you install a new version of Spotfire Analyst, the path to the user library is retained.</p> <ul style="list-style-type: none"> • Browse to the <i>version</i> directory for the older installation and move the packages to the new installation <i>version</i> directory . • Use TERR Tools to download and reinstall the packages.

For more information, see [Manage packages using TIBCO Spotfire and TIBCO Enterprise Runtime for R](#).

Installed by an SPK distributed in an update by Spotfire Server

Default Installation location	Description	Updating to a new version of Spotfire Analyst
<i>SPOTFIRE_HOME/Modules/TERR Packages/libraryversion</i>	Custom packages provided to you through an update when you connect to Spotfire Server are placed in this directory.	The packages provided to you through an update by Spotfire Server should be reinstalled by the Spotfire Server.

For more information, see [The Spofire SPK](#).

Installed on Spotfire Statistics Services

Default Installation location	Description	Updating to a new version of Spotfire Analyst
<i>SPSERVER_HOME/data/appdata/library</i> for a single server, or <i>SPSERVER_SHARE/data/appdata/library</i> for a cluster.	Packages that are uploaded to Spotfire Statistics Services for use by any client connecting to the Spotfire Statistics Services server. Your server administrator is responsible for setting the package location property, if necessary.	Your server administrator should preserve the data directory during updating.

Setting JAVA_HOME


Some packages that you use with TERR require access to Java on your system. If you call the TERR function `Sys.getenv("JAVA_HOME")` and it returns an empty string, you must set *JAVA_HOME* so the packages can access Java.

Perform this task on your Windows Or Linux system.

Prerequisites

The following list describes a few of the packages that are either provided with TERR or that you can use with TERR, but they require a bit-matching 32-bit or 64-bit version of Java, version 6 or later. (You might find other packages that require Java. These instructions can help you prepare your TERR session for those packages, too.)

Package name	Provided in your TERR installation
parallel	yes
sjdbc	yes
terrJava	yes

Package name	Provided in your TERR installation
rJava	no
	 See Installing the rJava package for more information.

Procedure

1. Locate your Java installation and make a note of it.



Your system can have more than one version of Java. Generally, use the latest version. On Windows, you can find this path in the registry. On Linux, you can usually find a link to it in the `\user\bin` directory.

For example, on Windows, this path might be `C:/Program Files/jdk-11.0.1`.

2. Start a session of the TERR console.
3. At the TERR command prompt, type the command `Sys.setenv(JAVA_HOME="path_to_your_Java_installation")` where *path_to_your_Java_installation* is the path you noted in Step 1.

For example, on Windows, this call might look like the following.

```
> Sys.setenv(JAVA_HOME="C:/Program Files/jdk-11.0.1")
```

On Linux, this call might look like the following.

```
> Sys.setenv(JAVA_HOME="/usr/lib/jvm/java-11-sun/")
```

Your system environment `JAVA_HOME` is now set to the specified Java installation.

4. Optional: Check the setting for `JAVA_HOME` from TERR by typing `Sys.getenv("JAVA_HOME")`. For example, on Windows, it might look like the following.

```
> Sys.getenv("JAVA_HOME")
[1] "C:/Program Files/jdk-11.0.1"
```

What to do next

Install the package that requires setting `JAVA_HOME`. For an example, see [Installing the rJava package](#).

Installing the rJava package

The rJava package gives access to low-level R functions to the Java interface, but it is not provided with TERR. These instructions help you prepare your computer to use rJava.

Prerequisites

The rJava package requires the following.

- A bit-matching 32-bit or 64-bit version of Java, version 6 or later, is installed. (Tested with version 11.0.1.)
- The system variable `JAVA_HOME` is set. Follow the instructions for [Setting JAVA_HOME](#) if you are unsure.

These instructions are for installing the rJava package for use with TERR 4.2 or later. If you are using an earlier version, and you cannot update your version of TERR, see the release notes for more information for the version of TERR you are running.

- For TERR version 3.1 and earlier, the rJava package does not work. To use rJava, update your version of TERR.
- For TERR version 3.2, you must use a build of rJava from TRAN. See that version's release notes for more information.

Perform this task in the TERR console or in TERR running under RStudio.

Procedure

1. At the command prompt, type `install.packages("rJava")`.
Note that `rJava` is installed from CRAN to the `site-library` directory.

2. At the command prompt, type `library(rJava)`.

For example:

```
> library(rJava)
The following object(s) are masked _from_ 'package:utils':
  head, str, tail
The following object(s) are masked _from_ 'package:methods':
  new, show
The following object(s) are masked _from_ 'base':
  anyDuplicated, duplicated, rev, sort, unique
```

The `rJava` package is now in your search path.

3. At the command prompt, type `searchpaths()`.

For example:

```
> searchpaths()
[1] ".GlobalEnv"
[2] "C:/Program Files/TIBCO/terr50/site-library/rJava"
[3] "C:/Program Files/TIBCO/terr50/library/stats"
[4] "C:/Program Files/TIBCO/terr50/library/graphics"
[5] "C:/Program Files/TIBCO/terr50/library/grDevices"
[6] "C:/Program Files/TIBCO/terr50/library/utils"
[7] "C:/Program Files/TIBCO/terr50/library/methods"
[8] "C:/Program Files/TIBCO/terr50/library/base"
```

The absolute file path is returned for each package in the current environment. Note that by default, the newly-loaded `rJava` is listed second in the search path.

Manage your packages when you install a new version of TERR

When a new version of TERR is released, you might want to install it to take advantage of the changes. You can run older and newer versions of TERR on the same computer, or you can uninstall the older version(s). In either case, you probably want to make sure any custom-created packages or packages downloaded from a repository are kept available to the TERR version(s) you are running.



This topic refers to the TERR Developer Edition, a stand-alone version that you can use from the console or in RStudio. Managing packages you use with updates to Spotfire or Spotfire Statistics Services follow a different process and are installed in different locations.



Uninstalling TERR does not remove the packages you installed. However, we recommend that you check for updates to any packages you have downloaded from package repositories such as TRAN or CRAN after you install a new version of TERR. You can check for updated versions by calling `update.packages()`. By default, this function checks the TRAN and CRAN repositories. See the help topic for `update.packages()` in the *TERR Language Reference* for more information.

As of TERR version 3.2, the TERR installation includes the directory `TERR_HOME/site-library`, which is used by default. If you want to use another directory, you can define the environment variable `TERR_LIBS_SITE` and set it to the directory of your choice.

Initially, the `site-library` directory is empty. If you have permission to write to the `TERR_HOME` directory, any packages you create or download are installed in `TERR_HOME/site-library`.

Installing packages to the `site-library` directory provides the following advantages.

- It provides you with the means to protect and manage the packages you installed and want to keep, separate from the new installation.

- It separates the packages shipped with TERR so they can be updated with new releases, and so you do not accidentally change or remove them.



You should avoid changing any entries in the `TERR_HOME/library` directory. Doing so can cause TERR to behave in unexpected ways.

The directory `TERR_HOME/site-library` is added to the head of the search path, which is returned by `.libPaths()`. For example, on a Windows computer where you have permission to write to the `TERR_HOME` directory, this function call would appear as follows.

```
> .libPaths()
[1] "C:/Program Files/TIBCO/terr50/site-library"
[2] "C:/Program Files/TIBCO/terr50/library"
```

After installing the new version of TERR, you can just copy the packages from the older `TERR_HOME/site-library` directory to the new `TERR_HOME/site-library` directory.

If you are downloading packages to a computer where you do not have permission to write to the directory `TERR_HOME/site-library`, then packages are stored in the user directory. For example, on Windows, this directory is `[My Documents]/TERR/x86_64-pc-windows-library/<version>`, and, calling `.libPaths()` would appear as follows.



```
> .libPaths()
[1] "C:/users/jdoe/Documents/TERR/x86_64-pc-windows-library/terr5.0"
[2] "C:/Program Files/TIBCO/terr50/site-library"
[3] "C:/Program Files/TIBCO/terr50/library"
```

In this case, you can ignore the `site-library` directory (which remains empty) and manage your packages by copying them from the older `<version>` to the new `<version>` in the user directory location.

Manage packages using Spotfire and TERR

Analysts and data scientists can download existing packages, or they can build and test their own packages. They can wrap them in a Spotfire SPK for distribution to analysts in their organization. Spotfire Server administrators can distribute the packages using Spotfire Server, and Spotfire web client users can view Spotfire analyses that use the packages.



Because this documentation specifically addresses using TERR in Spotfire Analyst, the instructions demonstrate using the TERR tools available from the Spotfire Analyst user interface. Experienced R or TERR programmers, or programmers using TERR from the stand-alone console can perform many of these tasks directly from the console or from within RStudio. See the *TERR Technical Guide* and *Language Reference* for more information.

For the following tasks, Spotfire products each play a role in building, distributing, using, and maintaining packages in the Spotfire predictive analytics platform.

Package management tasks and tools

I want to...	Available tools
Discover the packages available in TERR.	TERR in Spotfire Analyst and the TERR stand-alone console.
Review the differences between TERR and open-source R.	TERR in Spotfire Analyst and the TERR stand-alone console.
Review the packages on CRAN that have been tested with TERR.	TERR in Spotfire Analyst and the TERR stand-alone console.

I want to...	Available tools
Find and install a package from a repository, such as CRAN.	TERR in Spotfire Analyst and the TERR stand-alone console.
Create an SPK so the package can be shared with other Spotfire analysts and data scientists in the organization.	TERR in Spotfire Analyst and the TERR stand-alone console.
Remove a package from a Spotfire installation.	TERR in Spotfire Analyst
Change the version number for a package distributed to Spotfire analysts and data scientists.	Spotfire Server
Distribute a package to all Spotfire analysts and data scientists in your organization.	Spotfire Server
Distribute a package to a small group of Spotfire analysts and data scientists for testing.	Spotfire Server
Provide access to a package for Spotfire web client users who want to view data visualizations that rely on that package.	Spotfire Statistics Services

Development tools for creating packages

Spotfire Analyst provides both a console application and access to an installation of RStudio so experienced R developers can create and test packages to use in Spotfire analyses.

Both the TERR console application and RStudio access are available from the Spotfire menu **Tools > TERR Tools**. Also you can find links to all of the TERR documentation from the **TERR Tools** dialog.

Before you develop your package, review the limitations and considerations, the list of components that make up a package, and the guidance for testing the package

Limitations and considerations

Before you begin developing packages to use with TERR, you should familiarize yourself with some basic limitations.

Platform considerations

You can develop or download open-source R packages to run on either LINUX® or Microsoft Windows® platforms. Therefore, when you write your code to run on the server, take into account the platform you expect it to run on.

- If you plan to run packages locally on a TERR engine in Spotfire Analyst, your package must be a Windows binary package.
- If you plan to deploy a package to run TERR in Spotfire Statistics Services, the package type must be binary, and it must match the platform of the Spotfire Statistics Services deployment.

Graphical limitations

The TERR engine provides no graphical functions itself. However, it can run as a statistical engine in Spotfire Analyst, which provides data visualizations. Alternatively, you can use some CRAN packages (such as `htmlwidgets`, `dygraph`, or `leaflet`) that are compatible with TERR and that provide graphical

displays in the browser, or you can use the RinR package, provided with your TERR installation, to call open-source R functions for graphic displays. You should always test package functions to make sure they run as expected. See "Graphics in TIBCO Enterprise Runtime for R" in the *TIBCO Enterprise Runtime for R Technical Documentation* for more information.

Licensing limitations

To make a data function available in an analysis published to a Spotfire library, you must have the appropriate licenses. If you are not sure of your licenses, or if you do not see the **Tools > Register Data Functions** menu in the Spotfire Analyst installation, see your Spotfire Administrator for more information.

R package anatomy

Open-source R Packages running in the TERR engine are binary, and they must follow the standard package component design and contain version information.

The TERR engine is designed to be highly compatible with the open-source R engine. To develop packages using open-source R, see its documentation, available on the Comprehensive R Archive Network (CRAN).



Open-source R is available under separate open source software license terms and is not part of TERR. As such, open-source R is not within the scope of your license for TERR. Open-source R is not supported, maintained, or warranted in any way by TIBCO Software Inc. Download and use of open-source R is solely at your own discretion and subject to the free open source license terms applicable to open-source R.

A typical package is available as a binary file or as source code. You can use TERR to install a package from source. If the package has C/C++ code, you must first install the package `rinclude` (`install.packages(rinclude)`) before calling `install.packages(packagename)`. TERR does not support installing packages that have Java source code. See [Installation options for packages](#) on page 24 for more information.

Package components

A source package can contain any number of directories, including `html` (for the help index), `libs`, `help`, and so on. The simplest package requires the following files and directories:

File or Directory	Description
<code>mypkg</code>	The top-level directory name, which is also the package name (in this case, <code>mypkg</code>).
<code>mypkg/NAMESPACE</code>	Required. You must specify the NAMESPACE.
<code>mypkg/DESCRIPTION</code>	The file containing a description, the title, the author, date, the dialect, and version information, along with other information.
<code>mypkg/R</code>	The directory containing *.R files with R language functions as ASCII files.
<code>mypkg/R/mycode.R</code>	The R language code.

Your source package can also contain the following optional folders:

- `data` directory containing data files in a dump format.
- `man` directory containing help files in the `.Rd` help file format.

- `inst` directory contains files and directories to be copied, recursively, into the main package directory when the package is compiled. Any informational files that the end user should see should be included in the `inst` directory. For example, if you have a PDF containing a vignette, you can include it in the `inst/doc` directory.
- `src` directory, containing C, C++, or FORTRAN code.
- `tests` directory can contain package-specific tests. This directory can contain test code (that is, `.S`, `.ssc`, `.q`, and `.R`).



The TERR engine does not support the following:

- Packages using graphics devices or containing graphics functions. (However; we have implemented stub functions to allow the non-graphical portions of many packages to run without error.)
- Source packages with `src` directories that contain Java code.

Package versioning

Package version information is kept in the `DESCRIPTION` file. With every package revision, remember to revise the version number upward. This version number is an important part of your package management strategy.

Packages running in a local TIBCO Enterprise Runtime for R engine in Spotfire

Running functions locally requires making the TERR engine available under Spotfire.

To run the functions in a package using the TERR engine that is installed with Spotfire Analyst, you must have the appropriate license enabled by your Spotfire administrator.

Remember that other people in your organization might be running analyses that depend on data functions stored in the library and that take advantage of the version of the package you are using.

In Spotfire, these scenarios are possible because Spotfire Analyst includes an embedded TERR engine. With enough resources, you can efficiently test and run analyses using data functions locally.



Although the package is in your Spotfire installation, it is not loaded into the engine. To use the package, you must load the library as part of your data function script or include it in the `Packages` field of your data function.

Opening the TIBCO Enterprise Runtime for R console from Spotfire Analyst

TERR is provided in your installation of Spotfire Analyst so you can script and run data functions or create predictive models. TERR Tools is provided to give you access to the TERR console to test scripts and functions, to launch the RStudio interactive development environment for script authoring, and to the TERR Language Reference for help with installed packages. TERR Tools also provides an interface to download and install packages from the Comprehensive R Archive Network (CRAN). Perform this task from Spotfire Analyst.

Prerequisites

To work with TERR, you must have the appropriate license in Spotfire Analyst.

Procedure

1. From the menu, click **Tools > TERR Tools**.
2. Click **Launch TERR Console**.
The TERR engine console that is included in the Spotfire Analyst `Modules` folder is displayed in a separate window.

Checking installed packages

You can find a list of packages included in the installation of TERR.

Check the packages that are installed and available to the TERR engine from TERR Tools in Spotfire Analyst.

Procedure

1. From the menu, click **Tools > TERR Tools**.
2. Click **Open TERR Language Reference**.



The language reference is included in the installation. The links to the technical guides and readme files open these documents on the TIBCO documentation website <https://docs.tibco.com/products/tibco-enterprise-runtime-for-r>.

A web browser launches and displays the landing page with links to the documentation for TERR.

3. In the resulting browser window, under **References**, click **Included Packages**.

Packages for use with Spotfire

TERR includes a set of packages specifically for working with Spotfire.

Package Name	Description
Spotfire	This package is deprecated. See SpotfireData.
SpotfireConnector	This package contains functions used between Spotfire and Spotfire Statistics Services. You would probably not call these functions directly.
SpotfireData	This package contains functions for managing datasets between Spotfire and TERR. (In particular, the functions are for reading and writing files in the Spotfire Binary Data format (SBDF).) Advanced users might consider importing their Spotfire data directly in the TERR engine for debugging purposes.
SpotfireSPK	<p>This package contains two functions:</p> <ul style="list-style-type: none"> • buildSPK • buildServerSPK <p>Use this package to create an SPK to contain your R language packages that you want distributed to other Spotfire Analyst users or on Spotfire Server. (The packages in your SPK can be automatically distributed to Spotfire Analyst users who are working with data functions with local TERR engines.)</p>
SpotfireStats	This package contains the helper functions used by Spotfire predictive analytics. You would probably not call these functions directly.
SpotfireUtils	This package contains utility functions for interfacing between Spotfire and the TERR engine.

Package repositories

You can find or store packages in a variety of repositories. You can create your own in-house repository using the drat package.

You can download a package from a public repository, such as CRAN or TERR Archive Network (TRAN) repository, or you could use the drat package to create and maintain an in-house repository that is managed by your organization's [package curator](#).



TIBCO does not warrant, deliver, or support code or other material provided by the R Project for Statistical Computing, including but not limited to development tools and packages, and such code and other material does not constitute a part of TERR. Such material therefore is not within the scope of your license for TERR. Download and use of such material is solely at your own discretion and subject to the free open source license terms applicable to such material. TIBCO recommends that you consult a legal professional concerning compliance with any free open source license terms applicable to such material, particularly if you plan to engage in redistribution of TERR and/or such material. (Please note that TERR may be redistributed solely pursuant to a license that expressly grants such redistribution rights.)

Public repositories

You can download open-source packages from a repository, and then use the functions in the package in your data functions in Spotfire. Be sure to review the list of differences between the TERR engine, available from the TERR landing page. Also review the [limitations and considerations](#).

TRAN packages

A number of popular CRAN packages occasionally require customization to work with TERR. TIBCO hosts a package network, TRAN, where you can find these customized packages. Individual users can download these packages using the TERR console application by calling the function `install.packages()`. Optionally, these customized packages can be built into a Spotfire SPK and placed on a Spotfire Server to be available to Spotfire users throughout your organization.



Some packages customized and placed on TRAN require other packages not available on TRAN. Some of these packages cannot be installed using the TERR function `install.packages`, so the TRAN package cannot be successfully installed. If you encounter this situation, try building and installing the package using open-source R.

Company repositories

One skilled package developer in your organization should have the role of package curator to oversee the package version integrity within the company's package repository, whether these are packages downloaded and tested from a public repository or developed internally and kept in a CRAN-like in-house repository.


One easy way to establish and maintain a company repository is with the drat package, which is available in the CRAN repository. Using the drat package, you can create an in-house repository (on either a web server or a network share), you can upload packages to the in-house repository, and you can install packages from the in-house repository. For more information on using drat, install it into your TERR or open-source R console application, and then refer to its language reference and vignettes.

For more information about installing existing packages, see [Installation options for packages](#).

Installation options for packages


Use the function `install.packages()` to install packages to use in TERR either in the stand-alone console, or in Spotfire. You can find packages in a variety of locations, including repositories, on reliable

web sites, or stored locally. See the TERR help topic for `install.packages()` for more detail and examples.

Package Location	Description	Example
TERR Archive Network (TRAN) repository	<p>The default location, <code>https://tran.tibco.com</code>, for installing packages using <code>install.packages()</code>. Used for packages that have been customized to work specifically with TERR. Requires no further arguments.</p> <div>  <p>Some packages customized and placed on TRAN require other packages not available on TRAN. Some of these packages cannot be installed using the TERR function <code>install.packages</code>, so the TRAN package cannot be successfully installed. If you encounter this situation, try building and installing the package using open-source R.</p> </div> <p>For information on the installation differences between TERR and open-source R, see Specifying an older package from TRAN.</p>	<pre># install R Datasets Package found on TRAN: install.packages("datasets")</pre>
Comprehensive R Archive Network (CRAN) repository	The secondary location for installing packages using <code>install.packages()</code> . Requires no further arguments. (See the following note for important information about installing source packages from CRAN.)	<pre># install rpart package from CRAN: install.packages("rpart")</pre>
In-house repository	You or someone in your organization has set up a CRAN-like repository (either on a network share or on a web server) using a tool like the drat package. All TERR or open-source R users in the organization can access the same package version from the repository.	<pre>#download and install from a # local web service using a URL: URL <- "https:// mycompanysvc/mypackage" install.packages(URL)</pre>
Locally-available packages	If a trusted source gives you a package as a zip archive, you can put it in on your computer and install it using <code>install.packages()</code> .	<pre># install local newtree package # from a zip file in the working directory: install.packages("newtree_1. 2.zip")</pre>

Package Location	Description	Example
Trusted URL	If you are given a URL that contains a package you might want to use, and you trust the URL, you can pass the URL as the only argument to <code>install.packages()</code> .	<pre>#download from a custom URL and install # a custom package URL <- "https://customurl/ mypackage") install.packages(URL)</pre>

The CRAN repository contains binary packages (for Windows) and source packages (for Linux and Windows). You can easily install most binary and source packages in TERR. If you have problems building from source, then build the packages using open-source R before installing them into TERR. Note that TERR does not build binary packages from source packages that contain Java source code.

Platform	Package type	Notes
Linux, Windows	Binary	Call <code>install.packages(pkgname)</code> . TERR installs the binary package into your specified package directory.
Linux	Source; no Java code, no C/C++ or Fortran code	Call <code>install.packages(pkgname)</code> . TERR builds the source package into a binary package and installs it into your specified package directory.
Linux, Windows	Source; C/C++ or Fortran code (no Java code)	<div>  <p>On Windows, first you must install the Rtools utilities, which is maintained by Duncan Murdoch, from CRAN, and then update your PATH to specify the location of the utilities.</p> <ol style="list-style-type: none"> 1. If you have not already done so, install the package <code>rinclude</code> by calling <code>install.packages(rinclude)</code> 2. Call <code>install.packages(pkgname)</code>. <p>TERR builds the source package into a binary package and installs it into your specified package directory.</p> <p>If the package does not build and install, then try building it with open-source R, and then installing the binary as described here.</p> </div>
Linux	Source; Java code	<ol style="list-style-type: none"> 1. Build the package using open-source R tools for building packages from source. The tools compile the source code to create the binary package. 2. Call <code>install.packages(pkgname)</code>.

See the help for `install.packages(pkgname)` for more information.

Due to changes in open-source R version 3.5 and resulting compatibility changes in TERR 5.0, packages that are built with a version of TERR prior to 5.0 must be rebuilt.

- To install a binary package from a repository, always call `install.packages(pkgname)` from TERR. The `install.packages` function finds the correct binary version in the repository for your version of TERR. Manually downloading the binary package from CRAN can result in errors when you use it with TERR.
- To install a package from source, try installing it first with TERR (with `install.packages` in TERR or with `TERR CMD INSTALL` from a command line).

- To install a package from source that you cannot build with TERR, install the package with the version of open-source R tested with TERR.

To get more information about the packages on TRAN, run the following code in TERR:

```
ap <- available.packages(contrib.url(getOption("repos")[1],
  getOption("pkgType")))
# to print the entire matrix
ap
# to print just the package names
row.names(ap)
```

Specifying an older package from TRAN

TERR and open-source R search for packages differently. By default, the repository search `{{options()}$repos}}` is set to `{{c("https://tran.tibco.com/terr##", "https://cloud.r-project.org")}}`. If a package exists in both repositories, then TERR selects the version in TRAN, but open-source R selects the newest one based on the package version number.

This search-order difference is by design, because if a newer package on CRAN causes problems when tested with TERR, TRAN contains an older version of a package that has been tested successfully with TERR.

In some cases, you must use open-source R to install a package to use with TERR. For example, under Linux, packages with source code for use in TERR often need to be installed using open-source R. If you encounter the search-difference issue with such a package, and an older version is available on TRAN, then you must take additional steps to make sure you get the working package version.

Procedure

1. From open-source R, run `install.packages(pkgname)`.
The newest version of the package specified by `pkgname` is installed, along with its dependencies.
2. Set `options()$repos` to `c("https://tran.tibco.com/terr##")`.
The repository search option is set to check only the TRAN site and the version of TERR specified by the version number `##`.
3. Reinstall the needed package.
The package is installed according to the TRAN site search option.

Testing packages locally

After you have either created a package or downloaded a package from a repository such as the CRAN, you can test the package functions by running the example files in the TERR engine. Then you can write your own scripts using the package functions, and test them using Spotfire Analyst.

Prerequisites

Your Spotfire Analyst installation should be configured to use the local TERR engine. You can check this option from the Spotfire Analyst menu by clicking **Tools > Options**. In the Options dialog, click **Data Functions**, and make sure **Use locally-installed TIBCO Enterprise Runtime for R engine** is selected.

Perform this task from your Spotfire Analyst installation.

Procedure

1. From the menu, click **Tools > TERR Tools**.
2. Click **Launch TERR Console**.
The TERR engine console that is included in the Spotfire Analyst Modules folder is displayed in a separate window.

3. At the prompt, type `library(packagename)` to load the package.
The package is loaded to the package directory.
4. Type or paste the script or example using the package functions that you want to run.

Testing a CRAN package example

```
install.packages("survival")
library(survival)
Surv(heart$start, heart$stop, heart$event)
```

Getting package help

Each package you download or build should have help files. If you are writing data functions or developing analyses that use the functions in a package, you might want to see Help topics associated with them.

Procedure

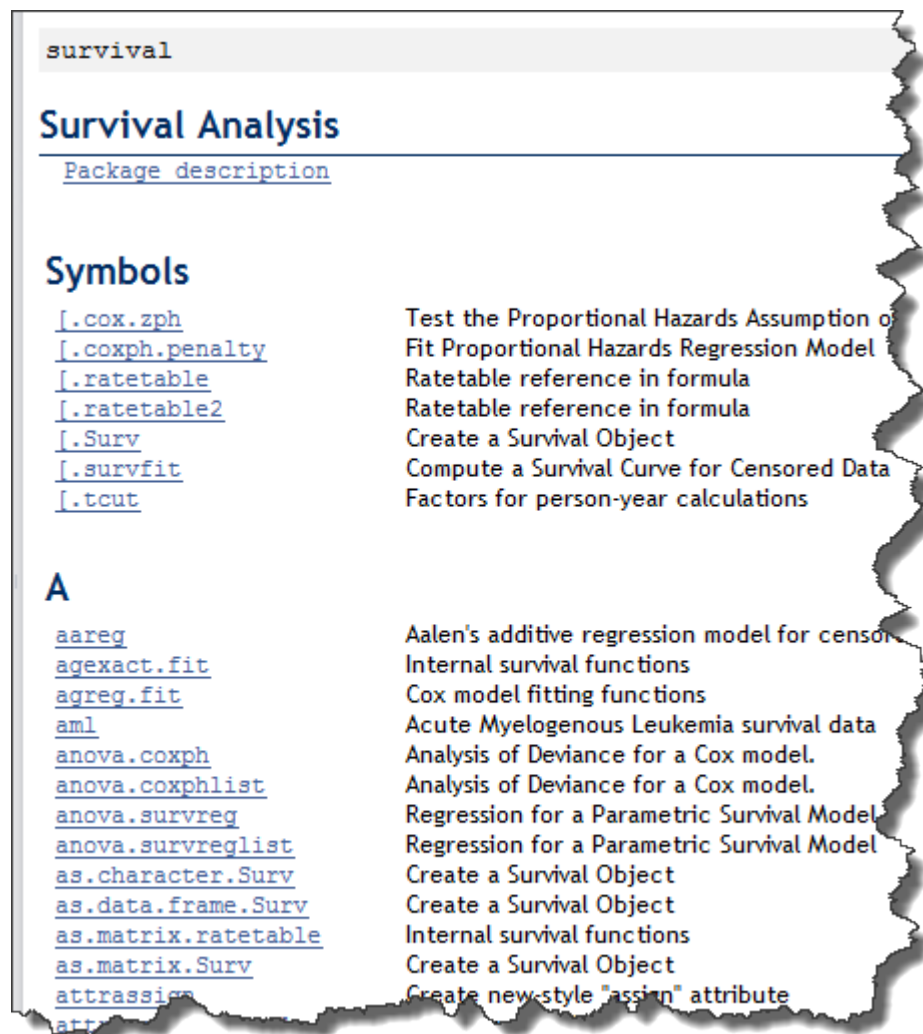
1. In Spotfire, open a sample DXP file.
2. From the menu, click **Tools > TERR Tools**.
3. Click **Open TERR Language Reference**.



The language reference is included in the installation. The links to the technical guides and readme files open these documents on the TIBCO documentation website <https://docs.tibco.com/products/tibco-enterprise-runtime-for-r>.

A web browser launches and displays the landing page with links to the documentation for TERR.

4. Under **Reference**, click **Included Packages**.
5. In the resulting **Available Packages** page, note that your package is listed. Click its name.
6. From the resulting page, select the function for which you want help.



Removing a package from a Spotfire installation

You might decide that you no longer need a package in your Spotfire installation. Perform this task using Spotfire Analyst.



Take care to remove only packages that you have downloaded and installed. If you remove a package that is installed with TERR, you could cause serious problems with the installation. Likewise, if you remove a package that was distributed by the Spotfire Server, any scripts or data functions that use the package will cease to work, and you will not be prompted to download and install it by the Spotfire Server distribution mechanism. See [Troubleshooting TERR and Spotfire packages](#) for more information.

Procedure

1. From the menu, click **Tools > TERR Tools**.
2. In the TERR Tools dialog, click the tab **Package Management**.
If you are working through a proxy, select the check box **Use IE Proxy Settings**.
3. Review the list under **Installed Packages**.
4. Select the package to remove, and then click **Remove**.
The package is removed from the list and from your installation.

The Spotfire SPK

A Spotfire SPK is usually created and tested by developers to package and deploy third-party extensions to the Spotfire Server, which can then be distributed to Spotfire Analyst users.



Even though they are both called “packages”, the Spotfire package (SPK) and the R package are different. The SPK is a means to deploy extensions to the Spotfire Server, which then distributes its contents to Spotfire Analyst users. We use the Spotfire SPK deployment mechanism to distribute R packages to Spotfire Analyst users who want to use the R package functionality to write data functions used by Spotfire data visualizations.

TERR contains an R package with one function (`buildSPK`) that you can use to build an SPK that contains one or more R packages. See [Creating the Spotfire SPK](#) for detailed instructions.

The R packages bundled into an SPK are distributed to Spotfire Analyst users who want to write data functions and create analyses using data functions.

This documentation does not cover the Spotfire SPK in general, only those built using the SpotfireSPK package and containing R packages. For more general information about extending Spotfire using the SPK, see the Spotfire SDK documentation available at docs.tibco.com.

Package workflow

Using packages with the TERR engine, Spotfire, and Spotfire Statistics Services is a two-part process.

1. [Stage the packages](#) (including developing or downloading, testing, and deploying).
2. [Distribute and use the packages](#).

Stage the packages

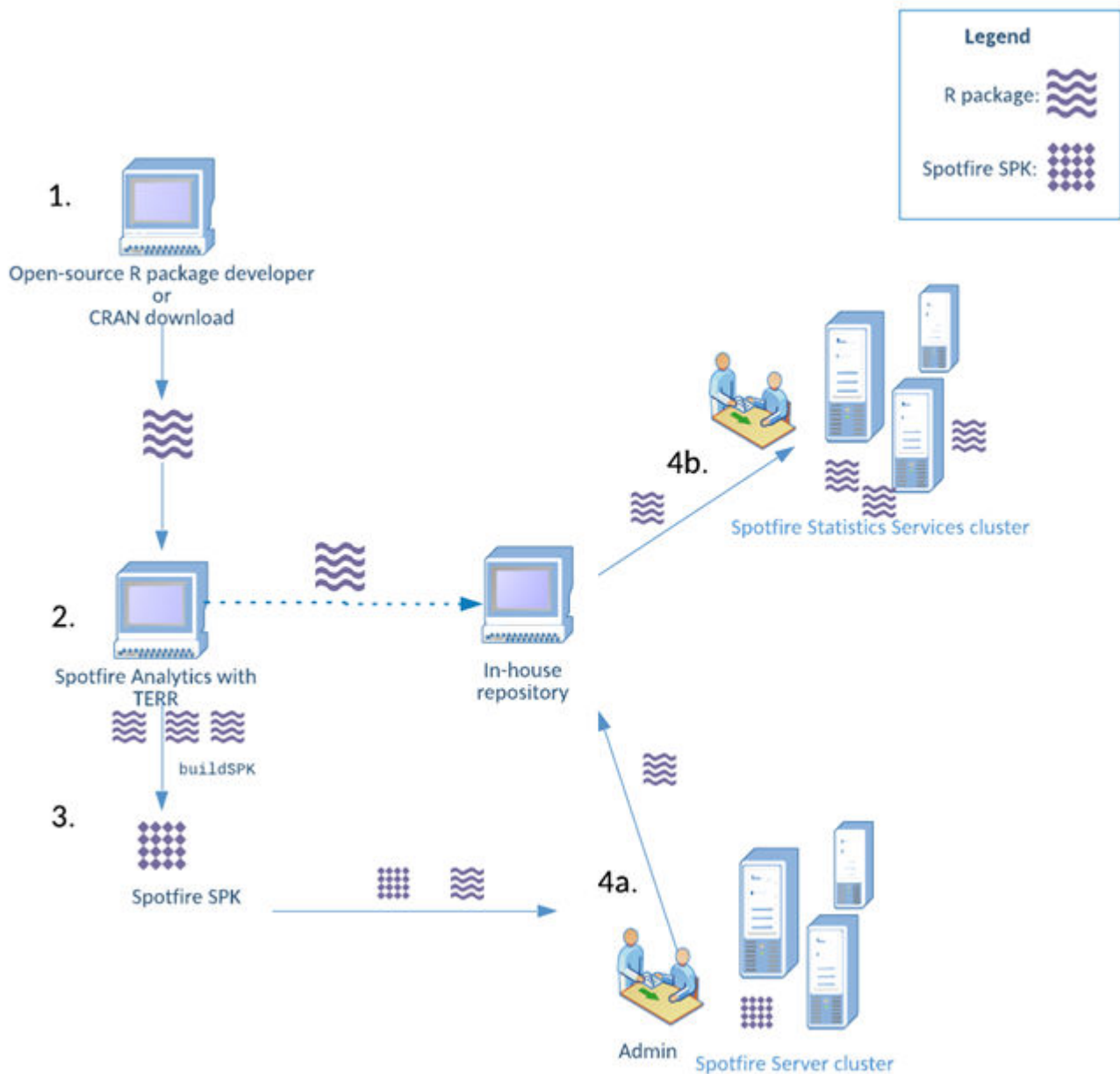
Displayed here is a high-level outline of the processes of staging packages, including deploying them to a Spotfire Server package repository and, optionally, to a Spotfire Statistics Services package repository.

To stage and deploy a package in the Spotfire Server cluster and optionally the Spotfire Statistics Services cluster, follow the pictured workflow. These steps describe the numbers in the image.

1. From your local TERR console, produce or download the R language package that you want to deploy and use. (Open the console from the Spotfire Analyst installation, from the **Tools > TERR Tools** menu.)
2. Using your local TERR engine, test the package.
3. From your local TERR engine, load the SpotfireSPK package, and then call the function `buildSPK` to build the SPK (passing in the DCF name, the SPK name, and any additional parameters you require). For more information, see the section [Creating the Spotfire SPK](#).
4. Hand off both the SPK and the R package to your Spotfire Server administrator. (Optionally, if you have the permissions, you can upload the R package to an in-house repository using the `drat` package.)
 - a. The administrator places the SPK on the Spotfire Server and if required, using the TERR console, copies the package to the in-house repository.
 - b. The administrator logs in to Spotfire Statistics Services and, using the TERR console, installs the R package from the in-house repository to Spotfire Statistics Services.



Optionally, the administrator can use the Eclipse plugin **TSSS Connector** to install the package to Spotfire Statistics Services. (For more information, see the section [Uploading a package using TSSS Connector](#).)



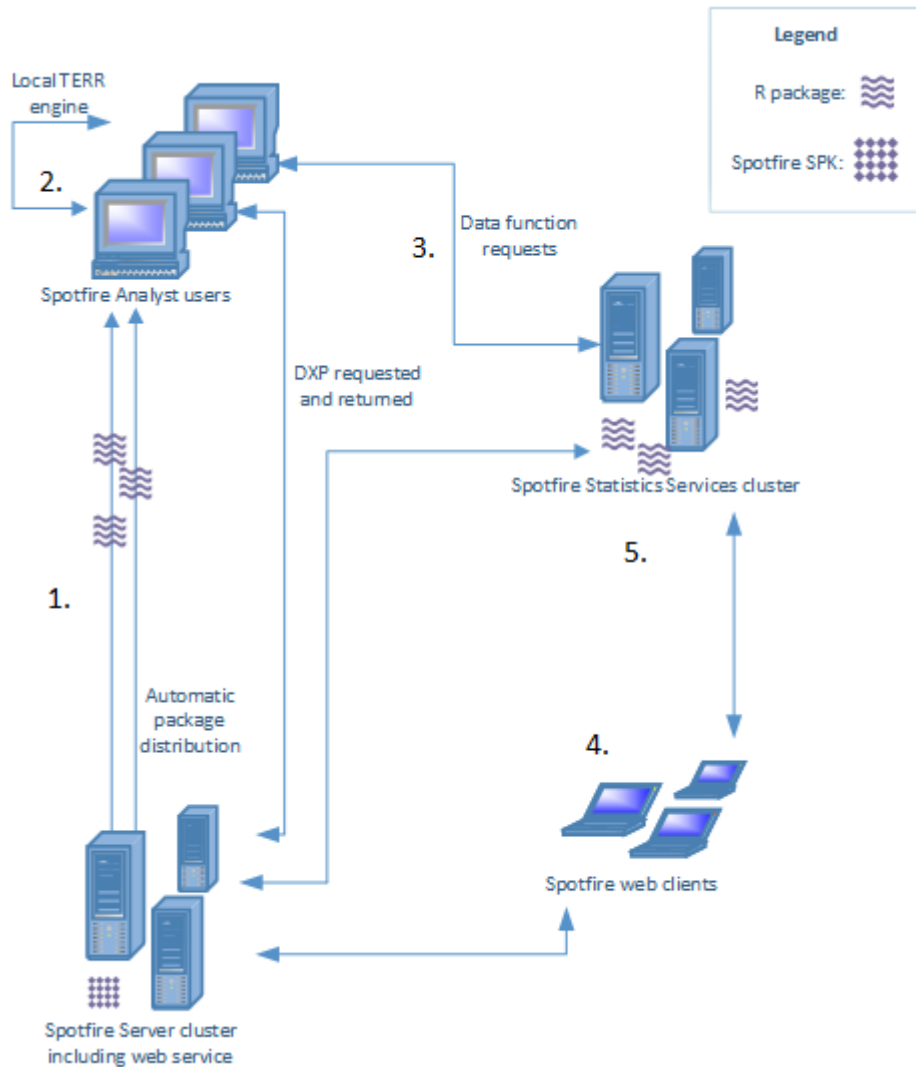
Distribute and use the packages

The following image provides a high-level picture of package distribution and use after it has been placed on TIBCO Spotfire® Server and TIBCO Spotfire® Statistics Services.

After the packages are placed on the Spotfire Server and, optionally, Spotfire Statistics Services, they are automatically distributed to Spotfire analysts for use in their advanced analytics, and they are available to Spotfire web client users who access Spotfire visualizations through a browser. These steps describe the numbers in the image.

1. When the Spotfire users launch Spotfire Analyst, they are notified by the server that a new distribution of the package is available. They accept the update.
2. If this Spotfire Analyst user has the Data Function license, he can run data functions using the packages and the local TERR engine.
3. Spotfire web client users can run TERR data functions remotely using the TERR engine on Spotfire Statistics Services.

4. Spotfire web client users access the Spotfire DXP (stored in the Spotfire Library) through the Spotfire Server.
5. If the Spotfire DXP includes data function(s), the Spotfire web client can access Spotfire Statistics Services to run the data function on the TERR engine to return the results to the Spotfire web client, and the results are rendered in the Spotfire web client browser.



Obtaining the SpotfireSPK toolset

Using the SpotfireSPK package, you can generate a Spotfire SPK containing your R or TERR packages, and then automatically distribute them to Spotfire Analyst installations in your organization. You must have installed Spotfire Analyst. You must have a license to use TERR in your Spotfire Analyst installation.

Procedure

1. From the Spotfire Analyst menu, click **Tools > TERR Tools**.
2. On the **Engine** tab, click **Launch TERR Console**.
The TERR console in the Spotfire installation is displayed.
3. At the console prompt, type `library("SpotfireSPK")`.
You can provide

The package is loaded in the TERR session, and you are now ready to build the SPK to contain the TERR-compatible binary packages.

4. Review the Help for its only function, `buildSPK`, by typing the following in at the command-line in the TERR console:

```
?buildSPK
```

This function takes several optional arguments; check with your server administrator regarding certificate and password requirements.

Creating the Spotfire SPK

Practice building an SPK that contains only one package: the survival package downloaded from the Comprehensive R Archive Network (CRAN).

Prerequisites

You must have access to the TERR engine, either through your Spotfire installation, or the stand-alone TERR console.



The survival R package is compatible with TERR; however, note that some of its examples might use the `plot` function, which is not supported in version 4.3.

When you prepare an SPK to be deployed to the Spotfire Server, remember the following rules:

- The Spotfire SPK can contain as many R packages as you need.
- Only one Spotfire SPK is allowed in a deployment area on the server; if you redeploy it, it overwrites your previously-deployed SPK.
- To distribute an updated Spotfire SPK to the Spotfire Analyst installations using the server, the Spotfire SPK must have its `BuiltVersion` incremented.

This example walks you through installing a package, generating the list, Install your package and make sure it works.

Procedure

1. Load the Spotfire SPK package.
`library(SpotfireSPK)`
2. Install a package to be in your Spotfire SPK file.
`install.packages("survival")`
The package survival is downloaded from the CRAN repository.
3. Generate the Debian Control File (DCF).
The DCF contains the package list to build into the Spotfire SPK.
`writeLines("Packages: survival", "SpotfireSPK.dcf")`



See [Spotfire SPK versioning](#) for information about creating the DCF and versioning the SPK.

4. Build the SPK.



In this example, you are building the SPK without passing any arguments for certificates or passwords, and the resulting output specifies that your resulting SPK is unsigned. If you do not include the arguments `certificate` and `password` in your `buildSPK` function, when the package is distributed, Spotfire Analyst users see a message warning of an unsigned file, and they are prompted to accept or reject the installation. This message appears for every update of the unsigned package. See your server administrator for a certificate and password to include.

```
buildSPK("SpotfireSPK.dcf", "SpotfireSPK.spk")
```

Note that "SpotfireSPK.spk" is not signed.
building SpotfireSPK.spk.
Done.

5. Print the new list file.

```
cat(readLines("SpotfireSPK.dcf"), sep="\n")
```

```
Packages: survival
Built: TERR 4.3.0; (includes date and time)
BuiltName: TIBCO Enterprise Runtime for R Packages
BuiltId: F13B9A7E-783A-432a-8676-42FCD3022D70
BuiltVersion: 1.0.0.0
BuiltPackages: survival (>=2.38-3)
```

(Your output will vary.)

6. Browse your computer for the SPK file and the DCF file.

By default, both of these files are written to the your Users\user name directory.

Remember that you can have only one SpotfireSPK package deployed to your Spotfire Server, and you can use the DCF list to keep track of the packages in the SPK. We recommend you keep this list where you can update it as necessary. See [Packages deployed for a small group](#) for more information.

Spotfire SPK versioning

You create the file SpotfireSPK.spk containing the packages you want to distribute to others using the Spotfire Server. You might need to change or update the packages that you distribute, which requires changing the version of the Spotfire SPK.

After you create the SpotfireSPK.spk, you give it to Spotfire Server administrator, who places it in a distribution area on Spotfire Server. It is used to distribute packages to other analysts who are using Spotfire Analyst. If you create a package and distribute it using the Spotfire SPK mechanism, and you need to update the package code, you must create a new SpotfireSPK.spk

You can create or change a Spotfire SPK using TERR, as described in [Creating the Spotfire package](#). The TERR function buildSPK creates the Spotfire SPK using the versioning rule details for the following tasks.



The SPK property BuiltVersion is NOT the same as the package version. That information is stored in the package DESCRIPTION file. BuiltVersion is always specified as four components (N.n.n.n).

You can learn more about the TERR function buildSPK by reading its help file. See [Obtaining the SpotfireSPK toolset](#) for more information.

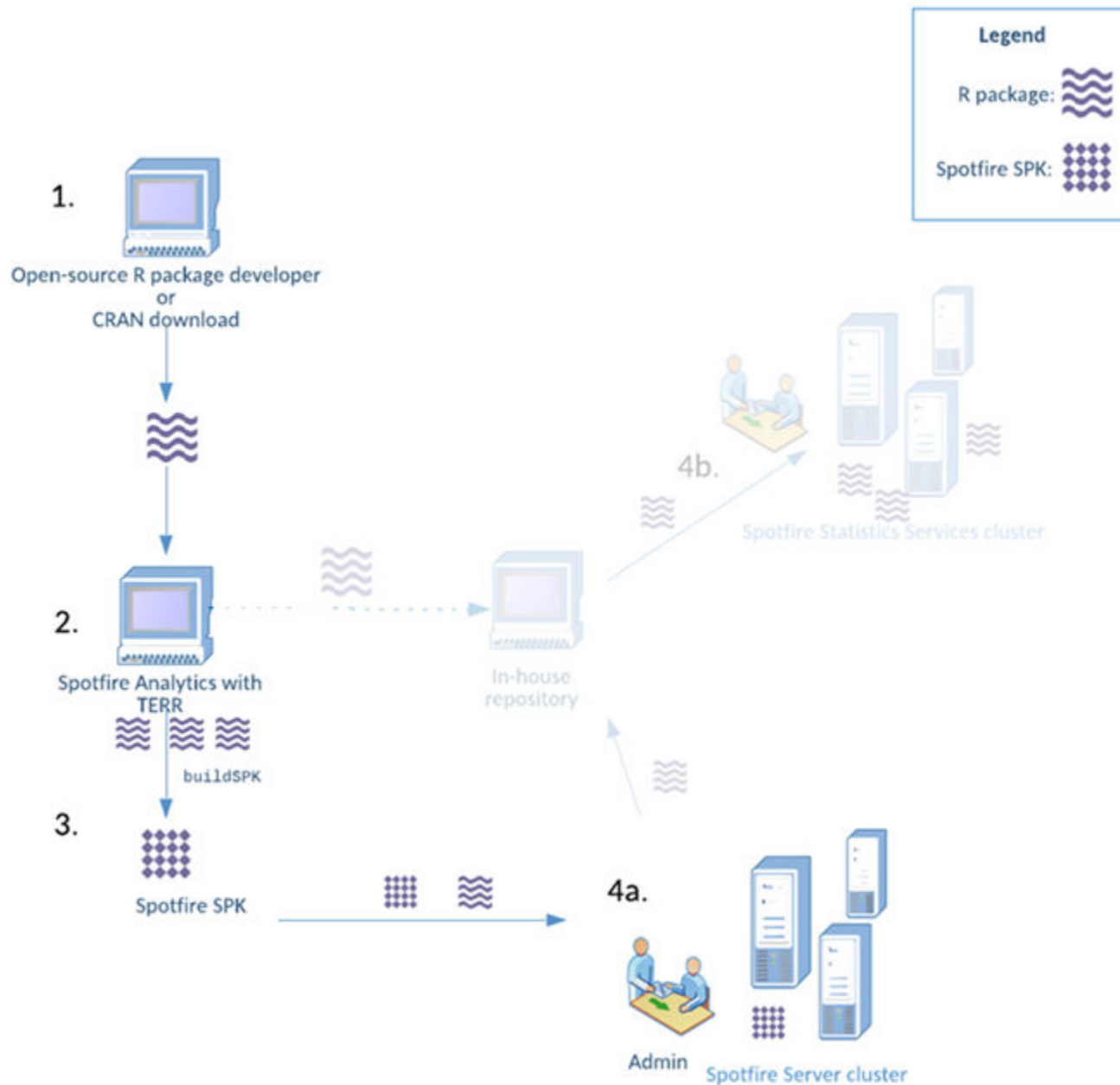
Task	Rule details
Generating a new DCF	If you generate a new DCF using writeLines or a text editor, and then you pass the new DCF to the buildSPK function, the function overwrites the old DCF, and the list contains only the packages you provide in the text argument of writeLines. The BuiltVersion is always set to 1.0.0.0.
Recreating a new DCF using the same version	If you do not need to increment or keep the older DCF, you can simply regenerate it using writeLines. Creating a new DCF always sets BuiltVersion to 1.0.0.0 by default. However, if you use this method of generating the DCF, Spotfire Server does not register the package as a new one, so it does not distribute the package to the users.

Task	Rule details
Adding package names to an existing DCF	If you only add packages to the existing DCF using a text editor, and then run the <code>buildSPK</code> function, passing in the edited DCF, the <code>buildSPK</code> function increments the <code>BuiltVersion</code> to a minor version number (for example, 1.1.0.0).
Removing package names from an existing DCF	If you remove a package from the existing DCF using a text editor, and then run the <code>buildSPK</code> function, passing in the edited DCF, the <code>buildSPK</code> function increments the <code>BuiltVersion</code> to a major version number (for example, 2.0.0.0).
Assigning a specific version number to a DCF	If you create a DCF and want to assign a specific version number to it, run the <code>buildSPK</code> function and pass in the DCF argument, along with the argument <code>version</code> , setting it to the version you want. This argument is a character string or a <code>numeric_version</code> object containing four components (for example, <code>version = "1.2.3.4"</code>).

Deploy the SPK to the Spotfire Server

The file `SpotfireSPK.spk` created by `buildSPK` is deployed to the Spotfire Server the same way any SPK is deployed.

1. Using the Administration Console tools, the Spotfire Server administrator places the SPK in the Spotfire Server distribution area.
2. When Spotfire Analyst users start the application, they are informed that an upgrade is available. (Alternatively, the server can be configured to force an upgrade so the new SPK is distributed automatically.)



If you did not provide a certificate when you build the SPK, the users are asked to accept the unsigned file. See your server administrator for more information, or to get a copy of a certificate and its password.

Packages deployed for a small group

The package curator can deploy packages to a small group, rather than to an entire organization.

Any Spotfire Server deployment area can contain only one `SpotfireSPK.spk`, and recreating the SPK and redeploying it overwrites the version that was previously deployed there. The package curator might not want to overwrite the broadly-distributed SPK. For example, the package curator might want to create a small, contained package or set of packages for testing. Luckily, the Spotfire Server can contain multiple deployment areas, and the `SpotfireSPK.spk` deployed there applies only to that area.

To deploy a package, or a set of packages in an SPK to a limited group, Ask the Spotfire Server administrator to create a new deployment area on the Spotfire Server and grant access to this deployment area to the designated group.

When the designated users open Spotfire and connect to the test deployment area, their installation is updated to the version of Spotfire with the package, and then they can use it with the local engine.



Even when testing a new package, make sure to follow your organization's rules to keep official package versions synchronized.

Spotfire package maintenance

Package maintenance includes updating package versions, removing obsolete packages, or adding new packages to the SPK distribution. It also includes ensuring that everyone using a package is using the same version.

We recommend that you assign a person in your organization the task of maintaining the R packages and versions distributed via the Spotfire SPK deployment mechanism, as well as those uploaded to Spotfire Statistics Services. (See [Curator role](#) for additional guidance.)

If you need to add one or more binary packages to the SPK, or if you need to update an existing package, you can recreate the `SpotfireSPK.spk` as described in [Creating the Spotfire SPK](#).

There can be only one `SpotfireSPK.spk` at a time on the Spotfire Server deployment area. Redeploying the SPK overwrites the SPK currently in a deployment area. Spotfire Analyst users who start a session that connects to that deployment area are prompted to update.



If you have analysts running analyses using the local TERR in Spotfire Analyst, be sure to inform them of any changes to packages distributed to the team by Spotfire Server.

Install packages on Spotfire Statistics Services

You can install packages onto the Spotfire Statistics Services server. If you want to use packages from a repository on the internet (such as CRAN), you need a computer with a connection to the internet.



If you use this method in a cluster environment, you must upload the same package on each server in the cluster.

- If you have administrative privileges, then you can log on to the Spotfire Statistics Services server with administrative credentials, and then use the TERR function `install.packages` to upload packages from either the CRAN repository or from an in-house repository. See [Uploading a package to Spotfire Statistics Services from a repository](#) for more information.
- If you have administrative privileges, then you can copy a package from an in-house computer to the Spotfire Statistics Services server. See [Uploading a package to Spotfire Statistics Services from another computer](#) for more information.
- If you have Eclipse installed on your computer, and you can get access to the Spotfire Statistics Services server, then you can use the tool TSSS Connector to connect to Spotfire Statistics Services and install packages to either a single server deployment of Spotfire Statistics Services or a cluster. See [Uploading a package to TIBCO Spotfire Statistics Services using TSSS Connector](#) for more information.

After installing packages on Spotfire Statistics Services, you can [validate the upload](#).



If you are using TERR with Spotfire Statistics Services, be aware that by default, certain Spotfire Statistics Services capabilities are disabled. If you created or used functions in a previous release, or if you install packages that contain functions that TERR deems to be potentially malicious, you might find that they do not work as expected. Any expression that TERR determines to be potentially malicious is disabled. For more information, see the *Spotfire Statistics Services User's Guide* and talk to your server administrator.

Uploading a package to Spotfire Statistics Services from a repository

If you have access to the internet from your installation of Spotfire Statistics Services, you can log on to the server and install packages directly.

Perform this task on the computer where Spotfire Statistics Services is installed.



If your Spotfire Statistics Services installation is a cluster, you must perform this task on every computer in the cluster.


Prerequisites

You must be able to log on with administrative credentials to the server where Spotfire Statistics Services is installed.

You must have built the package archive or downloaded a compatible package from a trusted web site or package repository.

The CRAN repository contains binary packages (for Windows) and source packages (for Linux and Windows). You can easily install most binary and source packages in TERR. If you have problems building from source, then build the packages using open-source R before installing them into TERR. Note that TERR does not build binary packages from source packages that contain Java source code.



Platform	Package type	Notes
Linux, Windows	Binary	Call <code>install.packages(pkgname)</code> . TERR installs the binary package into your specified package directory.
Linux	Source; no Java code, no C/C++ or Fortran code	Call <code>install.packages(pkgname)</code> . TERR builds the source package into a binary package and installs it into your specified package directory.
Linux, Windows	Source; C/C++ or Fortran code (no Java code)	 <p>On Windows, first you must install the Rtools utilities, which is maintained by Duncan Murdoch, from CRAN, and then update your PATH to specify the location of the utilities.</p> <ol style="list-style-type: none"> 1. If you have not already done so, install the package <code>rinclude</code> by calling <code>install.packages(rinclude)</code> 2. Call <code>install.packages(pkgname)</code>. <p>TERR builds the source package into a binary package and installs it into your specified package directory.</p> <p>If the package does not build and install, then try building it with open-source R, and then installing the binary as described here.</p>
Linux	Source; Java code	<ol style="list-style-type: none"> 1. Build the package using open-source R tools for building packages from source. The tools compile the source code to create the binary package. 2. Call <code>install.packages(pkgname)</code>.

See the help for `install.packages(pkgname)` for more information.

Due to changes in open-source R version 3.5 and resulting compatibility changes in TERR 5.0, packages that are built with a version of TERR prior to 5.0 must be rebuilt.

- To install a binary package from a repository, always call `install.packages(pkgname)` from TERR. The `install.packages` function finds the correct binary version in the repository for your version of TERR. Manually downloading the binary package from CRAN can result in errors when you use it with TERR.
- To install a package from source, try installing it first with TERR (with `install.packages` in TERR or with `TERR CMD INSTALL` from a command line).
- To install a package from source that you cannot build with TERR, install the package with the version of open-source R tested with TERR.

Spotfire Statistics Services accepts .zip archives (Microsoft Windows servers only) or tar.gz archives (Linux servers only).

- If you plan to download an R package from CRAN to run on the TERR engine, we recommend that you check the package compatibility list for the version of TERR on your installation of Spotfire Statistics Services, and test the package with the version of TERR that is in your installation of Spotfire Statistics Services. See the [Documentation](#) page for TERR for more information.

Procedure

1. Log on with administrator credentials to the computer where Spotfire Statistics Services is installed. If you do not have administrative credentials, ask your server administrator to help you.
2. Open the `bin` directory of the server's engine.
This path should look like `SPSERVER_HOME/engines/eng/bin` where `SPSERVER_HOME` is the installation and server context, and `eng` is the language engine, such as open-source R or TERR.
Windows example: `C:\Program Files\TIBCO\statsvcs711\TERRServer\engines\Terr\bin`
3. Right-click the language engine executable, and from the menu, click **Run as administrator**.
A console for the language starts.
4. At the console command prompt, type the command `install.packages("pkgname")`, where `pkgname` is the package you want to install.
If your installation of Spotfire Statistics Services is configured to be able to install packages from the internet, then `install.packages()` installs from the RStudio mirror of CRAN. If you want to install a package from another repository, such as an in-house package repository, provide the path. See the *TERR Language Reference* for more information.



By default, Spotfire Statistics Services is configured to restrict access to file I/O, downloading from the internet, and any other operation that can be considered a potential malicious action. If your installation of Spotfire Statistics Services is not configured to be able to install packages from the internet, then see your system administrator for guidance on how to get the packages you need.

Example: `install.packages("h2o")`

The current version of the package and all of the packages it requires to work are downloaded and installed into the directory `SPSERVER_HOME/engines/eng/library` from the specified repository.



Spotfire Statistics Services does not use the TERR `site-library` directory that the developer edition uses.

What to do next

[Validate the package upload.](#)

Uploading a package to Spotfire Statistics Services from another computer

If your installation of Spotfire Statistics Services is not connected to the internet, but you have administrative privileges to log on to the server, you can still install packages on it. You can upload two types of packages by copying them to Spotfire Statistics Services.

- You can upload packages that have been created and built for your Spotfire Statistics Services installation.
- You can use another computer that is running the same operating system, and that is connected to the internet to download a package from a repository.

Prerequisites

You must have either a built binary package or access to a computer on the internet.



Binary (built) packages are required. Source packages must be built before they can be uploaded to Spotfire Statistics Services.

Spotfire Statistics Services accepts `.zip` archives (Microsoft Windows servers only) or `tar.gz` archives (Linux servers only).

You must have administrative credentials to log on to the server where Spotfire Statistics Services is installed (or you must give the package archive to a server administrator to upload them).

Procedure

1. From the internet-connected computer, make sure that TERR or open-source R (and optionally RStudio) is installed.



The computer you use to download packages must be running the same operating system as the Spotfire Statistics Services server. (That is, Windows or Linux. If Linux, it must be the same version of Linux.)

2. Start a console or, if available, an RStudio session.



If you are running the console from Microsoft Windows, from the **Start** menu, right-click the application, and then select **Run as administrator** so you can install the package into the engine's `site-library` directory. Otherwise, you are prompted to create a personal library. If you select that option, make a note of the location so you can find the package.

3. Open the directory where the package (and any dependent packages) downloaded, and using an archiving utility, create a `.zip` or a `tar.gz` archive that contains all files and directories included with the download.

4. Copy the resulting archive to a medium (such as a flash drive) or a network location that your server can access.

5. If you have access, log on with administrator credentials to the server where Spotfire Statistics Services is installed.

If you do not have administrative credentials, ask your server administrator to help you.

6. Insert the flash drive into a port on the server, or from the server, map a network drive to the package archive location.

7. Browse to the installation of Spotfire Statistics Services and open the `library` directory for the appropriate language (TERR or R).

This path should look like `SPSERVER_HOME/engines/eng/library` where `SPSERVER_HOME` is the installation and server context, and `eng` is the language engine, such as R or TERR.

Windows example: `C:\Program Files\TIBCO\statsvcs75\TERRServer\engines\Terr\library`

8. Copy the zipped archive of the package(s) from the medium or network drive to the `library` directory, and then unzip them there.

What to do next

[Validate the package.](#)

Uploading a package using TSSS Connector

To upload a package to Spotfire Statistics Services, if you have server administrative privileges and an Eclipse installation, you can establish a connection to Spotfire Statistics Services and upload TERR or R packages using the TSSS Connector.

Prerequisites

To complete this task, you must have completed the following steps.

1. Built the package archive or downloaded a compatible package from a trusted web site or a package repository, such as CRAN or an in-house repository.



Binary (built) packages are required. Source packages must be built before they can be uploaded to Spotfire Statistics Services.

Spotfire Statistics Services accepts `.zip` archives (Microsoft Windows servers only) or `tar.gz` archives (Linux servers only).

2. Installed the Eclipse integrated development environment.



We tested versions 3.6 to 4.2.2 (Juno) of Eclipse for the remote submission tools.

3. Downloaded the TSSS Connector Eclipse plug-in from the update page on Spotfire Statistics Services.

For detailed information about updating or installing this plugin, browse to the landing page for the Spotfire Statistics Services installation available to you, and then see the update page.

```
http://SName:P#/SC/update
```

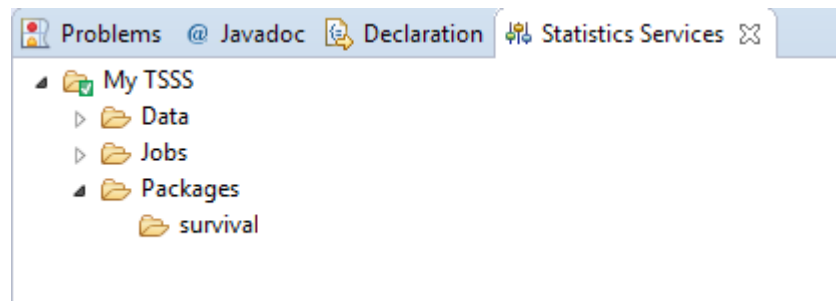
SName is the server name, *P#* is the port number, and *SC* is the server context (for example, `http://CoTSSS:8080/TERRServer/update`).

If you are uploading a package that is also provided to Spotfire Analyst users from a Spotfire Server deployment, then the package version must match the version distributed in the SPK to the Spotfire Analyst users. See [Develop packages for use with the TIBCO Enterprise Runtime for R engine](#) for more information.

Perform this task using the **Statistics Services** plugin for Eclipse.

Procedure

1. Right-click the **Packages** folder.
2. From the menu, select **Upload Package**.
3. Supply the location of your package archive.



4. Add the archive. The package is now on Spotfire Statistics Services.



Although the package is on Spotfire Statistics Services, it is not loaded into the engine. With each call to Spotfire Statistics Services, the engine is started anew.

To use the package, you can either load the library as part of your function scripts, or you can include it in the **Packages** field of your Spotfire data function.

Maintaining a package in TIBCO Spotfire Statistics Services

Keep the package versions in synch across your Spotfire Statistics Services deployments.

Prerequisites

You must have the Eclipse IDE and the TSSS Connector configured and installed.

After you add a package to the server, you can check its properties using the Statistics Services view in Eclipse.

Procedure

1. In the Statistics Services view, connect to the service if it is not already connected.
2. Expand the folder structure to see the packages listed in the Packages folder.
3. Right-click the package name, and from the menu, select **Package Properties**.
The Properties dialog box appears, and information on the package, drawn from its DESCRIPTION file, is displayed.
4. Update the package as needed.
5. Follow the steps to upload the updated package.
See [Uploading a package to TIBCO Spotfire Statistics Services](#).

What to do next

After the package has been uploaded, follow the steps in [Validating the package upload](#).

Validating the package upload

After you upload a package, run a quick validation to ensure that your package is on the server.

Prerequisites

You must have administrative privileges to perform this task.

Perform this task from the TERR console in your installation of Spotfire Statistics Services

Procedure

1. From the browser, type the following command.
`installed.packages()`
A list of installed packages is printed in the console.
2. Review the list (including the path) to ensure that the package you uploaded is installed.



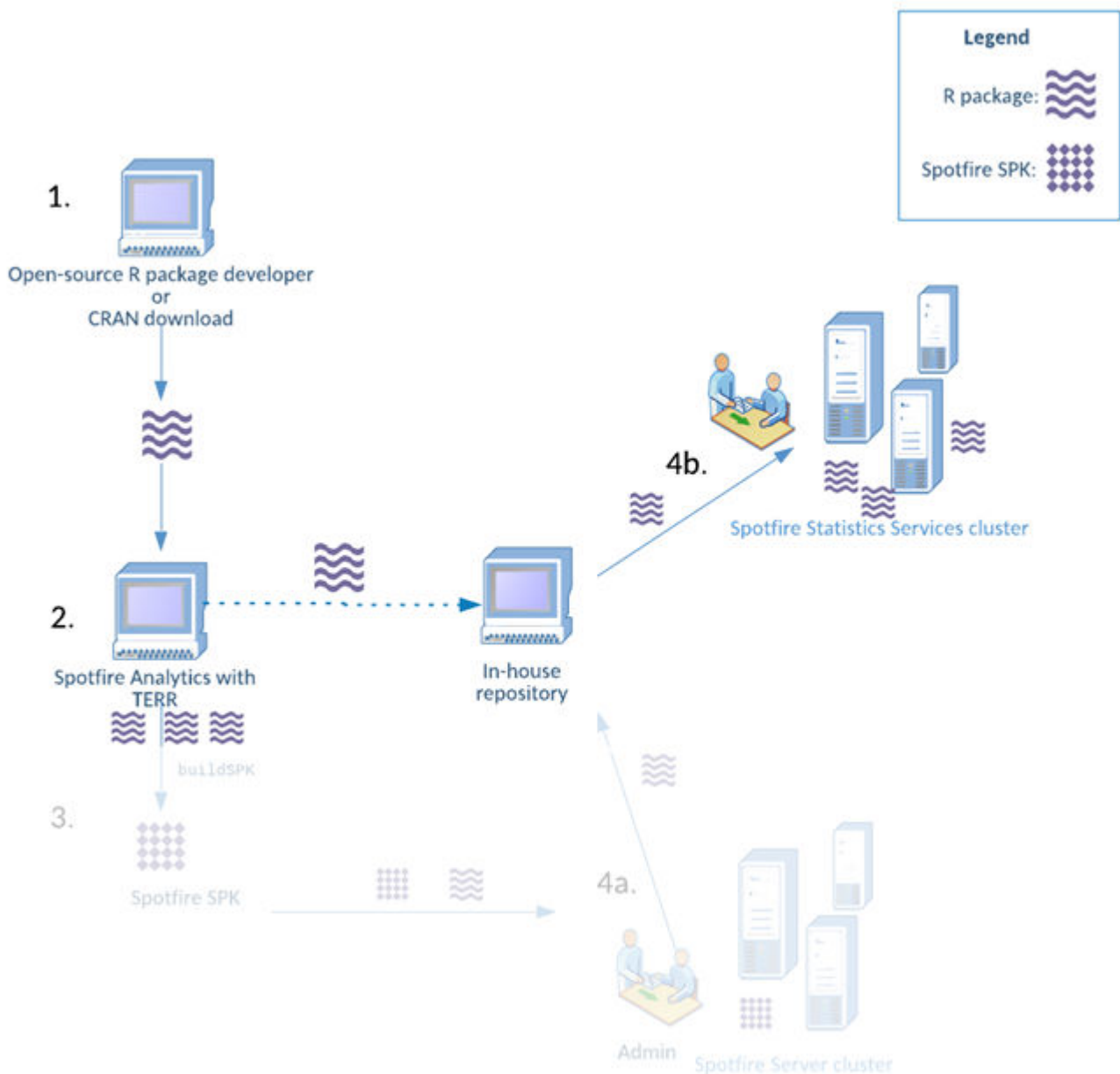
Although the package is on Spotfire Statistics Services, it is not loaded into the engine. With each call to Spotfire Statistics Services, the engine is started anew.

To use the package, you can either load the library as part of your function scripts, or you can include it in the Packages field of your data function.

Manage packages between Spotfire and Spotfire Statistics Services

You can share Spotfire visualizations that use R language packages. To share such visualizations widely in a web browser, your server configuration must include the Spotfire web client and Spotfire Statistics Services deployed and configured to work with Spotfire Server.

Spotfire Statistics Services includes a repository for the packages containing functions that can be used by Spotfire analyses. These packages must be identical to those packages distributed to the installations of Spotfire Analyst. See [Package Management through Roles](#) for more information about keeping packages synchronized.



You can add packages to your Spotfire Statistics Services installation. For more information, see [Install packages on Spotfire Statistics Services](#).

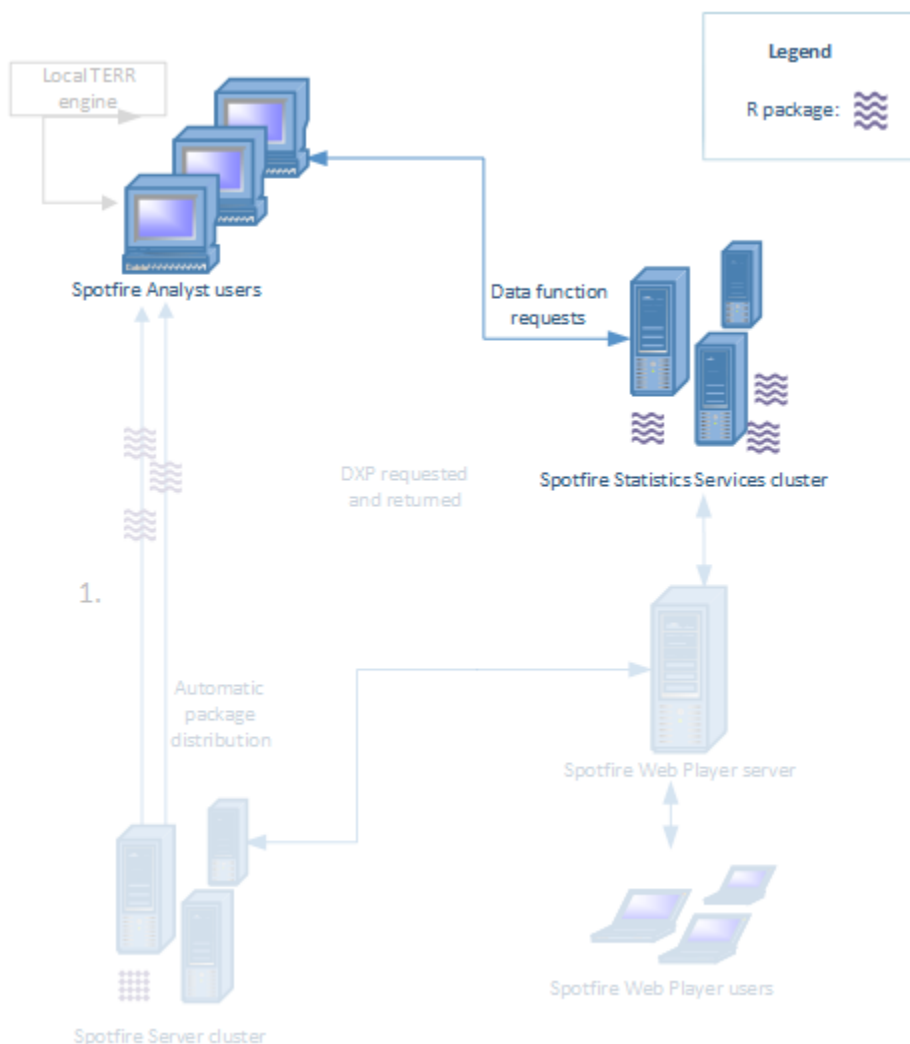
Changing the local engine option

You can configure Spotfire Analyst to use the TERR engine that is installed in your organization's Spotfire Statistics Services deployment.

Prerequisites

Remember that you must have the same package version on your local installation and on the server. See your organization's package curator for help.

Spotfire Statistics Services is required in deployments where users access Spotfire analyses using the Spotfire web client. Changing from the local TERR engine to the one on Spotfire Statistics Services is useful for testing the analyses the Spotfire web client users access.



Procedure

1. In Spotfire Analyst, click **Tools > Options**.
2. In the left pane, scroll down and select **Data Functions**.
3. In the **Data Functions** pane, provide the URL to the Spotfire Statistics Services (for example, <http://CoTSSS:8080/TERRServer>).
4. Clear the checkbox **Use locally installed TIBCO Enterprise Runtime for R**, and then click **OK** to accept.



The following advanced analytic tools available in the Spotfire Analyst installation always use the local engine, regardless of this setting.

- Classification modeling.
- Regression modeling (linear regression or regression tree).

Troubleshooting TERR and Spotfire packages

If you remove a package accidentally, cannot install a package, get different results than you expect in your analysis, or cannot distribute an SPK, try these techniques to solve the problem.

I generated a new SPK using `writeLines` but it is not being distributed.

It is most likely that the SPK version number is not being properly revised. Review the rules for how the SPK version is determined in [SpotfireSPK versioning](#) for more information.

I have copied a package from CRAN but it is not working with TERR

Always make sure to use the version of open-source R that was tested with the version of TERR you are using.

Due to changes in open-source R version 3.5 and resulting compatibility changes in TERR 5.0, packages that are built with a version of TERR prior to 5.0 must be rebuilt.

- To install a binary package from a repository, always call `install.packages(pkgname)` from TERR. The `install.packages` function finds the correct binary version in the repository for your version of TERR. Manually downloading the binary package from CRAN can result in errors when you use it with TERR.
- To install a package from source, try installing it first with TERR (with `install.packages` in TERR or with `TERR CMD INSTALL` from a command line).
- To install a package from source that you cannot build with TERR, install the package with the version of open-source R tested with TERR.

I accidentally removed a package from my Spotfire installation that I need. How do I get it back?

We recommend never removing a package from Spotfire TERR Tools, unless you have downloaded and installed the package yourself from a repository such as CRAN. You must be very careful to not remove a package that is installed with TERR or distributed by the Spotfire Server using the SPK mechanism. However, should you accidentally remove a needed package, you can recover.

If you deleted a distributed package, follow these steps.

1. Open Windows Explorer and browse to `%SPOTFIRE_HOME%\Modules`.
For example, `C:\Program Files (x86)\TIBCO\Spotfire\<version#\Modules`.
2. Delete the folder `TIBCO Enterprise Runtime for R Packages_<version#\>`.
3. Restart Spotfire and accept the prompt to update your installation.

All of the TERR packages contained in the SPK on the Spotfire Server are redistributed.

If you deleted a package required by TERR to operate, follow these steps.

1. Open Windows Explorer and browse to `%SPOTFIRE_HOME%\Modules`.
For example, `C:\Program Files (x86)\TIBCO\Spotfire\<version#\Modules`.
2. Delete the folder `TIBCO Enterprise Runtime for R_<version#\>`.
3. Restart Spotfire and accept the prompt to update your installation.

The package containing TERR is redistributed.

I am generating a different result from those seen by others. What could cause this?

Check your package version numbers to make sure everyone is using the same package version. See your package curator for more information.

I am trying to use a package on Spotfire Statistics Services but it's not working.

Be aware that by default, certain functions are restricted by Spotfire Statistics Services. If you see the error "Error: restricted call to Native[tempfile]", and execution of the expression is terminated, then the default restrictions have not been changed by your server administrator. Restricted behavior includes the following non-exhaustive list of operations.

- Performing any I/O to the file system or the internet.
- Loading new packages, except for the libraries included with TERR (stats, terrUtils, and so on).
- Spawning new OS processes (calling `system`).
- Calling `.Call`, which is used to call Rapi code in CRAN packages.
- Calling `.C` or `.Fortran`.
- Calling into Java using the `terrJava` package (which allows executing arbitrary Java methods).
- Calling any functions in the `parallel` package (which uses `terrJava`).
- Accessing any function environments in the stack above the call to `evalREX` using `sys.frame` or `parent.frame`. (This prevents malicious code from installing functions or expressions that could be executed after leaving restricted execution mode.)
- Changing the variable lookup path by setting `parent.env` of an environment, or reading or setting the environment of a closure.
- Defining S4 classes and methods using `setClass` or `setMethod`.

Be sure you are loading the package in your TERR script. Although the package is on Spotfire Statistics Services, it is not loaded into the engine. With each call to Spotfire Statistics Services, the engine is started anew. To use the package, you can either load the library as part of your function scripts, or you can include it in the `Packages` field of your data function.

I cannot install a package from TRAN on Linux because it requires another package that is not installing correctly.

Some packages customized and placed on TRAN require other packages not available on TRAN. Some of these packages cannot be installed using the TERR function `install.packages`, so the TRAN package cannot be successfully installed. If you encounter this situation, try building and installing the package using open-source R.

If you get a warning that the `rinclude` package is not installed, try installing that package from TRAN and trying again.



When you install a package using TERR, by default, TERR first checks for the package on TRAN, and then checks on CRAN. TERR installs the first version it finds. This is different than open-source R, which installs packages according to the newest version number available on CRAN. This difference is by design, because occasionally a CRAN package update causes a break with TERR compatibility, so we make available a tested version of the package on TRAN. If you need to install one of these packages using open-source R, you can install the CRAN package, and then set `options()$repos` to install from only TRAN before reinstalling the package. See [Specifying an older package on TRAN](#) for more information.

Index

.libPaths 18
[package installation 17

A

administering packages 13

B

best practices 10
binary 21
buildSPK 30

C

console 22
CRAN 10, 11, 19, 24, 27, 33, 34, 38, 46
curating packages 13

D

DCF 33, 34
developer 12
differences with R 11
differences, open-source R 9
documentation 9
drat 24

E

Eclipse 41, 42
Eclipse plugin 30

G

graphics 20

H

help 9, 28
HTTPS 10

I

installation 37
installed packages 23

J

java 16
JAVA_HOME 16

L

limited package deployment 36

local engine 22, 44

O

open-source R 10

P

package 43
package compatibility 11
package library 18
package location 13
package overview 30
package synchronizing 12
packages 16, 19, 43
platform 20

R

R package 12, 21
R security recommendations 10
recovering packages 46
redeploying packages 37
removing packages 29
repositories 24, 27
rJava 17
RStudio 20

S

SBDF 23
server 35
site-library 18
site.library 40
source 21
SPK 12, 19, 23, 30, 32–36
Spotfire 23
Spotfire configuration 44
Spotfire license 22
Spotfire licenses 20
SpotfireConnector 23
SpotfireData 23
SpotfireSPK 23, 32–34
SpotfireStats 23
SpotfireUtils 23
synchronizing packages 13

T

TERR Tools 22
testing 27
TRAN 24, 27, 46
troubleshooting 46
TSSS Connector 30, 41, 42

U

updating packages 13, 37, 38
updating TERR 18
uploading 37
uploading packages 13, 40

V

validation 43

W

writeLines 46