Spotfire® Server and Environment Security

Software Release 14.0 LTS (14.0.4)

Contents

Environment Overview

Understanding the components, and the communication between the components of the Spotfire environment is key to understanding how to build a more secure environment.

1. The Spotfire Server is the central component of the Spotfire environment, to which all Spotfire clients connect. From a Spotfire Server start page, entities in the Spotfire environment can be configured and monitored.

   For more information about the Spotfire Server, see its documentation.

2. Multiple nodes are installed and connected to Spotfire Server. The Spotfire Web Player service, Spotfire Automation Services, the Spotfire Enterprise Runtime for R – Server Edition, Spotfire Service for R, and Spotfire Service for Python can be installed on nodes to enable the use of Spotfire web clients, running Spotfire Automation Services jobs, and running data functions and scripts.

   For more information about the components installed on nodes, see their help:

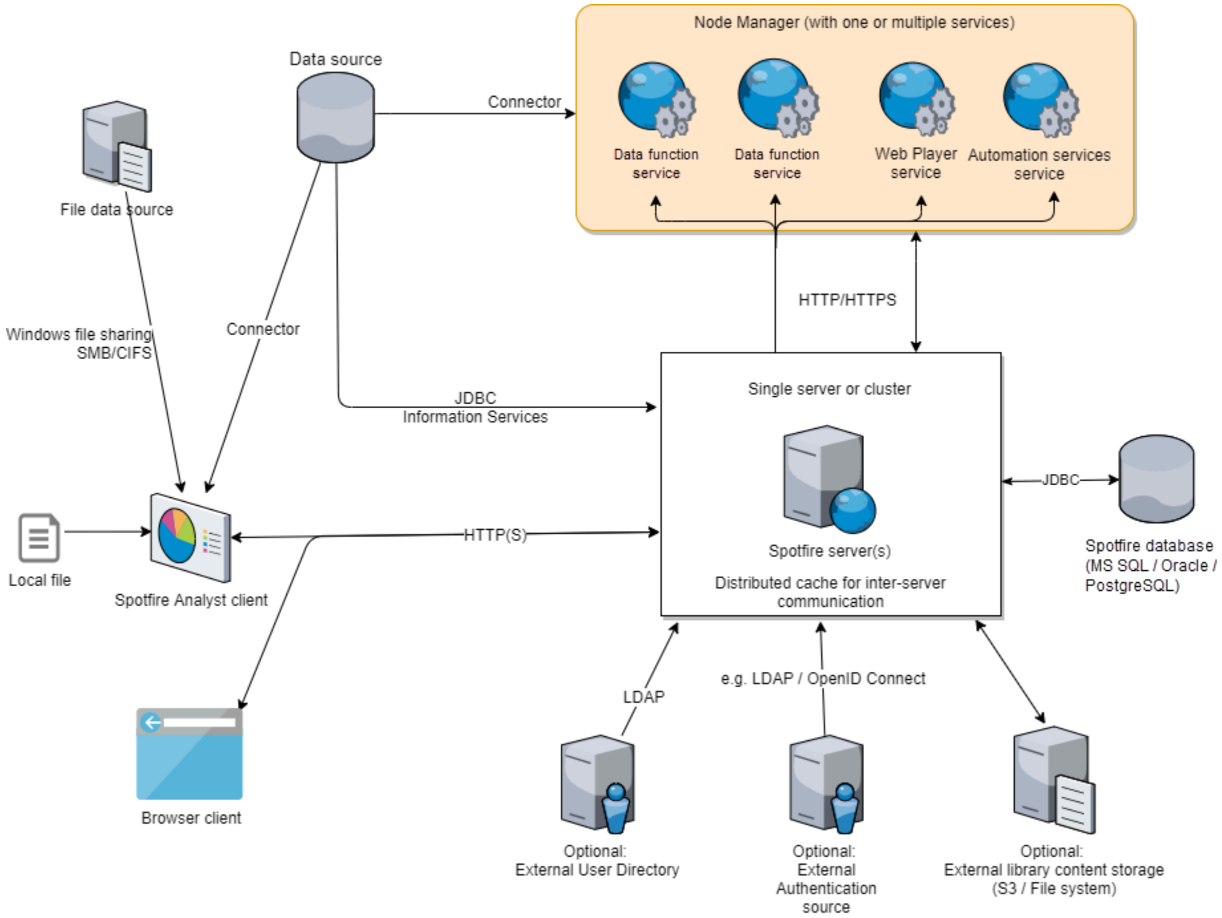
   - Node manager (installation and configuration in *Spotfire® Server and Environment Installation and Administration*)
   - Spotfire® Web Player (service installation and configuration in *Spotfire® Server and Environment Installation and Administration*)
   - Spotfire® Automation Services (service installation and configuration in *Spotfire® Server and Environment Installation and Administration*)
   - Spotfire® Enterprise Runtime for R - Server Edition (a/k/a the TERR™ service)
   - Spotfire® Service for R
   - Spotfire® Service for Python

3. The server is connected to a Spotfire database that contains a user directory and stores analyses and configuration files. For more information, see its documentation.

4. After the node is installed, the node performs a join request to a specific, unencrypted Spotfire Server HTTP port that handles only registration requests. The node remains untrusted until the administrator approves the request by trusting the node. The Spotfire Server start page provides the tools to add nodes to the environment by explicitly trusting them, thereby issuing the certificates. When the node receives its certificate, it can send encrypted communication over the HTTPS/TLS ports, and with this, the node can start to send more than registration requests.

   The secured back-end communication is based on certificates. After an administrator has approved the new server or node, the certificates are issued automatically. Without a certificate, a server or a service on a node cannot make requests to, or receive requests from, other entities, except for when requiring a certificate. For more information, see Ports and firewall configuration in *Spotfire® Server and Environment Installation and Administration*.

This diagram shows all of these components, as well as how data flows and network protocols are used in a typical Spotfire environment.


Node Manager (with one or multiple services)
Data function service
Data function service
Web Player service
Automation services service
Data source
Connector
File data source
Windows file sharing SMB/CIFS
Connector
JDBC
Information Services
HTTP/HTTPS
Single server or cluster
Spotfire server(s)
Distributed cache for inter-server communication
JDBC
Spotfire database (MS SQL / Oracle / PostgreSQL)
Local file
Spotfire Analyst client
HTTP(S)
Browser client
LDAP
e.g. LDAP / OpenID Connect
Optional: External User Directory
Optional: External Authentication source
Optional: External library content storage (S3 / File system)

Ports and Protocols

You can use the following ports, connections, and protocols to secure Spotfire.

Ports

Spotfire Server, the node manager, and related services reserve the following ports for various communication tasks.

Public-Facing Client Connection Ports

Name	Default Port and Protocol	Function Description	Secure/Encrypted
Public HTTP port	80/tcp, if enabled	Non-secure communication with installed clients and web clients.	No
Public HTTPS port	443/tcp, if enabled	Secure communication with installed clients and web clients.	Yes

The HTTP connector port and the HTTPS connector port are configured independently and are exposed externally for client connection. You can use either of them or, in some cases, both.

Spotfire Server

Name	Default Port and Protocol	Function Description	Secure/Encrypted
Back-end registration port	9080/tcp	Establishing trust between the Spotfire Server and nodes only.	No
Back-end communication port	9443/tcp	Monitoring secure traffic between nodes. (Spotfire Server monitors secure traffic from services on the nodes.)	Yes
First clustering port	5701/tcp	Secure communication within the environment. This port is the same for all servers in the cluster.	Yes
Second clustering port	5702/tcp	A second clustering port for secure communication within the environment.	Yes
Third clustering port	5703/tcp	A third clustering port for secure communication within the environment.	Yes
Fourth clustering port	5704/tcp	A fourth clustering port for secure communication within the environment.	Yes

Name	Default Port and Protocol	Function Description	Secure/Encrypted
JMX RMI port	1099/tcp, if enabled	If JMX RMI access is enabled, Spotfire Server opens a separate port for this purpose. Might be considered a "public-facing" port.	See config-jmx

Node Manager

Name	Default Port and Protocol	Function Description	Secure/Encrypted
Registration port	9080/tcp	Establishing trust between node managers and Spotfire Server.	No
Communication port	9443/tcp	Secure communication within the environment.	Yes

Services

Name	Default Port and Protocol	Function Description	Secure/Encrypted
Communication port (Spotfire Web Player/Spotfire Automation Services)	Next available general purpose 950<*x*>/tcp. (for example, 9501/tcp, or 9502/tcp, and so on, depending on the other services installed.)	Spotfire Web Player and Spotfire Automation Services for secure communication.	Yes
Communication port (TERR)	Next available general purpose 950<*x*>/tcp. (for example, 9502/tcp, or 9503/tcp, and so on, depending on the other data function services installed.)	TERR service, for secure communication. This port assignment is needed internally on the computer that the service is running on, but is not needed on other computers.	Yes
Communication port (Spotfire Service for Python)	Next available general purpose 950<*x*>/tcp. (for example, 9502/tcp, or 9503/tcp, and so on, depending on the other data function services installed.)	Spotfire Service for Python, for secure communication. This port assignment is needed internally on the computer that the service is running on, but is not needed on other computers.	Yes
Communication Port (Spotfire Service for R)	Next available general purpose 950<*x*>/tcp. (for example, 9502/tcp, or 9503/tcp, and so on, depending on the other data function services installed.)	Spotfire Service for R, for secure communication. This port assignment is needed internally on the computer that the service is running on, but is not needed on other computers.	Yes
Communication port (Information Services)	9445	Internal communication between Information Services and Spotfire Server.	Yes
TERR engine ports	61001/tcp -> 62000/tcp, if the TERR service is installed	Host-internal communication between the TERR service and the TERR engines.	No

Name	Default Port and Protocol	Function Description	Secure/Encrypted
Spotfire Service for Python engine ports	62001/tcp -> 63000/tcp, if the Spotfire Service for Python is installed	Host internal communication between the Spotfire Service for Python and the Python interpreter.	No
Spotfire Service for R engine ports	63001/tcp -> 64000/tcp, if the Spotfire Service for R is installed	Host internal communication between the Spotfire Service for R and the R engines.	No

The back-end ports need exposed only for Spotfire Server connection to services available from the node manager.

Outbound Connections

The following outbound connections might differ from your deployed system, because connections depend on the configuration of the particular environment. For example, the Spotfire Server creates LDAP connections only if the system is configured to use LDAP.

Spotfire Server

Type of connection	Default	Function	Secure/ Encrypted
Database communication	Oracle database: 1521 SQL Server: 1433 PostgreSQL: 5432	The Spotfire database server monitors this port.	If configured
LDAP	LDAP over TLS: 389 LDAPS: 636.	An optional number that indicates the TCP port that the LDAP service is monitoring.	If configured
LDAP > Global Catalog	LDAP: 3268 LDAPS: 3269	Active Directory LDAP servers also provide a Global Catalog that contains forest-wide information, instead of domain-wide information only.	If configured
TIBCO Enterprise Message Service (EMS)	Non-secure connection: 7222 Secure connection: 7243	This service can be used to trigger scheduled updates. EMS monitors this port.	If configured
Kerberos/GSSAPI	Fixed port 88 on the Active Directory domain controllers	Used by the Kerberos authentication method, as well as when authenticating to an LDAP server using the GSSAPI method.	Yes
Microsoft Net Logon, SMB, and CIFS	Fixed port 445 on the Active Directory domain controllers	Used by the NTLM v2 authentication method.	Yes
Open ID Connect providers	443	Used by the web authentication method.	Yes

Type of connection	Default	Function	Secure/ Encrypted
Data sources (Information Services)	Oracle database: 1521 SQL Server: 1433 Netezza: 5480 Otherwise, varies.	JDBC-compliant data sources and other services used by Information Services monitor these ports.	Varies

Node manager/Services

Type of connection	Default	Function	Secure/ Encrypted
Spotfire® Web Player & Spotfire® Automation Services > Map/tiles server connections	The default map layer uses https://geoanalytics.tibco.com/	The map chart downloads map tiles and other information from external servers.	Yes
Spotfire Web Player & Spotfire Automation Services > SMTP	25, 2525, or 587 Secure SMTP: 465, 25, or 587	Used by Spotfire Automation Services for sending e-mails.	Secure if configured
Spotfire Web Player & Spotfire Automation Services > Data sources (Connectors)	Varies	For information on available connectors, see "List of Connectors in this Version" in the Spotfire Analyst User Guide. Data connectors listen to these ports.	Varies

HTTP Cookies

Spotfire Server can set the following HTTP cookies on clients that connect over the public HTTP port (default 80/433).

The `Secure` attribute is set only if the connection is HTTPS, not HTTP. To protect against cross-site request forgery (CSRF) attacks, Spotfire does not rely on using the `SameSite` attribute on cookies.

Name	Description	Comment
`JSESSIONID`	Session cookie for Spotfire Server.	`HttpOnly` attribute is set.
`SF_REMEMBER_ME`	Cookies used for persistent sessions ("remember me") feature when running Spotfire in a web browser.	`HttpOnly` attribute is set. See config-persistent-sessions.
`SUID`	Contains the ID of the last authenticated user. It is used to determine whether or not an anonymous session should be created.	`HttpOnly` attribute is set.
`XSRF-TOKEN`	Holds CSRF token.	`HttpOnly` is not set. A cookie that holds a CSRF token is passed to JavaScript using a cookie value. This behavior is intended.
`zoneCheck`	Cookie the JavaScript API uses for identifying browser incompatibilities with Spotfire.	`HttpOnly` is not set. It is not needed, because it is used by client-side JavaScript code and does not contain sensitive information.

Node Trust and Back-End HTTPS Communication

Node managers and Spotfire Server use encrypted HTTPS for communication. All endpoints are authenticated using either server or client certificates issued by the Spotfire Server root certificate, which acts as a certificate authority for a particular Spotfire environment.

Neither the Spotfire Server nor the client certificates used by the various components of the system are self-signed. They are all signed by the certificate authority that is part of the Spotfire Server. Each Spotfire Server installation generates its own unique root certificate. You cannot provide your own.

The signing algorithm, which is used for both certificate authentication (CA) and end-entry certificates, is configurable. By default, it is set to `SHA256withRSA`.

If want to get new CA certificates that use the default `SHA256withRSA`, then you must generate new CA certificates. If you generated certificates using a version earlier than Spotfire Server 10.1.0 (which used the default `SHA1withRSA`), then you must revoke all certificates using the `reset-trust` command, and then generate new CA certificates, which use the new algorithm, and retrust all existing nodes.

The key length is also configurable. The default is set to `2048`, in accordance with the current Mozilla recommendations.

Any changes to the configured value affects only new certificates, so first reconfigure, and then run `reset-trust`, generate new CA certificates, and retrust all existing nodes.

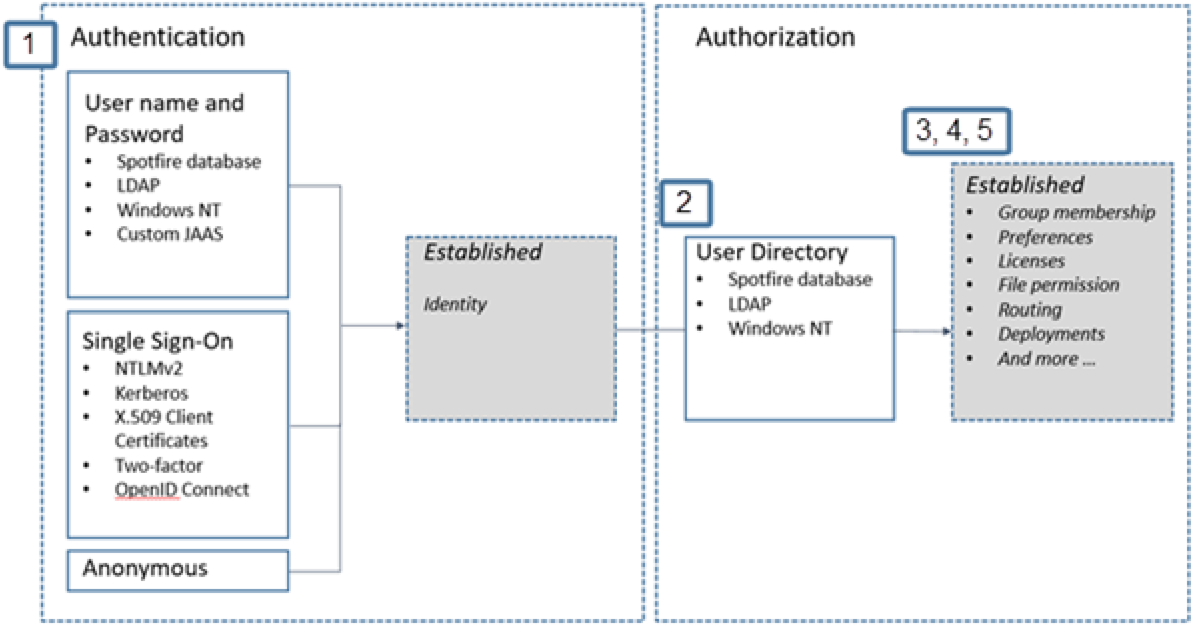
Certificate configuration property	Description
`security.ca.cert-signature-algorithm`	Configuration property to set the signing algorithm. Default: `SHA256withRSA`
`security.ca.rsa-key-strength`	Configuration property key length. Default: `2048`

The node manager and Spotfire Server registration ports (9080/tcp) are used to establish the trust. These ports use plain HTTP and are used only when new nodes are added to the cluster. After trust is established, any further communication is done over a secured HTTPS connection using the communication port (9443/tcp). For a node to become trusted, a member with the role of Spotfire administrator must manually trust the node, enabling the Spotfire Server certificate authority to issue server and client certificates to it. If a node is untrusted by an administrator through the web administration interface, the Online Certificate Status Protocol (OCSP) is used to communicate that the certificate for the untrusted node has been revoked.

Node managers running a Spotfire Web Player service or Spotfire Automation Services on Windows install the three certificates into the Windows certificate store under the machine level.

Authentication and Authorization

The following image provides an overview of the available authentication and authorization options for Spotfire.



You can implement other methods using APIs.

Generally, authentication and authorization occur in the following sequence, as shown in the illustration.

1. Authentication established: determined by one of the configurations shown in the left panel of the illustration.
2. User directory authorization.
3. Groups and roles authorization.
4. Licenses authorization.
5. Preferences authorization.

Authentication

Spotfire provides several standard authentication methods, as well as custom authentication using APIs.

Authentication method	Description
User name and password	The default method. User name and password specifies authentication using HTML forms (`POST - application/x-www-form-urlencoded`) or BASIC access authentication. The credentials are checked against the Spotfire database or another external authentication source (such as LDAP, Windows NT Domain, or Custom JAAS). See External directories and domains and User name and password authentication methods.
Two-factor	You can combine the chosen primary authentication method with X.509 client certificates. See Two-factor authentication.

Authentication method	Description
NTLMv2	Note that NTLMv1 is not supported. See NTLM authentication.
Kerberos	See Kerberos authentication.
Anonymous	If enabled, limited access to view Spotfire files is allowed for unauthenticated sessions. See Configuring anonymous authentication.
X.509 client certificates[1]	Spotfire Server requires the client to provide a valid X.509 certificate. Requires HTTPS. See Authentication using X.509 client certificates.
OpenID Connect (OIDC)	Goes under the label "Web Authentication" in Spotfire. Provides integration with external authentication providers that support OpenID Connect. See Configuring OpenID Connect. You can configure OpenID Connect to enable single logout (SLO). See the topic " Single Logout (SLO)" in the Spotfire® Server and Environment - Installation and Administration for information.
External authentication	See APIs and extension points.
Custom Web Authentication	See APIs and extension points.
Custom Authentication	See APIs and extension points.

OIDC or generic single logout configuration

The easiest way to configure single logout (SLO) is by using the OpenID Connect authentication configuration. Alternatively, you can configure Spotfire Server to use a generic (non-OIDC) option for single logout. Spotfire Server supports either a generic RP-initiated single logout, or a generic front-channel single logout. For more information about these options, see the topic "Single Logout (SLO)."

User Directory Options

Spotfire features the following user directory sources for authentication. Users and groups can also be provisioned through several types of APIs, including SOAP and SCIM.

User directory sources

User directory source	Description
Spotfire database	Users are stored in a database and managed using the Spotfire administrative tools.
Windows NT	Legacy. Users are managed in a Windows NT domain. This option does not apply to Linux installations.
LDAP	Users (and groups, optionally) are managed in an LDAP server (such as Active Directory) and are synchronized with Spotfire database.

[1] Combining X.509 client certificates with another authentication method such as user name and password provides a type of two-factor authentication.

Groups and user API, commands

API	Description
Java	PostAuthenticationFilter (See the API reference for more information.) Can be specified by the command-line command config-post-auth-filter.
SOAP Web Service	Listed under APIs and Extension Points. See Package com.spotfire.ws.pub for more information.
SCIM group management	Introduced in Spotfire 11.5.0. Used to synchronize users, groups, or group memberships, from one provider to another, by using the System for Cross-domain Identity Management (SCIM) standard APIs. See Configure Spotfire for cross-domain identity management for more information.
Command-line commands	The command-line commands import-users and import-groups are used to import users or groups (respectively) into the user directory.

APIs and Extension Points

To create a custom authentication experience for your Spotfire users, you can use one of the following types of APIs or extension points.

Type	Description
Post-authentication filter	Use a Java class to implement the com.spotfire.server.security.PostAuthenticationFilter interface, perform additional checks, or create automation steps to perform after completing authentication but before logging the user in. See Spotfire Server API for more information.
Custom JAAS module	Customize a user name and password authentication method with a JAAS module, which is implemented using the com.spotfire.server.jaas API. For example, instead of checking the end-user credentials for the Spotfire database or LDAP, you can implement a custom login. See Spotfire Server API for more information.
External authentication	Use external authentication to provide custom authentication flows where the user's identity can be derived from the incoming HTTP request (for example, using a cookie or a header). External authentication should be combined with a (reverse) proxy or Java class (Custom Web Authentication) that implements the logic that the custom authentication scheme requires.
Custom Web Authentication	Implement custom web-based authentication flows using the com.spotfire.server.security.CustomWebAuthenticator API. A typical use case is to implement an OAuth2-based authentication flow. See Spotfire Server API and Configuring custom web authentication.
Custom Authentication	Implement custom authentication by implementing the com.spotfire.server.security.CustomAuthenticator interface. See Spotfire Server API.
Custom login page	Create a custom login page for the Spotfire Server to enable a fully customizable look and feel. If the authentication method is based on user name and password, and if additional information must be collected from the user, you can combine a custom login page with a PostAuthenticationFilter and possibly a custom JAAS login module. See Replacing the default Login page in the Spotfire Server web UI in the Spotfire Cobranding manual.
Authentication Filter API	This feature is deprecated and should no longer be used.

Additional information about custom authentication methods is available on the Community.

- Create a Custom Login Page

- Spotfire Server API for Custom Authentication
- External Authentication in Spotfire 7.11 and later versions

Password Policy and Password Complexity Enforcement

As of version 12.2.0, Spotfire Server provides built-in support for minimum and maximum password policies for passwords stored in the Spotfire database.

For more information about configuring Spotfire for minimum and maximum password lengths in the Spotfire Database, see Authentication towards the Spotfire database.

Alternatively, you can implement support by configuring Spotfire for Kerberos, NTLM, OpenID Connect, or User name and Password authentication (together with an LDAP/Active Directory). If the external authentication source enforces a password policy, it also applies to Spotfire.

Authorization

Group assignments can authorize users' permissions with Spotfire Server and should be granted only to users who are fully trusted. The list is only a subset of all available Spotfire groups.

For a full list, see Roles.

Roles

Groups define standard roles for administering and using Spotfire. Each special group enables a set of licenses that correspond to an administrative or user role. To assign a role to a user, just add the user to one of the special groups in the following list.

Group	Description
Administrator[1]	Members of this group can set library permissions, preferences, licenses, manage users and memberships on the system. Only users who need administrator privileges on Spotfire Server, including the ability to manage users and groups, should belong to this group.
Library Administrator[1]	Members of this group are granted full permission to the library. It overrides all folder permissions set in the library, granting full control over content. It also includes the permission to import and export library content. Only users and groups that need administrative privileges in the library should belong to this group.
Deployment Administrator[1]	Members of this group have permission to use the Deployments & Packages user interface in the Spotfire Server console. A deployment area is a collection of software packages intended for a specific Spotfire group and client type (Spotfire client, Spotfire Web Player and Spotfire Automation Services) and are used to push hotfixes and other software updates.
Diagnostics Administrator[1]	Members of this group have permission to use the Monitoring & Diagnostics page in the Spotfire Server web administration pages.
Scheduling and Routing Administrator[1]	Members of this group have permission to use the Scheduling & Routing page in the Spotfire Server web administration pages to create and manage scheduled updates and routing rules.
Scheduled Updates Users	The account that runs scheduled updates must be a member of this group. By default, the account `scheduledupdates@SPOTFIRESYSTEM` is a member of this group.
Automation Services Users	Members of this group have permission to execute Spotfire Automation Services jobs on the server, using the Job Builder or the Client Job Sender. By default, the account `automationservices@SPOTFIRESYSTEM` is a member of this group.
Custom Query Author[2]	Members of this group have permission to save scripts written in custom query languages as trusted to the library.

Group	Description
Script Author[3]	Members of this group have permission to save scripts as trusted to the library. For more information about scripts see Usage of Scripts and Data Functions in the Spotfire Analyst help.
Everyone	This group always contains all users in the Spotfire implementation. No users can be removed from this group, but you can set licenses for the group if you want to.
System Account	This group cannot be edited. It contains the system accounts that are used internally in the Spotfire environment.

[1] Members of these groups have almost unrestricted access to the system. Only fully trusted users should be added to any of the administrator groups.

[2] Provides the ability to create data connections that contains arbitrary and unrestricted query language constructs (typically SQL).

[3] Scripts are very powerful. A script author can, but is not limited to, run arbitrary commands on the Web Player server. See Scripts in Spotfire on page 25 for a description of the different types of scripts in Spotfire and what capabilities they bring.

Licenses

Generally, licenses do not grant further permissions to Spotfire users (as opposed to groups). Rather, licenses provide a way to toggle certain functionality on or off for groups of users in the user interface. This topic discusses exceptions.

See the License feature reference in the *Spotfire® Server and Environment - Installation and Administration manual*, or the Spotfire® Administration Manager User Manual, for more information about licenses.

License name	Description
Spotfire Analyst: Create Information Link	Users that have this license can author information links containing arbitrary SQL code.
Spotfire Information Modeler: Administration	Users that have this license have permission to modify data sources, joins, and other elements when they are working with information links.

Preferences

Preferences are usually set by administrators. Some preferences can have an impact on security, and these should be set only after considering what possible security impact changing the preference might have. A non-exhaustive list of such preferences are listed below.

See the *Spotfire® Administration Manager User Manual*, available on the documentation site, for more information about preferences.

Application > ApplicationPreferences

Preference name	Default	Description
Additional File Extensions	`.html`, `.htm`	In Spotfire clients, `file://` links are passed to the operating system, and the default open action for the file type is performed. For example `.html` files are opened in the default browser, `.jpg` files are opened in the application associated with the `.jpg` file extension. By adding extensions such as `.bat`, `.py`, `.exe` (that can contain code), as allowed file extensions in Spotfire, opening files from untrustworthy sources can be dangerous if dangerous file types are allowed.
Additional URI Schemes	Empty	Controls which URI schemes can be used, in addition to `http://` and `https://`.
Allow copying refresh token for credentials profile	False	Controls whether users can copy the OAuth refresh token, for use in a credentials profile, when they open a Microsoft SharePoint Online connection in a Spotfire web client.
AllowSharingOfCachedDataBetweenUsers		Controls whether users are allowed to select the check box **Share cached data between all concurrent users of Spotfire web clients** on the Cache Settings tab in the Data Connection Properties dialog. Setting this preference to False will disable the check box control.
Blocked System Types	Empty	Specifies an array of system types that cannot be used when users save or load documents and bookmarks. The purpose of this restriction preference is to provide the administrator a way to block yet-unknown security issues with insecure deserialization of .NET types or classes, as an environment option. Any classes found to be insecure classes can be blocked without using this preference. Also see Use Blocked System Types in the Application Preferences topic in the *Spotfire Administration Manager - User Guide*.
EnableAllowSavingDatabaseCredentials	True	If enabled, users have the option to include embedded credentials to a data source used in the file when saving Spotfire analyses. Embedding credentials is not recommended because it is possible for anyone with access to the file to read the credentials. By setting this value to False, you can ensure that credentials are not embedded in files by mistake.
Sandbox Attribute for iframe Components	`allow-forms allow-popups allow-same-origin allow-scripts`	You can restrict the content of iframe components in the application (such as the Web page panel) using the standard sandbox attribute rules. Enter values that removes the specified sandbox restrictions, as a space-separated list.

Preference name	Default	Description
Whitelist for Allowed URIs	Empty	You can specify an array of URIs that should be allowed to use in links within Spotfire analyses but also in the "Web Page panel". For security reasons, only trusted sources should be whitelisted. By controlling the whitelist, you can ensure that only approved web servers and other external resources are allowed to interact with analysis files in the Spotfire environment. See Use Whitelist for Allowed URIs in the Application Preferences topic in the *Spotfire Administration Manager - User Guide*.

TextArea > TextAreaPreferences

Preference name	Default	Description
`PerformHtmlSanitation`	`True`	The HTMLSanitization is a whitelist feature that works by only allowing a small subset of HTML in the text area. If disabled, the author or others can create or open analyses that include text areas without HTML sanitation. Setting the preference to False makes the system susceptible to cross-site scripting (XSS) attacks if files from untrustworthy sources are opened.

DataFunctions

Preference name	Default	Description
`IgnoreTrustCheck`	`False`	Allows you to switch off the trust checking of data functions so that data functions that are not approved by a member of the Script Author group can execute without prior approval. Introduced in Spotfire 10.3.

MapChart > MapChartPreferences

Preference name	Default	Description
`DefaultWebMapServiceListUrl`	http://geoanalytics.tibco.com/	The default map chart resource server URL can be overridden so the map chart can be used in an environment without Internet access. See Offline Maps in Spotfire on the Community.

Preference name	Default	Description
`DefaultHttpsWebMapServiceListUrl`	http://geoanalytics.tibco.com/	The default map chart resource server URL can be overridden so the map chart can be used in an environment without Internet access. See Offline Maps in Spotfire on the Community.

Connectors > <Connector name>

Preference name	Default	Description
`AllowEmbeddingCertificatesWithPrivateKeys`	`False`	To be able to create encrypted connections with some connectors, you must add and embed a certificate file in the connection data source. This preference determines whether users are allowed to embed certificate files that contain private keys. Embedding certificate files with private keys is not recommended, because it is possible for anyone with access to the file to extract the certificate. By setting this value to **False**, you can ensure that certificates with private keys are not embedded in files by mistake.

Logging and Monitoring

Spotfire provides different logs for monitoring, diagnostics, and accountability purposes.

Type	Description
User Action Logging	See Action logs and system monitoring.
Monitoring & Diagnostics	See Monitoring and diagnostics.
JMX	See Server monitoring using JMX and JMX configuration security features.

Logs can contain personal identifiable information such as IP numbers, e-mail addresses, and user names. Logs do not contain hashed, encrypted or clear text passwords, session tokens, authentication/authorization tokens.

Session Management

When a user accesses the Spotfire Server, from a web browser or from a client such as Spotfire Analyst, a session is created. A session ID is valid across most of the Spotfire Server environment but the public APIs do not use sessions.

Session IDs

The Spotfire session IDs are 16 bytes/128 bit IDs that are randomly generated by Tomcat. See the Tomcat documentation about session ID generation for more information.

The JSESSIONID cookie holds the session ID. All information associated with the session is stored server-side. See also HTTP Cookies on page 9.

Session rotation

Session IDs are rotated (replaced) when a user authenticates and when users change their own passwords in the Spotfire Server database (username/password authentication). Changes to credentials in external authentication systems have no effect on active sessions.

When a session expires, the session is invalidated (deleted) server-side. The session ID may still remain in client cookie stores and similar but it will no longer refer to any active session (and any subsequent attempts to use it will simply be ignored).

Session timeouts and configuration

The Spotfire Server allows you to configure different aspects of the session management. For example, you can change the absolute and idle timeouts, and restrict the number of concurrent sessions. Absolute session timeout is a recommended security feature, while idle session timeout is mainly a resource management feature. See the documentation about Absolute session timeout and idle session timeout to change the default timeout values.

See also Spotfire Server Security Configuration and Administration Activities on page 37.

Cryptography

Most authentication data and cryptographic keys for user-facing services are configurable by the administrator.

A Spotfire system also uses cryptographic keys to bind together the internal components and services using connections requiring TLS client authentication. These keys are randomly generated by the services when the system is set up, therefore, they are unique to each Spotfire system. They cannot be modified by the purchaser, but the keys can be replaced by new random keys at any time.

Data At Rest

Data at rest is data stored, either temporarily or permanently. Data at rest has certain encryption types, or no encryption, depending on where it is being stored.

- Data in memory on the Spotfire Server, Spotfire Web Player or in the Spotfire Analyst clients is never encrypted.
- Data stored in the Spotfire database is not encrypted, except for especially sensitive data like passwords for service accounts, which are encrypted using AES-128 (Kerberos or LDAPS). User passwords are always hashed (by default, using PBKDF2) and never encrypted.
- Temporary files stored in the attachment manager on the Spotfire Server file system are encrypted. (One exception: the Information Services component's temporary pivot cache is not encrypted.) The default encryption algorithm is AES-128. Other possible options are AES-192 or AES-256. See config-attachment-manager (`--encryption-enabled` and `--encryption-key`) for more information.
- Temporary files stored on the Spotfire Web Player file system are not encrypted.
- Temporary files stored on the Spotfire Analyst file system are not encrypted.
- "Save my login information" stores the user's Spotfire login in an encrypted form using Microsoft's ProtectedData API (DPAPI) protected with the user scope.

Data In Motion

Data in motion is moving through the Spotfire environment. Data in motion has certain encryption protection, depending on how and where it is moving.

- Communication between the Spotfire Server and any backend services, like Spotfire Web Player, is always encrypted using Transport Layer Security (TLS).
- Data that is transported over the HTTP, LDAP, and JMX protocols can be secured by TLS. The TLS protocol version, the encryption algorithm, and the key strength is configurable using standard Java procedures. See Test or Revert changes to Oracle's JDK and JRE Cryptographic Algorithms in the Java documentation for more information. Also see:
  - Configuring HTTPS
  - Configuring LDAPS
  - config-jmx (`--tls-enabled`, `--need-client-auth`)
- Communication with the Spotfire database can be secured by either TLS or vendor-specific encryption protocols. See the documentation for your database vendor for more information about configuring the database server to accept only secured / encrypted connections.
- Communication with databases used as Information Services data sources can also be secured by either TLS or vendor-specific encryption protocols. See the vendor documentation for your database.

Standards and Algorithms

Spotfire provides the following standards and algorithms for encryption.

Purpose	Encryption/Hashing algorithm	Comment
Backend HTTP over TLS (HTTPS)	Default (with modern protocols and cipher suites enabled): TLS_AES_128_GCM_SHA256, TLS_AES_256_GCM_SHA384, TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 The following cipher suites are supported for backwards compatibility only: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256, TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384, TLS_DHE_RSA_WITH_AES_128_CBC_SHA256, TLS_DHE_RSA_WITH_AES_256_CBC_SHA256, TLS_AES_CBC_128_SHA256, TLS_AES_CBC_256_SHA256	The TLS protocol for Spotfire Server 11.4 and forward is TLSv1.3, when communicating with the node manager or the Java-based services, and TLSv1.2 when communicating with .NET-based services. (Previous versions of the node manager can use TLSv1.2, TLSv1.1 or TLSv1 before being upgraded.) If all (modern) protocols and cipher suites are enabled on the computer running the Spotfire Web Player service, then the cipher suite chosen for all communication is TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256. Support for the TLS_DHE_RSA_WITH_AES_* and TLS_AES_CBC_* cipher suites are kept only for backwards compatibility and will be removed in a later version.
Backend certificates	Asymmetric keys: automatically generated 2048-bit RSA keys (configurable for certificates representing TSS instances, but not configurable for other components). Signature algorithm: SHA256withRSA (configurable).	Keystore: PKCS12.
Data transfers	SHA-512, but also supports SHA-256, SHA-1 and MD5	For error-detection checksums in the Digest/Content-MD5 HTTP headers, as defined by RFC 3230 and RFC 1864.
Encryption of service passwords	AES-128	
External actions	SHA-512 and ASiC-E Containers (Associated Signature Containers)	External actions can be trusted based on hash value or signer certificates.
HTTP over TLS (HTTPS)	The TLS protocol version, the encryption algorithm and the key strength are configurable using standard Java procedures.	See JDK Providers Documentation.
Hashing of user passwords	PBKDF2	SHA-512, SHA-256 or SHA-1 can be used for password hashes created by older versions of Spotfire Server.
Information Link cache	SHA-256	For calculation of cache keys used for comparison.
JDBC over TLS	The TLS protocol version, the encryption algorithm and the key strength are configurable using standard Java procedures.	See JDK Providers Documentation.
JDBC using vendor-specific cryptography	The Oracle Database JDBC driver supports the following algorithms: Legacy: RC4-40, RC4-56, RC4-128, RC4-256, DES-40-CBC, DES-56-CBC, 3DES-112 and 3DES-168. Recommended: AES-128, AES-192 and AES-256.	See JDK Providers Documentation.

Purpose	Encryption/Hashing algorithm	Comment
JMX over TLS	The TLS protocol version, the encryption algorithm and the key strength are configurable using standard Java procedures.	See JDK Providers Documentation.

Kerberos/GSSAPI	Legacy: DES-CRC, DES-MD5, RC4-HMAC and AES-128-CTS-HMAC-SHA1-96.
	Recommended: AES-256-CTS-HMAC-SHA1-96.
		Uses the built-in Java support for the Kerberos and GSS-API protocols. See JDK Providers Documentation.
		If you must use RC4_HMAC (which is disabled in newer versions of Java), set `allow_weak_crypto = true` in the `[libdefaults]` section of `krb5.conf` on each Spotfire Server and specify the algorithms to use in `permitted_enctypes` similar to the following:
		`default_tkt_enctypes = <add other ciphers here> rc4-hmac`
		`default_tgs_enctypes = <add other ciphers here> rc4-hmac`
		`permitted_enctypes = <add other ciphers here> rc4-hmac`

LDAP over TLS (LDAPS)	The TLS protocol version, the encryption algorithm and the key strength are configurable using standard Java procedures.	See JDK Providers Documentation.
NTLM v2	According to the protocol specification.	
OAuth2	RSA-OAEP-256	For encryption of access and refresh tokens according the JWE standard (RFC 7516).
OAuth2	A128GCM	For encryption of access and refresh tokens according the JWE standard (RFC 7516).
OAuth2	SHA-256	For client verification according to the PKCE standard (RFC 7636).
Script trust hashes	SHA-512	JavaScript, custom queries, TERR scripts, R scripts, Python scripts, IronPython scripts, and other data functions are trusted based on hash value.
Server configurations	SHA-1	For error-detection checksums.
Software distributions files ("deployments")	SHA-1	For error-detection checksums.
Temporary data files	AES-128, AES-192 and AES-256	
Visualization mods	SHA-512 and ASiC-E Containers (Associated Signature Containers)	Visualization mods can be trusted based on hash value or signer certificates.

Spotfire Analyst, Spotfire Web Player, and Spotfire Automation Services

The applications in the Spotfire environment use the following encryptions.

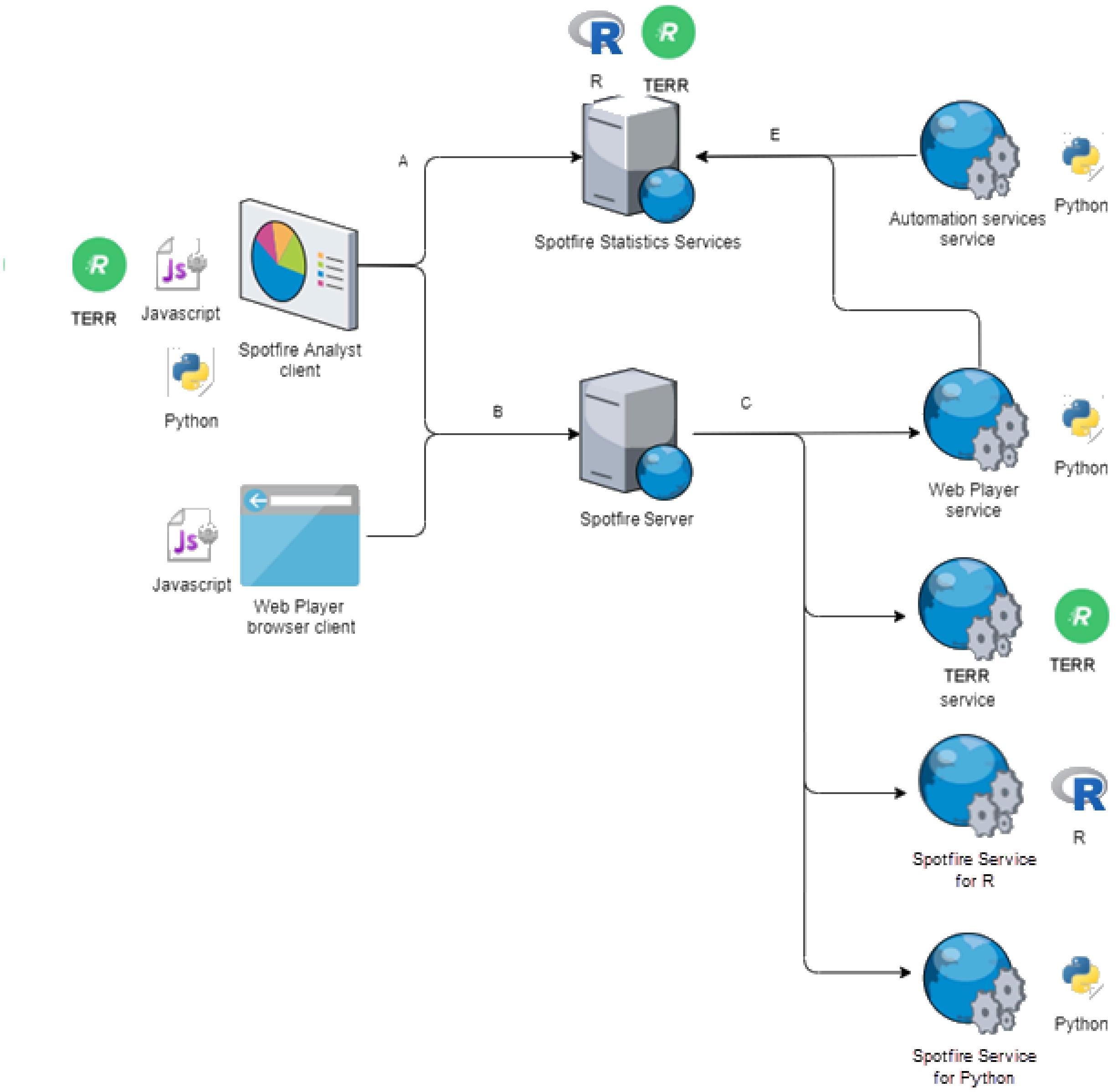
Purpose	Encryption/Hashing algorithm	Comment / References
Digital signatures and encryption of sensitive data and credentials	Strength and algorithm dependent on operating system version and configuration	On Windows: The Microsoft `ProtectedData` API (DPAPI) • Protected Data. Protect Method • Windows Data Protection • Data Protection API (Wikipedia) On Linux (Web Player and Automation Services): ASP.NET Core Data Protection Overview
Hash calculation (not for security purposes)	SHA-1 (160 bit)	The Microsoft `SHA1CryptoServiceProvider` • SHA1Crypto Service Provider Class
Script trust hashes	SHA-512	JavaScript, custom queries, TERR, R, Python, and IronPython scripts, and other data functions are trusted based on hash value.
Hash calculation (for security purposes)	SHA-256	The Microsoft `SHA256CryptoServiceProvider` • SHA256Crypto Service Provider Class
Hash calculation (for security purposes)	SHA-512	The Microsoft `SHA512CryptoServiceProvider` • SHA512Crypto Service Provider Class
Encryption of data that crosses computer boundaries	AES-256	The Microsoft `EncryptedXml.Encrypt` API • Encrypt(XmlElement, X509Certificate2)

Scripts in Spotfire

Spotfire supports a number of execution environments for a several programming languages: JavaScript, Python, R, TERR, or IronPython scripts, and custom queries (different database query languages). In addition, a limited subset of HTML is available in text areas of Spotfire files.

This diagram demonstrates the following.

- Local Python script execution by Spotfire Analyst.
- Local TERR script execution by Spotfire Analyst.
- Local execution of IronPython from Spotfire Analyst, Spotfire Web Player, or Spotfire Automation Services.
- Remote TERR script execution on Spotfire® Statistics Services on behalf of Spotfire Analyst.
- Remote R script execution on Spotfire® Statistics Services on behalf of Spotfire Analyst.
- Remote TERR script execution on TERR service, on behalf of Spotfire Analyst, Spotfire Web Player, and Spotfire Automation Services.
- Remote Python script execution on Spotfire Service for Python on behalf of Spotfire Analyst.
- Remote R script execution on Spotfire Service for R on behalf of Spotfire Analyst.
- JavaScript execution on Spotfire Analyst and in a web browser (file opened in Spotfire Web Player).
- Remote R script execution on Spotfire Service for R, on behalf of Spotfire Analyst, Spotfire Web Player, and Spotfire Automation Services.
- Remote Python script execution on Spotfire Service for Python, on behalf of Spotfire Analyst, Spotfire Web Player, and Spotfire Automation Services.



Not pictured: Spotfire Web Player and Spotfire Automation Services execute a data function on Spotfire® Statistics Services.

Spotfire Statistics Services can support one of four types of scripts, but they are not all not described further in this document. Use TERR service and Spotfire Service for R instead of Spotfire Statistics Services to run TERR and R data functions from Spotfire Web Player and Spotfire Automation Services.

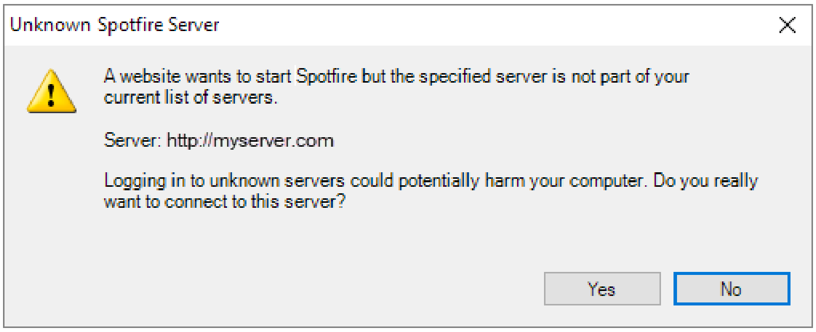
For this Spotfire release, the Spotfire Statistics Services server is discontinued and will no longer be updated. You can continue to use it but no bug fixes or security patches will be provided, and it will eventually be removed.

Script Trust

Only members of the Script Author group can save Spotfire files with scripts that are marked as trusted.

A file containing a trusted script is automatically executed when needed without first asking for end user consent. If the script is not trusted, the user is prompted to approve and manually trust the script for execution to prevent potentially harmful scripts.

Because the Spotfire Server tells a Spotfire client which scripts are trusted and which are not, a Spotfire client must not connect to unknown servers that the user does not trust. For this reason, the following pop-up is displayed if the user tries to connect to a server that has not been manually added to the list of known servers.



Unknown Spotfire Server

A website wants to start Spotfire but the specified server is not part of your current list of servers.

Server: http://myserver.com

Logging in to unknown servers could potentially harm your computer. Do you really want to connect to this server?

If the user does not trust the administrator of the Spotfire Server, then the user should click **No**. To limit the exposure of the infrastructure to the TERR, R, or Python script, you can configure the TERR service, Spotfire Service for R, or Spotfire Service for Python to run data functions in a Docker container on Linux. Alternatively, for TERR, you can run scripts in restricted execution mode.

Script Types

If the correct trust is in place, you can run any of these script types in Spotfire.

IronPython

IronPython scripts can access the capabilities available in the Spotfire Analyst API, and also other APIs provided by the operating system. These capabilities include running arbitrary commands; therefore, strict control must be employed to those users who are allowed to author and mark scripts as trusted in the library (such as members of the Script Author group).

Component	Description
Authorization	Members of the Script Author group can mark scripts as trusted to be executed by others.
Execution context	• The script is executed on the computer that opens the file, which can be either on the computer running Spotfire Web Player, the Spotfire Analyst client or Spotfire Automation Services, depending on where the file is opened. • The script is executed with privileges of the user who is currently logged in, or of the service account for which the service is set to run. In some cases where Kerberos with delegation is configured, the script executes in the end users' context.

JavaScript in Text Area

To customize parts of the application that cannot be done using sanitized HTML in the Spotfire text area, you can add snippets of JavaScript.

Component	Description
Authorization	Members of the Script Author group mark scripts as trusted for execution by others.

Component	Description
Execution context	JavaScript runs in a web browser that does not have direct access to the operating system API. It can use a subset of the functions provided by the Spotfire application for the user who is currently logged in. If a user opens a file containing trusted JavaScript on the Spotfire Web Player, then the script can access anything the user has permission to access in the domain running the Spotfire Server (according to a security policy in browsers referred to as same origin policy). For this reason, only trusted users should be members of the Script Author group.

HTML in Text Area

A subset of HTML is allowed in the text area visualization.

Component	Description
Authorization	By default, arbitrary HTML is not allowed in Spotfire because it would enable running JavaScript in the text area. The preference `PerformHTMLSanitation` can be set to `false`, which allows creating and viewing any HTML. Setting this preference to `false` is not recommended, because doing so allows any user to create a file with JavaScript code, bypassing all script trust mechanisms. See Supported HTML in the Text Area.
Execution context	If `PerforHTMLSanitation` is set to `false`, then HTML or JavaScript runs in a web browser that does not have direct access to the operating system API. It can use a subset of the functions provided by the Spotfire application for the user who is currently logged in. If a user opens a file containing trusted JavaScript on the Spotfire Web Player, then the script can access anything the user has permission to access in the domain running the Spotfire Server (according to a security policy in browsers referred to as same origin policy). For this reason, only trusted users should be member of the Script Author group.

Custom Queries

A normal query (not custom) issued by a Spotfire data connection can use only allowed constructs (for example `SELECT column FROM table`) in a way that is tightly controlled by the Spotfire connector. A data connection with a custom query does not limit the types of language constructs that are allowed, and enables use of any language construct (for example `INSERT`, `UPDATE`, `CREATE`), as well as other functions specific to the data source.

Component	Description
Authorization	• Only users that are members of the Custom Query Author group can create custom queries to be trusted by other users. • The database server normally allows only connections that are authenticated and authorized. Spotfire must provide the connection with credentials to the database server.
Execution context	• A custom query is executed on the database server and initiated from Spotfire Analyst, Spotfire Web Player, or Spotfire Automation Services. • The query runs with the permissions assigned to the currently-authenticated user by the database server.

TERR Data Functions

Spotfire® Enterprise Runtime for R (a/k/a TERR™) is an implementation of the R programming language that provides restricted and unrestricted execution environments. TERR data functions running in unrestricted mode have access to the operating system and can run arbitrary commands.

Component	Description
Authorization	- Members of the Script Author group can save data functions as trusted to be executed in unrestricted mode for other users. - Spotfire Statistics Services can be configured to require authentication. It runs as a separate product. - Spotfire® Enterprise Runtime for R - Server Edition (a/k/a the TERR™ service) runs in a node manager and is called using the Spotfire Server acting as a reverse proxy. It requires an authenticated Spotfire session.
Execution context	A TERR data function runs locally or remotely. Local execution takes place on the Spotfire client itself. Remote execution is when a TERR data function is sent off from a Spotfire client, Spotfire Web Player service, or Spotfire Automation Services service to a Spotfire Statistics Services service (a stand-alone product) or to TERR™ Server Edition (TERR service) (which runs on a node manager). For this Spotfire release, the Spotfire Statistics Services server is discontinued and will no longer be updated. You can continue to use it but no bug fixes or security patches will be provided, and it will eventually be removed.

R Data Functions

The R programming language provides execution environments for running R scripts in an R engine that you have installed on a node manager. R data functions have access to the operating system and can run arbitrary commands.

Component	Description
Authorization	- Members of the Script Author group can save data functions to be executed by other users. - Spotfire Service for R runs in a node manager and is called using the Spotfire Server acting as a reverse proxy. It requires an authenticated Spotfire session.
Execution context	An R data function runs remotely only. Remote execution is when an R data function is sent from a Spotfire client, Spotfire Web Player service, or Spotfire Automation Services service to Spotfire Service for R (which runs on a node manager).

Python Data Functions

The Spotfire Service for Python provides execution environments for running Python scripts in a Python interpreter installed on the node manager. Python data functions can have access to the operating system and can run arbitrary commands.

Component	Description
Authorization	- Members of the Script Author group can save data functions to be executed for other users. - The Spotfire Service for Python runs in a node manager and is called using the Spotfire Server acting as a reverse proxy. It requires an authenticated Spotfire session.

Component	Description
Execution context	A Python data function runs locally or remotely. Local execution takes place on the Spotfire client itself. Remote execution is when a Python data function is sent off from a Spotfire client, Spotfire Web Player service, or Spotfire Automation Services service to Spotfire Service for Python (which runs on a node manager).

Spotfire Visualization Mods

Spotfire visualization mods are visualizations created using web technologies such as JavaScript or TypeScript, that run in the provided framework within Spotfire clients. Running a mod involves code execution, therefore, provisions are in place to help users make trust decisions. Mods can be created and uploaded to a Spotfire library by any user with sufficient privileges, and trust for mods can be handled either by the server administrator or by end users, depending on how the environment has been configured.

The trust for mods is based on code signing by the developer of the mod. When mods are developed for a particular Spotfire environment, they can be signed by the user account that loads the mod project into Spotfire, but mods can also be signed using a certificate created by a certificate authority (CA). See Trusting custom content in the Spotfire environment in the Server and Environment - Installation and Administration manual for details about trust. For information about signing see Signing a visualization mod using Package Builder in the Spotfire Developer documentation.

Component	Description
Licenses	The license features for working with Spotfire visualization mods are located under Spotfire Extensions. • To create new visualization mods you need the **Develop Visualization Mod** license feature. • To open .mod files from the library you need the **Open Visualization Mod from Library** license feature. • To use local .mod files you need the **Open/Save Local Visualization Mod** license feature. • To be able to save .mod files to the library you need the **Save Visualization Mod to Library** license feature. • To be allowed to trust mods developed by others you need the **Trust Mods** license feature.
Execution context	Visualization mods run in a sandboxed iframe within the Spotfire clients. If a signer is trusted, mods developed by that signer will work the same way as native Spotfire visualizations. If a user opens a file containing a trusted visualization mod, then the code can access anything the user has permission to access. For this reason, only trusted users should be allowed to develop mods. If an untrusted visualization mod is accessed by a user who is allowed to trust mods, the user will be asked whether to trust the mod. It is then possible to choose to trust either that particular mod or to trust the signer. Once trusted, the mod will run for this particular user. Users who lack the permission to trust mods will not be able to use any untrusted mods at all.

If you suspect that a signature or a specific mod has been misused, there are several actions that can be taken depending on the situation:

Option	Description
Remove previous trust decisions	Any trust decision, taken by either the administrator or by an end user, can be withdrawn. If an administrator has configured a signer to be trusted for a specific group, this trust can be removed by clicking Revoke trust on the Trusted signers page for the group in the administration pages on the server. See Removing trusted signers from a group. Administrators can also remove trust using the `remove-code-trust` command. End users can also remove trust for any mods or signers that they previously trusted on their My account page, which can be reached via the Manage trust dialog in the client.

Option	Description
Invalidate signature (revoke certificate from server)	If there are suspicions that a user on the Spotfire Server has signed unsafe mods, it is possible to revoke the user's certificate, which renders signatures invalid. This prevents other users from making a trust decision based on false premises. When a certificate has been revoked, any mods that have been signed (after a specified time) will be considered invalid. An end user who tries to add a mod with an invalid signature will be informed that the signature has been invalidated. By default, mods with invalid signatures cannot be trusted in on-premises systems. An administrator can revoke the certificate for a user through the `revoke-code-signing-certificate` command, whereas an end user can revoke their own signatures on their My account page, reached from the Manage trust dialog.
Block certificate, user or item	If there are suspicions that a certificate from a CA or a specific visualization mod is being used for malicious purposes, it should be blocked from the system. An administrator can block either the certificate, a Spotfire user or a specific visualization mod through the `block-code-trust` command. If you select to block a specific mod then it might still be possible to trust and use an updated version of that mod. Note that any modification will be seen as an update from a trust perspective. See Blocking certificates, users or custom items for more information.

Spotfire External Actions

External actions in Spotfire are predefined configurations that allow end users to trigger actions to be executed on, or send data to, external systems made available using TIBCO Cloud™ Integration (TCI). Because the actions can affect external systems, and act on behalf of the user who runs an action, provisions are in place to help users make trust decisions.

The trust for actions is based on code signing by the configurator of the action.

When actions are configured within a Spotfire analysis, they are automatically signed by the user account that adds the action to the analysis, and others can use that signature to determine whether this signer is trusted. The actions in the external systems themselves are created using TCI, and the Spotfire configurator basically maps the parameters in TCI to some input provided from within the analysis. The person configuring the action must know how the endpoints in TCI work, and should add a proper description to help the end users understand what the action will do. Inputs can be either prompts for values to be entered by the end users, or based on some data selection in the analysis. Based on the information added, and the signature of the configurator, the end user can provide consent to run the action.

See Trusting custom content in the Spotfire environment in the Server and Environment - Installation and Administration manual for details about trust.

Component	Description
Licenses	The following license features for working with external actions are available: • To run an external action in Spotfire, you need the **Use External Actions** license feature, located under Spotfire Consumer. • To add and configure an external action, you need the **Configure External Actions** license feature, located under Spotfire Analyst. • To be allowed to trust external actions configured by others, you need the **Trust External Actions** license feature, located under Spotfire Extensions.
Execution context	If a signer is trusted for a group by an administrator, actions configured by that signer can be triggered by users in that group (with permission to run actions), and all actions by that signer will be executed without asking the user to trust the action. If an untrusted external action is triggered by a user who is allowed to trust external actions, the user will be asked whether to trust the action. It is then possible to choose to trust either that particular action or to trust the signer. Once trusted, the action will run for this particular user. Users who lack the permission to trust external actions will not be able to use any untrusted external actions at all. An end user always has the opportunity to view the consent dialog (Run external action) and get a preview of the data or information sent to the external system.

If you suspect that a signature or a specific action has been misused, there are several measures that can be taken depending on the situation:

Option	Description
Remove previous trust decisions	Any trust decision, taken by either the administrator or by an end user, can be withdrawn. If an administrator has configured a signer to be trusted for a specific group, this trust can be removed by clicking Revoke trust on the Trusted signers page for the group in the administration pages on the server. See Removing trusted signers from a group. Administrators can also remove trust using the `remove-code-trust` command. End users can also remove trust for any external actions or signers that they previously trusted on their My account page, which can be reached via the Manage trust dialog in the client.

Option	Description
Invalidate signature (revoke certificate from server)	If there are suspicions that a user on the Spotfire Server has signed unsafe external actions, it is possible to revoke the user's certificate, which renders signatures invalid. This prevents other users from making a trust decision based on false premises. When a certificate has been revoked, any actions that have been signed (after a specified time) will be considered invalid. An end user who tries to run an action with an invalid signature will be informed that the signature has been invalidated. By default, actions with invalid signatures cannot be trusted in on-premises systems. An administrator can revoke the certificate for a user through the `revoke-code-signing-certificate` command, whereas an end user can revoke their own signatures on their My account page, reached from the Manage trust dialog.
Block certificate, user or item	If there are suspicions that a specific external action is being used for malicious purposes, it should be blocked from the system. An administrator can block either a Spotfire user, or a specific external action through the `block-code-trust` command. If you select to block a specific action, then it might still be possible to trust and use an updated version of that action. Note that any modification will be seen as an update from a trust perspective. See Blocking certificates, users or custom items for more information.

Components

The Spotfire environment is composed of servers, services, applications, and tools that communicate and interact to produce visualizations and dashboards that can be shared through a web browser and exported to different formats.

Securing communication between the components of the Spotfire environment require planning and an understanding of each component. This section provides information about each component, its authentication protocols, and how it executes requests.

Spotfire Server

The Spotfire Server is the central component of the Spotfire environment, to which all Spotfire clients connect.

These tables provide reference for the security considerations for the Spotfire Server.

Spotfire Server component	Description
Service account	By default, the service is installed under the following, for the specified operating system: • Linux (RPM): `spotfire` • Linux (tar file): `root` • Windows: `NT AUTHORITY/System`
Ports and protocols	External communication port: • HTTP over 80/tcp • HTTPS over 443/tcp if enabled
Logs	`<spotfire server installation>/tomcat/logs`, See Spotfire server logs.

A non-extensive inventory of data that may contain credentials and other sensitive information

Type	(Default) location	Comments
Spotfire library exports	`<spotfire server installation>/tomcat/application-data/library/`	Default library export path. Can contain old export or backups of library content.
Spotfire server logs	`<spotfire server installation>/tomcat/logs`	See Logging and monitoring.
Spotfire temporary attachments	`<spotfire server installation>/tomcat/temp/AttachmentManager`	Encrypted attachments. Temporary storage for data uploaded and downloaded to the server by Spotfire clients.
Encrypted Spotfire database password for Spotfire Server	`<spotfire server installation>/tomcat/webapps/spotfire/WEB-INF/bootstrap.xml`	Used by Spotfire server during startup process to connect to database.
Spotfire library data	External library storage location, S3 or local file system, or in Spotfire database.	Only used if enabled. Default setting is to store library data in the Spotfire database.

Type	(Default) location	Comments
HTTPS keystore password	`<spotfire server installation>/tomcat/conf/server.xml`	If HTTPS is enabled, `server.xml` contains the password to the keystore (pkcs12 or jks) that contains the private certificate required to create a HTTPS listener.
Keystore for HTTPS certificates	`<spotfire server installation>/tomcat/certs`	PKCS12 (`.pfx`) or Java keystore (`.jks`) with private keys needed for HTTPS configuration.
Password hashes for end users	Spotfire database	Users' password hashes needed when Spotfire database is used as the authentication source. Default algorithm since Spotfire Server 7.5 is PBKDF2 (using HmacSHA512), 100000 iterations, 32 bytes of salt. Older algorithm still supported for upgraded system. From version 3.3 to 7.5: SHA-512, 2323 iterations, 16 bytes of salt. Default in 3.0 to 3.2: SHA-1, one iteration.
Encryption password	`<spotfire server installation>/tomcat/webapps/spotfire/WEB-INF/boostrap.xml`	The password is stored encrypted using AES-128 symmetric encryption using a static secret key. The password is used to encrypt service accounts passwords stored in Spotfire database. See config-encryption. If not set, a static password is used.
Service account passwords	Spotfire database and `configuration.xml`	Passwords for service accounts for services such as LDAP configuration, S3 configuration, OpenId Connect, Action Log database are encrypted AES-128 using an encryption password as secret key. `configuration.xml` is an exported copy of the effective configuration that resides in the Spotfire database. The file can safely be removed from the file system after having changed the Spotfire configuration in the database.
Information Services data source credentials	Spotfire database	Credentials for data sources used by Information Services (created using the Spotfire Analyst > Information Designer tool) are encrypted AES-128 using an encryption password as secret key.
Hashed passwords for JMX users	Spotfire database	If JMX is used, users credentials are stored in the Spotfire database.
Kerberos keytab	`<spotfire server installation>/Spotfire.keytab`	Used if Spotfire is configured for Kerberos authentication. The keytab file contains encrypted credentials that can be used to authenticate to remote systems.
Spotfire Server Backend trust keystore	`<spotfire server installation>/nm/trust/keystore.p12`	Keystore needed for back-end trust encrypted TLS communication. The keystore is locked with a static password.

Type	(Default) location	Comments
Passwords embedded in Spotfire files	Spotfire database (library)	The Spotfire database may contain Spotfire files (`.dxp`) with embedded credentials to data sources. Passwords are not encrypted because the password must be made available to end users who access the file. We do not recommend embedding credentials in the file. The preference `EnableAllowSavingDatabaseCredentials` can be used to disable the option to embed credentials in Spotfire files.
Library exports	`<spotfire server installation>/tomcat/application-data/library`	Can contain zip-files containing exported library content. Data source passwords for information services data sources are not included in the library exports. However, Spotfire analysis files (`.dxp`) in the exported zip can contain embedded passwords.
Database installation script	No default location. From where they were run.	Database installation scripts will contain credentials and connection information to the Spotfire Server database when they are run. These files will contain sensitive information and should be deleted when no longer needed or stored in a safe location.
OAuth2 API Clients credentials		The credentials are encrypted.

Spotfire Server Security Configuration and Administration Activities

This table provides information about configuration activities, security settings, and links into the documentation and community site.

Activity	Description or references
"Remember me" in Spotfire Analyst	Default: Enabled. See config-login-dialog --allow-remember-me. Controls whether users can select to store the log in information for future automatic login, or if they must always provide username and password when logging in.
Apache Ignite - TLS (Spotfire server clustering communication)	Default: Enabled. TLS can be disabled or enabled. See config-cluster --secure-transport=<true|false>.
Backend communication - Auto-trust	Default: Disabled - If enabled, node managers are automatically trusted by the server cluster. See Automatically trusting new nodes for more information.
Configure Encryption password	The encryption password is used to encrypt service account passwords stored in the Spotfire database. If not set, a static password is used. See config-encryption for more information.
Configure Spotfire server database security	See the following help topics for more information. • Using Kerberos to log in to the Spotfire database • Setting up the Spotfire database (SQL Server with Integrated Windows authentication)
Cross-site request forgery (CSRF) - Public web services	See config-csrf-protection for more information.
HTTP - Security headers	See Security HTTP headers.

Activity	Description or references
HTTPS (TLS over HTTP) for front end port	See HTTPS (TLS over HTTP) for Front End Port.
JMX Security	JMX Security
LDAP - SASL authentication	Spotfire Server supports two Simple Authentication Socket Layer (SASL) mechanisms for authentication towards LDAP: DIGEST-MD5 and GSSAPI. See Authentication towards LDAP.
Session handling - Maximum concurrent sessions	Default: unlimited. See Managing active user sessions for information on configuring a limited number of active sessions for named users and guest (anonymous) users. See also Session Management on page 20.
Session handling - Persistent sessions	Default: Enabled. See config-persistent-sessions for information on configuring persistent sessions for browser clients. See also Session Management on page 20.
Session handling - Timeouts	Default: 30 minutes (session), 24 hours (absolute). See Absolute session timeout and idle session timeout for more information. See also Session Management on page 20.
Setting LDAP - LDAP over TLS	Configuring LDAPS. In an LDAP environment, where the Spotfire system communicates with an LDAP directory server, administrators often secure the LDAP protocol using TLS, if the LDAP directory supports this. See Authentication towards LDAP.
Upgrade Java	If desired, you can manually upgrade Java. See Switching to another Java Development Kit for the Spotfire Server for more information.
Upgrade Spring	See Upgrade Spring for Spotfire Server 7.5 and later on the Community.
Upgrade Tomcat	See Upgrade Apache Tomcat for Spotfire Server 7.5 and later on the Community.

Changing a Windows Service Account for Spotfire Server

The service account running the Spotfire Server under Windows can be changed to a user with more restricted rights.

Change this setting from the Windows Services user interface.

Procedure

1. Right-click **Spotfire Server** > **Properties** > **Login On tab** > **Log on as**.

   The specified user must have read and write permissions to the files in the Spotfire Server installation path.

2. Set file system permissions restrictively and apply minimal permissions.

   The service account must have both read and write permissions in the installation folder. Other users on the system do not need access to files in the installation folder.

HTTPS (TLS over HTTP) for Front End Port

The file `<spotfire server installation>/tomcat/conf/server.xml` contains the TLS configuration for the HTTPS.

In this version of Spotfire, the `server.xml` file is aligned with Mozilla's Modern Compatibility configuration. The ciphers and protocols in the configuration file can be adjusted to accommodate for environment specific needs.

```
<Connector port="443"
             [...]
              SSLEnabled="true"
              scheme="https"
              secure="true">
     <SSLHostConfig certificateVerification="none"
                   [...]
                    sslProtocol="TLS"
                    protocols="TLSv1.2+TLSv1.3"
                    honorCipherOrder="true"
                    ciphers="<cipher-suites>"
                    [...]
     </SSLHostConfig>
   </Connector>
```

See Configuring HTTPS for how to enable HTTPS for front end communication, between Spotfire clients and Spotfire Server.

See Authentication using X.509 client certificates for how to enable HTTPS client certificate authentication.

Security HTTP Headers

The HTTP headers listed in this topic can be set using Spotfire configuration settings.

See the header help topics, linked from the table, for detailed instructions for configuring the header.

Header	Default value	Comment
X-Frame-Options	Not set	Prevents clickjacking and framing of the Spotfire Server web interface by other web sites. If enabled (set to DENY), then the Spotfire Web Player JavaScript API stops working. See Mozilla's reference for X-Frame-Options for more information.
Strict-Transport-Security (HSTS)	Not set	Instructs the client that it should be accessed only using HTTPS, instead of using HTTP. See Mozilla's reference for Strict-Transport-Security for more information.
Cache-Control		Sets directives for caching mechanisms in requests and responses. See Mozilla's reference for Cache-Control for more information.
X-Content-Type-Options	nosniff	Prevents browser mime-sniffing in some cases. See Mozilla's reference for X-Content-Type-Options for more information.
SameSite Cookie Attribute	Unset	Used in cases where Spotfire Server cookies are used as third-party cookies. For example, it might be needed when external web sites and Spotfire are interacting. See the W3C specification and related documents of rfc6265bis for more information.
Content-Security-Policy	A default is set but is subject to change. Current policy is logged on INFO level during startup of the server.	Can be used to detect and mitigate some types of attacks, including Cross-Site Scripting (XSS) and data injection attacks. See Mozilla's reference for Content Security Policy for more information.

Adding Custom HTTP Headers in the Spotfire Server Configuration

Other HTTP headers, such as `Referrer-Policy`, and `Public-Key-Pins` (HTTP Public Key Pinning / HPKP), do not have built-in commands to configure. They can be added as custom headers in the Spotfire Server configuration by using the following steps.

Procedure

1. Export the configuration to an XML file.
2. Open the configuration XML file in a text editor, and then add the following tag with content.

```
<security>
   ...
   <headers>
     <directives>
       <directive>
         <action>add</action>
         <enabled>true</enabled>
         <name>headername</name>
         <value>value</value>
       </directive>
     </directives>
     <properties />
   </headers>
```

3. Replace *`headername`* with the name of the HTTP header and *`value`* with the header value, and, if needed, replace `add` with another action type.

   The allowed values for `<action>` are:

   - `add`
   - `append`
   - `set`
4. Issue `config import-config --comment "HTTP header <`*`action`*`>"` (with `<`*`action`*`>` reflecting the appropriate action type).
5. Restart the server.

Spotfire Node Manager

A node manager is a container for setting up, running, or tearing down services such as Spotfire Automation Services, Spotfire Web Player, the TERR service, or the Spotfire Service for Python. A service running on a node manager runs in a separate process, can open service ports and the service installation files resides under `<`*`nm installation path`*`>/services/`.

Spotfire node manager component	Description
Service account	- Windows default: `NT Authority\system` - Linux default: `spotfire`

Spotfire node manager component	Description
Ports and protocols	• Registration port on node manager computer: HTTP/9080 • Communication port on node manager computer: HTTPS/9443

A non-extensive inventory of data that might contain credentials and other sensitive information.

Type	(Default) Location	Comments
Node manager and service logs	`<node manager installation>/logs`	Contains the node manager and the service logs. It can also contain minidumps, and memory process dumps for Spotfire Web Player, if these are created. See also `enabledMiniDumpCreationOnError` in the `Spotfire.Dxp.Worker.Web.config` help topic.
SMTP configuration credentials	Spotfire.Dxp.Worker.Automation.config	When Spotfire Automation Services is configured with an SMTP server that requires authenticated connections.
Node manager backend trust keystore	`<node manager installation>/trust/keystore.p12` (node manager).	Keystore containing keys for the following: • internal node manager <-> Spotfire Server • node manager <-> service • service <-> Spotfire Server • service <-> service The keystore is locked with a static password.
Spotfire Web Player / Spotfire Automation Services proxy server credentials	Spotfire.Dxp.Worker.Host.exe.config	`ProxyUsername` and `ProxyPassword` hold credentials to a network proxy if one is configured.
Spotfire Statistics Services configuration for Spotfire Web Player and Spotfire Automation Services.	`Spotfire.Dxp.Worker.Host.exe.config`	`TibcoSpotfireStatisticsServicesUsernames` and `TibcoSpotfireStatisticsServicesPasswords` in contains credentials to Spotfire Statistics Services servers if one or more is configured. For this Spotfire release, the Spotfire Statistics Services server is discontinued and will no longer be updated. You can continue to use it but no bug fixes or security patches will be provided, and it will eventually be removed.
Credentials Profiles for Connectors used by Spotfire Web Player and Spotfire Automation Services	`Spotfire.Dxp.Worker.Host.exe.config`	A configuration file containing user names and passwords to data sources used by data connectors.
Spotfire Automation Services Kerberos identity	`Spotfire.Dxp.Worker.Automation.config`	On Windows only: The Windows user specified by `<kerberosIdentity userName="domain\username" password="password" />` is used to run Spotfire Automation Services.

Type	(Default) Location	Comments
Spotfire Web Player > Scheduled updates identity	Spotfire.Dxp.Worker.Web.config	On Windows only: The Windows user specified by `<kerberosIdentity userName="`*`domain`*`\`*`username`*`" password="`*`password`*`" />` is used to run Spotfire Web Player.

Credentials are encrypted in the configuration files that are installed with the service. To modify the configuration, you must export the configuration from the database, make modifications, import it back into the database, and then set the configuration for the service.

Node Manager Configuration Tasks

A Java Development Kit and Spring Framework are bundled with the node manager. They are upgraded to newer versions together with the regular node manager upgrades; however, to run the most recent and most secure version, you can review the versions and upgrade as necessary. The articles listed in this topic can help guide you.

- Switching to another Java Development Kit for the node manager in the *Spotfire® Server and Environment - Installation and Administration* Guide.
- Upgrade Spring for Spotfire Node Manager on the Community.

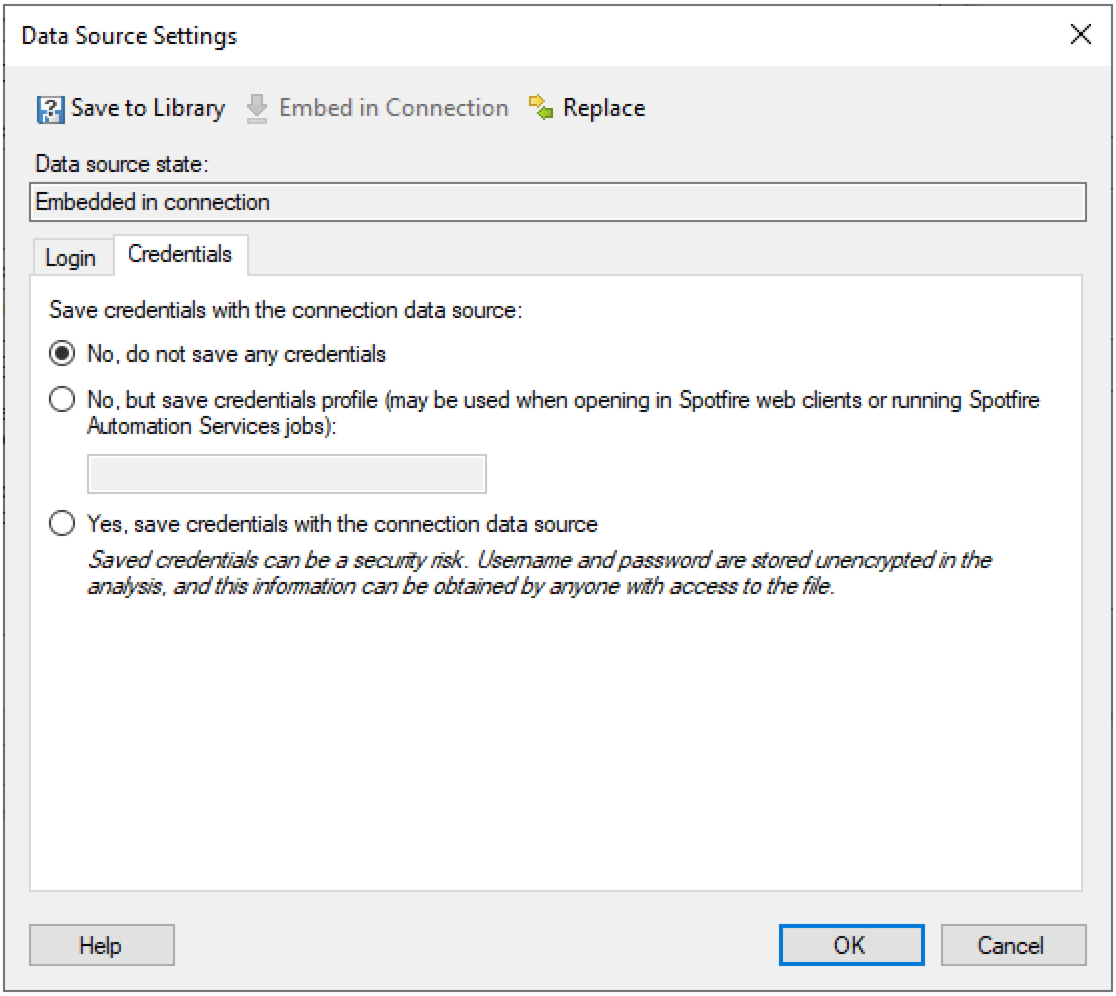
Spotfire Connectors

Spotfire connectors support a variety of authentication and transport security options.

See the documentation for each connector to see available security options.

Database Credentials for Connectors

Because the database connections using Spotfire connectors are initiated directly from the Spotfire client (Spotfire Analyst, Spotfire Web Player, or Spotfire Automation Services), it is important to understand that any database credentials must be available to the client to establish the connection. Spotfire connector data source credentials settings control whether credentials are embedded in the connection or not.



Data Source Settings

Data source state:

Embedded in connection

Save credentials with the connection data source:

● No, do not save any credentials

○ No, but save credentials profile (may be used when opening in Spotfire web clients or running Spotfire Automation Services jobs):

○ Yes, save credentials with the connection data source

Saved credentials can be a security risk. Username and password are stored unencrypted in the analysis, and this information can be obtained by anyone with access to the file.

Option	Description
No, do not save any credentials	Use this option if you do not want to save credentials with the connection data source. If the connection data source uses database authentication, all users of the data source are prompted for user name and password for the database when this data source (or a data connection using it) is opened.
No, but save credentials profile (may be used when opening in Spotfire web clients or running Spotfire Automation Services jobs)	Use this option if you want to save a credentials profile instead of saving the actual credentials with the connection data source. See Details on Data Source Settings - Credentials in the Spotfire documentation for more info how to use credentials profiles. See `Spotfire.Dxp.Worker.Host.exe.config file > DataAdapterCredentials` for information on how to configure Spotfire Automation Services services and Spotfire Web Player to use a credentials profile. Use `<Spotfire.Dxp.Web.Properties.Settings>` and `<Spotfire.Dxp.Data.Access.Adapters.Settings>` in Spotfire.Dxp.Worker.Host.exe.config.

Option	Description
Yes, save credentials with the connection data source	Saving credentials with the connection data source can be a security risk because the user name and password are stored as part of the analysis file, and anyone with access to the file can obtain this information. Use this option carefully. If you do save credentials with the connection data source, a recommended practice is to use a database user that has only the minimum required privileges for reading the data that you want to analyze in Spotfire. Select this check box if you want the connection data source to remember the specified username and password. This means that users will not be prompted for credentials when opening a data connection which uses this data source or an analysis which includes such a data connection. This option can only be used if the connection data source is set to use database authentication. See also the preference `EnableAllowsavingDatabaseCredentials`. When this preference is set to `False`, then the option **Yes, save credentials with the connection data source** is disabled in the user interface.

Client ID and secret for the Microsoft SharePoint Online Connector

For the Microsoft SharePoint Online connector, Spotfire installed clients use a default client ID and client secret from a Spotfire app in Microsoft Azure AD. For the web clients, you add your own client ID and secret.

You can add the client ID and client secret information in two ways:

- With settings in the web player service configuration, in the file Spotfire.Dxp.Worker.Host.exe.config.
- With preference settings in the Administration Manager.

Information that you add in preference settings is not secret, so do not add sensitive client IDs and client secrets as preferences.

For more information, see Enabling the Microsoft SharePoint Online Connector in Spotfire Web Clients.

Spotfire Web Player

Spotfire Web Player is a service that runs on a node. It provides a web service for sharing and distributing analyses inside and outside of an organization.

See Manually Editing the Service Configuration Files for more information.

Spotfire Web Player component	Description
Service account	Default: `NT Authority\System`.
Ports and Protocols on page 6	Next available general purpose `950<x>/tcp`. (For example, 9501/tcp, or 9502/tcp, and so on, depending on the other services installed.)

Configuration File Settings for Spotfire Web Player

These tables provide information about the configuration files for Spotfire Web Player and its interactions with Spotfire Server and Spotfire Automation Services using APIs.

Spotfire.Dxp.Worker.Web.config

For more information, see Spotfire.Dxp.Worker.Web.config help.

Setting	Default value	Description
/javascriptApi -`<javaScriptApi enabled="true" domain="domain1.com,domain2.com">`	JavaScript API enabled, all domains allowed	Controls whether the use of the JavaScript API is enabled or not enabled, and from which domains it is possible to use the JavaScript API. A non-empty domain whitelist indicates that only listed domains are able to embed Spotfire files in their web site using the JavaScript API. The list is a comma-separated list of domain names.
`/analysis/inactivityTimeout`	2 hours	Timeout for inactive analyses. A Spotfire file is closed after the `inactivityTimeout` is reached. In practice, a session timeout is not shorter than the `inactivityTimeout` value because an open analysis file in a web browser continuously renews the session, so the session timeout is not met. Only after the session has no open files left, and the user session is not actively connected to Spotfire Server, the session timeout starts counting. This design ensures that every HTTP request renews the session.

Spotfire.Dxp.Worker.Core.config

This configuration file specifies settings for the service's communication with the Spotfire Server, and if sections in configuration files should be encrypted. For more information, see Spotfire.Dxp.Worker.Core.config help.

Setting	Default value	Description
`/cryptography@ encryptConfigurationSections`	`true`	Set to `true` to encrypt sections of configuration files containing sensitive information.
`/cryptography@ DataProtectionConfiguration Provider`	`DataProtectionConfiguration Provider`	On Windows: By default the `DataProtectionConfiguration Provider` uses Windows Data Protection API (DPAPI) to encrypt sections of the configuration with a machine-specific secret key which means that the encrypted sections can only be decrypted from the same machine as the service is running on. See Encrypting Configuration Information Using Protected Configuration for more information. On Linux: Our own provider is used.

Spotfire.Dxp.Worker.Host.exe.config or Spotfire.Dxp.Worker.Host.dll.config

`Spotfire.Dxp.Worker.Host.exe.config` is the configuration file for both Spotfire Web Player and Spotfire Automation Services on Windows. When running on Linux, the config file is called `Spotfire.Dxp.Worker.Host.dll.config`. See Spotfire.Dxp.Worker.Host.exe.config file and Spotfire.Dxp.Worker.Host.dll.config help for more information.

Setting	Default value	Description
`/Spotfire.Dxp.Internal.Properties.Settings/AllowedTlsVersions`	`Tls, Tls11, Tls12`	Determines which versions of the TLS security protocol are allowed. Specify the values separated by a comma ",". For information about the possible values for this setting, refer to the .NET enum `SecurityProtocolType`. If you leave the value for this setting blank, the allowed TLS versions are set to `SystemDefault`. If you remove the setting from the configuration file, the allowed TLS versions are set to the default value.
`/Spotfire.Dxp.Data.Properties.Settings/AllowedFilePaths`	Empty	A list of directories that Spotfire Web Player or Spotfire Automation Services are allowed to use as file data sources. Add only approved network shares or other paths that contain files that should be possible to load in a Spotfire file. For security reasons, you should not add entire drive letters such as C:\ because that would allow Spotfire users to read local files from the Spotfire Web Player service. The names are checked in a case-insensitive manner.
`/system.net/defaultProxy`		On Windows: If the Spotfire Web Player or Spotfire Automation Services should use a proxy server to reach internal and external networks, one can be enabled in this file.

Spotfire Automation Services

Spotfire® Automation Services is a web service for automatically executing multi-step jobs within your Spotfire® environment. You can, for example, use Spotfire® Automation Services to deliver an analysis file to specific people, in a particular format, at specified times.

Spotfire Automation Services component	Description
Service account	Default: `NT Authority\System`
Ports and protocol	Next available general purpose `950<`*`x`*`>/tcp`. (For example, 9501/tcp, or 9502/tcp, and so on, depending on the other services installed.)
Log files	*`<node manager installation directory>`*`/logs/,`*`<node manager installation directory>`*`/services/`*`<automation services service directory>`*`/logs`

Configuration File Settings for Spotfire Automation Services

These tables provide information about the configuration files for Spotfire Automation Services and its interactions with Spotfire Server and Spotfire Automation Services using APIs.

Spotfire.Dxp.Worker.Automation.config

This configuration file is used for configurations that are specific to Spotfire Automation Services.

Setting	Default value	Description
`/Spotfire.Dxp.Automation.Framework/security/allowedFilePaths@allowAll`	`True`	By default, Spotfire Automation Services tasks can read files from, and write files to any directory in the file system. Set this to `False` to allow only tasks to read from, and write to, directories specified in the \ `<allowedFilePaths>` section.
`/spotfire.dxp.automation.tasks/smtp` - SMTP Configuration	Not enabled	An SMTP server can be set up to use TLS (`useTls`) or different methods of authentication.
`/Spotfire.Dxp.Automation.Framework/allowedFilePaths`	All paths are allowed	By default, Spotfire Automation Services tasks can read files from, and write files to, any directory in the file system. Set this to `False` to allow only tasks to read from, and write to, directories specified in the `<allowedFilePaths>` section. (Not to be confused with `<allowedFilePaths>` in `Spotfire.Dxp.Worker.Core.config`.)

Spotfire.Dxp.Worker.Core.config

This configuration file specifies settings for the service's communication with the Spotfire Server, and if sections in configuration files should be encrypted.

Setting	Default value	Description
`/cryptography@encryptConfigurationSections`	`true`	Set to `true` to encrypt sections of configuration files containing sensitive information.
`/cryptography@DataProtectionConfigurationProvider`	`DataProtectionConfigurationProvider`	On Windows: By default the `DataProtectionConfigurationProvider` uses Windows Data Protection API (DPAPI) to encrypt sections of the configuration with a machine-specific secret key which means that the encrypted sections can only be decrypted from the same machine as the service is running on. See Encrypting Configuration Information Using Protected Configuration for more information. On Linux: Our own provider is used.

Spotfire.Dxp.Worker.Host.exe.config or Spotfire.Dxp.Worker.Host.dll.config

`Spotfire.Dxp.Worker.Host.exe.config` is the configuration file for both Spotfire Web Player and Spotfire Automation Services on Windows. When running on Linux, the config file is called `Spotfire.Dxp.Worker.Host.dll.config`. See Spotfire.Dxp.Worker.Host.exe.config file and Spotfire.Dxp.Worker.Host.dll.config help for more information.

Setting	Default value	Description
`/Spotfire.Dxp.Internal.Properties.Settings/AllowedTlsVersions`	`Tls, Tls11, Tls12`	Determines which versions of the TLS security protocol are allowed. Specify the values separated by a comma ",". For information about the possible values for this setting, refer to the .NET enum `SecurityProtocolType`. If you leave the value for this setting blank, the allowed TLS versions are set to `SystemDefault`. If you remove the setting from the configuration file, the allowed TLS versions are set to the default value.
`/Spotfire.Dxp.Data.Properties.Settings/AllowedFilePaths`	Empty	A list of directories that Spotfire Web Player or Spotfire Automation Services are allowed to use as file data sources. Add only approved network shares or other paths that contain files that should be possible to load in a Spotfire file. For security reasons you should not add entire drive letters such as C:\ because that would allow Spotfire users to read local files from the Spotfire Web Player service. The names are checked in a case-insensitive manner.
`/system.net/defaultProxy`		On Windows: If the Spotfire Web Player or Spotfire Automation Services should use a proxy server to reach internal and external networks, one can be enabled in this file.

Client Job Sender (Spotfire Automation Services)

The Client Job Sender command-line tool which can be used for executing Spotfire Automation Services jobs.

The tool has a number of security configuration options. See section Configuring the Client Job Sender in the *Spotfire Automation Services User Manual* for a full list of settings.

Spotfire Enterprise Runtime for R - Server Edition

Spotfire® Enterprise Runtime for R (a/k/a TERR™) provides Spotfire clients with the ability to execute R code, using Spotfire Enterprise Runtime for R, on the TERR service node.

A Spotfire Enterprise Runtime for R - Server Edition service (a/k/a, a TERR™ service) is required to execute data functions in Spotfire files from Spotfire Automation Services and Spotfire Web Player, because those services do not have TERR engines.

The TERR service itself is running the service as the same user account as is running the node manager on which the service runs. See Node Manager.

By default, scripts executed by the TERR service on behalf of its users are executed in a different execution context, as explained here.

TERR service component	Default	Description
Service account	`NT Authority\System` or `spotfire` (Linux)	Default: `NT Authority\System` or `spotfire` (Linux)

TERR service component	Default	Description
Log files		*<node manager installation>*/logs See the topic Service Logs in the *Spotfire® Enterprise Runtime for R - Server Edition Installation and Administration* guide for more information.

The Spotfire Enterprise Runtime for R – Server Edition ports and protocols

Name	Default Port and Protocol	Function Description	Secure/Encrypted
Communication port	Next available general purpose `950<x>/tcp`. (For example, 9502/tcp, or 9503/tcp, and so on, depending on the other data function services installed.)	For secure (HTTPS) internal communication. Cannot be accessed directly.	Yes
TERR engine ports	`61001/tcp` -> `62000/tcp`	Host-internal communication between the TERR service and the TERR engines.	No

Settings and Configuration Tasks for Spotfire Enterprise Runtime for R - Server Edition

You can use these settings to limit the capabilities of running data functions based on Spotfire® Enterprise Runtime for R (a/k/a TERR™).

Setting / Configuration task	Default value	Description
`terr.restricted.execution.mode` (Enforce restricted execution)	`TRUE`	Enforce restricted execution mode for all scripts. Restricted execution mode in the service allows executing arbitrary scripts without worrying that the script could do malicious things, such as deleting files or uploading confidential data to a server over the internet. For more information, see the topic Safeguarding Your Environment in the *Spotfire® Enterprise Runtime for R - Server Edition Installation and Administration* guide.
`use.engine.containers`	- Windows: `FALSE` - Linux: `TRUE`	Available on Linux only. If your deployment is on a Linux server, then the default configuration for the service is to use containers (the property `use.engine.containers: TRUE`). Running the service with containers enabled prevents the engines from having access to the host system. See the topic Containerized Service in the *Spotfire® Enterprise Runtime for R - Server Edition Installation and Administration* guide for more information.
`disable.spotfire.trust.checks`	`FALSE`	Disable the trust check only if the service is installed on Linux, with Docker containers, where extra means have been taken to secure the container environment, or if all Spotfire users in the environment can be trusted.

Setting / Configuration task	Default value	Description
Set file size upload limit	100MB	See the topic File Size Upload Limit in the *Spotfire® Enterprise Runtime for R - Server Edition Installation and Administration* guide for more information.
Set TERR engine ports range	61001 - 62000	See Engine Ports in the *Spotfire® Enterprise Runtime for R - Server Edition Installation and Administration* guide for more information.
Enable JMX Monitoring	OFF	See Monitoring the service using JMX in the *Spotfire® Enterprise Runtime for R - Server Edition Installation and Administration* guide for more information.

Restrict Network Access for TERR Scripts in Containers

By default, the containers in which Spotfire® Enterprise Runtime for R (a/k/a TERR™) scripts are running have access to network resources given to it.

If Spotfire Enterprise Runtime for R scripts are not running in restricted execution (REX) mode, then any TERR scripts can connect to the network. To restrict external network access for the container, and therefore any scripts running within it, the node manager computer must be configured in such a way that the containers cannot reach the network. One way to do this is by implementing iptables rules that block traffic from Docker containers to outside networks.

Use a Custom Docker Image for Containerized TERR

If the node manager is running on a Linux computer, then you can run Spotfire® Enterprise Runtime for R - Server Edition (a/k/a the TERR™ service) in a Docker container.

For more information, see the following help topics in the *Spotfire® Enterprise Runtime for R - Server Edition Installation and Administration* guide.

- Configuring a Custom Docker Image on a Node with Internet Access
- Pulling a Custom Docker Image from an Authenticated Repository

Script Security & Restricted Execution Modes

The following mechanisms control security of the Spotfire® Enterprise Runtime for R - Server Edition (a/k/a the TERR™ service) and prevent users from running malicious scripts on the server.

- Restricted execution mode (REX).
- TERR engine in Docker containerization.
- Script trust and access control.

Only users in the Spotfire license group `Script Author` can create and mark Spotfire Enterprise Runtime for R scripts as trusted. For other users to run the scripts, the scripts must be trusted (through the Manage Trust mechanism in Spotfire). Trusted scripts run in an unrestricted execution environment (no REX or container) unless the Spotfire Enterprise Runtime for R - Server Edition enforces all scripts to be run in restricted mode. Untrusted scripts always run in REX mode or in a container.

Docker Containerization for TERR Scripts

Spotfire® Enterprise Runtime for R (a/k/a TERR™) scripts running in a container but not using restricted execution mode have full access to the Docker container and have permission to do anything that is possible to do from within the container. The level of isolation a container provides depends on the Docker installation and the privileges given to these containers.

Configuration	Description
TERR service host isolation	Scripts are prohibited from accessing the file system of the host computer running the service.
User isolation	The use of engine containers ensures that the same execution environment is not re-used for multiple data functions initiated by different users.
Network isolation	Depending on configuration, the Spotfire Enterprise Runtime for R scripts can access external network and other Docker containers that are available from within a container. In many cases, a default installation with engine containers lets scripts access the external network, including the internet, and to access other Docker containers. To restrict access to the network, the Docker containers must be configured to restrict network access. The container options should not be used without `terr.restricted.execution.mode=true` or additional network configuration, if network isolation is needed.

TERR Restricted Execution Mode (REX)

Scripts running in restricted execution mode (REX), but without container isolation, are running directly on the Spotfire Enterprise Runtime for R - Server Edition host using the same user account as is running the node manager on which the service runs.

The scripts are restricted in their capabilities (see `terr.restricted.execution.mode` under Safeguarding Your Environment in the *Spotfire® Enterprise Runtime for R - Server Edition Installation and Administration* guide). Enforcing all scripts to be running in both restricted execution mode and in container isolation provides an extra level of security and is recommended to achieve the highest level of security.

Impact of Relaxing the Spotfire Enterprise Runtime for R - Server Edition Security Settings

If you have scripts that cannot run in restricted mode because they need access to resources on the system or network, then you can change the settings to enable those scripts to run.

This table shows the resulting execution mode, given user role, service configuration, and whether the script is marked as trusted in the library.

Script Author	terr.restricted.execution.mode	disable.spotfire.trust.checks	Trusted Script	Use evalREX
*	`True`	*	*	Yes
Yes	`False`	*	*	No
No	`False`	`True`	*	No
No	`False`	`False`	True	No
No	`False`	`False`	`False`	Yes

A data function based on Spotfire Enterprise Runtime for R runs without `evalREX` only if `terr.restricted.execution.mode` is `False` and one of the following conditions also exists.

- The data function is trusted in the Spotfire library.
- The request to run the data function originates from a member of the **Script author** group.
- The Spotfire Enterprise Runtime for R - Server Edition is configured with `disable.spotfire.trust.checks=True`.

Spotfire Service for R

The Spotfire Service for R provides Spotfire clients with the ability to execute open-source R code, using the user-installed R engine, on the R service node.

Spotfire Service for R is required to execute data functions in Spotfire files from Spotfire Automation Services and Spotfire Web Player, because those services do not have R engines.

The Spotfire Service for R itself is running the service as the same user account as is running the node manager on which the service runs. See Node Manager.

By default, R scripts executed by the Spotfire Service for R on behalf of its users are executed in a different execution context, as explained here.

Python service component	Default	Description
Service account	`NT Authority\System` or `spotfire` (Linux)	Default: `NT Authority\System` or `root` (Linux)
Log files		`<node manager installation>/logs` See the topic Service Logs in the *Spotfire® Service for R Installation and Administration* guide for more information.

The Spotfire Service for R ports and protocols

Name	Default Port and Protocol	Function Description	Secure/Encrypted
Communication port	Next available general purpose `950<x>/tcp`. (For example, 9502/tcp, or 9503/tcp, and so on, depending on the other data function services installed.)	For secure (HTTPS) internal communication. Cannot be accessed directly.	Yes
R engine ports	`63001/tcp` -> `64000/tcp`	Host-internal communication between Spotfire Service for R and the R engine.	No

Settings and Configuration Tasks for Spotfire Service for R

You can use these settings to limit the capabilities of running R data functions.

Setting / Configuration task	Default value	Description
`use.engine.containers`	- Windows: `FALSE` - Linux: `TRUE`	Available on Linux only. If your deployment is on a Linux server, then the default configuration for Spotfire Service for R is to use containers (the property `use.engine.containers: TRUE`). Running Spotfire Service for R with containers enabled prevents the engines from having access to the host system. See the topic Containerized Service in the *Spotfire® Service for R Installation and Administration* guide for more information.
Set file size upload limit	100MB	See the topic File Size Upload Limit in the *Spotfire® Service for R Installation and Administration* guide for more information.
Set R engine ports range	`63001/tcp` -> `64000/tcp`	See the topic Engine Ports in the *Spotfire® Service for R Installation and Administration* guide for more information.
Enable JMX Monitoring	OFF	See the topic Monitoring the Service using JMX in the *Spotfire® Service for R Installation and Administration* guide for more information.

Restrict Network Access for R Scripts in Containers

By default, the containers in which R scripts are running have access to network resources given to it.

R scripts can connect to the network. To restrict external network access for the container, and therefore any scripts running within it, the node manager computer must be configured in such a way that the containers cannot reach the network. One way to do this is by implementing iptables rules that block traffic from Docker containers to outside networks.

Use a Custom Docker Image for Containerized R

If the node manager is running on a Linux computer, then you can run Spotfire Service for R in a Docker container.

For more information, see the following help topics in the *Spotfire® Service for R Installation and Administration* guide.

- Configuring a Custom Docker Image on a Node with Internet Access
- Pulling a Custom Docker Image from an Authenticated Repository

Script Security for R

The following mechanisms control security of Spotfire Service for R and to prevent users from running malicious scripts on the server.

- R engine in Docker containerization.
- Script access control.

Only users in the Spotfire license group Script Author can create R scripts. For other users to run the scripts, the scripts must be trusted (through the Manage Trust mechanism in Spotfire).

Docker Containerization for R Scripts

Scripts running in a container have full access to the Docker container and have permission to do anything that is possible to do from within the container. The level of isolation a container provides depends on the Docker installation and the privileges given to these containers.

Configuration	Description
R service host isolation	Scripts are prohibited from accessing the file system of the host computer running the Spotfire Service for R.
User isolation	The use of engine containers ensures that the same execution environment is not re-used for multiple data functions initiated by different users.
Network isolation	Depending on configuration, the R scripts can access external network and other Docker containers that are available from within a container. In many cases, a default installation with engine containers lets scripts access the external network, including the internet, and to access other Docker containers. To restrict access to the network, the Docker containers must be configured to restrict network access. The container options should not be used without additional network configuration, if network isolation is needed.

Conditions for Running R Scripts on the Spotfire Server

If you have Spotfire® Statistics Services configured to work with R, and you are moving to Spotfire Service for R, then your R script authors need to apply trust to the R scripts so they can run when the new service is active.

If Spotfire Statistics Services is installed and configured to use R, and if the user running the script has an Author Scripts license feature, then an untrusted R script created with Spotfire Analyst will run using Spotfire Statistics Services even if the user running the script is not a member of the Script Author group.

If Spotfire Service for R is installed and configured, then only members of the Script Author group can run untrusted R scripts created using Spotfire Analyst. Script Authors must trust the scripts so they run for other users and services such as the Web Player and Automation Services. The general rule to remember is this:

- For users (or other services, such as Automation Services and Web Services) to run an R script in Spotfire Service for R, that script can run ONLY if it has been saved and trusted by an authorized script author.

If you have both Spotfire Statistics Services and Spotfire Service for R installed and configured, R scripts always default to use Spotfire Service for R, even if you specify a URL for your installation of Spotfire Statistics Services.

Optionally, Spotfire Server administrators can set the property `disable.spotfire.trust.checks=true`; however, administrators must use caution when considering this option, because when this property is set, Spotfire does not apply any trust checks to scripts.

Conditions for scripts to run when Spotfire Service for R is installed

Are you in the Script Author group?	Is your script trusted using the Manage Trust option?	Has the server admin set disable.spotfire.trust. checks=true?	The script runs without trust errors
Yes	No	No	Yes
No	No	Yes	Yes
No	Yes	No	Yes
No	No	No	No

In this version of Spotfire Analyst, there is no option to run R scripts locally.

Spotfire Service for Python

The Spotfire Service for Python provides Spotfire clients with the ability to execute Python code, using the available Python interpreter, on the Python service node.

Spotfire Service for Python is required to execute data functions in Spotfire files from Spotfire Automation Services and Spotfire Web Player, because those services do not have Python interpreters.

The Spotfire Service for Python itself is running the service as the same user account as is running the node manager on which the service runs. See Node Manager.

By default, Python scripts executed by the Spotfire Service for Python on behalf of its users are executed in a different execution context, as explained here.

Python service component	Default	Description
Service account	`NT Authority\System` or `spotfire` (Linux)	Default: `NT Authority\System` or `root` (Linux)
Log files		`<node manager installation>/logs` See the topic Spotfire Service for Python Logs in the *Spotfire® Service for Python Installation and Administration* guide for more information.

The Spotfire Service for Python ports and protocols

Name	Default Port and Protocol	Function Description	Secure/Encrypted
Communication port	Next available general purpose `950<x>/tcp`. (For example, 9502/tcp, or 9503/tcp, and so on, depending on the other data function services installed.)	For secure (HTTPS) internal communication. Cannot be accessed directly.	Yes

Name	Default Port and Protocol	Function Description	Secure/Encrypted
Python engine ports	`62001/tcp` -> `63000/tcp`	Host-internal communication between Spotfire Service for Python and the Python interpreter.	No

Settings and Configuration Tasks for Spotfire Service for Python

You can use these settings to limit the capabilities of running Python data functions.

Setting / Configuration task	Default value	Description
`use.engine.containers`	• Windows: `FALSE` • Linux: `TRUE`	Available on Linux only. If your deployment is on a Linux server, then the default configuration for Spotfire Service for Python is to use containers (the property `use.engine.containers: TRUE`) . Running Spotfire Service for Python with containers enabled prevents the engines from having access to the host system. See the topic Containerized Spotfire Service for Python in the *Spotfire® Service for Python Installation and Administration* guide for more information.
Set file size upload limit	100MB	See the topic File Size Upload Limit in the *Spotfire® Service for Python Installation and Administration* guide for more information.
Set Python engine ports range	62001 - 63000	See the topic Python Engine Ports in the *Spotfire® Service for Python Installation and Administration* guide for more information.
Enable JMX Monitoring	OFF	See the topic Monitoring Spotfire Service for Python using JMX in the *Spotfire® Service for Python Installation and Administration* guide for more information.

Restrict Network Access for Python Scripts in Containers

By default, the containers in which Python scripts are running have access to network resources given to it.

Python scripts can connect to the network. To restrict external network access for the container, and therefore any scripts running within it, the node manager computer must be configured in such a way that the containers cannot reach the network. One way to do this is by implementing iptables rules that block traffic from Docker containers to outside networks.

Use a Custom Docker Image for Containerized Python

If the node manager is running on a Linux computer, then you can run Spotfire Service for Python in a Docker container.

For more information, see the following help topics in the *Spotire® Service for Python Installation and Administration* guide.

- Configuring a Custom Docker Image on a Node with Internet Access
- Pulling a Custom Docker Image from an Authenticated Repository

Script Security for Python

The following mechanisms control security of Spotfire Service for Python and to prevent users from running malicious scripts on the server.

- Python engine in Docker containerization.
- Script access control.

Only users in the Spotfire license group `Script Author` can create Python scripts. For other users to run the scripts, the scripts must be trusted (through the Manage Trust mechanism in Spotfire).

Docker Containerization for Python Scripts

Scripts running in a container have full access to the Docker container and have permission to do anything that is possible to do from within the container. The level of isolation a container provides depends on the Docker installation and the privileges given to these containers.

Configuration	Description
Python service host isolation	Scripts are prohibited from accessing the file system of the host computer running the Spotfire Service for Python.
User isolation	The use of engine containers ensures that the same execution environment is not re-used for multiple data functions initiated by different users.
Network isolation	Depending on configuration, the Python scripts can access external network and other Docker containers that are available from within a container. In many cases, a default installation with engine containers lets scripts access the external network, including the internet, and to access other Docker containers. To restrict access to the network, the Docker containers must be configured to restrict network access. The container options should not be used without additional network configuration, if network isolation is needed.

Spotfire Analyst

With Spotfire Analyst, analysis authors can develop web-based and Windows client-based analyses. Spotfire Analyst provides authoring tools for sharing analyses and dashboards. It is installed on the Windows desktop.

Documentation

You can find documentation for Spotfire Analyst on the documentation portal at Spotfire Analyst Documentation. Alternatively, you can find the documentation from the Spotfire Analyst **Help** menu.

Installation directory

By default, Spotfire Analyst is installed in `C:\Program Files (x86)\Spotfire\`. Other information and settings are stored in the directory `C:\Users\[`*`username`*`]\AppData`.

Ports & Protocols

The default HTTP port is 8000. The protocol is tcp HTTP. Spotfire opens a web server on port 8000. It accepts connections only from localhost.

Spotfire Documentation and Support Services

For information about the Spotfire® products, you can read the documentation, contact Spotfire Support, and join the Spotfire Community.

How to Access Spotfire Documentation

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

The website is updated frequently and is more current than any other documentation included with the product.

Spotfire Documentation

The documentation for all Spotfire products is available on the Spotfire Documentation page. This page takes you directly to the latest version of each document.

To see documents for a specific Spotfire product or version, click the link of the product under 'Other versions', and on the product page, choose your version from the top right selector.

Release Version Support

Some release versions of 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://spotfi.re/lts.

How to Contact Support for Spotfire Products

You can contact the Support team in the following ways:

- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the support portal at https://spotfi.re/support.
- For creating a Support case, you must have a valid maintenance or support contract with Cloud Software Group, Inc. You also need a user name and password to log in to https://spotfi.re/support. If you do not have a user name, you can request one by clicking **Register** on the website.

System Requirements for Spotfire Products

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

How to join the Spotfire Community

The Spotfire Community is the official channel for Spotfire customers, partners, and employee subject matter experts to share and access their collective experience. The 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 Spotfire products. In addition, users can submit and vote on feature requests from within the Ideas Portal. For a free registration, go to https://spotfi.re/community.

Legal and Third-Party Notices

SOME CLOUD SOFTWARE GROUP, INC. ("CLOUD SG") SOFTWARE AND CLOUD SERVICES EMBED, BUNDLE, OR OTHERWISE INCLUDE OTHER SOFTWARE, INCLUDING OTHER CLOUD SG SOFTWARE (COLLECTIVELY, "INCLUDED SOFTWARE"). USE OF INCLUDED SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED CLOUD SG SOFTWARE AND/OR CLOUD SERVICES. THE INCLUDED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER CLOUD SG SOFTWARE AND/OR CLOUD SERVICES OR FOR ANY OTHER PURPOSE.

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

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 Cloud Software Group, Inc.

Spotfire, the Spotfire logo, TERR, TIBCO, Enterprise Message Service, and Hawk are either registered trademarks or trademarks of Cloud Software Group, Inc. in the United States and/or other countries. A list of Cloud SG's trademarks and trademark guidelines is available at https://www.cloud.com/legal.

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. You acknowledge that all rights to these third party marks are the exclusive property of their respective owners. Please refer to Cloud SG's Third Party Trademark Notices (https://www.cloud.com/legal) for more information.

This document includes fonts that are licensed under the Apache License, Version 2.0, which is available at https://www.apache.org/licenses/LICENSE-2.0 and reprinted in the Addendum below.

Copyright (c) Christian Robertson / Google, Roboto font.

Cloud SG 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. See the "readme" file for the availability of a specific version of Cloud SG software 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. CLOUD SG MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S), THE PROGRAM(S), AND/OR THE SERVICES DESCRIBED IN THIS DOCUMENT AT ANY TIME WITHOUT NOTICE.

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 "README" FILES.

This and other products of Cloud SG may be covered by registered patents. For details, please refer to the Virtual Patent Marking document located at https://www.tibco.com/patents.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

   You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS