TIBCO ActiveMatrix® Spotfire Enabler User's Guide

User's Guide

Software Release 1.0.0

November 2012

Two-Second Advantage®

Important Information

Contents

Preface

TIBCO ActiveMatrix Spotfire Enabler lets you use TIBCO Spotfire to visualise your ActiveMatrix environment. The TIBCO ActiveMatrix Spotfire Enabler provides the necessary tools and samples needed to enable TIBCO Spotfire analysis of TIBCO ActiveMatrix enterprise components.

TIBCO Spotfire software and licenses must be obtained separately.

TIBCO Product Documentation

This section lists documentation resources you may find useful.

The following documents form the TIBCO ActiveMatrix® Spotfire Enabler documentation set:

- *User's Guide*: Read this manual to learn how to install and use the product.
- *Release Notes*: Read this manual for a list of new and changed features, steps for migrating from a previous release, and lists of known issues and closed issues for the release.

Typographical Conventions

Table 1: General Typographical Conventions

Convention	Use
TIBCO_HOME *ENV_NAME*	TIBCO products are installed into an installation environment. A product installed into an installation environment does not access components in other installation environments. Incompatible products and multiple instances of the same product must be installed into different installation environments. An installation environment consists of the following properties: • **Name** - Identifies the installation environment. The name is appended to the name of Windows services created by the installer and is a component of the path to the product shortcut in the **Windows Start** > **All Programs** menu. This name is referenced in documentation as *ENV_NAME*. • **Path** - The folder into which the product is installed. This folder is referenced in documentation as *TIBCO_HOME*.
CONFIG_HOME	The folder that stores configuration data generated by TIBCO products. Configuration data can include sample scripts, session data, configured binaries, logs, and so on. This folder is referenced in documentation as *CONFIG_HOME*.

`code font`	Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example:
	• Use `MyCommand` to start the foo process.
	• Code example:
	`public class HelloWorldImpl extends AbstractHelloWorldImpl {`
	`...`
	`public HelloResponseDocument sayHello(HelloRequestDocument firstName) {`
	`...`
	`System.out.println("--> Generating Java Hello Component Response...");`
	`String name = firstName.getHelloRequest()==null||firstName.getHelloRequest(). equals("")?"Friend":firstName.getHelloRequest();`
	`HelloResponseDocument resp = HelloResponseDocument.Factory.newInstance();`
	`resp.setHelloResponse("Hi " + name + "! " + "This is the Java component.\n");`
	`System.out.println("--> Java Hello Component Response: \n\t\t" +`
	`resp.getHelloResponse());`
	`...`
	`}`
	`}`
	• *CONFIG_HOME*`/admin/`*enterpriseName*`/samples/remote_props.properties`
	• Output example:
	`C:\Program Files\tibco\amx-3\studio\3.5\eclipse>amx_eclipse_ant.exe -buildfile "C:/helloworld1/build.xml" -data "C:/hws"`
	`Buildfile: C:/helloworld1/build.xml`
	`createApplicationDAA:`
	`[sds.createDAA] Waited for 47ms for workspace refreshes after building features.`
	`all:`
	`BUILD SUCCESSFUL`
	`BUILD SUCCESSFUL`
	`Total time: 2 minutes 18 seconds`

Convention	Use
`bold code font`	Bold code font is used in the following ways: • In procedures, to indicate what a user types. For example: **`Type admin`**. • In large code samples, to indicate the parts of the sample that are of particular interest. • In command syntax, to indicate the default parameter for a command. For example, if no parameter is specified, MyCommand is enabled: `MyCommand [enable | disable]`
italic font	Italic font is used in the following ways: • To indicate a document title. For example: See *TIBCO BusinessWorks Concepts*. • To define new terms. For example: A *keystore* is a database of keys and certificates. • To indicate a variable in a command or code syntax that you must replace. For example: `MyCommand` *pathname*.
Key combinations	Key name separated by a plus sign indicate keys pressed simultaneously. For example: **Ctrl+C**. Key names separated by a comma and space indicate keys pressed one after the other. For example: **Esc, Ctrl+Q**.
	The note icon indicates information that is of special interest or importance, for example, an additional action required only in certain circumstances.
	The tip icon indicates an idea that could be useful, for example, a way to apply the information provided in the current section to achieve a specific result.
	The warning icon indicates the potential for a damaging situation, for example, data loss or corruption if certain steps are taken or not taken.

Table 2: Syntax Typographical Conventions

Convention	Use
[ ]	An optional item in command syntax. For example: `MyCommand [optional_parameter] required_parameter`
|	A logical 'OR' that separates multiple items of which only one may be chosen. For example, you can select only one of the following parameters: `MyCommand param1 | param2 | param3`

{ }	A logical group of items in a command. Other syntax notations may appear within each logical group.
	For example, the following command requires two parameters, which can be either the pair `param1` and `param2`, or the pair `param3` and `param4`.
	`MyCommand {param1 param2} | {param3 param4}`
	In the next example, the command requires two parameters. The first parameter can be either `param1` or `param2` and the second can be either `param3` or `param4`:
	`MyCommand {param1 | param2}{param3 | param4}`

Convention	Use
	In the next example, the command can accept either two or three parameters. The first parameter must be param1. You can optionally include `param2` as the second parameter. And the last parameter is either `param3` or `param4`. `MyCommand param1 [param2] {param3 | param4}`

Connecting with TIBCO Resources

How to Join TIBCOmmunity

TIBCOmmunity is an online destination for TIBCO customers, partners, and resident experts. It is a place to share and access the collective experience of the TIBCO community. TIBCOmmunity offers forums, blogs, and access to a variety of resources. To register, go to http://www.tibcommunity.com.

How to Access TIBCO Documentation

After you join TIBCOmmunity, you can access the documentation here: http://docs.tibco.com.

How to Contact TIBCO Support

For comments or problems with this manual or the software it addresses, contact TIBCO Support as follows:

- For an overview of TIBCO Support, and information about getting started with TIBCO Support, visit this site:

  http://www.tibco.com/services/support
- If you already have a valid maintenance or support contract, visit this site:

  https://support.tibco.com

  Entry to this site requires a username and password. If you do not have a username, you can request one.

Chapter

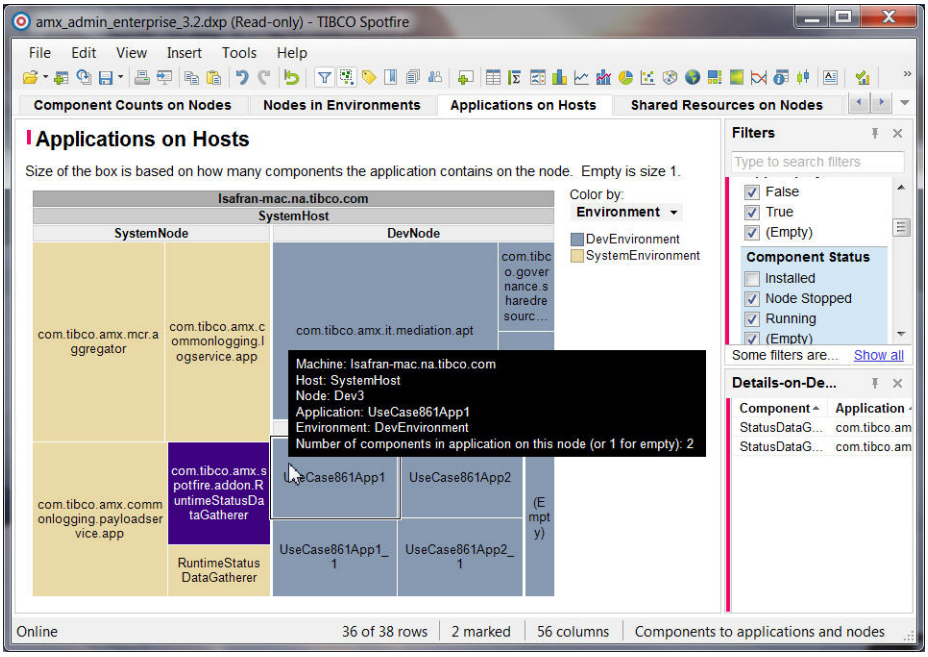
1

Overview

The TIBCO ActiveMatrix Spotfire Enabler makes it easy for you to graphically view, analyze and report on valuable information about the ActiveMatrix enterprise you administer. You can rapidly create concise, colorful visualizations in the form of heat maps, box plots, bar charts, pie charts and other useful views that make it easier to plan deployments as well as respond to events that affect your business.

With the Spotfire Enabler you can display different types of visualizations simultaneously and link them to each other. For example, you can readily filter a visualized issue to detect aberrations, or identify nodes meeting certain criteria to help with planning. You can quickly display different views across multiple dimensions, even for very large enterprises.

The Spotfire Enabler lets you quickly gain insight into both infrastructure artifacts and application artifacts. Filters let you drill down to the data of interest, such as the load on a particular node, or available resources and libraries as well as to see where an application instance is being created or how an application instance affects the rest of your ecosystem. You can readily identify candidate machines for creating new nodes, and establish maximum reuse of shared resource instances, shared libraries and application runtimes. You can instantly visualize the status of the nodes, component instances, applications, endpoints and resource instances. You can also view features and substitution variable usage.



The Spotfire Enabler also helps you verify that your organization's deployment rules and best practices are being followed. For example, you could verify and ensure that no more than *x* number of environments are created per enterprise, or no more than *y* number of nodes are created per environment. The application operator may want to ensure that all shared resources instances are created with the name of the machine as a suffix, or that no more than *z* number of applications are deployed per environment.

You can filter your data interactively, and derive instant answers. Sharing your results is easy, including in the form of dynamic reports as well as instant presentations.

This document provides information about visualizations and usage patterns, as well as filtering in various ways, so that you can identify and take action on relevant ActiveMatrix enterprise data. You can extend the functionality of the Spotfire Enabler by creating your own visualizations or modifying the provided ones. See the Spotfire documentation for details on how to define your own visualizations and filters.

This user manual contains information about all functionality available with the Spotfire Enabler. If you do not have access to all licenses, some functions described, such as the ability to edit your analysis, may be unavailable. For details on how to access the full range of functionality, please visit the support website.

Chapter

2

Installation

You install the Spotfire Enabler by extracting the zip file and running the DDL script for your ActiveMatrix Administrator database server. This works with all database types supported by ActiveMatrix Administrator: DB2, MS SQL Server and Oracle. After creating the views from your database you need to create an ODBC data source on the Spotfire client machine.

Topics

Installing The Spotfire Enabler

The TIBCO ActiveMatrix Spotfire Enabler is provided in a zip file that includes DDL scripts for DB2, MS SQL Server or Oracle. The userid running the DDL must have permission to create the Spotfire Enabler views.

About this task

You install the Spotfire Enabler by extracting the zip file. You will then run the DDL script for your database server and then create and deploy the Runtime Status Data Gathering application, as described in Using the Runtime Status Data Gathering Application on page 18. This application is for ActiveMatrix Administrator 3.2 or higher.

After creating the views from your database you need to create an ODBC data source on the Spotfire client machine.

The Spotfire Enabler zip file contains the following items:

Folder	Files	Comments
app	dxpmodifer.jar	Executable jar file for the DXPModifier Utility
	RuntimeStatusDataGatherer.daa	DAA file for Runtime Status Data Gathering (3.2 or higher)
	RuntimeStatusDataGatherer.deployment-build.xml	XML deployment build file for Runtime Status Data Gathering application
	RuntimeStatusDataGatherer.deployment-config.xml	XML deployment configuration file for Runtime Status Data Gathering application
doc	html/index.htm	Spotfire Enabler User's Guide in the HTML format. This is a nested directory containing the User's Guide in the HTML format.
	pdf/tib_amx_spotfire_enabler_usersguide.pdf	Spotfire Enabler User's Guide in the PDF format
dxp	amx_admin_enterprise_3.1.dxp	Spotfire Enabler Analysis DXP file for ActiveMatrix Administrator 3.1 (DB2 and MS SQL)
	amx_admin_enterprise_3.1.oracle.dxp	Spotfire Enabler Analysis DXP file for ActiveMatrix Administrator 3.1 (Oracle)
	amx_admin_enterprise_3.2.dxp	Spotfire Enabler Analysis DXP file for ActiveMatrix Administrator 3.2 (DB2 and MS SQL)
	amx_admin_enterprise_3.2.oracle.dxp	Spotfire Enabler Analysis DXP file for ActiveMatrix Administrator 3.2 (Oracle)
license	TIB_amxs_1.0.0_license.pdf TIB_amxs_1.0.0_license.txt	Spotfire Enabler End User License Agreement
views	spotfire-views.oracle.sql	SQL file for Oracle view
	spotfire-views.sql	SQL file for DB2 and MS SQL views

Folder	Files	Comments
	TIB_amxs_1.0.0_relnotes.pdf	Spotfire Enabler Release Notes
	TIB_amxs_1.0.0_readme.txt	Spotfire Enabler README file

The userid for data tables must match that of the TIBCO ActiveMatrix Administrator database. For your Windows userid, there is no conflict as long as the ODBC data sources are defined as system data sources.

Procedure

1. Extract the zip file contents for the Spotfire Enabler to *TIBCO_HOME*.
2. Run the appropriate DDL script from the `views` folder to create the desired views in your database.
   - For DB2 or for Microsoft SQL Server use `spotfire-views.sql`.
   - For Oracle, use `spotfire-views.oracle.sql`.
3. On the Spotfire Client machine, create an ODBC data source named `amx_admin` that points to the ActiveMatrix Administrator server database.
4. Run the DXPModifier utility to change the userid and password to match those used by your ActiveMatrix Administrator server to access its database.
5. If you are using ActiveMatrix Administrator 3.2, instantiate the application that makes runtime status information available to Spotfire. (No applicable for ActiveMatrix Administrator 3.1). For details, see Using the Runtime Status Data Gathering Application on page 18.
   a) Copy the `RuntimeStatusDataGatherer.deployment-build.xml` file from `app/` to the samples directory of your ActiveMatrix Administrator server. Copying this file prevents having to edit it.

   Alternatively, edit the `RuntimeStatusDataGatherer.deployment-build.xml` file, and change the value of the property `admin.samples.directory` to point to the `CONFIG_HOME/admin/`*`enterpriseName`*`/samples` directory.

   b) For the `RuntimeStatusDataGatherer.deployment-config.xml`, change `${TIBCO_HOME}` to the actual path to *TIBCO_HOME*.
   c) Copy this file from the `app/` directory to the `CONFIG_HOME/admin/`*`enterpriseName`*`/samples` directory or

   edit the `RuntimeStatusDataGatherer.deployment-build.xml` file to replace in the `data.file` property value `admin.samples.directory` with `basedir`.

   d) In the directory with the file `RuntimeStatusDataGatherer.deployment-build.xml` run the command:
   `ant -f RuntimeStatusDataGatherer.deployment-build.xml`
6. In the Spotfire client, open the library file `amx_admin_enterprise_<`*`version`*`>.dxp` in the `dxp` folder.

   For Oracle use `amx_admin_enterprise_3.1.oracle.dxp` or `amx_admin_enterprise_3.2.oracle.dxp`. Make sure the version of the DXP file matches your version of ActiveMatrix Administrator.

   To look at different databases simultaneously, you can have multiple sessions running. For details, see Using Spotfire Enabler With Multiple Enterprises on page 20.

What to do next

You have an alternative to use the ActiveMatrix Administrator UI to deploy the Spotfire Enabler DAA file from the `app` directory. To do so, create the application in `SystemEnvironment` and deploy it to `SystemNode`.

Using the Runtime Status Data Gathering Application

The Runtime Status Data Gathering application provided with the Spotfire Enabler gathers deployment and configuration runtime status data. You can load the application by deploying the DAA file from the ActiveMatrix Administrator UI. There is also a pair of build and configuration files that use the ActiveMatrix Administrator command line ANT task to load the application: `RuntimeStatusDataGatherer.deployment-build.xml` and `RuntimeStatusDataGatherer.deployment-config.xml`. You can also use these files to cleanup resource instances and resource templates associated with the application as well as remove the application.

About this task

When the DAA file is uploaded to the Administration server, the application template and the feature are extracted and stored. The default application name in the `RuntimeStatusDataGatherer.deployment-config.xml` file is `com.tibco.amx.spotfire.enabler.RuntimeStatusDataGatherer`. By default this application is created in the `System` folder. The feature named `com.tibco.amx.spotfire.addon.RuntimeStatusDataGatherer.customfeature` provides the Java classes.

There is a property named `StatusRefreshInterval` whose default value is 60. This property controls how many seconds the application waits to refresh the status stored in the database. The longer the interval, the more time must pass before changes to status are available to Spotfire. The shorter the value, the more overhead you incur processing the list and updating the database.

After the Runtime Status Data Gathering application stores the updated status, you will need to reload data in Spotfire to pick up the latest values.

The Runtime Status Data Gathering application exposes a SOAP service binding that can be used to force a reload of the status data or to change the **Refresh** interval.

Changing the Status Refresh Interval

The Runtime Status Data Gathering application used with ActiveMatrix Administrator 3.2, has a default value of 60 seconds.

About this task

You can change this value by editing the `StatusRefreshInterval` property and redeploying the application.

Alternatively, you can use the following procedure.

Procedure

1. In the ActiveMatrix Administrator UI, select the Runtime Status Data Gathering application.
2. In the General tab, select the Status Gathering Service binding named `SOAPService_Binding1`.
3. Select the System Node binding instance that has a Generate WSDL button.

   With the WSDL you can use SOAPUI or another tool to control the **Refresh** interval without having to redeploy the application.

Launching the Spotfire Enabler

When you launch Spotfire you can either work offline or log into the Spotfire server.

About this task

You will load the Spotfire Enabler analysis file by opening the DXP file, which contains the default visualizations.

Procedure

1. Launch Spotfire and enter your login information or click Work Offline to work offline.
2. Load the Spotfire Enabler analysis by opening the DXP file.

   The default visualizations are loaded, with each page showing one or more visualizations. For details on the visualization pages and how to use them, see Understanding Visualization on page 21.
3. Click the **Reload** button to load your own enterprise data after you have run the appropriate DDL script.

   If you have not run the DXPModifier Utility, or if the password has changed since it was last run, you will be prompted to enter the database userid and password eight times.
4. Save the analysis.

   By saving the analysis, you avoid entering your password multiple times the next time you use it. The alternative is to run the DXPModifier utlity. If you change your password you can either run this utility again or follow the prompts to provide your new password. For details, see Using the DXPModifier Utility on page 19.

   By default, data is stored in the analysis for all data sources so you can open and work on the analysis without needing to interact with the database. As a result, data can be stale. You can select **Edit > Data Table Properties** to ensure that your data is always reloaded when you open the analysis. Select each data table and click the **Linked to Source** radio button in the **General** tab and click **Apply**. This will prevent you from opening the analysis if there is a problem connecting to the database.

Using the DXPModifier Utility

The DXPModifier utility simplifies managing the database connection used to load your enterprise data. It allows you to update the database password or point at a completely different ODBC data source.

Procedure

On the command line type `java-jar <path>/dxpmodifier.jar <args>`.

You can run the command with the arguments shown in the table below. For example, from the `amx_spotfire_enabler/1.0` directory `java -jar app/dxpmodifier.jar -dxp dxp/amx_admin_enterprise_3.1.dxp -odbc my_odbc_name -userid DB_USER -pw DB_USER_PW`.

The output shows what is changing (ODBC, user, and / or password) with the old value and the new value. There are eight entries in each of the two files.

The utility will back up the old file, appending `.bakX` to the name, where *X* is the lowest number that will create an unused file name. If only `.bak` is available, it will use that rather than `.bak0`. There is no limit on the number of backup files and you can determine the extent of periodic cleanup needed.

Argument	Description
`-dxp`	file system path to the DXP file

Argument	Description
`-odbc`	new name of the ODBC data source
`-userid`	database userid
`-pw`	database password
`-test`	do a dry run displaying what would be changed, without actually making any changes
`-help`	display usage

Using Spotfire Enabler With Multiple Enterprises

You can utilize Spotfire Enabler with multiple enterprises. You create copies of the appropriate DXP files, create an ODBC data source, and run the DXPModifier utility against the copies of the DXP file. You can then use the copied DXP files to start a separate instance of Spotfire for each enterprise.

Procedure

1. Copy the appropriate DXP file for each enterprise you want to view.
2. Create an ODBC data source for each enterprise.
3. Run the DXPModifier utility against each DXP file copy to point it to the appropriate ODBC data source.
4. Start separate instances of Spotfire for each enterprise by using the copied DXP file(s).

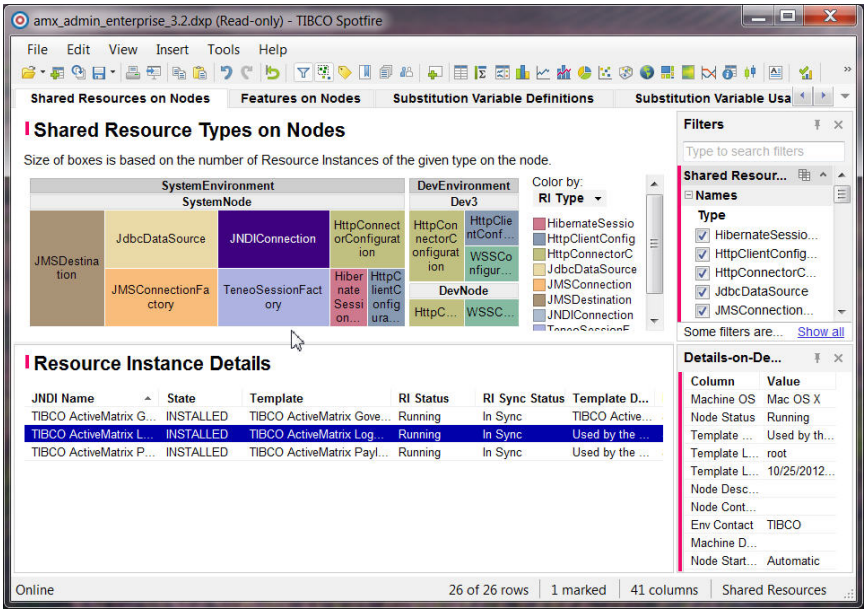
Chapter

3

Understanding Visualization

The Spotfire Enabler provides some predefined visualizations, which you can use with your own administrative data. These visualizations help you plan deployments and monitor your enterprise. You can filter your data interactively, and derive instant answers.

With these defined visualizations you can readily identify important enterprise data. For example, you can quickly identify which nodes have no user applications distributed to them, which nodes have the most components loaded on them, or which nodes have applications from a specific template. You can also determine which nodes have specific resource instances or use a particular template for resource instances. At a glance you can see which nodes have an older version of a feature, which nodes have the ActiveMatrix Mediation component installed, which ones have node names beginning with the environment name, or which nodes use particular substitution variables.



The Spotfire Enabler lets you readily filter to focus on what you care about across the full information set. You can quickly display different views across multiple dimensions in ways helpful to ActiveMatrix administrators, even with very large enterprises.

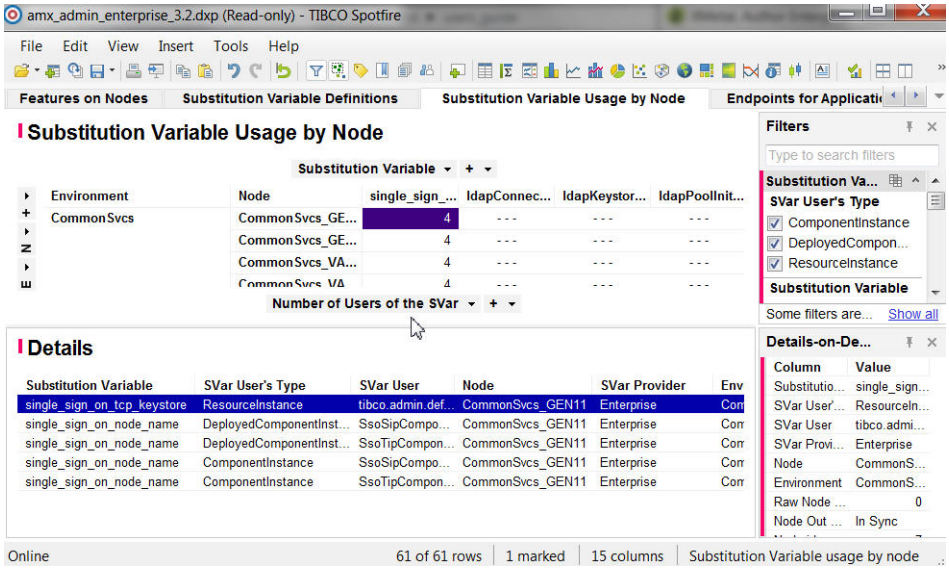
You can also verify that your organization's deployment rules and best practices are being followed. For example, you could verify and ensure that no more than *x* number of environments are created per enterprise, or no more than *y* number of nodes are created per environment. The application operator may want to ensure that all shared resources instances are created with the machine name as a suffix, or that no more than *z* number of applications are deployed per environment.

In the Endpoints in Environments, Applications in Environment, and Applications in Environment with Details pages, the status can have multiple values. This is seen some of the application's endpoint (in the Endpoints in Environments page) and the application's components (in the Applications in Environment and Applications in Environment with Details pages) are in one state and others are in a different state. In some cases the same color will be used for both individual and combination states. Hover over the application to see the exact state.

Topics

Visualization Pages

There are a number of predefined ActiveMatrix administrator visualization pages in the Spotfire Enabler, which you can use with your own data. This type of analysis can help you plan deployments and monitor your enterprise across multiple environments. For example, you can quickly identify which nodes have the most components loaded, or which resource instances, templates and substitution variables are in use. You can easily share your results dynamically, including with reports and instant presentations.



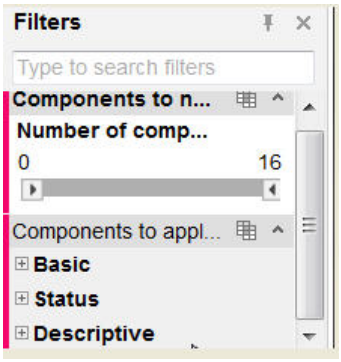
Create the views that enable loading and viewing your own enterprise data with your preferred database tool, as described in Installing The Spotfire Enabler on page 16.

There are a few differences between what you see using ActiveMatrix Administrator 3.1 versus what you see using ActiveMatrix Administrator 3.2. With 3.2 runtime status information is available for components, endpoints, nodes and resource instances. See Using the Runtime Status Data Gathering Application on page 18.

The data tables drive what you see in a specific visualization. When you make a selection in a visualization, this action changes the values in the other visualizations on the page. The new values are displayed below the chart, plot or graph, and also in the **Details-on-Demand** window. See Data Tables for details on column names and what the values mean.

There are groups of filters appropriate for each page, as you will see in the **Filters** panel. The filter contents change as you make selections in other filters so the filters can directly provide valuable information. See Using Filters on page 51 for tips on filtering.

If a visualization page displays no data, reset all filters for that page.



At the bottom of your visualization page is an indicator of the number of rows and columns in the entire analysis you are viewing, as well as the data table in use for your selection.

The status information stored in the database for ActiveMatrix Administrator 3.2 is refreshed in the visualization pages every 60 seconds by default. You will need to refresh the Spotfire data tables to pick up the latest status information. To change the Refresh interval, see Changing the Status Refresh Interval on page 18.

Component instances and endpoints that are deployed and still part of the current configuration will count as two when you are determining counts. For all pages (except the **Component Counts on Nodes** page), you can use the filters to view one or the other (`deployed = True or False`) and the boxes in the visualization will resize appropriately.

Component Counts on Nodes Page

The **Component Counts on Nodes** page shows the number of nodes that have a certain number of components.

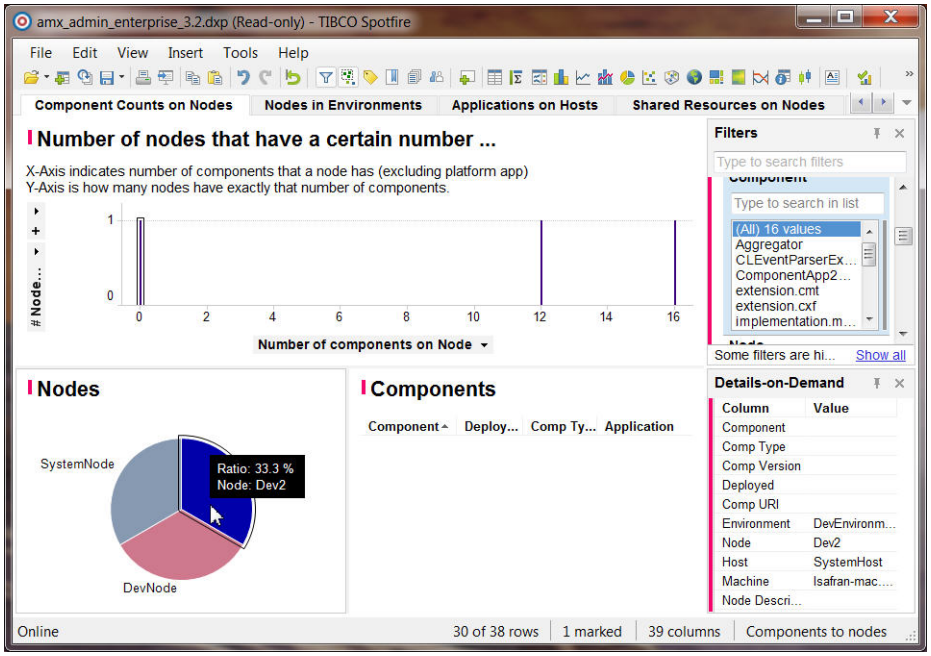
By default the **X-Axis** indicates the number of components in the node, excluding platform application components. The **Y-Axis** shows how many nodes have exactly that number of components. When you select an item on the top chart it turns purple.

When you select one or more bars in the top chart, you will see a pie chart under **Nodes**, which shows the nodes that match the selected bars, and entries for the component instances on those nodes in the **Details-on-Demand** window. By default, no nodes in the pie chart are selected. However, if you do make a selection (marked in dark blue), then that selection is maintained until you explicitly clear it.

You will see a table under **Components**, listing all components for the nodes selected in the pie chart. It displays the component names, whether the components are deployed, the component type and the application to which the component belongs.

Filter selections affect the contents of the Component table. They can also cause nodes to be completely hidden, but will not move nodes between buckets. If any component matches the filters, the node will contribute to the chart with its full number of components, not just those that match the filters..

The **Details-on-Demand** window will list more columns than the Components table. The exact columns shown depend on what was last selected (whether it was from the column table or from the node pie chart). When you right-click on the Node pie chart and unmark it, this information is no longer displayed.



The data tables used with this page are the **Components to Nodes** and the **Components to Applications and Nodes**.

The **Filter** groups are **Basic, Status**, and **Descriptive**. Notice the options available (e.g., you can click Running or Stopped under **Node Status**, and your pie chart will immediately show only the status chosen.)

- **Basic** filter group: Comp Type, Environment, Application, Component, Node, Host, Machine OS, Machine name.
- **Status** filter group: Component Status, Deployed - Comp, App Deployed, Node Status, Sync State Comp Inst, Sync Reason for Comp Inst, Node Installed, Sync State Node, Node Startup Mode, System Node.
- **Descriptive** filter group: App Last Deployed (dates), App Last Modified Date, App Description, App Contact, Node Contact, Env Contact, App Last Modifier, App Last Deployer, App Template, App Template Version, Component Version, Node Description, Node JVM Arguments, Machine Description, Raw Component Status, Raw Node Status, Component Status, Node Status.

The Component Status and Node Status filters and columns only display when using ActiveMatrix Administrator 3.2 or higher.

Nodes in Environments Page

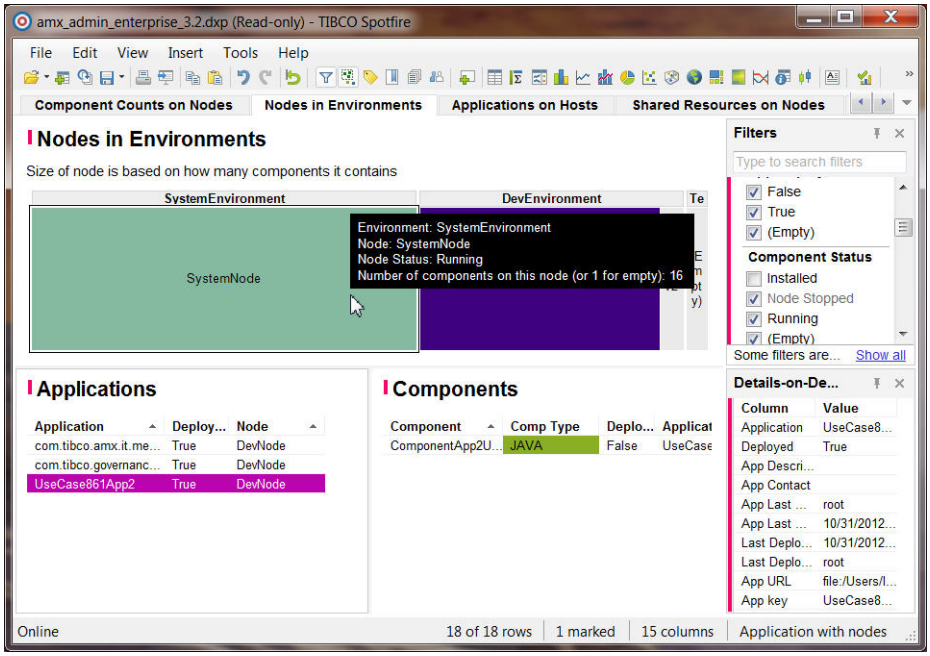
The **Nodes in Environments** page shows the number of components on a node.

This page provides an overview of all nodes in the system grouped by environment. It gives you a sense of how heavily utilized the node is, as the size of the box reflects a larger or smaller number of components. Empty nodes shows as the same size as nodes that have a single component. Environments with no nodes will have a single entry of size one, but having the label "Empty."

The color of the box depends upon whether you are displaying data for an ActiveMatrix Administrator 3.1 or 3.2 server. For a 3.2 server, the color indicates the node status: green for running, red for stopped, grey for empty, and so on. For a 3.1 server, a deeper color indicates a larger number of applications on the node. For 3.1 servers, displaying the legend will show the range of values; for 3.2 it will show what state each color represents.

Underneath the **Node** box you will see an **Applications** table, listing the applications on the selected nodes, whether they are deployed, and the node name. Related information is shown in the **Details-on-Demand** window as you select rows in the **Applications** table.

When you select an application from the **Applications** table, you will see component information for that application in the **Components** table beside it. When you select an item from **Components** you will see related information in the **Details-on-Demand** window. The color in the **Comp Type** column indicates the type of the component.



The data table used with this page is **Components to Applications and Nodes**.

The **Filter** groups are **Basic, Status**, and **Descriptive**.

- **Basic** filter group: Comp Type, Environment, Application, Component, Node, Host, the Machine OS,Machine name.
- **Status** filter group: Component Status (3.2 only), Deployed - Comp, App Deployed, Component Status, Node Status (3.2 only), Sync State Comp Inst, Sync Reason for Comp Inst, Node Installed, Node Installed, Sync Reason for Node, Sync State Node, Node Startup Mode, System Node
- **Descriptive** filter group: App Last Deployed dates, App Last Modified dates, the App Description, App contact, Node Contact, Env contact, App Last Modifier, App Last Deployer, App Template, App Template Version, Component Version, Node Description, Node JVM Arguments, Machine Description, Raw Component Status, Raw Node Status, Component Status, Node Status.

The Component Status and Node Status columns and filters only display when using ActiveMatrix Administrator 3.2 or higher.

Applications on Hosts page

The **Applications on Hosts** page shows where applications are loaded. In particular, each box represents the components an application contains on a node.

The size of the box on this visualization page reflects a larger or smaller number of components for the application on the specific node.

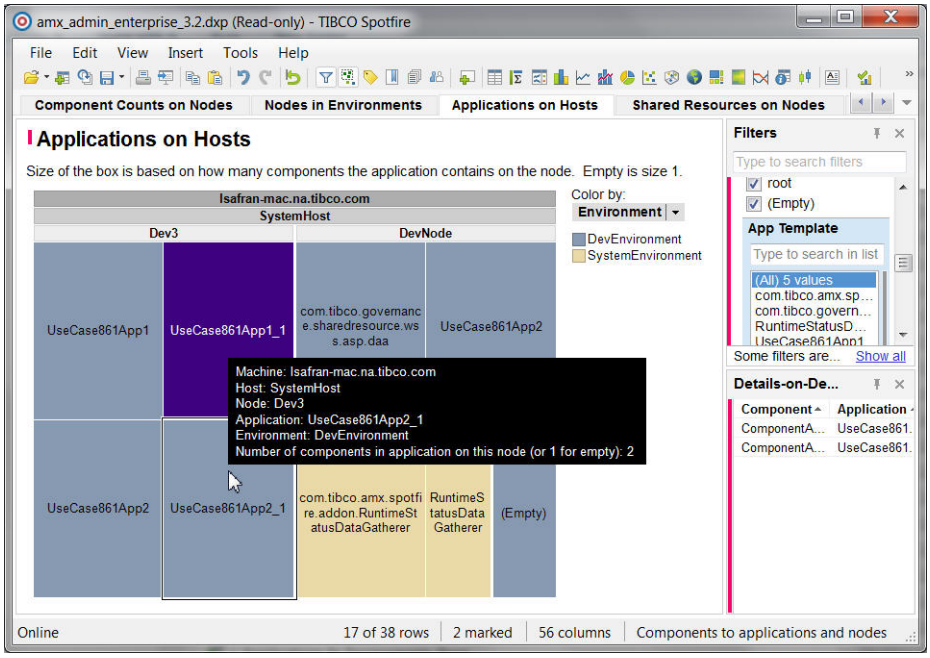
Organization is by machine, then host within the machine, then by node within the host and finally the applications with components located on the node.

Empty nodes shows as the same size as nodes that have a single component, but have the label "Empty."

You can narrow your focus by clicking on filters (e.g., a specific application or environment).

The color represents the environment the application is contained in. This allows you to quickly scan for applications deployed to hosts reserved for another environment.

The **Details-on-Demand** window will list the components for the selected applications. When you right-click on a box and unmark it, it returns to its appropriate color, and this information is no longer displayed.



The data table used with this page is **Components to Applications and Nodes**.

The **Filter** groups are **Basic, Status**, and **Descriptive**.

- **Basic** filter group: Comp Type, Environment, Application, Component, Node, Host, Machine OS, Machine name.
- **Status** filter group: Component Status, Deployed - Comp, App Deployed, Node Status, Sync State Comp Inst, Sync Reason for Comp Inst, Node Installed, Sync State Node, Node Startup Mode, System Node.
- **Descriptive** filter group: App Last Deployed dates, App Last Modified dates, App Description, App contact, the Node Contact, Env contact, App Last Modifier, App Last Deployer, App Template, App Template Version, Component Version, Node Description, Node JVM Arguments, Machine Description, Raw Component Status, Raw Node Status, Component Status, Node Status.

The Status columns and filters only display when using ActiveMatrix Administrator 3.2 or higher.

Shared Resources On Nodes Page

The **Shared Resources on Nodes** page shows the number of resource instances on a given node or set of nodes. Each box represents the number of resource instances of a specific type on a node. This tab addresses all questions related to resource instances and resource templates.

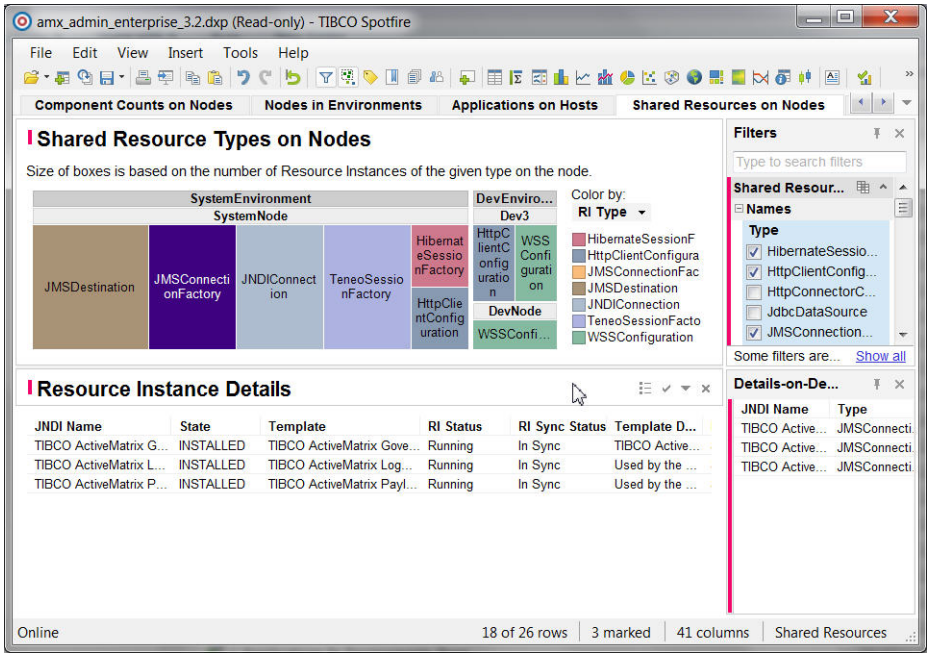
The size of the boxes reflects a smaller or larger number of resource instances.

The color of the box indicates the type of resource instance. By default a legend displays to explain which color represents which resource type.

By default, all nodes in the enterprise with at least one resource instance will be shown. To see only nodes from a specific environment, either click on the environment name or use the **Environment** filter.

You can use filters to look for resource instances of a particular type (Names -> Type), all instances with a particular name (Names -> JNDI Name) or all resources created from a particular resource template (Names -> Template).

Underneath the graph are the resource instance details. When you select one or more boxes in the graph, they turn purple (to indicate selection), and the resource instance information displays below the graph.



The data table used with this page is **Shared Resources**.

The **Filter** groups on this page are **Names**, **Status** and **Descriptions and Contacts**.

- **Names** filter group: Type, Template, JNDI Name, Environment, Node, Machine, Host, Machine OS
- **Status** filter group: State, RI Status, Sync State RI, Sync Reason for RI, System Node, Node Installed, Node Status, Sync State Node, Sync Reason for Node, Template Last Modifier, Template Last Modification Date, Node Startup Mode
- **Descriptions and Contacts** filter group: Template Description, Node Description, Node Contact, Env Contact, Machine Description, Node JVM Args

The RI Status and Node Status columns and filters only display when using ActiveMatrix Administrator 3.2 or higher.

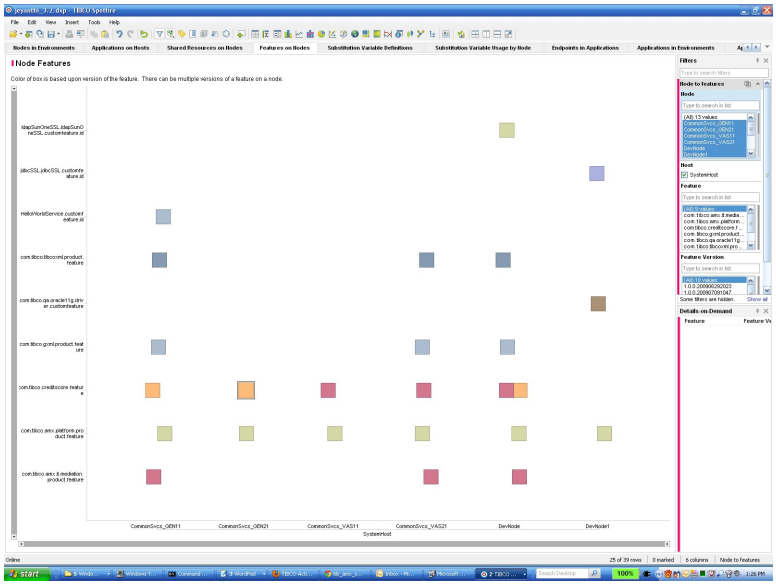
Features on Nodes Page

The **Features on Nodes** page shows the relationship between features and nodes, allowing you to visually identify which nodes have a specific version of a feature and which have multiple versions of a feature.

Feature names are listed along the Y-Axis and node names grouped by host along the X-Axis. The color of the box indicates the version of a feature. The **Details-on-Demand** window displays feature version and related values when you select one or more boxes. The selected boxes are marked in purple until you unselect them. Hovering over a box will show the version of the feature that it represents.

If there are multiple versions of a feature on a node, the boxes will be grouped very closely together and may partially overlap.

Colors are meaningful going across (e.g., same Feature id, different nodes) but not meaningful going up and down the graph (different Feature ids).



The data table used for this page is Node to Features.

The filters on this page are not grouped. The available filters are:

- **Node**: names for all nodes.
- **Host**: names for all hosts.
- **Feature**: names for all features.
- **Feature Version**: values for all feature versions.

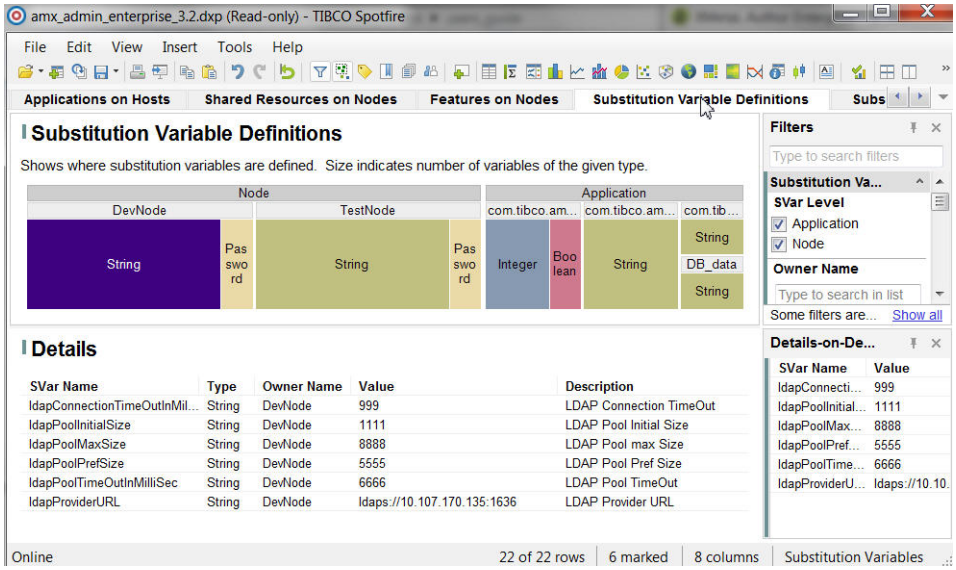
Substitution Variables Definitions Page

The **Substitution Variable Definitions** page shows where substitution variables are defined.

This page provides a global overview of all your substitution variables. It is useful for finding where specific variables are defined (by using the filters) and comparing the values at different levels. You can use it in conjunction with the **Substitution Variable Usage by Node** page to identify the impact of adding or deleting a substitution variable.

The size of the boxes indicates the number of variables of the specified type that are defined by that object. Grouping is by type of object, then name of the object and then type of variable. Only types of objects for which there is at least one entry will be shown, and only objects for which there is at least one entry will be shown. For example, if there are no node level substitution variables, the Node category will not be shown. Similarly, if there are no environment variables for environment `my_env`, but there are for `SystemEnvironment`, then Environment will be shown, but there will be no group for `my_env`.

The color of the box indicates the type of substitution variable.



The types of objects that can contain substitution variables are: Enterprise, Host, Node, Environment, and Application (Application Fragment level substitution variables will show up under Application).

The types of substitution variables are: String, Integer, Boolean and Password.

When you select a box, it turns purple to indicate it's selected. The **Details** table displays SVar Name, Type, Owner Name, Value and Description.

The **Details-on-Demand** window displays the SVar Name, Value, Type and Description and the Owner's name and level.

The data table used with this page is **Substitution Variables**.

The filters on this page are not grouped. The available filters are SVar Level, Owner Name, Type, SVar Name, Description, Value.

Substitution Variable Usage by Node Page

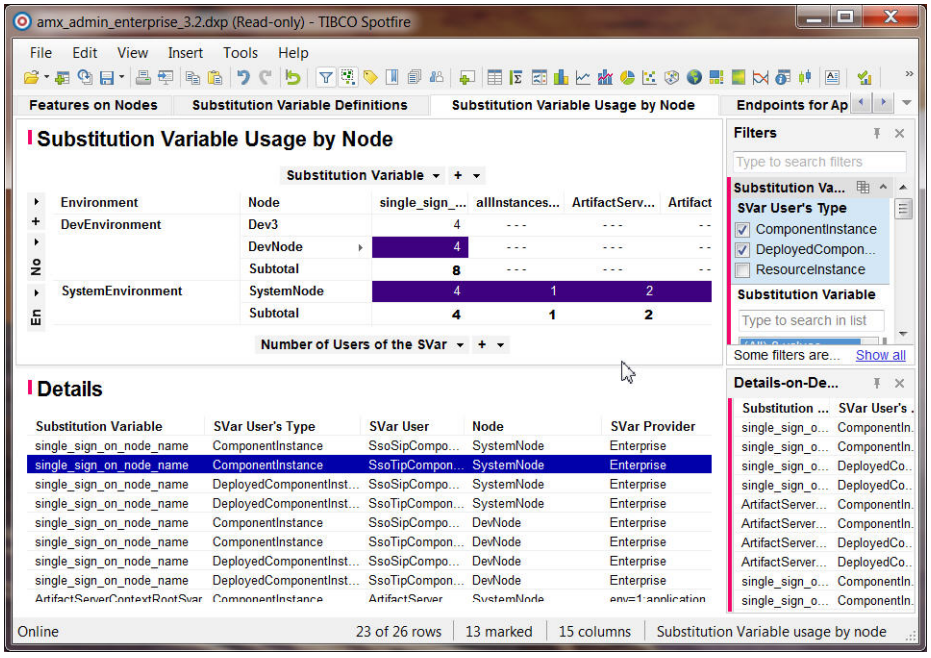
The **Substitution Variable Usage by Node** page shows how many and which components and resource instances use each variable on each node.

Each row of the table represents a node or a total. Usages of each substitution variable are totaled by environment and totaled for all environments. The first two columns in the table are the name of the environment containing the node and the name of the node. The remaining columns show usage counts for variables by node, except for the last column which shows the total usages of all substitution variables on the node or in the environment.

You may need to scroll to see all columns or all rows. Hovering just under the table brings up the horizontal scrollbar. Hovering just to the right of the table brings up the vertical scroll bar.

When a node has no substitution variable usages that match the current filters, the row for the node will not be shown. Similarly, if a substitution variable name has no usages on any node that meet the current filter selections, then that variable's column will not be shown. Therefore, as you narrow your search, columns that were previously scrolled out of view may appear.

When you select one or more variables in the chart, they turn purple. The **Details** table below shows details on what exactly is using the substitution variables. The SVar Provider is the object on which the substitution variable is defined (same as SVar Owner on the Substitution Variables Definitions page). For Enterprise level substitution variables, this is straightforward. For other substitution variables, it shows the path of objects of lower precedence with their internal id; the last value is the type of object on which the substitution variables is actually defined (along with it's id). You can enable the internal ids for filters or Details-on-Demand on other pages to find out which particular object the id refers to.



The data table used with this page is the Substitution Variables usage by node.

The filters on this page are not grouped. The available filters are:

- **SVar User's Type**: Component Instance (current configuration), Deployed Component (deployed component instance), Resource Instance.
- **Substitution Variable**: list of substitution variable names
- **Node**: list of node names
- **SVar Provider**: list of substitution variable provider values
- **SVar User**: names of the objects using the substitution variable
- **Environment**: list of environment names
- **Node Sync State**: In Sync, Out of Sync
- **Sync Reason for Node**: Empty for In Sync, or explanation of what is out of sync for the node

Endpoints for Applications Page

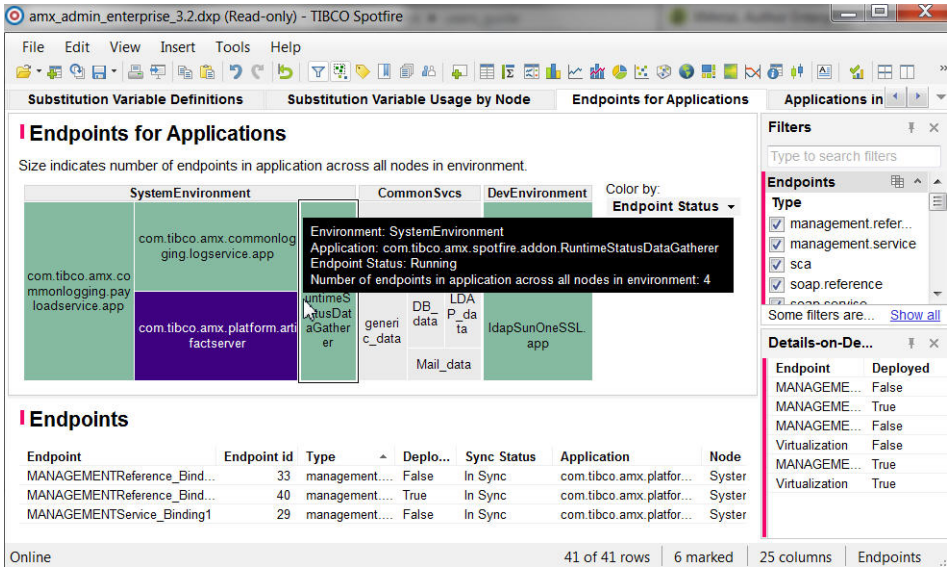
The **Endpoints for Applications** page shows the number of endpoints in applications across all nodes of an environment where the nodes are grouped by environment.

The size of the boxes indicates the number of endpoints the application has on the node. Hover your mouse over a selected box to see the exact number of endpoints for that application. Select an application to display information in the **Endpoints** table.

The **Endpoints** table underneath the graph lists the Endpoint name, id, type, whether deployed or not, runtime status, node, application name, sync status and environment. Selecting an applications fills in details in the **Endpoints** table below the graph. The **Details-on-Demand** window displays Endpoint and Deployed for any selected box(es).

The color of the box, for an ActiveMatrix Administrator 3.2 server, indicates the status of the endpoint. For 3.1, the color is associated with the environment.

If an application has endpoints but none of which are deployed, the status is displayed as `Empty` and the color grey.



The data table for this page is **Endpoints**.

The **Filter** group on this page is Status, and there are also ungrouped filters.

Status filter group: Deployed, Sync State, Sync Reason, Endpoint Status, Node Status.

The other available filters on this page are: Type, Endpoint, Node, Environment, Application, Interface - short, Interface, URI.

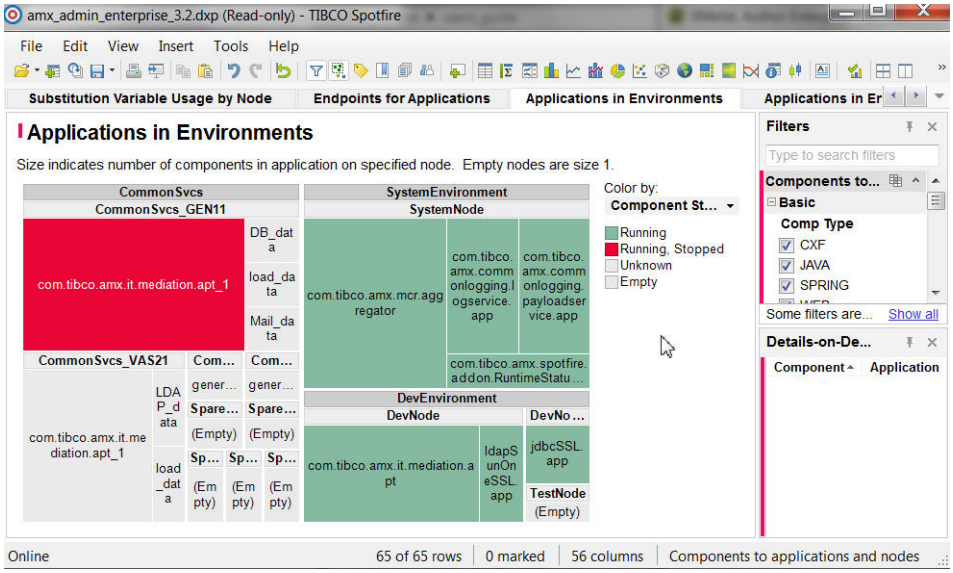
The Endpoint Status and Node Status filters and columns only display when using ActiveMatrix Administrator 3.2 or higher.

Applications in Environment Page

The **Applications in Environment** Page shows the number of components in applications on specified nodes. This page provides a "big picture" view of the size and health of your enterprise.

The size of the boxes indicates the number of components in an application. Hover your mouse over a selection to see the exact number of components for an application. Nodes with no components display as the same size as those that have a single component, but will contain the label "(Empty)". Environments with no nodes will simply have one node label "(Empty)".

A selection turns purple to mark it as selected. For unselected boxes the color indicates the Status, as shown in the legend (e.g., Running or Stopped). This applies only when using ActiveMatrix Administrator 3.2 or higher. For ActiveMatrix Administrator 3.1, the color indicates the environment.



The data table for this page is **Components and Applications to Nodes**.

The **Filter** groups for this page are: **Basic, Status**, and **Descriptive**.

- **Basic** filter group: Comp Type, Environment, Application, Component, Node, Host, Machine OS, Machine.
- **Status** filter group: Deployed - Comp, App Deployed, Component Status, Sync State Comp Inst, Sync Reason for Comp Inst, Node Status, Sync State Node, Sync Reason for Node, Node Installed, Node Startup Mode, System Node.
- **Descriptive** filter group: App Last Deployed, App Last Modified Date, App Description, App Contact, Node Contact, Env Contact, App Last Modifier, App Last Deployer, App Template, App Template Version, Component Version, Node Description, Node JVM Args, Machine Description.

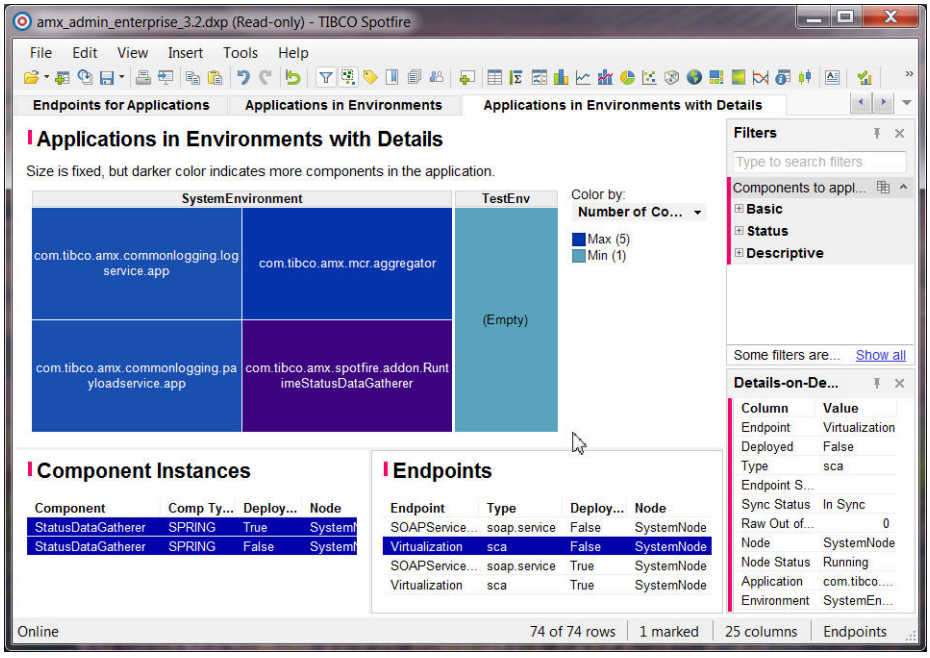
The Component Status and Node Status columns and filters only display when using ActiveMatrix Administrator 3.2 or higher.

Applications in Environments with Details Page

The **Applications in Environments with Details** page shows the number of components in the application with a darker color rather than a smaller or larger box. These are not grouped by node and can be useful for identifying large applications across your enterprise.

If an environment has any nodes without applications, or has no nodes, it will show the label "(Empty)". Selecting it will show the node name down in Component Instances.

A selection turns purple to mark it as selected. When one or more boxes are selected, values are shown in the Component Instances and Endpoints tables. If an application has no endpoints, then the Endpoints table will be empty. Selections in the Component Instances and Endpoints tables will cause all entries in the other with the same node to be selected, and the last thing selected controls the **Details-on-Demand**.



The data table for this page is **Components and Applications to Nodes**

The **Filter** groups for this page are: **Basic, Status**, and **Descriptive**.

- **Basic** filter group: Comp Type, Environment, Application, Component, Node, Host, Machine OS, Machine.
- **Status** filter group: Deployed - Comp, App Deployed, Component Status, Sync State Comp Inst, Sync Reason for Comp Inst, Node Status, Sync State Node, Sync Reason for Node, Node Installed, Node Startup Mode, System Node.
- **Descriptive** filter group: App Last Deployed, App Last Modified Date, App Description, App Contact, Node Contact, Env Contact, App Last Modifier, App Last Deployer, App Template, App Template Version, Component Version, Node Description, Node JVM Args, Machine Description

The Component Status and Node Status columns and filters only display when using ActiveMatrix Administrator 3.2 or higher.

Examples

The Spotfire Enabler lets you visualize relevant information. The visualization examples help you understand how to use a visualization to identify information about your enterprise.

There are many questions that the Spotfire Enabler can answer for you; the examples given are just a few to get started.

For details on each visualization page used in the examples, see Visualization Pages on page 23.

Example #1: Identifying Nodes with No User Applications

You can identify which nodes in your environment have no user applications distributed to them using the **Component Counts on Nodes** tab.

Procedure

1. Select the **Component Counts on Nodes** tab.
2. Select the bar above the number 0.
   The list of nodes will show up in a pie chart. (If 0 is not listed, then no nodes are empty).
3. Alternatively, narrow the range of the number of **Components on Node** filter to only include 0.
   The list of node names will display in the filter **Basic > Node**.

Example #2: Identify Which Nodes Have the Most Components

You can identify which nodes in your environment have the most components using the **Component Count on Nodes** tab.

Procedure

1. Select the **Component Count on Nodes** tab.
   The **X-Axis** indicates the number of components in the node, excluding platform application components and the **Y-Axis** shows how many nodes have exactly that number of components.
2. Select the rightmost bar in the chart.

Example #3: Identifying which Nodes use a Template

You can identify which nodes in your environment use a specific resource template for the resource instances by using the **Shared Resources on Nodes** tab.

Procedure

1. Select the tab **Shared Resources on Nodes**.
2. In the **Filters** panel in **Names > Template**, select the template of interest.
3. To see only nodes from a specific environment, use the **Environment** filter.

Example #4: Identifying Nodes With Older Feature Version

You can identify which nodes in your environment use an older version of a custom feature using the **Features on Nodes** tab.

Procedure

1. Select the **Features on Nodes** tab.

2. Left click on a specified name in the **Feature** filter box.
3. Select all versions in the **Feature Version** filter.
4. Deselect the current version in the **Feature Version** filter.

Example #5: Identifying Which Nodes Have ActiveMatrix Mediation

You can identify which nodes in your enterprise have the ActiveMatrix Mediation component installed by using the **Nodes in Environments** tab.

Procedure

1. Select the **Nodes in Environments** tab.
2. Select the application named `com.tibco.mediation.it`(ActiveMatrix Administrator 3.1) or `com.tibco.amx.it.mediation.opt` (ActiveMatrix Administrator 3.2 or higher).

Example #6: Verifying Node Names Begin with Environment Name for a Specific Host

You can identify which nodes in your environment for a given host begin with the environment name using the **Nodes in Environments** tab and the host filter.

Procedure

1. Select the **Nodes in Environments** tab.
2. Right click on the **Basic > Host** filter.
3. Left click on **Deselect All Values** and then click on the Host you want to view.
   The visualization will display the environment names across the top of the table and the node names in the boxes.

Example #7: Identifying Where a Substitution Variable Is In Use

You can identify where a substitution variable is being used in your environment using the **Substitution Variable Usage by Node** tab.

Procedure

1. Select the tab **Substitution Variable Usage by Node**.
2. Click on the name of the substitution variable in the filter **Substitution Variable**.
3. Draw a box around the **SVar** column and drag it down to the bottom before releasing.
   The **SVar User Type** and **SVar User** indicate the type and name of the object where a specific substitution variable is being used.

Example #8: Identifying Which Applications Are Deployed to Stopped Nodes

You can see which of your ActiveMatrix 3.2 applications are deployed to stopped nodes by using the **Nodes in Environments** tab.

Procedure

1. Select the **Nodes in Environments** tab.
2. In the **Status** filter group, uncheck Running in the **Node Status** filter.
3. View the **Application** list box in the **Basic** filter group. Alternatively, select all nodes being displayed in the top visualization and look at the **Applications** table.

Data Tables

The following table shows which views and tables from the ActiveMatrix Administrator database are used by each Spotfire data table.

Name	View(s) used	Table used
Components to Nodes	COMPONENT_NODES_VIEW	
Components to Applications and Nodes	COMPONENT_NODES_VIEW, APPLICATIONS_VIEW	STATUS_CACHE (3.2 only)
SharedResources	RESOURCES_VIEW, NODES_VIEW	STATUS_CACHE (3.2 only)
SubstitutionVariables	SVARS_VIEW	
Application with Nodes	APPLICATIONS_VIEW, COMPONENT_NODES_VIEW	
Substitution Variable Usage by Node	SVAR_RESOLVED_NODE_VIEW	
Node to Features	NODE_FEATURES_VIEW	
Endpoints	ENDPOINT_VIEW	STATUS_CACHE (3.2 only)

Sync State Bitmap

Data tables reference this sync state bitmap. The meaning of each bit is as follows:

1 = Configuration modified

2 = Logger modified

4 = Features have been added or removed from node

8 = Distribution modified

16 = Dependencies modified

32 = Upgrade defined but not applied

Application with Nodes Data Table

The following table identifies the column name and description for the Application with Nodes data table.

Column Name	Description
Application	Application name
App Contact	Contact for application
App Description	Description of application
App id	Internal identifier
App key	Text based internal identifier

Column Name	Description
App Last Modification Date	Date that the application configuration or distribution was last modified
App Last Modifier	Userid who last changed the application configuration or distribution
App Template	Name of the application template
App Template Version	Version of the application template used to create the component
App URL	Internal identifier
Deployed	Is application deployed (`True` or `False`)
Env id	Internal identifier
Last Deployer	Userid who last initiated deployment or redeployment of the app
Last Deployment Date	Date the application was last deployed or redeployed
Node	Node name

Components to Applications and Nodes Data Table

The following table identifies the column name, description, and meaning of values for the Components to Applications and Nodes data table.

Column Name	Description	Meaning of values
App Contact	Contact for application	
App Deployed	Is the application deployed	
App Description	Description of application	
App id	Internal identifier for application	
App Key	Internal identifier	
App Last Deployer	Userid who last initiated deployment or redeployment of the app	
App Last Deployment Date	Date the application was last deployed or redeployed	
App Last Modifier	Userid who last changed the application configuration or distribution	
App Last Modification Date	Date that the application configuration or distribution was last modified	
App URL	Internal identifier	
App Template	Application template name	
App Template Version	Application template version	
Application	Application name	
Application id	Internal identifier for application - same as App id	
Comp id	Internal identifier for component	

Column Name	Description	Meaning of values
Comp Instance id	Internal identifier for component instance	
Component	Component name	
Component Status	Runtime status of component (3.2 only)	
Component Version	Component version	
Comp URI	Internal identifier	
Comp Type	Shortened version of component's implementation	
Deployed - Comp	Is component instance deployed	`True` or `False`
Env Contact	Contact for the environment	
Env id	Internal identifier	
Environment	Environment name	
Env ref id	Internal identifier (same as Env id)	
Host	Host name	
Host id	Internal identifier	
Host Out of Sync	Bit mask for host's sync state	See Sync state bitmap
Impl Type - full	Component implementation's full name	
Machine id	Internal identifier	
Machine	Machine name	
Machine Description	Description of machine	
Machine OS	Operating system on which node is located	
Node	Node name	
Node Contact	Contact for node	
Node Description	Description of node	
Node id	Internal identifier	
Node Installed	Is node installed	`True` or `False`
Node JVM Args	Node's JVM arguments	
Node Startup Mode	Controls Node start when Host is restarted	`Automatic | Manual | Disabled`
Node Status	Runtime status of node (3.2 only)	
Raw Sync - Component	Bit mask for component instance's sync state	See Sync state bitmap
Raw Deployed	Is component instance deployed	`1 = true, 0 = false`
Raw Component Status	Status of component as reported by runtime (3.2 only)	

Column Name	Description	Meaning of values
Raw Installed	Is node installed	`1 = true, 0 = false`
Raw Node Status	Status of node as reported by runtime (3.2 only)	
Raw Node Out of Sync	Bit map for node's sync state	
Raw System Node	Does node contain an ActiveMatrix Administrator server instance	`1 = true, 0 = false`
Size	Count of component instances for use in visualization	Note: 0 is changed to 1
Sync Comp	Text description of component instance's sync state	
Sync Node	Text description of node's sync state	
Sync State Comp Inst	Sync state for component instance	
Sync State Node	Sync state for node	
Sync Reason for Comp Inst	If component instance is out of sync, describes what has changed	
Sync Reason for Node	If node is out of sync, describes what has changed	
System Node	Does node contain an ActiveMatrix Administrator server instance	`True` or `False`

Components to Nodes Data Table

The following table identifies the column name, description, and meaning of values for the Components to Nodes data table.

Column Name	Description	Meaning of values
Appl Deployed	Is application deployed	
App id	Internal identifier	
Comp id	Internal identifier	
Comp Instance id	Internal identifier	
Component	Component name	
Comp Version	Component version	
Comp URI	Internal identifier	
Comp Type	Shortened version of component's implementation	
Deployed	Is component instance deployed	`True` or `False`
Env Contact	Environment contact	
Env id	Internal identifier	
Environment	Environment name	

Column Name	Description	Meaning of values
Host	Host name	
Host id	Internal identifier	
Host out of Sync	Bit mask for host's sync state	See Sync state bitmap
Installed	Is node installed	`True` or `False`
Is Comp Inst in Sync	Is component instance in sync	
Is Node in Sync	Is node in sync	
Node	Node name	
Node Contact	Contact for node	
Node Description	Description of node	
Node id	Internal identifier	
Nodes JVM Args	Node's JVM arguments	
Number of Components on Node	Number of components on node	
Node Startup Mode	Controls Node start when Host is restarted	`Automatic | Manual | Disabled`
Machine	Machine name	
Machine Desc	Machine description	
Machine id	Internal identifier	
Machine OS	Operating system on which node is located	
Raw Impl Type	Component implementation's full name	
Raw Installed	Is node installed	`1 = true, 0 = false`
Raw is System	Does node contain an admin server instance	`1 = true, 0 = false`
Raw Node Out of Sync	Bit mask for node's sync state	See Sync state bitmap
Raw Sync - Component Instance	Bit mask for component instance's sync state	See Sync state bitmap
Sync Component Instance	Text description of component instance's sync state	
Sync Node	Text description of node's sync state	
Sync Reason for Comp Inst	If component instance is out of sync, describes what has changed	
Sync Reason for Node	If node is out of sync, describes what has changed	

Column Name	Description	Meaning of values
System Node	Does node contain an admin server instance	`True` or `False`

Node to Features Data Table

The following table identifies the column name and description for the Node to Features data table.

Column Name	Description
Feature	ID (i.e., name) of the feature
Feature Version	Version of the feature
Host	Host name
Host id	Internal identifier
Node	Node name
Node id	Internal identifier

Shared Resources Data Table

The following table identifies the column name, description, and meaning of values for the Shared Resources data table.

Column Name	Description	Meaning of values
Env id	Internal identifier	
Environment	Environment name	
Env Contact	Environment contact	
Host	Host name	
Host Out of Sync	Bit mask for host's sync state	
JNDI Name	Resource instance name	
Machine	Machine name	
Machine Description	Machine description	
Machine OS	Machine operating system	
Node	Name of the node on which the resource instance is defined	
Node Contact	Contact for node	
Node Description	Description of node	
Node id	Internal identifier	
Node id Dup	Duplicate of node id	
Node Installed	Is node installed	
Node JVM Args	Node JVM arguments	

Column Name	Description	Meaning of values
Node Startup Mode	Controls Node start when host is restarted	`Automatic | Manual | Disabled`
Node Status	Status of node (3.2 only)	
Node Sync Status	Sync status of node	
Raw is System Node	Does node contain an admin server	`1 = true, 0 = false`
Raw Node Installed	Is node installed	`1 = true, 0 = false`
Raw Node Status	Runtime status for node (3.2 only)	
Raw RI Status	Runtime status for resource instance (3.2 only)	
Raw Sync - Node	Bit mask for node's sync state	See Sync state bitmap
Raw Sync - RI	Bit mask for resource instance's sync state	See Sync state bitmap
Raw Type	Resource Instance full type Qname	
Res id	Internal identifier	
RI Status	Status of resource instance (3.2 only)	
RI Sync Status	Sync status of resource instance	
Sync Reason for Node	If node is out of sync, describes what has changed	
Sync Reason for RI	If resource instance is out of sync, describes what has changed	
Sync State Node	Sync state for node	
Sync State RI	Sync state for resource instance	
State	Is it installed or not	
System Node	Does node containActiveMatrix Administrator instance?	
Template	Resource Template name	
Template Description	Resource Template description	
Template Last Modifier	Userid which last changed the Resource Template used by this Resource Instance	
Template Last Modification Date	When the resource template's configuration was last changed	
Type	Type of resource instance	

Substitution Variables Data Table

The following table identifies the column name, description, and meaning of values for the Substitution Variables data table.

Column Name	Description	Meaning of values
SVar name	Name of substitution variable	
Description	Description of substitution variable	
Type	Type of substitution variable	
Value	Value of the substitution variable	
Owner id	Internal identifier	
Owner Type	Internal name for type of object on which the substitution variable is defined	`AdminConfig` = Enterprise `Environment` = Environment `Platform` = Host `TibcoNode`= Node `AmfApplication` = Application
Owner Name	Name of the object on which the substitution variable is defined	
SVar Level	Type of object on which the substitution variable is defined	Enterprise, Environment, Host, Node, Application

Substitution Variables Usage by Node Data Table

The following table identifies the column name, description, and meaning of values for the Substitution Variables Usage by Node data table.

Column Name	Description	Meaning of values
Env id	Internal identifier	
Environment	Environment name	
Node	Node name	
Node id	Internal identifier	
Node Out of Sync	Text description of node's sync state	
Owner id	Internal identifier	
Raw Node Out of Sync	Bit mask for node's sync state	See Sync state bitmap
Raw SVar Providers	A string showing the object currently providing the substitution variable and the set of objects with lesser	`urn:` prefixes everything `;` = separator

Column Name	Description	Meaning of values
	precedence which could provide a value if the substitution variable were removed from where it is currently defined	`ent` = Enterprise `env` = environment `platform` = host `node` = node `app` = application
Substitution Variable	Substitution variable name	
SVar Provider	More readable value	Enterprise or raw value minus the leading `urn:ent;`
SVar User	Name of an object that is using the substitution variable	
SVar User's Type	The type of the object which is using the substitution variable	`ResourceInstance, ComponentInstance, DeployedComponentInstance`

Endpoints Data Table

The following table identifies the column name, description, and meaning of values for the Endpoints data table.

Column Name	Description	Meaning of values
App id	Internal identifier	
Application	Application name	
Deployed	Is endpoint deployed?	`True` or `False`
Endpoint	Binding name	
Endpoint id	Internal identifier	
Endpoint Status	Runtime status of endpoint (3.2 only)	
Env id	Internal identifer	
Environment	Environment name	
Interface	Full QName of service or reference interface	
Interface - short	Local part of interface name	
Node	Node name	
Node id	Internal identifier	
Node Status	Runtime status of node (3.2 only)	
Raw Deployed	Is endpoint deployed	`1 = true, 0 = false`
Raw Endpoint Status	Runtime status of endpoint (3.2 only)	
Raw Node Status	Runtime status of node (3.2 only)	
Raw Out of Sync	Bit mask for endpoint's sync state	See Sync state bitmap

Column Name	Description	Meaning of values
Raw Type	Full Qname of binding type	
Ref Interface	Reference's interface (PortType)	null for service endpoints
Svc Interface	Service's interface (PortType)	null for reference endpoints
Sync Reason	Reason for sync	
Sync State	Text description of endpoint's sync state	`in sync` or `out of sync`
Sync Status	Text description of endpoint's sync state	`in sync` or `out-reason`
Type	Short version of type of Binding	
URI	Internal identifier used by runtime to identify the endpoint	

Appendix

A

Spotfire Visualizations

Visualizations are the key to analyzing data from your Spotfire analysis. To use the Spotfire Enabler, it is helpful to understand the basics of Spotfire visualization patterns and how to use the four filter types.

Topics

Spotfire Visualization Usage Patterns

Visualizations are the key to analyzing data from your Spotfire analysis.

You can quickly identify important information, such as candidate machines for creating new nodes, or establish maximum reuse of shared resource instances, shared libraries and application runtimes. You can also readily determine if your organization's deployment rules and best practices are being followed.

You can use a variety of visualization types to provide the best view of the data. These include tables, cross tables, graphical tables, bar charts, line charts, combination charts, pie charts, scatter plots, 3D scatter plots, map charts, treemaps, heat maps, parallel coordinate plots, summary tables, and box plots.

Different types of visualizations can be shown simultaneously. By letting values control visual attributes such as size, color, shape, and so on, your visualizations can reflect many dimensions of data. Visualizations can be linked to each other, but note that they may or may not dynamically update when you manipulate the corresponding filters on the page.

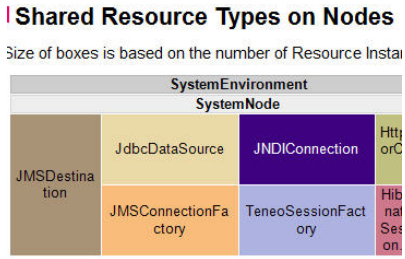
To understand what you see in a given visualization and how to make adjustments, it's helpful to know how to:

- select, unselect and reselect
- bookmark
- use Details-on-Demand
- move through tree maps
- use list boxes
- use tabs
- use tables

For more details on using Spotfire visualizations, see the Spotfire documentation.

Selecting

Here are some tips on how to select, unselect and reselect within a Spotfire visualization:



- To select a bar, left click on it or draw a box around it (this is called "marking".) The selection displays in purple.
- To select a single value, left click on it.
- To select multiple values, hold down the left mouse button and draw a box. All items that the box includes (even if only partially) will be selected and display in purple.
- To unselect a value, right click, move the cursor over the top item (**Marked Rows**), and then move the cursor to the right to the top item (Unmark) and left click. Unselecting limits the data showing in your visualization.
- **Details-on-Demand** shows values for a selected item. If only one value is selected in **Details-on-Demand**, it displays vertically; multiple values are shown as rows in a table with one value per row.
- To reselect everything in a single filter, right click on the filter and select **Reset Filters** from the toolbar.
- To reselect everything in all filters, right click on the filter and select **Reset Filters** from the toolbar.

Bookmarking

Bookmarking lets you return to a previous view.

A bookmark saves all marked rows in every marking, the active page and visualization, and any specified filters. You can link and modify bookmarks as well as organize them in folders.

Using Details-on-Demand

The **Details-on-Demand** window generally shows more data than the Details tables do. This window is below the **Filters** panel.

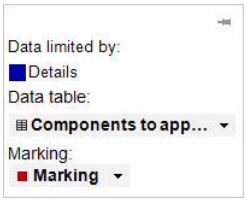
This window can display the exact values of a row or a group of rows. By clicking an item in a visualization, or marking several items by clicking and dragging the mouse around them, you can see the numerical values and textual data they represent directly in the **Details-on-Demand** window.

Details-on-Demand shows values for a selected item. If only one value is selected then it displays vertically; multiple values are shown as rows in a table with one value per row. When only one item is selected, **Details-on-Demand** presents a 2-column table for easy viewing. When multiple items show in this window, each item has its own row.

You can undock **Details-on-Demand** by left clicking on the push pin. This makes for easier viewing. For example, this enables viewing a reasonable number of columns for multiple values. You can redock the window by clicking on the pushpin at the top right of the window. If you click away, the undocked window disappears; retrieve it by selecting **View > Details-on-Demand**.

Showing The Legend

You can show and hide the legend.



You can show a hidden legend by hovering over the title bar and then left-clicking left the icon that displays. You will see a popup menu. Use the pushpin to dock it to your visualization. You can hide visible legends by clicking on same icon in the title bar.

Moving through Tree Maps

Tree maps provide useful views of your enterprise data.

You can drill into a tree map by clicking on a hierarchy header. Hovering over the header underlines it. Once drilled in, you can go back out by clicking on higher level element in the header. Hover on a Tree map's box to see a tooltip.

Using List Boxes

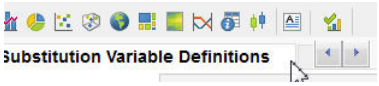
At the top of a List you see how many values match all other filters' selections.

List boxes have a search field at the top to limit the shown values to only those that contain the typed in string. This does not affect the selections of the filter.

The values matching all other filters' selections will show as (All) *X* values, where *X* is the number of values. Clicking on the (All) *X* values line will select all values (this has the same effect as using **Reset Filters** from the toolbar).

Using Tabs

The Spotfire Enabler tabs show at the top of a visualization. Since there are more tabs than fit on the screen, use the arrows to the right of the tabs to scroll through them.



See the section on Using Visualizations for more details on the individual pages and how to use them.

Using Tables

In tables, clicking on a column heading will sort the table by that column. If a table in a visualization is wider or longer than the space allocated to it, hover over it to display the appropriate slider bars.

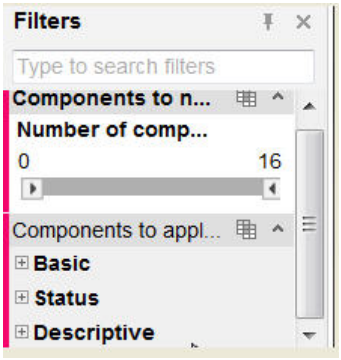
Using Filters

Filters are powerful tools to isolate data and make discoveries. Filters are critical to obtaining the views you want of your enterprise data.

By adjusting filters, you can reduce the data seen in the visualizations to isolate the data that interest you. For example, a filter could be adjusted so that data is only shown for a certain range of dates or for a certain number of components. When you manipulate a filter, you can instantly see how the current setting affects the visible data in the visualizations. You can instantly reset any or all filter changes you make. The visibility of filters is controlled per page, and the filter visibility of new pages is inherited from the active page. You can view and manage filters in a popover, a docked panel, or as a floating window.

When you manipulate a filter by moving a slider or by selecting a check box, all linked visualizations are immediately updated to reflect the new selection of data. Other filters are also immediately updated, and you can use the contents of the filters to answer some questions about your data.

Filters in some Spotfire Enabler pages may be organized into groups. When a group is closed, changes to the filters still remain active. As filters are selected, the other filters' contents change to reflect only values that match the selected filter.



There are several types of filters available with the Spotfire Enabler, each suited for a particular data type: **Range Filter**, **List Box Filter**, **Radio Button Filter** and **Check Box Filter**.

Filters on most of the Spotfire Enabler visualization pages are organized into groups. When a group is closed, changes to the filters still remain active. As filters are selected, the other filters' contents change to reflect only values that match the selected filter. See Using Filters on page 51Spotfire Visualizations on page 47for more information on using filters and filter types.

Filters panel

The Filters panel lets you perform actions related to filters. You can search using the Search function or scroll the list of available filters in the panel. You can find all filters that have been modified using `status:modified` as your search. You can find all filters for a specific data table using `datatable:<datatablename>` as your search.

You can also show and hide filters per page, change filter names, sort filters, group filters, show and hide groups of filters, sort filters within filter groups and reset any/all filters. Renaming a filter is equivalent to renaming the corresponding column in the data table.

Filter types

There are several different filter types to use in Spotfire Enabler visualizations: the range filter, the radio button filter, the check box filter and the list box filter.

Range filter

The range filter lets you narrow down the data shown to a range of values. You can adjust the handles using your mouse or use the arrow keys on your keyboard to adjust the range when the filter device is active. Only rows with values within the range will remain visible in the visualization.

You can narrow the range filter span, filter the data range, set the data range without filtering, and change the scale.

Clicking and dragging the middle part of the slider to move the range sideways lets you sweep over different "slices" of a data table. Observing the reactions of the other filters to such a sweep can provide interesting clues to correlation between parameters in the data table.

Radio button filter

In a radio button filter each value is represented by a radio button. This filter type is available for columns containing no more than 500 values.

The radio buttons are mutually exclusive, that is, only one of the alternatives in the filter can be set at a time. However, an (All) option is always present, letting you select all values. A (None) option is also available, letting you filter out all the values, showing nothing. If there are empty values present, a radio button named (Empty) will be available. Selecting this radio button will filter to the empty values.

Values that have been filtered out by other filters are indicated with gray text. If you select a radio button that is grayed out, nothing will be shown in the visualizations since that value has been filtered out already. When the filter is active, you can use the arrow keys on the keyboard to change the selected radio button.

Check box filter

In a check box filter, each unique value in the column is represented by a check box. This filter type is available for columns containing no more than 500 values.

One or several check boxes can be selected or cleared to determine which values are to appear in the visualizations. If there are empty values present, a check box called "Empty" will be available, letting you filter to those values. Values that have been filtered out by other filters are indicated with gray text.

When the filter is active, you can use the arrow buttons and the spacebar on the keyboard to select and clear check boxes. You can quickly select or clear all the values by resetting the filter or deselecting all values.

List box filter

The list box filter lets you select a few values from a long list of values present in the column. Select items in the list to filter to those items. Press Ctrl or Shift and click to select multiple values. If you click on the first alternative in the list (All), no filter is applied, and all values are shown.

You can narrow down the list of values by typing a string of text in the search field above the list. The values that do not match the string are removed from the list. As you type the string, the list is continuously updated with the values that match the current substring.

For list box filters in the text area, press Enter to search. By narrowing down the list, you can more easily find the values of interest, but this does not affect the applied filtering in any way. To apply filtering you must select values from the list. You can also enter an expression in the field to narrow down the list, applying the searching rules described in the in the Spotfire documentation.

Remove the text string from the search field to make all the values reappear in the list. The grey indicator on the right hand side of the filter shows where your selected values are located in the whole list. When you hover the mouse, a tooltip appears, showing how many values have been selected in the list. If fewer than six values are selected, you also see those values in the tooltip.