



# **TIBCO Spotfire® Application Profiler Tech Note**

*Software Release 12.0.0*

# Contents

---

Introduction.....	3
Architecture of Application Profiler.....	5
Extend Application Profiler with IronPython scripts.....	7
Use case: Monitor Spotfire library state.....	8
Use case: Assessing the likely effects of an upgrade.....	10
Use case: Collecting specific data.....	11
Resource considerations for test operations.....	12
Example: Resource considerations.....	14
Troubleshooting.....	17
Frequently Asked Questions.....	18
TIBCO Documentation and Support Services.....	20
Legal and Third-Party Notices.....	21

# Introduction

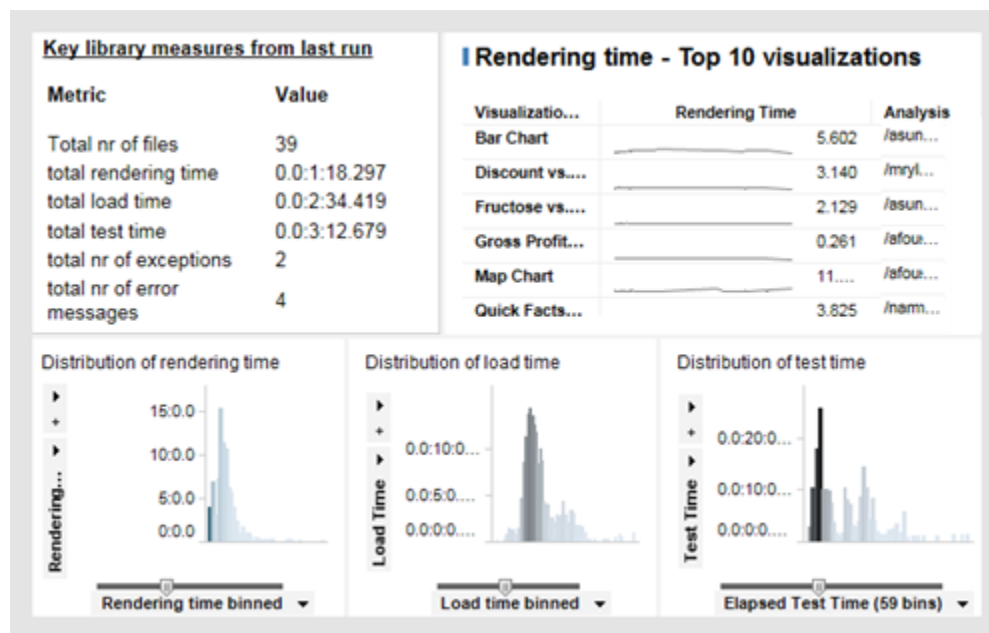
This is a document that describes how the Application Profiler works and how it is used. It also elaborates on a few example use cases, and finally, provides a few words of advice related to running the utility in your Spotfire library.

Business users love the ability to easily create new reports, duplicate reports, add more data to reports, and fundamentally change the contents and behavior of a report. But as a report goes through numerous changes, or new copies are created, a systems administrator can find it rather difficult to understand and maintain the different analytic applications of the system. The Application Profiler is a utility that allows a Spotfire administrator or a superuser to scan the Spotfire library and acquire an overview of its analysis files.

The Application Profiler helps answering some important questions:

- How many analysis files are there in the library?
- How long time does it take to load them?
- How many and which visualizations do they contain?
- How long time does it take to render the visualizations?
- Were there any configuration messages reported for any of the analysis files?
- How have these measures changed over the previous six months?
- Will an upgrade to another Spotfire version have any effect on the above measures?
- How many of all the analysis files make use of a specific custom expression?
- How many of my bar charts use horizontal bars AND stacked bars in combination?

One of the key features of the Application Profiler is to monitor the state of the library over time. Below is an image of a use case providing a view of how certain metrics of the library have developed over time.



The value in this use case is that a Spotfire administrator may identify trends before they grow serious:

- If the author of an analysis has configured a visualization in a way that it requires unreasonably long time to render.

- If the author of an analysis has configured a visualization in a way that is erroneous and generates error messages.
- If there is a sudden increase of visualizations of the same name this may indicate that a lot of copies of an analysis file has been created.

# Architecture of Application Profiler

---

This is an overview of the Spotfire Application Profiler, along with the inputs it accepts and the outputs you can expect.

The Application Profiler is accessed from the **Tools > Diagnostics** menu of the installed Spotfire client, and it is available to users with the appropriate license.

You specify which analysis files should be used as input in the Application Profiler dialog – this could be anything from a single analysis file to the entire Spotfire library. Then click **Start** to make the Application Profiler navigate through all pages in the analysis files to gather useful information.

In addition to the standard functionality, the Application Profiler offers a set of operations to enhance the output:

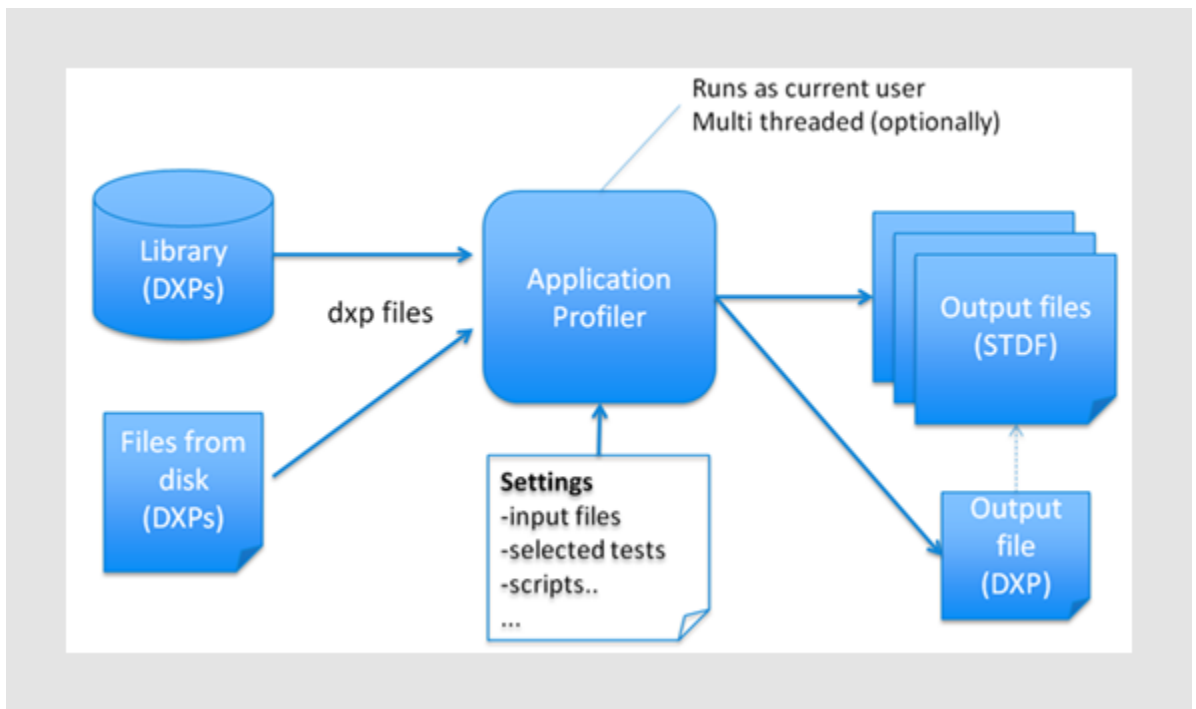
- Render each visualization
- Interact with visualizations through marking and filtering
- Creating new visualizations
- Apply bookmarks
- Remove/rename columns and data tables
- Save and reload files

The Application Profiler then runs each analysis file, records the time it takes to load the analysis file and to finish all requested operations, as well as any warnings or errors that occurred during the process.



The Application Profiler runs as the current user of the computer it is running on. This means that if the analysis file requires an external data connection, and the credentials are not stored in the DXP file itself, the connection is made using the current user's credentials. The Application Profiler also runs as a headless instance of Spotfire; meaning, there is no Main Form. However, it does require the ability to create a window.

When the test run has been completed, the output data is stored in a set of files using the STDF format. The files can then be visualized with Spotfire in a default dashboard by simply clicking a button. As with any other DXP file, you then have the possibility to create one or more visualizations focusing on the aspects of the data that are most important to you.



## Extend Application Profiler with IronPython scripts

---

You can create IronPython scripts to extend the Spotfire Application Profiler tests.

The Application Profiler is capable of performing a number of standard tests by default. However, it is also possible to create and add a custom IronPython script to the test process, to provide specific measures that are not covered by the standard options.

These scripts may perform additional operations, record additional information, and can be specified to run once for each one of the following:

- The entire document
- For each data table
- For each visualization (visual)

It is also possible to write data to the output data tables for further analysis.

Creating and adding IronPython scripts enables answering specific questions that you might have about your analysis files.

## Use case: Monitor Spotfire library state

---

The Application Profiler lets you monitor the entire library for key aspects of each analysis.

Spotfire administrators might find themselves in an organization with a large number of employees. Imagine that hundreds of these employees are Spotfire authors, modifying and creating analyses on a daily basis. As the analyses are under constant change, the library can be very difficult to maintain. Because of this challenge, Spotfire administrators often find it important to keep an eye on how the analyses develop over time. The Application Profiler provides a way of monitoring key figures per analysis and of the library as a whole.

With this in mind, let us assume that we want to monitor the library.

First, we determine a set of key measures to follow over time:

- The total number of analyses in the library.
- The total loading time of all analyses.
- The total rendering time for the visualizations in the analyses.
- The top 10 analyses with the longest rendering and loading time.
- Whether there are any errors or exceptions that occur.

Next, we must make a couple of configurations in the Application Profiler to acquire any results.

Most importantly, the Application Profiler must be run against all the analyses in the library. This is just a matter of selecting and adding the folders in the library where the analyses are stored in the **Included paths**.

Secondly, we must select what types of tests to run. In our case, we need to make sure we get information about the total loading time of all analyses, the total rendering time for the visualizations in the analyses, and possible errors or expectations that occur during the test.



The total loading time of all analyses as well as exceptions are recorded automatically during the test. If we would like to acquire the total rendering time for the visualizations in the analyses, and are curious about exceptions and errors that occur only during rendering, we must also select the test option **Render original visualizations**.

Now we have reached the point of running the test and pointing out the results displayed on our dashboards.

In our case, we want to identify the output for the total loading time of all analyses, and the total rendering time for the visualizations in the analysis. We also want to monitor errors and exceptions:

- The column **Elapsed Load Time** is found in the data table `ApplicationProfiler_Files`.
- The column **Rendering Time** is found in the data table `ApplicationProfiler_Visualizations`.
- The column **Errors**, which contains any error messages recorded, is found in `ApplicationProfiler_Visualizations`.
- The columns **Exception During Load** and **Exception During Test** are found in `ApplicationProfiler_Files`.

To compare our key measures over time, we use the column **Start Time**. This is the same column for all rows that belong to the same test run in the Application Profiler. To perform a comparison of when the test of a specific analysis started, we use the column **Analysis Start Time**.

To finally meet our objectives, we have to configure the following visualizations:

- Bar chart: Total number of files in the library: `UniqueCount(Analysis_Path)` over Start Time
- Line chart: Total Load Time: `Sum(Elapsed Load Time)` over Start Time



- Line chart: Total Rendering Time: Sum(Rendering Time) over Start Time

General advice: When monitoring the library over time, you should consider reviewing the test coverage that you need in relation to the run time of the different test options. This means that the more test coverage you need, the more test options you select, the higher the total run time needed. See the chapter [Resource considerations for test operations](#) for a discussion about this topic. At minimum, **Render original visualizations** should be selected in a monitoring use case.

## Use case: Assessing the likely effects of an upgrade

---

You can use the Application Profiler to automate testing on all the analysis files in a library to determine compatibility between the current and the new version of Spotfire.

Let us assume that a large enterprise organization is using Spotfire. There are several thousands of analyses located in their Spotfire library. An upgrade to a new version of Spotfire is scheduled. This is an example of a situation where business and IT are both eager to make sure the process runs as smooth as possible. But, when assessing the likely effects of an upgrade, testing that all analyses in the library works with the installed new version is the most daunting task.

In this case, the Application Profiler is very helpful.

The Application Profiler can in an automated fashion test several thousands of analyses. The tests can be done for both the original version of Spotfire and the new version, to compare different parameters and spot differences.

Proceed through the following steps to analyze library items for version compatibility:

1. Specify which analyses should be used as input.
2. Select a location where to save the output.
3. Select which test options to run, and save this configuration.
4. Run the tests with the Application Profiler in the current version of Spotfire.
5. Install the newer version of Spotfire on the computer used for the tests.
6. Load the saved configuration.
7. Change the prefix for the output files to identify the new version of Spotfire.



This is to make sure that, if the test is unsuccessful, the output files containing the data from the test run with the previous version of Spotfire does not contain irrelevant errors.

8. Run the test using the new version of Spotfire.

The result from the version compatibility test is that the output files, at the specified location, contain results from both versions of Spotfire. Each row contains information about the version of Spotfire used for the particular test.

It is also possible to compare various aspects of the test results for each Spotfire version. In the default dashboard, there are visualizations that compare various aspects for each Spotfire version, but you can also create your own comparisons.

To get a comprehensive test coverage in this type of use case, it can be useful to have all test options selected, if possible with regards to time and performance.

## Use case: Collecting specific data

---

Learn how to use the Application Profiler to evaluate library items.

In this use case, we would like to determine how many bar charts in the library are configured to display horizontal bars. Below are two scripts that we need to apply to the Application Profiler to perform this task.

The script in step one is used to write data to a column based on the orientation of the bar chart.

The script in step two declares a column for the new output called *BarChartOrientation*. This particular column is of the data type **string**.

### Procedure

1. To run the following script, it must be run from the "for each visual" context:

```
import Spotfire.Dxp.Application.Visuals.BarChartOrientation as barChartOrientation
from Spotfire.Dxp.Application.Visuals import BarChart
from Spotfire.Dxp.Application.Visuals import VisualTypeIdentifiers
if (Visual.TypeId == VisualTypeIdentifiers.BarChart):
    if (Visual.As[BarChart]().Orientation == barChartOrientation.Horizontal):
        OutputColumns["BarChartOrientation"] = "HORIZONTAL"
    else:
        if (Visual.As[BarChart]().Orientation == barChartOrientation.Vertical):
            OutputColumns["BarChartOrientation"] = "VERTICAL"
```

2. To write data to the new output column through the `OutputColumns["BarChartOrientation"]`, you must declare the `OutputColumn` through a script in the "Available OutputColumns for each Visual" context:


```
from System import Tuple, String
from Spotfire.Dxp.Data import DataType
OutputColumnDataTypes.Add(Tuple.Create[String,DataType]("BarChartOrientation",
    DataType.String))
```

# Resource considerations for test operations

This topic describes the test options available to the Application Profiler and how these options impact the run time of the tests.

## Test operations

Before selecting test operations, each option has a specific resource cost that you should take into consideration. For example, when selecting **Render original visualizations**, each of the visualizations in an analysis file is rendered. This operation can cost a lot of time. Many of the other test operations will cause several additional renderings of all visualizations, meaning longer run times of the tests.

Test	Description	Run time impact
<b>Render original visualizations</b>	Render all visualizations and stores the rendered image and the rendering time, as it was when the analysis was first opened.  This is the only time that images are stored, any subsequent renderings of an image will not be stored in the output files.	Depends on the number of visualizations and their configuration.
<b>Interact with visualizations</b>	Marks and highlight in all visualizations, which renders them again.	Renders all visualizations again, possible several times.
<b>Perform filtering</b>	Filters in all types of filters available, and in all unique combinations of filters. Dependent on the amount of unique filter combinations, the test may be very time consuming. There is an option to limit the number of filter combinations applied if desired.	Renders all visualizations several times, may increase test time with orders of magnitude.
<b>Apply Bookmarks</b>	Applies all bookmarks (one at a time) and renders all visualizations again for each bookmark. There is an option to limit the amount of bookmarks applied.	Renders all visualizations once for each bookmark.
<b>Remove/rename columns</b>	Removes columns and re-renders all plots. Renames columns and re-renders all plots.	Renders all visualizations once per removed or renamed column, may increase test time with orders of magnitude.
<b>Remove/rename tables</b>	Removes and renames tables and re-renders all plots.	Renders all visualizations once per removed data table.
<b>Create new visualizations</b>	Creates one new visualization of each supported type (in each file of the test run).	Only the time it takes to create and render the visualizations.
<b>Save and reload</b>	After having completed all selected tests, and if this option is selected, Application Profiler saves the file in the latest version of Spotfire and reruns the tests. Application Profiler always works on a temporary copy so the original file is not affected.	Approximately doubles the test time since it runs the test two times.

## Increase test speed

To increase the speed of your tests, you can configure the Application Profiler to run several parallel instances of Spotfire. In the Application Profiler main window, adjust the **Max number of test threads** value to control the number of instances of Spotfire that are in use.



The more test threads you run in parallel, the more memory Spotfire consumes. If the Application Profiler runs out of RAM memory, the test execution will slow down. If it runs out of disk memory, the tests are aborted and you will receive an undefined error message. When you run several test threads in parallel, it is recommended that you run the Application Profiler on a computer with a lot of RAM and disk space.

## Example: Resource considerations

This topic provides discoveries on how selecting different test options can impact the test run time.

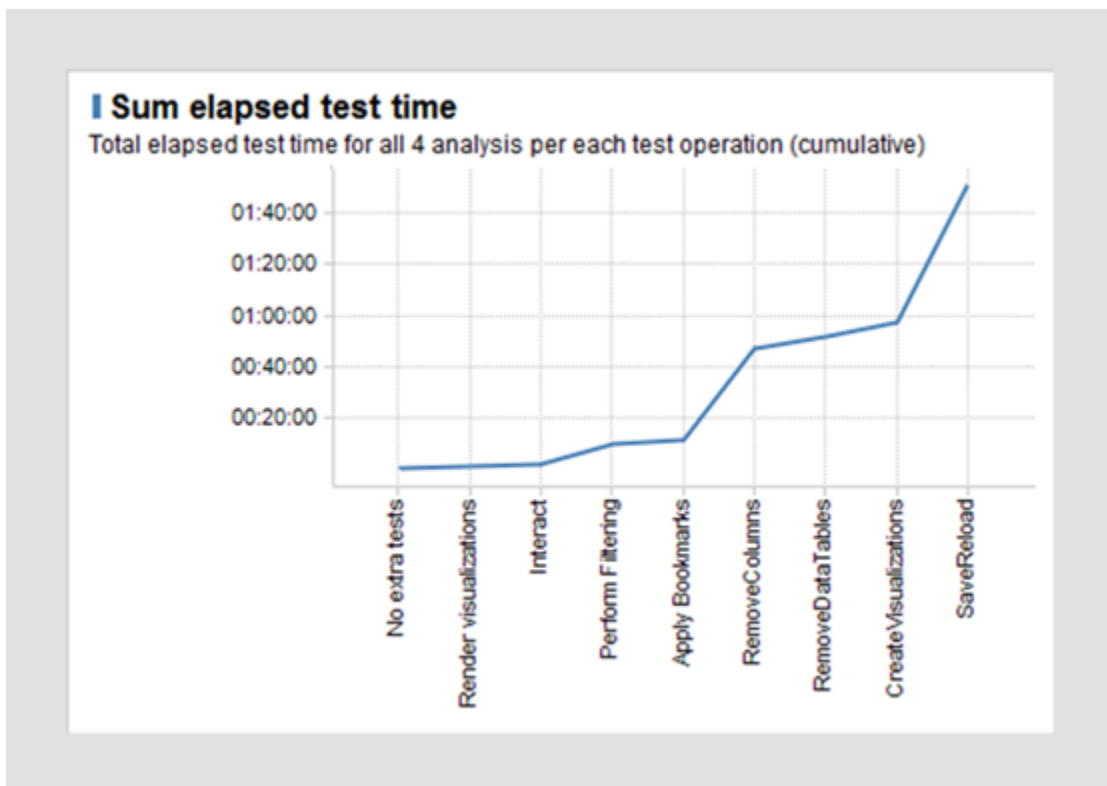
During the development of the Application Profiler, one test run contained 6,500 files and 70,000 visualizations. It took several days to complete the tests in this run. The run created 8.5 GB of output files and it took five minutes to visualize the results.

Below are a set of graphs from a test run on four files to demonstrate how different test options can impact the test run time.

The four files have the following characteristics:

Rows	Columns	Visualizations	Data tables
7,000	20	14	1
750,000	21	8	3
450,000	80	22	6
1,000	33	6	3

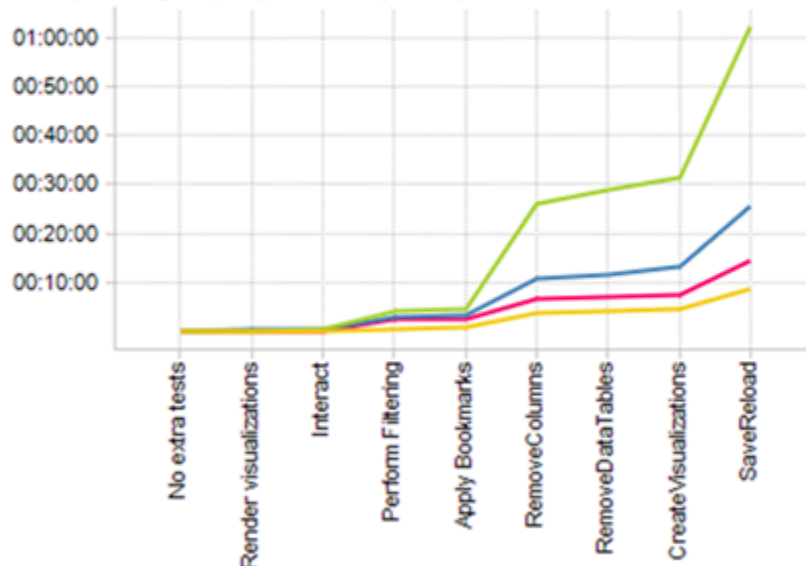
*Total elapsed time for all four analysis per each test operation (cumulative)*



*Elapsed test time per analysis per (cumulative) test operation*

### Sum elapsed test time

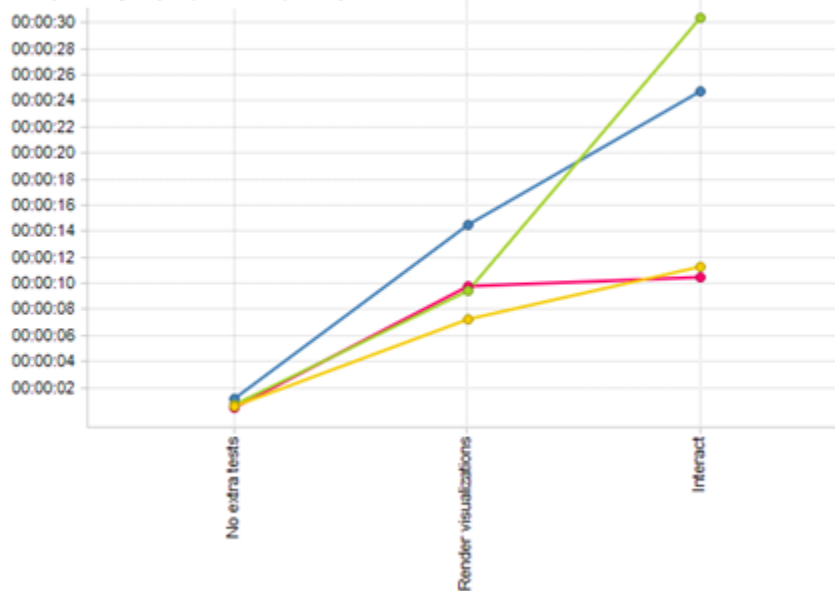
Elapsed test time per analysis per (cumulative) test operation



Elapsed test time per analysis per (cumulative) test operation that only include the test operations: No extra tests, Render visualizations, and Interact

### Sum elapsed test time

Elapsed test time per analysis per (cumulative) test operation



Below are a number of other discoveries regarding impact on the test run time that were made during the development of the Application Profiler:

- No **Extra test options** is much faster than selecting **Render original visualizations**.
- **Render original visualizations**, **Interact with visualizations**, and **Apply bookmarks** are much faster than **Perform filtering**, **Remove/rename columns**, and **Remove/rename tables**.
- **Save and reload** doubles the test time.



# Troubleshooting

---

This topic provides recommended and required actions when receiving unexpected results using the Application Profiler.

If a test run of the Application Profiler returns unexpected results, you should first investigate if this is manually reproducible.



The Application Profiler interacts with the analysis through the API, as opposed to the normal user interface (UI), so it is possible, though uncommon, that there are differences in behavior between normal usage and the Application Profiler tests results.

Below are some guidelines to actions that are needed when manually reproducing certain issues:

- If the Application Profiler detects a visualization error, you should be able to reproduce the error by opening the analysis and inspecting the visualization. If this can be repeated, you can investigate the error in more detail. If you cannot reproduce the issue manually, it might be impossible or very unlikely to occur during manual interaction.
- If the Application Profiler detects an exception while loading a file, you should be able to reproduce the error by opening the file manually and verifying whether it loads or whether the exception occurs. If the exception can be reproduced, contact TIBCO support. You will be asked to describe the issue and the steps required to repeat the issue.
- If Application Profiler detects an exception during a test, look into the Actions column for the file that had the exception, and identify the last actions that were performed before the exception. You should start by attempting to repeat that action. If performing that step does not trigger the exception, you might have to repeat several steps. If it is possible to reproduce the exception the steps manually, contact TIBCO Support.



Since the Application Profiler uses the public API rather than the normal user interface, it is possible, though not common that the Application Profiler triggers exceptions that cannot be triggered through normal UI interaction.

- If the rendering time of a visualization looks extreme, open the analysis and navigate to the page with the visualization. This triggers a rendering of the visualization.

General advice: If a problem can be reproduced using the normal Spotfire client and manual interaction, and if it is thought to be an error in the code, it should be reported to TIBCO support.

# Frequently Asked Questions

## Can Application Profiler run as a task from a task scheduler?

Yes, you can employ the Application Profiler command line interface to start the Application Profiler from the command line, using a task scheduler, or a batch file.



Use the setting **Run only when user is logged on** when scheduling the Application Profiler using the Microsoft Windows Task Scheduler. Running the Application Profiler with the task scheduler setting **Run whether the user is logged on or not** is currently not supported.

For example, you could customize the following command to launch the Application Profiler:

```
Spotfire.Dxp.exe /server:"http://myservername" /area:Production /username:user1 /
password:mypassword /applicationprofiler:"C:\Users\%USERNAME%\Documents\Application Profiler
\appprofileconfig.xml"
```

Switch	Description
/server	The URL to the Spotfire Server.
/area	The deployment area of the Spotfire Server that contains the analysis files to analyze.
/username	A user with access to Spotfire Server.
/password	The password for user you provided in /username.
/applicationprofiler	<div> Switch to invoke the Application Profiler. <div> <p>The config.xml file must have already been saved from the Application Profiler dialog to this folder. The path to the config.xml must be run from a command-line.</p> </div> </div>

## Can I run the Application Profiler in Web Player mode?

No, the Application Profiler runs as the current user in the installed client. However, it is very likely that issues detected using the installed client will also occur in the web client. So any testing that you perform with Application Profiler also benefits web client users.

## What authentication mechanisms can I use when I run Application Profiler?

Application Profiler supports Basic (username/password) and Windows Integrated Authentication. In both cases, the Application Profiler is authenticated as the current user.

## Will custom extensions in my DXP file work in the Application Profiler tests?

This depends on what the custom extension requires. The Application Profiler runs as a "headless" version (for example, it has no Main Form) of Spotfire Analyst. If the custom extension works in such an environment, it should also work when tested by the Application Profiler.

## The Application Profiler discovered an exception in a DXP file, should I contact TIBCO Support?

File a service request with TIBCO Support if you can reproduce the exception using manual interaction. Provide as much contextual information as possible, such as, whether you performed the Application

Profiler tests in different Spotfire versions, or only in one. And, if possible, provide the **Exception during load**, **Exception during test**, and **Actions** columns from the output.

### How to Contact TIBCO Support

Get an overview of [TIBCO Support](#). You can contact TIBCO Support in the following ways:

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

# TIBCO Documentation and Support Services

---

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

## How to Access TIBCO Documentation

Documentation for 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.

## TIBCO Spotfire Documentation

The documentation for all Spotfire products is available on the [TIBCO 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 TIBCO Spotfire products are designated as long-term support (LTS) versions. LTS versions are typically supported for up to 36 months from release. Defect corrections will typically be delivered in a new release version and as hotfixes or service packs to one or more LTS versions. See also [https://docs.tibco.com/pub/spotfire/general/LTS/spotfire\\_LTS\\_releases.htm](https://docs.tibco.com/pub/spotfire/general/LTS/spotfire_LTS_releases.htm).

## How to Contact TIBCO Support

Get an overview of [TIBCO Support](#). You can contact TIBCO Support in the following ways:

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

## System Requirements for Spotfire Products

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

## How to join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the [TIBCO Ideas Portal](#). For a free registration, go to <https://community.tibco.com>.

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

## Legal and Third-Party Notices

---

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

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

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

TIBCO, the TIBCO logo, the TIBCO O logo, TIBCO Spotfire, TIBCO Spotfire Analyst, TIBCO Spotfire Automation Services, TIBCO Spotfire Server, TIBCO Spotfire Web Player, TIBCO Enterprise Runtime for R, TIBCO Enterprise Runtime for R - Server Edition, TERR, TERR Server Edition, and TIBCO Spotfire Statistics Services are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

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

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

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

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

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

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

Copyright © 1994-2022. TIBCO Software Inc. All Rights Reserved.