

# IS Service Profiler



## User Guide

<b>Project</b>	ISSPROF
<b>Document File</b>	ISSPROF - User Guide.doc
<b>Version</b>	1.5
<b>Version Date</b>	17-09-2014
<b>Status</b>	FINAL
<b>Author</b>	António Abreu

## TABLE OF CONTENTS

<b>1</b>	<b>INTRODUCTION .....</b>	<b>8</b>
1.1	Document Objectives .....	8
1.2	Target Audience .....	8
1.3	Pre-requisites .....	8
1.4	Environment Requirements & Compatibility .....	8
<b>2</b>	<b>INSTALLATION AND SETUP.....</b>	<b>11</b>
2.1	First-time Installation .....	11
2.2	Intentionally preventing the Profiler from loading.....	12
2.3	Re-installing or Upgrading .....	13
2.4	Uninstalling .....	14
<b>3</b>	<b>SECURITY AND CONTROLLED ACCESS.....</b>	<b>14</b>
<b>4</b>	<b>FUNCTIONALITY GUIDE.....</b>	<b>15</b>
4.1	Administration Page.....	15
4.2	Licensing Pages .....	17
4.3	View Per Service.....	18
4.4	Package Coverage .....	21
4.5	Browse Snapshot.....	24
4.6	Browse Running Services.....	26
4.7	Export To File.....	29
4.7.1	CSV Format .....	30
4.7.2	XML Format .....	31
4.8	Load Snapshot from file .....	33
<b>5</b>	<b>EXTENSIBILITY .....</b>	<b>34</b>
5.1	Overview .....	34
5.2	Public API.....	35
5.2.1	Documents .....	35
5.2.2	Services .....	36
<b>6</b>	<b>TROUBLESHOOTING.....</b>	<b>39</b>
<b>7</b>	<b>SUPPORT.....</b>	<b>43</b>
7.1	Disclaimer.....	43
7.2	Reporting a Service Profiler issue.....	43
<b>8</b>	<b>FURTHER READING.....</b>	<b>44</b>
<b>APPENDIX A</b>	<b>GLOSSARY.....</b>	<b>1</b>

## FIGURE INDEX

Figure 1 - Administration option for preventing the Service Profiler from loading.....	12
Figure 2 - Preventing the Service Profiler from loading.....	12
Figure 3 - Reactivate the Service Profiler .....	13
Figure 4 - Start/Stop the Profiler .....	15
Figure 5 - The Settings & Options panel .....	16
Figure 6 - Managing Package Exclusion Patterns.....	17
Figure 7 - Licensing main page.....	17
Figure 8 - Licensed Service Volume breached .....	18
Figure 9 - The View Per Service report.....	19
Figure 10 - Timing display format .....	19
Figure 11 - The Package Coverage report – Snapshot View.....	22
Figure 12 - The Package Coverage report – Service Usage Statistics View .....	22
Figure 13 - The Package Coverage service usage detail .....	23
Figure 14 - The Browse Snapshot page.....	24
Figure 15 - The Browse Snapshot example of detail for a selected Service tree node .....	25
Figure 16 - Browse Running Services dashboard.....	26
Figure 17 - Browse Running Services with thread info .....	28
Figure 18 - Thread lock detail info.....	29
Figure 19 - Exporting Snapshot as a file .....	30
Figure 20 - Load snapshot file (invalid) .....	33
Figure 21 - Load snapshot file (valid) .....	33
Figure 22 - The Snapshot as a Document structure.....	34

## TABLE INDEX

Table 1 - Compatibility Categorization .....	8
Table 2 - Compatibility Matrix .....	9
Table 3 - Service Profiler version for Operating System.....	10
Table 4 - Setting to safeguard before installing an upgrade.....	14
Table 5 - Administration / Settings & Options .....	16
Table 6 - Package Exclusion Patterns form fields .....	17
Table 7 - View Per Service table for Snapshot Header .....	20
Table 8 - View Per Service table for Snapshot Details .....	20
Table 9 - Snapshot Browser options .....	25
Table 10 - Snapshot Browser node icons visual hints.....	25
Table 11 - Browse Running Services node icons additional visual hints.....	27
Table 12 - Browse Running Services thread icons when With thread info option .....	28
Table 13 - Snapshot XML dictionary.....	32
Table 14 - Documents defined as public .....	35



Table 15 - Admin request Services .....36

Table 16 - Snapshot request Services .....36

Table 17 - Out-of-the-box Snapshot Analysis Services.....36

Table 18 - Export Snapshot to file Services.....37

Table 19 - Utility Services .....37

Table 20 - Script timeout browser related info .....41

## 1 INTRODUCTION

The Wrightia's webMethods IS Service Profiler is a tool to collect and analyze runtime information about Integration Server Services. For the remaining of this document this tool is referenced as Service Profiler.

It is intended to have a small footprint while running, be easy and fast to install and uninstall, and to require no configuration over the Services being profiled.

### 1.1 Document Objectives

---

This document is the User Guide for v1.1, v1.2, v1.3 and v1.4 of the tool.

Its main objective is to be used as the reference for the user of the tool. The guide contains information spanning from installation and setup to advanced usage.

### 1.2 Target Audience

---

This document is targeted to any user of the tool, independently of the level of intervention, responsibility or technical ability.

### 1.3 Pre-requisites

---

This document will not discuss usability subjects that are specific to the webMethods platform and/or its tools, such as Integration Server administration, Developer, Service details, Package management and usage, etc.

So, a basic knowledge of the Integration Server structure and administration, Package management and Service development is a fundamental requirement.

For the purpose of installation, some basic knowledge of the underlying operating system is also required.

### 1.4 Environment Requirements & Compatibility

---

The tool has been constructed to be restricted only by the Integration Server requisites. Therefore, the only environment it needs to run is the Integration Server, and there is no dependency on an existing Broker, Audit DB, Modeler, Developer, Workflow or any other tool.

There is no additional memory requirement above the required by the Integration Server.

The technical specifications for the tool are defined for Hardware & Operating System platform, Integration Server and JVM compatibility. The compatibility can be categorized in a combination of Expected, Tested and Available, as explained in Table 1, below.

Table 1 - Compatibility Categorization

Category	Description
Expected	According to the technical specifications and/or documented behavior, it is expected to be 100% compatible.
Tested	The specific version has been successfully tested.
Available	A version for the specific configuration is readily available.

Table 2 - Compatibility Matrix

Item	Version	CPU	Data Model	Expected	Tested	Available	Comment
OS	Windows 2000 & XP	Intel	32	☑	☑	☑	
	Windows 2003 Server	Intel	32	☑	☑	☑	
	Windows 2003 Server x64	AMD64	64	☑	☑	☑	
	Windows 2008	Intel	32	☑	☑	☑	
	Windows 2008 x64	AMD64	64	☑	☑	☑	
	Sun Solaris	SPARC	32/64	☑	☑	☑	Tested on Solaris 8 & 9. The 64-bit version only runs on Solaris 9, or above.
		AMD64	32/64	☑	☑	☑	Tested on Solaris 10.
	AIX	PPC	32/64	☑	☑	☑	A small, but fundamental, part of the tool is distributed has a native <i>Dynamic Library</i> . Specific OS platform availability is only limited by the existence of the generated binaries, or the existence an ANSI C compiler in the target environment for a 1 <sup>st</sup> time compilation and test.
	HP-UX	PA-RISC	32	☑	☑	☑	Tested on HP_UX v11.11B.
		PA-RISC 64	32/64				
	Linux	Itanium	32/64	☑	☑	☑	
		Intel	32	☑	☑	☑	Tested with Red Hat and Suse. Tested also with Red Hat ES 4.0.
	OS X	AMD64	32/64	☑	☑	☑	Tested with Red Hat ES 4.0.
		Intel	32/64	☑	☑	☑	Only for Mac OS X v10.5 and above.
IS	v6.0			☑	☐	☑	
	v6.0.1 SP2			☑	☑	☑	
	v6.1 FP2			☑	☑	☑	
	v6.1 SP1			☑	☑	☑	
	v6.5.x			☑	☑	☑	Interface available only through the IS administration pages.
	v7.0.x			☑	☑	☑	
	v7.1.x			☑	☑	☑	
	v8.0.x			☑	☑	☑	
	v8.1.x			☑	☑	☑	
	v8.2.x			☑	☑	☑	
	v9.0			☑	☑	☑	
	v9.5			☑	☑	☑	
	v9.6			☑	☑	☑	
JVM	1.3.1		32	☑	☑	☑	When using a JRE other than the one bundled with the Integration Server, be sure to make the <code>tools.jar</code> library available. Default JRE distributions do not include this library. It can be retrieved from the JDK and placed in the <code>ext</code> folder of the JRE.  The only tested manufacturers were HP, IBM and SUN.
	1.4.0, 1.4.1		32	☑	☐	☑	
	1.4.2			☑	☑	☑	
	1.5.0		32/64	☑	☑	☑	
	1.6.x		32/64	☑	☑	☑	
	1.7.x		32/64	☑	☑	☑	

The **Integration Server** and **JVM** version listed in **Table 2 (above)** have been supported and tested since the first available version of the **Service Profiler**. However, the same cannot be alleged for the supported **Operating Systems**. The **OSs** have been tested and supported has the tool evolved, being introduced in specific tool versions, as stated in **Table 3 (below)**.

Table 3 - Service Profiler version for Operating System

OS	CPU Arch	Profiler version
Windows 2000 & XP	Intel	v1.0
Windows Server 2003	Intel	v1.0
Linux	Intel	v1.0
HP-UX	PA	v1.0
Sun Solaris	SPARC 32	v1.0
HP-UX	Itanium	v1.0.1
Linux	AMD64	v1.0.2 patch #3
Windows Server 2003 x64	AMD64	v1.0.2 patch #3
Sun Solaris	SPARC 64	v1.0.3
AIX	PPC / PPC64	v1.2.1
Sun Solaris	AMD64	v1.2.1 patch #1

## 2 INSTALLATION AND SETUP

The installation steps are detailed at a customized page from the tool's page at the website.

### 2.1 First-time Installation

---

Basically, the installation and setup is comprised of the following steps, in order:

- Install the packages;

Mind the order by which the packages are installed:

- WiaRoot;
- WiaUtilities;
- WiaRuntimeEcosystem;
- WiaServiceProfiler.

When this package is installed it may have load errors leaving it as **partial**. This is because some its **Java** services reference classes that reside in libraries in the `static` folder and these are only mounted when the **Integration Server** (re)starts.

The installation of this package does not give immediate access to the tools functionality and it will be partially disabled (as in *Figure 3, on page 13*). To fully enable the tool, the installation needs extra steps to be complete (*please read further*).

- Edit the startup scripts;

Depending on the version of the **Integration Server** and which is the **Operating System** the scripts and nature of the changes vary.

The purpose of this editing is to add to the server's **Java** command line the instructions to load the profiler classes as part of the **JVM**.

Refer for detailed steps available from the tool's download page at our website.

- Set the **License Key**.

When the tool is first installed there is no **License Key** set and any attempt to perform an action will result in a '**License not set**' exception. The license is set in the licensing pages of the tool.



## 2.2 Intentionally preventing the Profiler from loading

The **Service Profiler** can be intentionally prevented from being by the **Integration Server**, without the need to uninstall it. This can be achieved by means of putting a file named `ISSPROF_VOID` in the `<wm_dir>/IntegrationServer` folder (See [Figure 2, below](#)). The presence of this file causes the loading sequence to be skipped. A direct result is having the functionality links disabled in the tool pages menu.

This operation can be done and removed from the tool's admin page (see [Figure 1, below](#)) without the need to go to the file system and/or command-line prompt. As is shown in [Figure 1](#), upon pressing the **[Save Changes]** button, if the check-box is checked the before mentioned file is created, otherwise it is removed. Any option that requires an **Integration Server** restart to be effective is marked with the **!** icon by its side. After saving changes the icon starts glowing to indicate that the option is set but a restart is still required to make it effective.

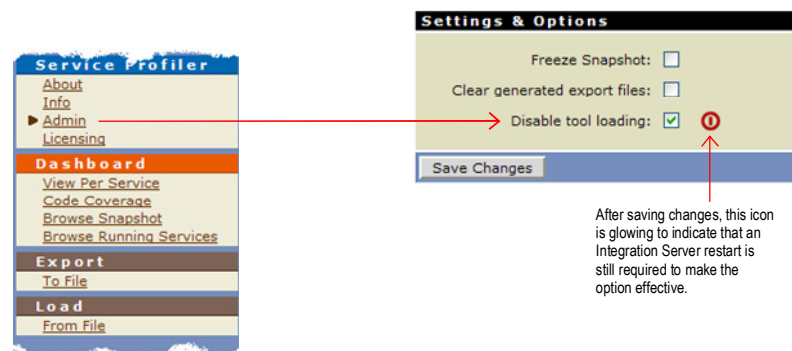


Figure 1 - Administration option for preventing the **Service Profiler** from loading

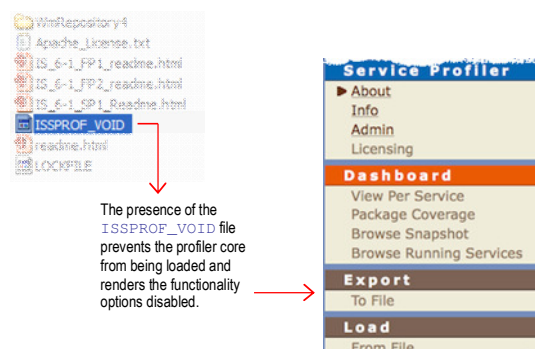


Figure 2 - Preventing the **Service Profiler** from loading

**IMPORTANT:** The functionality menus will be disabled whenever the tool core libraries are prevented from loading... or an error prevented them to be successfully loaded.

The reactivation of the tool can be done in the same administration option. When tool is deactivated, only the menu options that can be used while in this state are enabled. For the same reason, the admin page only contains options that are valid when the tool is deactivated (See [Figure 3, below](#)).

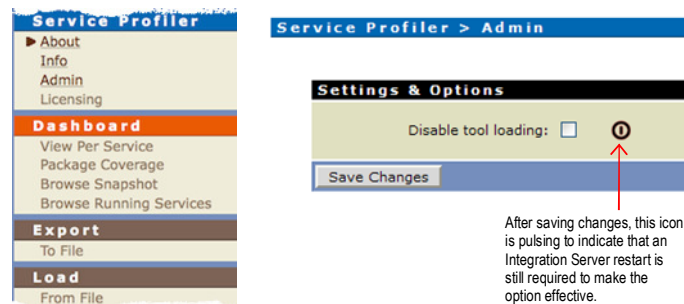


Figure 3 - Reactivate the Service Profiler

## 2.3 Re-installing or Upgrading

Due to the nature of the technology involved and the way it interacts with the **Integration Server**, not all updates can be installed while it is running:

- There are libraries being loaded even before the **Integration Server** starts and those can only be replaced by exiting the **JVM** completely.
- Even though an **IS Package** has its own class loader and could be replaced with the server running, the **WiaServiceProfiler** contains libraries that are directly loaded by the server class loader and thus cannot be unloaded without shutting it down. If you try to remove, or reinstall, this package using the package management pages it will not be successful... simply because the server will be locking some library files and the package files will not be deleted/moved/replaced.

As a rule-of-thumb, a reinstallation would only assured by first uninstalling the current version. However, this is not always needed and if so it should be referenced in the release notes specific to the new version or upgrade.

Some patches or **FIXes** will have their own install instructions, which could be highly simplified and not involve a complete uninstall, not even a server shutdown.

There are a few settings (*summarized in [Table 4](#), on page 14*) that you might be interested to safeguard before attempting an upgrade:

- Configuration and administrative settings;  
Make a copy of the **.cnf** files in the **config** folder of the **WiaServiceProfiler** package, to some place out of the package structure.  
If they happen to be removed by the (re)installation, copy those files back after the upgrade (a fresh installation does not (re)create them, but the start-up of the package does create empty versions).
- **ACL** settings.  
If you have customized the **ACL** settings you may have to redo those settings in the new installation.

Table 4 - Setting to safeguard before installing an upgrade

Setting	Action
Configuration and administrative	Copy the packages/WiaServiceProfiler/config/*.cnf files to a folder outside the package structure. These files contain specific settings customized after installation.
Security	Any ACLs settings customized after installation may have to be configured again in the server administration pages. Check those settings and write them down.

## 2.4 Uninstalling

Uninstalling the **Service Profiler** means the complete removal of its components and files. However, if the sole intent is to keep the tool from being loaded and have it performing no work at all, this can easily be accomplished without uninstalling (*please refer to [Intentionally preventing the Profiler from loading](#), on page 12*).

To completely uninstall the tool:

- Shutdown the **Integration Server**;  
If the **Integration Server** is running as an **Windows NT Service** you must unregister it before the next step.
- Revert the **Integration Server** launch script to the original;  
Copy the saved original scripts or just remove the lines inserted upon installation.  
Depending on the **Integration Server** version, which files these are may vary. Please refer to the installation procedure for details on the saved file and/or the inserted lines;
- Delete the folder <wm\_dir>/IntegrationServer/serviceprofiler;
- Delete the folder for the package WiaServiceProfiler;
- If the **Integration Server** is configured to run as a **Windows NT Service**, unregister the service and register it again.
- Start the **Integration Server**;
- Delete support **IS Packages**;  
If no longer needed (*e.g.*; not referenced by other packages), delete the packages WiaRuntimeEcosystem, WiaUtilities and WiaRoot.
- Remove the WiaServiceProfilers **ACL**.

## 3 SECURITY AND CONTROLLED ACCESS

The **Service Profiler** has the access controlled through its own **ACL** definition:  
WiaServiceProfilers.

By default, during installation, the **Developers** and the **Administrators User Groups** are associated with this **ACL**.

You can customize the access to the tool pages and services through the association of **User Groups** to this **ACL**. However, only users that are also associated with the **Administrators ACL** can use the **Licensing** pages of the tool.

## 4 FUNCTIONALITY GUIDE

### 4.1 Administration Page

This page is accessed through the **Admin** menu entry and contains the editing of settings that affect the overall tool behavior.

- Administration;

Direct administration actions over the **Service Profiler**. These actions are resumed to the **Start** and **Stop** of the tool.

**Started** means that the tool is actively collecting information about the currently running **IS Services**.

When the package is (re)loaded, the tool is **Stopped** and therefore any of the functionality based on snapshots will issue an (**Profiler Not Started**) error if used under this condition.

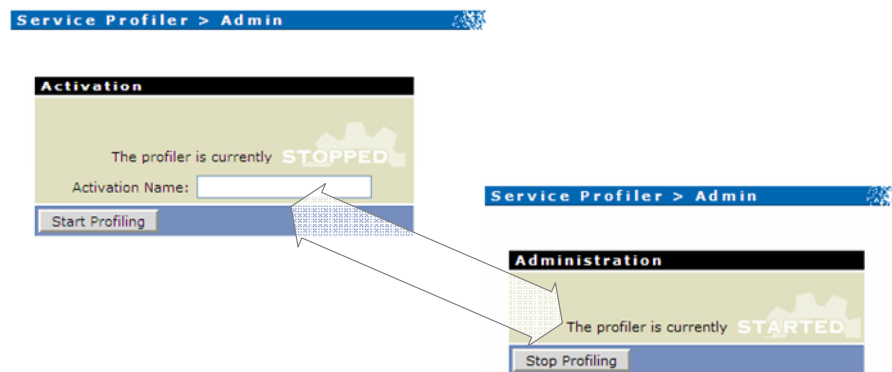


Figure 4 - Start/Stop the Profiler

The tool needs to be **Started**... and the service counters are reset.

Each time the tool is started, it is the same as saying it is activated, and the activation can be given a name of your choice. This name will have appended a timestamp and be placed as part of the snapshot's identification in its header. If no value is provided, it defaults to the hostname and port number.

- Settings & Options;

This panel commands the setting of some property-like settings and also the execution of available administration related options.




Some of the properties/options may require a server restart to take effect. These are identified by having the  icon by its side. After saving changes the icon starts glowing to indicate that the option is set but a restart is still required to make it effective.

Table 5 - Administration / Settings &amp; Options

Setting	Description
<b>Freeze Snapshot</b>	<p>Take a snapshot of the current profiling data and hold it in memory, using it as cached value for the next snapshot requests. This setting allows performing all the <b>Data Analysis</b> over the same snapshot, in opposition to always getting a new snapshot every time an operation is performed or a refresh is requested.</p> <p>If activated, the tool continues to gather profiling information about the running Services, but when a snapshot is requested the one cached (<i>aka</i>, frozen) is returned.</p> <p>When the tool is <b>Stopped</b>, this option is automatically set <b>ON</b> to allow the analysis to continue working over the last know snapshot before the <b>Stop</b> action.</p> <p>When this setting is <b>ON</b>, an  icon is shown in the affected pages, working as a visual hint to the source of the data.</p> <p>If the icon is accompanied by an  icon, this means that the freeze was triggered by a snapshot loaded from an XML file.</p>
<b>Clear generated export files</b>	<p>Deletes all files generated as a consequence of the <b>File Export</b> operation with the <b>Generate file &amp; link</b> option (<i>see Export To File, on page 29</i>).</p> <p>Each a file export of this type creates a file in the package's (outbound) area.</p> <p>Even though these files are automatically deleted every time the package is reloaded, this option allows deleting those files without having to reload the package.</p>
<b>Disable tool loading</b>	<p>This option prevents the Service Profiler from loading its libraries when the Integration Server starts (<i>See Intentionally preventing the Profiler from loading, in page 12</i>).</p> <p>When this option is activated, the tool's menus are disabled and it has absolutely no effect over the hosting Integration Server.</p> <p>A server restart is required to make the changes effective.</p>
<b>Auto-start upon package (re)load</b>	<p>If checked (and saved), the tool will be automatically started upon its package (re)load. When started, it will assume the activation name given in the <b>Auto-started activation Name</b> field.</p> <p>By default it is <u>not</u> checked, and the default <b>Activation Name</b> is <b>AUTOSTARTED</b>.</p>
<b>Publish snapshot on shutdown</b>	<p>If checked (and saved) and if the tool is started, the package unload/shutdown will trigger the publication of the current snapshot.</p> <p>No subscription and/or processing of this publication are implemented. It exists solely to give you a chance of saving the last state of the <b>Service Counters</b> before deleting it from memory. With this publication, it can be any persistence mechanism you can build, triggered by the subscription of the published document.</p>

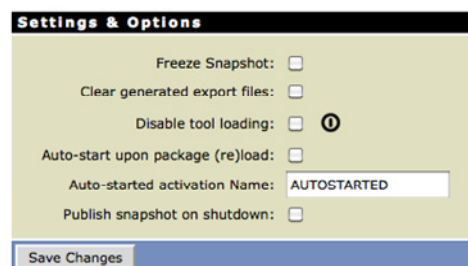


Figure 5 - The Settings &amp; Options panel

- Package Exclusion Patterns;

While running, the **tool** establishes no filters: it collects info for all running Services, because they all contribute to your system performance.

However, these may not all be under your control or even the object of interest for your analysis. So, you can establish which packages you wish to rule out through **Regular Expression** patterns over the package names (the result of the applied pattern can be immediately checked, to avoid runtime expression evaluation errors).

Table 6 - Package Exclusion Patterns form fields

Field/Item	Description
New Pattern	New Regular Expression Pattern to add or test. When the [Save Changes] button is clicked this expression is added to the list.
Check	Click on the icon to see the list of packages it defines.
Remove	Indicate which Name Patterns are to be removed from the list by checking the corresponding checkbox. When the [Save Changes] button is clicked all checked expressions are removed from the list.
Disable	This checkbox allows disabling the evaluation of the pattern without actually having to remove it. This is very useful for temporarily removing the filter.

1

Define the Regular Expression you want to add.

2

Test the RE by checking which its result is.

3

The list of Package Names resulting from the RE is shown here.

Package Exclusion Patterns

New pattern: W.\*Demo\$

Remove

Name Pattern

Check

Disabled

☐

^Wm.\*

☒

☒

^WiaServiceProfiler\$

☐

☐

^Wia[^P].\*

☐

Save Changes

Package List (W.\*Demo\$)

WiaServiceProfilerDemo

4

Save it.

A pattern can be disabled without actually having to remove it.  
The setting is effective after clicking [Save Changes].

The already defined REs can also be checked and/or selected for removal before clicking [Save Changes].

Figure 6 - Managing Package Exclusion Patterns

4.2 Licensing Pages

This functionality is only accessible to users associated with the Administrators ACL

The entry page shows the current licensing information (Expiration Date, Status, etc.) and can be used to set a new valid License Key.

ServiceProfiler > Licensing

Set New License Key

Reset 'Services Top Volume' Counter

Licensing Information

Licensed To

License Level

Expiration Date

Evaluation Yes

Status OK

Licensed Service Volume 100

Services Top Volume 61

Figure 7 - Licensing main page

**IMPORTANT:** Take notice that the pointed functionality option is **Set New License Key** and not **Edit License Key**. There is a difference: for the first, the current License Key is never shown.

The name of the customer for whom the License Key was issued to is also part of the license information.

If the license defines a limit to the number of allowed profiled services, the Licensed Service Volume value reflects the maximum number of services that are identified in the snapshots. The Services Top Volume is a count of how many services have been considered to the already taken snapshots, since the tool is up and running or since the last reset. This value is updated each time a snapshot is taken and it will be kept between start/stop cycles of the tool.

When the Services Top Volume value exceeds the Licensed Service Volume, an icon is added to the first (see Figure 8, below). The count is an indication of the dimension of the universe of services being actually profiled, and also to help establish the adequate licensed value if an upgrade is required.

When the icon appears, it means that not all individual services will be shown in the dashboards and this icon is assigned to the counters of those unlicensed services.

Licensing Information	
Licensed To	
License Level	
Expiration Date	
Evaluation	Yes
Status	OK
Licensed Service Volume	100
Services Top Volume	105

Figure 8 - Licensed Service Volume breached

The Reset 'Service Top Volume' Counter option resets the counter.

### 4.3 View Per Service

The View Per Service dashboard page (see Figure 9, below) is a sample Analysis Tool that builds a view/report of the generated hierarchical snapshot as a flat representation, accumulating per Service.



Dashboard > View Per Service

Snapshot Identification				
Sample Key	Server Name	Begin Time	End Time	MTC
WIA-AA:5555-20061108182045	WIA-AA:5555	08-11-2006 18:20:45.758	08-11-2006 18:52:38.138	2

Snapshot Details per Service								
Minimum Own Elapsed Time (D HH:MM:SS.m): 0.100								
Reported Times are averages: <input type="checkbox"/>								
<input type="button" value="Refresh"/>								
Package	Service	Type	Count	Errors	Elapsed	Spent	With Child Code	Spent
WiaServiceProfilerDemo	wia.demo.issprofiler.usecases.uc2:createDocument		53	53	1:03.308	0.100	1:03.308	0.100
WiaServiceProfilerDemo	wia.demo.issprofiler.usecases.uc1:singleService		12	12	18.177	0.050	28.550	0.150
WiaServiceProfilerDemo	wia.demo.issprofiler.usecases.uc3:looping		1	0	1.811	1.452	22.132	16.374
WiaServiceProfilerDemo	wia.demo.issprofiler.usecases.uc3:createStringList		8412	0	2.245	1.602	5.429	3.856
WiaServiceProfilerDemo	wia.demo.issprofiler.usecases.uc1:looping		1	1	0.141	0.000	0.491	0.020
WiaServiceProfilerDemo	wia.demo.issprofiler.usecases.uc3:order		8412	0	10.444	8.012	20.341	14.921
WiaServiceProfilerDemo	wia.demo.issprofiler.usecases.uc2:twoLevels		53	53	53.207	0.170	1:56.545	0.270
WiaServiceProfilerDemo	wia.demo.issprofiler.usecases.uc3:orderStringList		8412	0	0.522	0.401	0.522	0.401
WiaServiceProfilerDemo	wia.demo.issprofiler.usecases.uc5.amabr_issue_nr9_svcPortType:queryEntity		15	0	0.130	0.010	2.151	0.090
WiaServiceProfilerDemo	wia.demo.issprofiler.ui.run_uc:getUseCasesMetadata		109	0	0.211	0.040	0.221	0.050
AA_Issue_9_DB_Connections	amabr.issue.nr9.DB:newInstance		1	0	0.581	0.010	0.581	0.010
WiaServiceProfilerDemo	wia.demo.issprofiler.usecases.uc3:orderNumericList		8412	0	1.394	0.991	1.394	0.991
AA_Issue_9_DB_Connections	amabr.issue.nr9.DB:getName		14	0	0.150	0.000	0.150	0.000
WiaServiceProfilerDemo	wia.demo.issprofiler.usecases.uc2:looping		1	1	0.251	0.000	0.541	0.010
WiaServiceProfilerDemo	wia.demo.issprofiler.ui.run_uc:triggerUseCase		108	0	0.170	0.040	0.210	0.080
WiaServiceProfilerDemo	wia.demo.issprofiler.util:randomString		168240	0	3.184	2.253	3.184	2.253
WiaServiceProfilerDemo	wia.demo.issprofiler.config:get		9697	0	0.120	0.120	0.120	0.120
WiaServiceProfilerDemo	wia.demo.issprofiler.ui.run_uc:getMenuMetadata		2	0	0.461	0.000	0.461	0.000
Grand Totals:			2:36.507	15.252	4:26.331	39.697		

[any]	[unlicensed]	
-------	--------------	--

Unlicensed nodes are collapsed together and shown like this.

Figure 9 - The View Per Service report

The source gathered data is an invocation tree where the same Service may be called from different parents, but this particular view disregards the service interrelations and sums all values for each individually identified Service into a single accumulator set.

All timing values are displayed in the data format explained in Figure 10 (below) with the leading zeros suppressed with exception to time value smaller than one second.

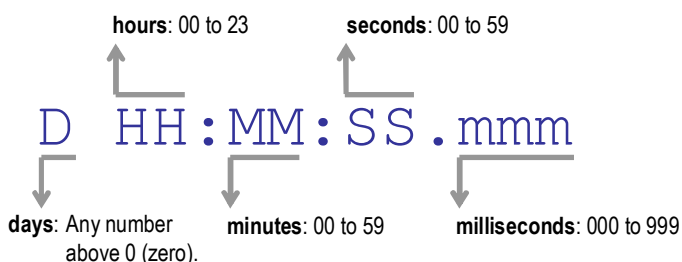


Figure 10 - Timing display format

The table can be sorted by any of the detail columns by clicking on the corresponding column header. Clicking multiple time toggles between ascending and descending sort order. Default (first click) sorting order depends on the data type: numeric and time columns start by being sorted descending while the others are ascending.

The information is laid in two separate tables:

- One for the **Snapshot Header**;

This table contains data that identifies the snapshot,












- One for the **Snapshot Details**.

This table contains the information about each Service found in the snapshot, on an *one-Service-per-line* layout.

Table 7 - View Per Service table for Snapshot Header

Snapshot Header	Description
<b>Sample Key</b>	A unique key that identifies the current snapshot. It reflects the <b>Activation Name</b> , if given when the tool was started ( <i>See Administration Page, on page 15</i> ).
<b>Server Name</b>	The net name of the <b>Integration Server</b> host, and its port.
<b>Begin Time</b>	The timestamp of when the profiling was started.
<b>End Time</b>	The timestamp of when the snapshot was requested.
<b>MTC</b>	<b>Maximum Thread Count.</b> This value contains the maximum number of threads that were used in parallel to execute the listed services. It is useful to give the user an idea of what percentage of defined <b>Service Threads</b> were (or are being) used during the profiling session. If the number of threads is equal, or close, to the maximum defined in the <b>Integration Server</b> , there may be services spending time waiting for a thread to become free.

Table 8 - View Per Service table for Snapshot Details

Snapshot Details	Description
<b>Package</b>	The name of the Package where the profiled Service is defined.
<b>Service</b>	The fully-qualified runtime name of the profiled Service.
<b>Type</b>	<p>The Service Type:</p> <ul style="list-style-type: none"> <li> Flow Service</li> <li> Java Service</li> <li> Adapter Service</li> <li> C/C++ Service</li> <li> Web Service</li> <li> XSLT Service<sup>1</sup></li> <li> Blaze Service</li> <li> Warning!</li> </ul> <p>Service name not found. This may happen if the package is disabled or the Service is renamed during the profiling period.</p> <p> Unlicensed count</p> <p>Aggregated counts for all found services that fall out of the <b>Licensed Service Volume</b> (<i>see Licensing Pages, on page 17</i>).</p> <p>The service type icon is also a link to the <b>Browse Snapshot</b> page (<i>see page 24</i>), where the occurrences of the corresponding service will be automatically highlighted.</p> <p>The link is made on the icon, and not on the Service column, to save horizontal space on the page: due to <b>webMethods</b> styling for links, their text appears as bold and this would widen considerably the Service table column.</p>
<b>Call Count</b>	<p>The accumulated number of times the Service has been invoked.</p> <p>The Service can be called from a number of other Services and has a Call Count for each of those calling nodes. However, this value sums all those counters.</p>
<b>Error Count</b>	<p>The accumulated number of exceptions the Service has raised.</p> <p>The Service can be called from a number of other Services and has an Exception Count for each of those calling nodes. However, this value sums all those counters.</p>
<b>Own Code</b>	<p>Timings for the profiled Service Own Code.</p> <p>This means that is does not include the time spent in Services it calls, only the time it took executing its own functionality code.</p>
<b>With Child Code</b>	<p>Timings for the profiled Service Own Code <u>plus</u> the time spent in Services it calls.</p> <p>The <b>With Child...</b> columns have defined a tip label showing the minimum and maximum timing values the corresponding Service took on a single call. Just place the mouse cursor over the table cell and the tip will appear for a short while.</p>
<b>Elapsed</b>	Clock time passed from the beginning of the call until it ends.
<b>Spent</b>	Actual time spend in the CPU doing work.

<sup>1</sup> This kind of service was introduced with **v6.5** of the **webMethods Integration Platform**.

Additionally, there are filtering options that further limits the amount of data when the page is refreshed, by means of clicking on the **[Refresh]** button. These options are:

- **Minimum Own Elapsed Time;**

Do not show Services that have an **Own Elapsed Time** less than the established value.

This allows you to concentrate the analysis only on the Services that take the longest... or significant timings.

- **Reported Times are averages.**

When checked, the values of **Elapsed** and **Spent** are the calculated averages for the **Call Count**.

## 4.4 Package Coverage

---

The **Package Coverage**<sup>2</sup> report (*Figure 11, below*) is another example of an **Analysis Tool**.

It presents a view over the **Snapshot** data, reporting the percentage of **Services** in a **Package** that have effectively been run:

- While the profile was active;

The results are based solely on the number of services for each package present in the snapshot against the universe of services defined by their respective packages. From this, the percentage of services used per package is calculated.

This is the default view, if a snapshot from the **Service Profiler** is available.

- Since the **Integration Server** was started.

The results are based on the **Integration Server**'s service usage statistics.

This is the default view, if a snapshot from the **Service Profiler** is not available. In this case the **Base coverage analysis on current server's usage statistics** option is automatically selected and cannot be changed.

Check the **Base coverage analysis on current server's usage statistics** option and press **[Refresh]** to change to this view.

In this view, the **Snapshot Identification** table is removed, because the information source is not a snapshot.

---

<sup>2</sup> This report was known as **Code Coverage** in versions of **Service Profiler** prior to v1.4.

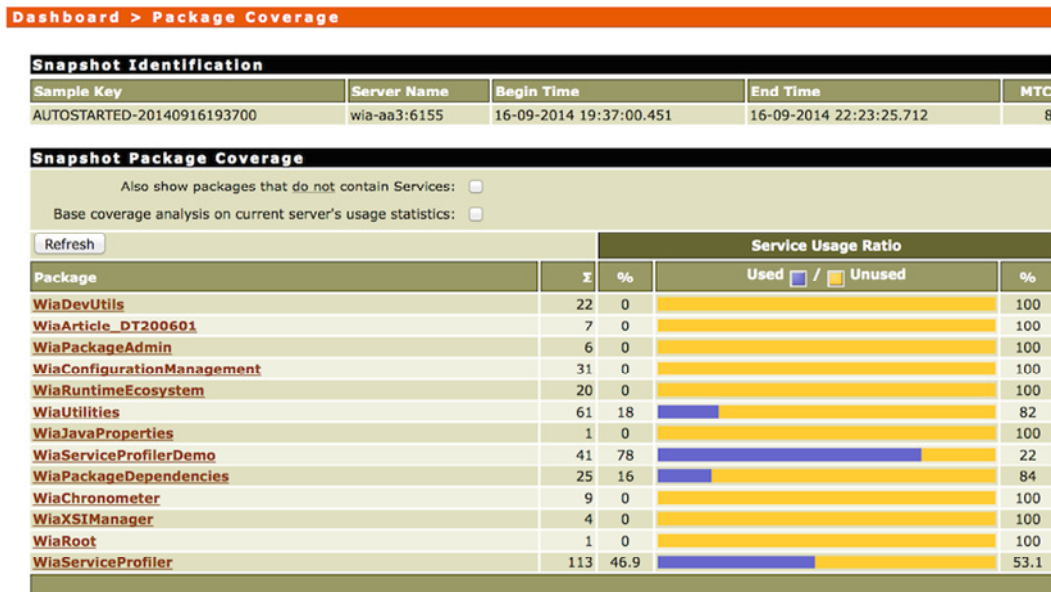


Figure 11 - The Package Coverage report – Snapshot View

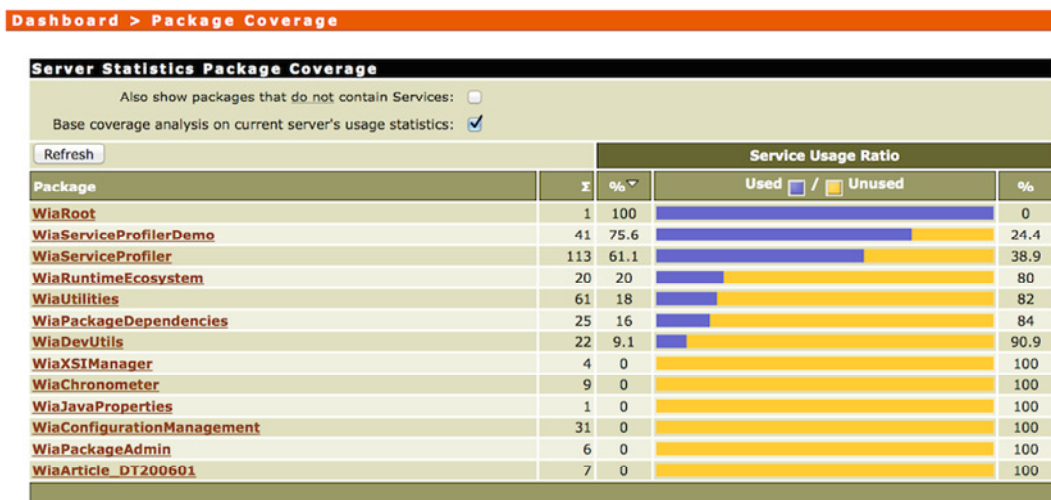


Figure 12 - The Package Coverage report – Service Usage Statistics View

The table can be sorted by clicking on the corresponding column header on any of the detail columns, with exception to the percentage bar. Clicking multiple times toggles between ascending and descending sort order.

Each package name (that contains Services defined) is presented as a link to a new page where the complete list of Services from that package is presented, together with an indication on whether it has been used/invoked or not while the profiler was *Active* (see *Figure 13*, below).

Additionally, there is a filtering option that changes the amount of data when the page is refreshed, by means of clicking on the **[Refresh]** button. The option is:

- Also show packages that do not contain Services.

When checked, the entire list of packages, even if they do not have any Services defined, is shown.

This is just a simple option to allow viewing the entire universe of Package names. The Packages that do not contain Services are not available as links for drilling-down.

**Dashboard > Package Coverage > Package Services**

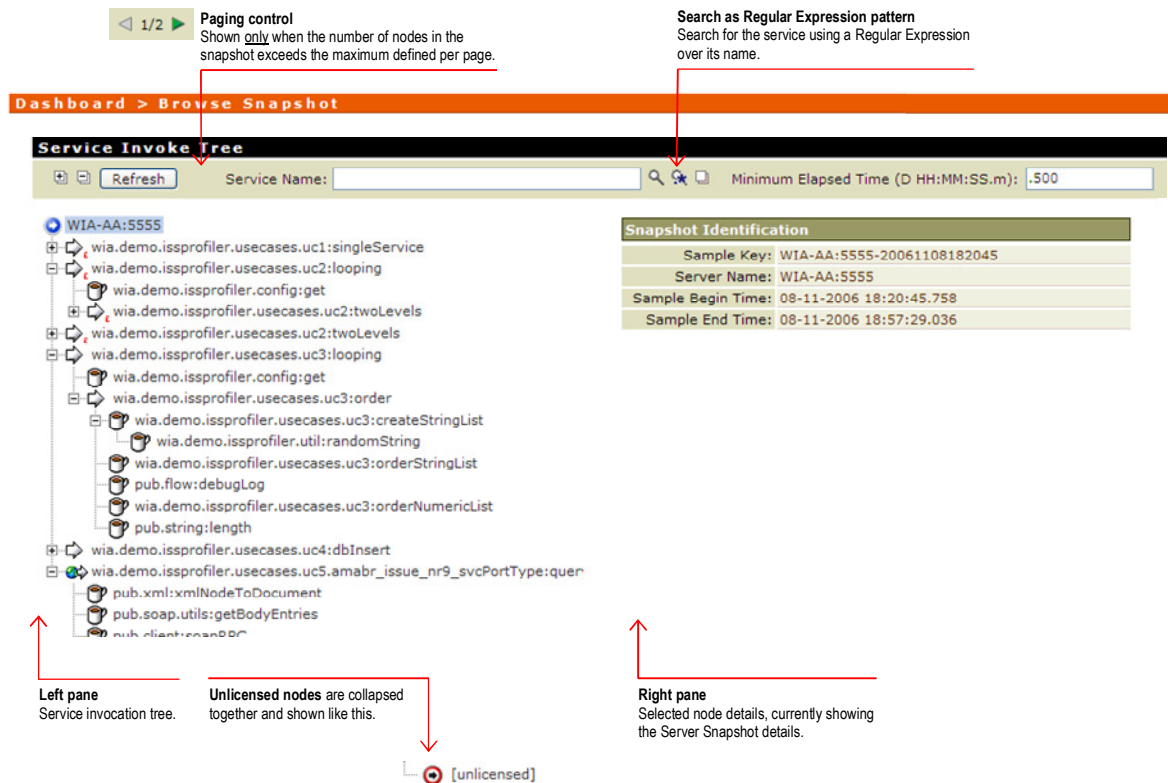
• [Back to Package Coverage](#)

Services From Package 'WiaServiceProfilerDemo'	
Service Name ^	Used?
wia.demo.issprofiler.admin:initialize	Yes
wia.demo.issprofiler.admin:startup	Yes
wia.demo.issprofiler.config:configFilename	No
wia.demo.issprofiler.config:get	Yes
wia.demo.issprofiler.config:list	Yes
wia.demo.issprofiler.config:loadConfig	Yes
wia.demo.issprofiler.config:set	Yes
wia.demo.issprofiler.ui.contact:send	No
wia.demo.issprofiler.ui.run_uc:getMenuMetadata	Yes
wia.demo.issprofiler.ui.run_uc:getUseCasesMetadata	Yes
wia.demo.issprofiler.ui.run_uc:triggerUseCase	Yes
wia.demo.issprofiler.usecases.uc1:looping	No
wia.demo.issprofiler.usecases.uc1:singleService	No
wia.demo.issprofiler.usecases.uc2.wsc.uc2:processEntry	No
wia.demo.issprofiler.usecases.uc2.wsc:processEntry	Yes
wia.demo.issprofiler.usecases.uc2:createDocument	No
wia.demo.issprofiler.usecases.uc2:looping	No
wia.demo.issprofiler.usecases.uc2:random	No
wia.demo.issprofiler.usecases.uc2:sleep	No
wia.demo.issprofiler.usecases.uc2:twoLevels	No
wia.demo.issprofiler.usecases.uc2:unzip	No
wia.demo.issprofiler.usecases.uc2:zip	No
wia.demo.issprofiler.usecases.uc3:createStringList	No
wia.demo.issprofiler.usecases.uc3:looping	No
wia.demo.issprofiler.usecases.uc3:order	No
wia.demo.issprofiler.usecases.uc3:orderNumericList	No
wia.demo.issprofiler.usecases.uc3:orderStringList	No
wia.demo.issprofiler.usecases.uc5.amabr_issue_nr9_svcPortType:queryEntity	No
wia.demo.issprofiler.usecases.uc6:recursiveL1	No
wia.demo.issprofiler.usecases.uc6:recursiveL1_1	No
wia.demo.issprofiler.usecases.uc6:recursiveL2	No
wia.demo.issprofiler.usecases.uc6:runCase	No
wia.demo.issprofiler.usecases.uc7.file:getFile	No
wia.demo.issprofiler.usecases.uc7.file:reader	No
wia.demo.issprofiler.usecases.uc7:looping	No
wia.demo.issprofiler.util:doThreadInvoke	Yes
wia.demo.issprofiler.util:getRunningInstances	Yes
wia.demo.issprofiler.util:getThisPackageHomeDir	No
wia.demo.issprofiler.util:randomNumber	No
wia.demo.issprofiler.util:randomString	No
wia.demo.issprofiler.util:sleep	No

Figure 13 - The Package Coverage service usage detail

## 4.5 Browse Snapshot

This **Analysis Tool** simply shows the Snapshot tree of called Services.



**Paging control**  
Shown *only* when the number of nodes in the snapshot exceeds the maximum defined per page.

**Search as Regular Expression pattern**  
Search for the service using a Regular Expression over its name.

**Dashboard > Browse Snapshot**


**Service Invoke Tree**

Refresh Service Name:  Minimum Elapsed Time (D HH:MM:SS.m):

**Snapshot Identification**

Sample Key:	WIA-AA:5555-20061108182045
Server Name:	WIA-AA:5555
Sample Begin Time:	08-11-2006 18:20:45.758
Sample End Time:	08-11-2006 18:57:29.036

**Left pane**  
Service invocation tree.

**Unlicensed nodes** are collapsed together and shown like this:  [unlicensed]

**Right pane**  
Selected node details, currently showing the Server Snapshot details.

Figure 14 - The Browse Snapshot page

Placing the mouse cursor over the node name will show a tip with context summarized information. By clicking on the node name, contextual detail information is shown in the panel on the right:

- The **Integration Server** node stems snapshot identification data (see Figure 14, above);  
The nature of this information is the same as the one presented on the other **Analysis Tools**.
- A service node stems details about that Service in that calling context (see Figure 15, below);  
The **Accumulated With Children** table presents timings that include the service own timing plus the timings of all the services in its child nodes.  
An **Aggregated Count** value may also appear in the contextual detail information if the **Licensed Service Volume** as been surpassed and the node is an aggregation of all service counts below the parent service.

The paging control is only shown if the number of nodes in the snapshot exceeds the maximum defined per page (for more information please read section *Timeout or long-running script alert on browser*, on page 41).

Service			
Package:	WiaServiceProfilerDemo		
Call Count:	52		
Exception Count:	52		

Own Code			
Elapsed:	53.067		
Spent:	0.160		

Accumulated With Children:	Minimum	Maximum
Elapsed:	1:56.255	0.240
Spent:	0.260	0.000

The minimum & maximum values timed on a single service call.






Includes the sum of the Service's Own Code values with the values of all its child nodes.

Figure 15 - The Browse Snapshot example of detail for a selected Service tree node

There are no sorting options: the root nodes are always ordered ascending.



The Package Exclusion Patterns (see *Administration Page, on page 15*) and the Minimum Elapsed Time filtering options are only applied to the root nodes.

Table 9 - Snapshot Browser options

Option	Description
	Expand all tree nodes.
	Collapse all tree nodes.
<b>Service Name</b>	Locates all occurrences of a service name in the call tree.
	Locates all occurrence of a fully qualified service name. The tree is expanded to where the service is found and the tree node(s) is(are) highlighted (but not selected).
	Locates all occurrence of a service in the given the Regular Expression entered in the Service Name text box. The tree is expanded to where the service is found and the tree node(s) is(are) highlighted (but not selected).
	Clear all highlighted tree nodes (does not deselect any currently selected tree node).
<b>Minimum Elapsed Time</b>	Applied only to root services. Only shows root nodes for which the Elapsed Time is above the stated value. The filter is applied by pressing the [Refresh] button.

On the tree nodes, which represent Service Calls, there may appear some visual hints overlaid to the node icon that give it some extra meaning. These tree node visual hints are explained in Table 10 (below).

Table 10 - Snapshot Browser node icons visual hints

Tree Node Hint	Description
	False root. This can only occur on the node that is representing a Top Level Service or Root Service, i.e., a Service at the top of the tree. A False root occurs when the Service Profiler is started, a service running path is already on the run and the first collected data happens in the middle of the call stack: for the Service Profiler this is a root because its has no recollection of the services up in the call stack. However, it is recognizably not a true Top Level Service because its parent is not null.
	This hint indicates that that Service has raised exceptions. Select it to view its details, where an exception count is also presented.

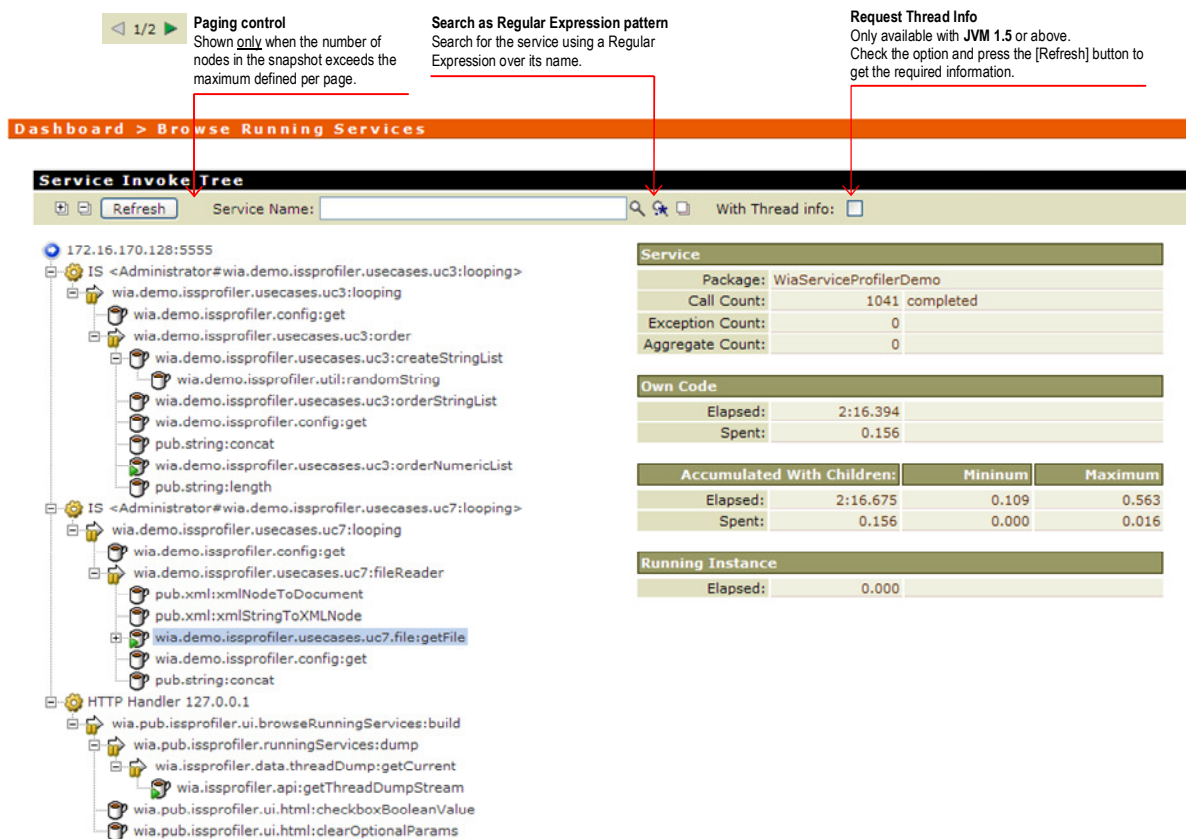


**IMPORTANT:** The **View Per Service** reports accumulated timings per service independently from its position in the calling tree. This may induce timing differences for a specific node by comparison because that service may be called by different parents which may even be currently filtered out.

**IMPORTANT:** Some service nodes that are presented in the root may in fact not be top-level services. They may happen to be on the root node just because they were first scanned after the time when the profiler was started.

## 4.6 Browse Running Services

The **Browse Running Services** dashboard is similar to the **Browse Snapshot** in format, but not in content.



**Service Invoke Tree**

Service Name:

**Service**

Package:	WiaServiceProfilerDemo
Call Count:	1041 completed
Exception Count:	0
Aggregate Count:	0

**Own Code**

Elapsed:	2:16.394
Spent:	0.156

**Accumulated With Children**

	Minimum	Maximum
Elapsed:	0.109	0.563
Spent:	0.000	0.016


**Running Instance**

Elapsed:	0.000
----------	-------

Figure 16 - Browse Running Services dashboard

Unlike all the other dashboards, which show data about services that have already finished, this dashboard shows the tree of services currently running where the 1<sup>st</sup> level node is the thread it is running in. Only when the thread's top level service exits are the entire branch service counts added to the snapshots shown in the other dashboards. That is why this dashboard completes the profiling services view: it allows checking how many **Service Threads** are being used, which service path is being executed in each on them, etc.

The **Browse Running Services** dashboard shares the same service search features available for the **Browse Snapshot**, as described in [Table 9 \(page 25\)](#), and the contextual detail information is also similar, but with one singular difference:

- A group of information identified as **Running Instance**.  
This group of information is only shown if the selected service is the running instance, *i.e.* it is the service that is currently (at the time the snapshot was taken) on the top of the call-stack. It is visually identified by having  over its icon.



The **Elapsed** value indicates the time that has been elapsed since the service was invoked.

Much like for the other dashboards, the counts for the services involved in the current execution path are only updated when they finish. These specific services are identified by visual hints overlaid to the node icon, and are explained in the [Table 11](#), below. This dashboard also presents the visual hints described in [Table 10 \(page 25\)](#).

This said, there are some specific characteristics about this dashboard snapshot that further distance it from the others:

- Service call count of 0 (zero);  
The service call count is only incremented when the call is completed and returns the control back to the caller. This is also true for the timings and other counts.  
So, the displayed call count value, exception count, elapsed time, etc., for each service is only for calls that have already finished.  
When the service is first called, and is still on the thread call-stack, its call count is 0 (zero).
- The **Package Exclusion Patterns** is only applied to the top-level services;  
In fact, this is also the case for the **Browse Snapshot** dashboard.
- This dashboard snapshot is not affected by the **Freeze Snapshot** setting.  
So, every time the page is loaded, or the **[Refresh]** button is clicked, a fresh snapshot of the threads is always taken.

Table 11 - **Browse Running Services** node icons additional visual hints

Tree Node Hint	Description
	This hint indicates that the Service is part of the thread call-stack. It is not the service currently running but it is on the call path and has itself called another service.
	The node with this hint is the Service actually running within the thread. It is the service that is currently (at the time the snapshot was taken) on the top of the call-stack.

If the **JVM** being used is **v1.5** or above, additional thread information can be requested, including **Java Stack Trace** and lock monitor information. When the **With thread info** checkbox is checked, pressing the **[Refresh]** button gets a (new) snapshot with the thread information is added to it (*see Figure 17, below*). Because this option has a slightly higher performance impact on the running services, it is unchecked by default.

**IMPORTANT:** It is important to point out that the viewed information is valid for the moment the snapshot was taken. So, for instance in case of simple blockage, the status may be transient and when viewed it may already have been changed.



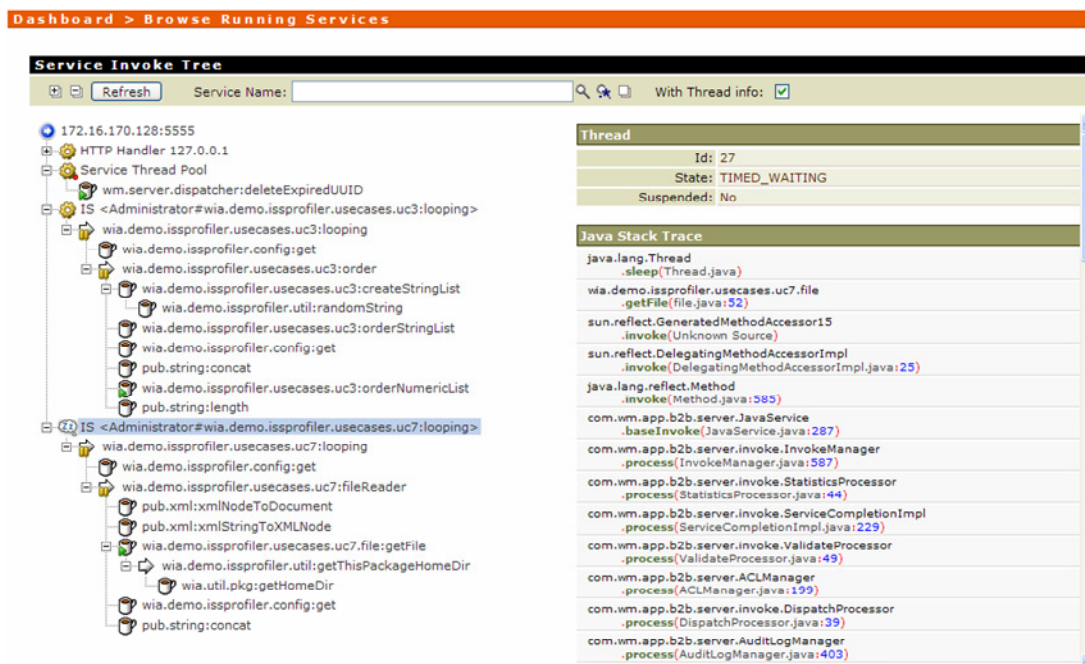








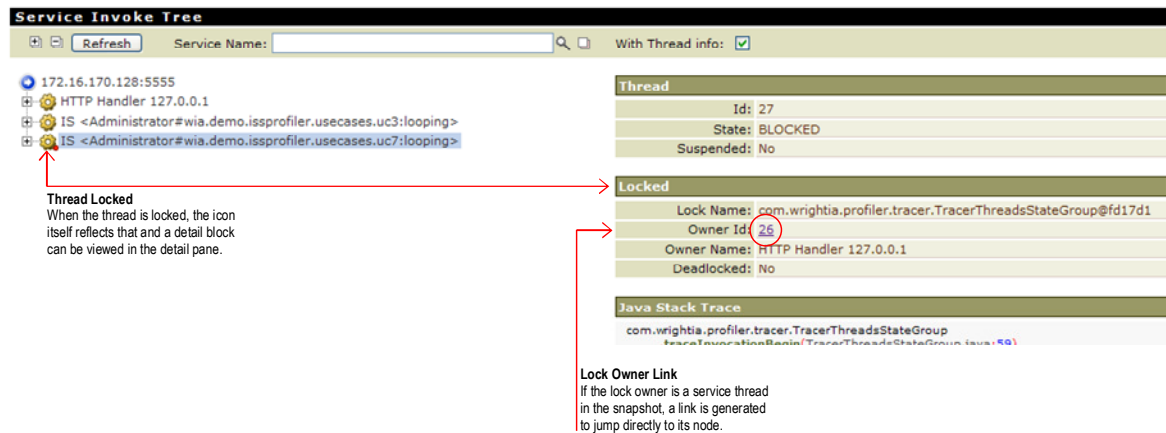
Figure 17 - Browse Running Services with thread info

When the **With thread info** option is used, the thread name link on the tree (left pane) changes from its default behavior (expanding/collapsing the node) to show thread details on the right pane. Additionally, the thread node icons suffer changes to visually reflect some (not all, but what can be considered important) state information.

Table 12 - Browse Running Services thread icons when **With thread info** option

Tree Node Hint	Description
	The thread is running normally ( <b>RUNNABLE</b> ) or no detail information about the threads was requested.
	The thread is blocked ( <b>BLOCKED</b> ) by the thread info gathering process. The lock is only for the duration of the information gathering and it is now already released. The detail panel contains information about the blocker, and if it is a thread in the current snapshot a link is available to directly locate it in the tree.
	The thread is blocked by another ( <b>BLOCKED</b> ). The detail information contains information about who is blocking, and if it is a thread in the current snapshot makes a link available to directly locate it in the tree.
	The thread is not only blocked. It is actually involved in a deadlock.
	The thread is suspended.
	The thread is waiting, either that some time has passed ( <b>TIMED_WAITING</b> ) or that a notification happens ( <b>WAITING</b> ).

When the thread is blocked, a specific block of information is placed the detail pane including the identification of its owner. If the block owner is a **Service Thread** that is part of the current snapshot, the thread Id is transformed into a link that jumps directly to its node in the tree (see [Figure 18](#), below).



**Service Invoke Tree**

Refresh Service Name: With Thread info: ☒

172.16.170.128:5555

- HTTP Handler 127.0.0.1
- IS <Administrator#wia.demo.issprofiler.usecases.uc3:looping>
- IS <Administrator#wia.demo.issprofiler.usecases.uc7:looping>

**Thread Locked**  
When the thread is locked, the icon itself reflects that and a detail block can be viewed in the detail pane.

**Thread**

Id:	27
State:	BLOCKED
Suspended:	No

**Locked**

Lock Names:	com.wrightia.profiler.tracer.TracerThreadsStateGroup@fd17d1
Owner Id:	26
Owner Name:	HTTP Handler 127.0.0.1
Deadlocked:	No

**Java Stack Trace**

```
com.wrightia.profiler.tracer.TracerThreadsStateGroup
    java.lang.Thread.run()
    com.wrightia.profiler.tracer.TracerThreadsStateGroup.run()
    java.lang.Thread.run()
```

**Lock Owner Link**  
If the lock owner is a service thread in the snapshot, a link is generated to jump directly to its node.

Figure 18 - Thread lock detail info

## 4.7 Export To File

This functionality allows exporting the current Snapshot to a file.

When you choose the file format/type, adequate options appear corresponding to that file format/type.

When the **[Execute]** button is clicked, the file generation process is started. Where the file is generated depends on kind of generation selected:

- **Stream to download;**

The generated file is streamed directly to the browser.

A dialog box is open to define where the file should be locally created. A default file name is automatically suggested, but (depending on the used browser) you can give it a name of your choice.

- **Generate file & link.**

The file is generated in the server file system and a link is generated to access it.

The link is intended to be used with the *Save as...* option of the browser, to copy it to your local disk.

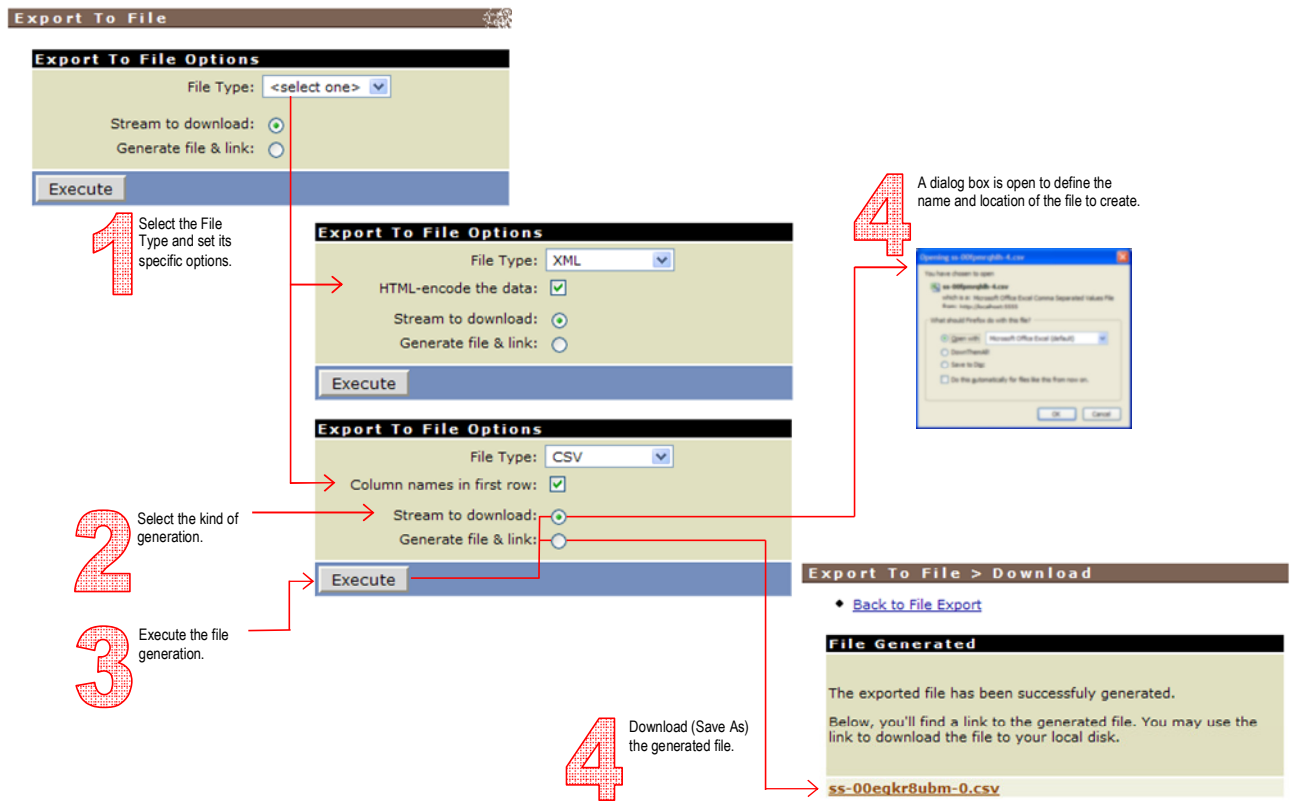


Figure 19 - Exporting Snapshot as a file

#### 4.7.1 CSV Format

The standard **CSV (Comma Separated Values)** format is widely know but still leaves some minor details for the implementation. Therefore, the **Service Profiler** implementation fixes these rules:

- The value separator is always the comma (",");
- The **Service Name** includes the package name as a prefix separated by a forward-slash ("/");

For example: `WiaServiceProfiler/wia.pub.issprofiler.analyze:viewPerService`

- Only values that include the value separator are quoted;  
When this happens the value is enclosed in double-quotes.
- Header names are not quoted;  
Header names are optional, and are not quoted in accordance with the rule to quote values.
- The record separator used is dependent of the **Operating System** the **Integration Server** is running under;  
The record separator is the new-line sequence used for normal text files, following the rules for the hosting **Operating System**.

- Column header names are:

Node Id  
Parent Node Id  
Sample Key  
Server Name  
Sample Begin Time  
Sample End Time  
Max Thread Count  
Service Name  
Calling Service Name<sup>3</sup>  
Call Count  
Exception Count  
Elapsed milliseconds  
CPU nanoseconds  
Max Elapsed milliseconds  
Min Elapsed milliseconds  
Max CPU nanoseconds  
Min CPU nanoseconds

#### 4.7.2 XML Format

---

The produced **XML** is syntactically formatted according to standard rules, which includes starting the file with the line:

```
<?xml version="1.0" encoding="UTF-8"?>
```

The one specificity that the produced **XML** contains is the used dictionary. Because the **API** to the core functions already uses **XML** to provide the Snapshot, the same structure is used here, both for simplicity and enhanced performance.

As an option, the data values can be **HTML**-encoded entities, *i.e.*, where values contain characters that are used by the tagging rules or may create incompatibilities these characters are encoded in the form `&<code>;`. Depending on the application that will be the target of the produced **XML**, decoding these entities may be automatic or not.

---

<sup>3</sup> The **Calling Service Name** is only filled for top-level-services. It is used to detect **False Root** (See the *Glossary* for details).

Table 13 - Snapshot XML dictionary

Tag	Level	Attribute	Comment
service-counters	0		Main enclosing tag.
		Version	Version of the snapshot XML structure.
		sample-key	Snapshot ID.
		server-name	Integration Server net-name (or IP address) and port number.
		sample-max-thread-count	The top count of threads that were used simultaneously.
		sample-begin-time	Timestamp of when the profiler started collecting data, formatted as "dd-MM-yyyy kk:mm:ss.SSS".
		sample-end-time	Timestamp of when the Snapshot was taken, formatted as "dd-MM-yyyy kk:mm:ss.SSS".
invocation	1..n		
		service	Name of the Service. It contains the name of the Package has a prefix separated by a forward-slash, for example: MyPackage/MyAPP.util:getMaxValue
		calling-service	Name of the Service calling this one. If filled, it is only for top-level-services. If filled, the service is a False Root.
		service-type-id	The type of service as referenced in the metadata/service-type element with the corresponding type-id attribute.
		call-count	Number of times this Service was called... in the context of this call path node.
		exception-count	Number of time this Service raised an exception.
		elapsed-ms	Sum of the elapsed milliseconds for the calls in the context of this call path node.
		cpu-ns	Sum of the CPU spent nanoseconds for the calls in the context of this call path node.
		min-elapsed-ms	Reference value. The minimum timed value on a single service call.
		min-cpu-ns	
		max-elapsed-ms	Reference value. The maximum timed value on a single service call.
		max-cpu-ns	
child-invocations	n+1		If defined, contains the invocations to Services from the current Service.
invocation	...	...	Recursive...
metadata			If defined, contains metadata about the services in this snapshot.
service-type	1-n		Reference data for the service types. The list of service types is dynamically generated for this specific snapshot, only with the service type that occurs in it. Its included IDs <u>must</u> not be considered valid of other snapshots.
		name	The name that describe the service type.
		type-id	The ID of the service type. The service nodes refer to this value to define their service type.

## 4.8 Load Snapshot from file

It is possible to load a previously exported snapshot back into the tool to view it in the dashboards. However, only snapshots exported as **XML** are accepted. This is validated in the file upload panel (see [Figure 20](#), below) and only valid files actually uploaded.

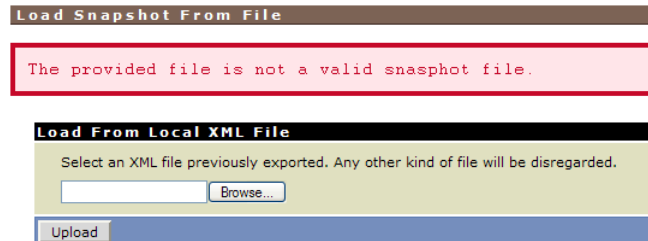




Figure 20 - Load snapshot file (invalid)

When the file is successfully uploaded (see [Figure 21](#), below), a corresponding message is shown together with the  icon which is also the visual clue indicating that the snapshot is from a loaded **XML** and not from the running services. This visual clue is shown in the dashboards and panels that show the loaded snapshot. Not all dashboards show/reflect the loaded snapshot, e.g., the [Browse Running Services](#).

The snapshots can be from **Integration Servers** other than the current one. For snapshots exported with older versions of the **Service Profiler** the service icons may appear incorrect or as an  icon. This is because metadata about the exported services is not included as part of the snapshot and the tool tries to collect it from the current host where it may not exist, or even worse, not be true.

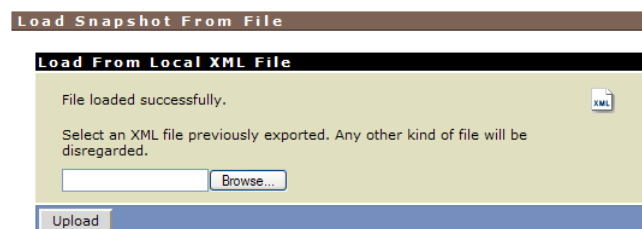



Figure 21 - Load snapshot file (valid)

As consequence of the file load, the snapshot is frozen (). If at the time of the file load the **Service Profiler** is started and collecting service information, it will continue doing so in the background.

To discard the loaded **XML** snapshot and access the running services snapshots, just go to the admin page and unfreeze the snapshot (see [Administration Page](#), on page15).

## 5 EXTENSIBILITY

**IMPORTANT:** The information presented in this section may change in the future to accommodate the evolution of the tool.

These changes will eventually be additions to the presented structures and extensibility mechanisms which, to the extent of the possible and reasonable, will be compatible with previous versions of the same mechanism. However, this compatibility compromise will be superseded when it impairs the progression of the tool evolution/betterment.

### 5.1 Overview

The **Service Profiler** core functionality is to gather raw information about the running **Services**. From that information, **Analysis Tools** may be implemented. The tool already includes some tools of that nature. However, to avoid limiting the analysis of the data to the tool's included functionality, external accessibility to the snapshot data is provided.

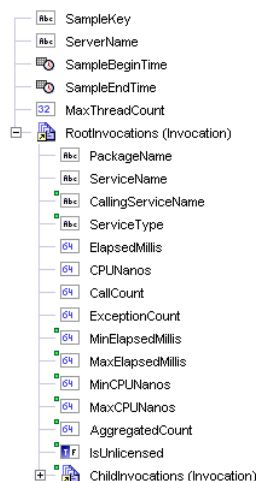


Figure 22 - The Snapshot as a Document structure

A **Snapshot** is a tree structure (see *Figure 22, above*) with common information on the root note plus the entire top-level **Services**, each of them being a node and with (possible) child nodes in them representing the **Services** they have called. These child codes can also have their own child nodes in them representing the **Services** they have called... and so on.

The **Snapshot** can currently be exported as an **XML** file or a **CSV** (see *Export To File* on page 29).

Beyond the export as file feature, and closer to the **Integration Server** environment, the **Service Profiler** provides a public interface (API) of **Services** and **Document Types** that can be used by any external **IS Service** to request **Snapshots** and process them.

The term public determines that the interface exposes controlled functionality and that the retro-compatibility of that interface will be taken into consideration when upgrades are issued, but is not guaranteed. The latter may be justified by the integration of new and incompatible functionality, but whenever it may occur, it will be clearly described in the **Release Notes**.

## 5.2 Public API

The **Service Profiler** specific functionality is enclosed in the `WiaServiceProfiler` package.

The `WiaRoot` and `WiaUtilities` are also distributed as part of the installation, but these are support packages that are shared by other **Wrightia** products. The first contains no usable components but the latter contains some generic use services, which are not object of this documentation. However, the included services are fairly self-commented, through the use of Service and Field comment entries.

The `WiaServiceProfiler` package public interface is rooted on the `wia.pub.issprofiler` folder. To execute the enclosed services, the user must be part of the **WiaServiceProfiler's ACL**. These components are fairly self-commented, through the use of Service and Field comment entries.

The `wia.pub.issprofiler.ui` folder contains the functionality that serves exclusively the product **DSPs**. These are made public for the sole reason that they provide the exact same information that is viewed on the corresponding pages and are therefore not object of the current documentation.

### 5.2.1 Documents

The several existing views of a **Snapshot** are defined as **Documents**, which are used as input and/or output parameters of the **Services** on the public **API**.

On **Table 14** (below) are listed the public interface **Documents**.

Table 14 - Documents defined as public

wia.pub.issprofiler.docs:		
Interface	Document Name	Description
Take Snapshot	ServiceCounters	This document represents a complete, raw, <b>Snapshot</b> as provided by the profiler core. This document is a translation to <code>IData</code> from the native XML dictionary... not a direct conversion of tags and attributes. This document has become a <b>Publishable Document</b> on v1.2.3 <sup>4</sup> of the <b>Service Profiler</b> .
	Invocation	This is the representation of a single <b>Service Invocation</b> node of the <b>ServiceCounters</b> document.
View Per Service	ViewPerService	The snapshot represented as a <b>View Per Service</b> result.
	InvocationAccumulator	<b>ViewPerService</b> child document representing an accumulator for a particular <b>Service</b> .
Package Coverage	CodeCoverage	The snapshot represented as a <b>Package Coverage</b> result.
	PackageCoverage	Child document of the <b>CodeCoverage</b> , representing the coverage statistics for a particular <b>Package</b> .
Browse Snapshot	BrowseSnapshot	The snapshot represented as a <b>Browse Snapshot</b> result.
	BrowseNodeAccumulator	<b>BrowseSnapshot</b> child document, representing a <b>Service</b> node in the invocation path where its invocation statistics are kept.
Browse Running Services	BrowseRunningServices	The snapshot of the services still running. Each service tree runs in its own <b>Service Thread</b> and depending on the JVM version and request options it may also contain thread info.
	ThreadInfo	State information of a specific running <b>Service Thread</b> and its <b>Java Stack Trace</b> .
	ThreadDump	Dump the state of all running <b>Threads</b> .
Export To File	ExportedFile	The link representation of the result of an <b>Export To File</b> .

<sup>4</sup> Even though this is the actual version when this feature became available, it was only officially released with the v1.3 of the tool.



For details on the structure of each of these documents, please use **Developer** (or any other capable tool) to view the **Document's** field comments.

## 5.2.2 Services

This section describes the public **API Services**.

**IMPORTANT:** The public **API** *per se* does not include **Services** for administrative purposes. However, that functionality can be accessed through the **UI Services** for the administration pages.

Table 15 - Admin request Services

wia.pub.issprofiler.admin:	
Service Name	Comments
startProfiler	Start profiling. It is equivalent to the <b>Start</b> operation in the <b>Service Profiler Admin</b> page
stopProfiler	Stop profiling. It is equivalent to the <b>Stop</b> operation in the <b>Service Profiler Admin</b> page

Table 16 - Snapshot request Services

wia.pub.issprofiler.snapshot:	
Service Name	Comments
take	Take a snapshot of the current (possibly frozen) counters.
freeze	Takes a snapshot and freezes it in memory, caching for the next requests. Doesn't need to check if already frozen...
unfreeze	Unlock freeze settings, allowing each new request to get the most current snapshot. Doesn't need to check if already unfrozen: if not frozen, it does nothing.
publishCurrent	Take a snapshot of the current (possibly frozen) counters, and publishes it as a <b>Publishable Document</b> (wia.pub.issprofiler.docs:ServiceCounters).

Table 17 - Out-of-the-box Snapshot Analysis Services

wia.pub.issprofiler.analyze:	
Service Name	Comments
viewPerService	Analyzes the statistical <b>Service Counter</b> tree structure and produces a flat (array) of accumulators: one element per service. Optionally, the accumulator list can be filtered to only include those that are equal or above a given <b>Minimum Elapsed Time</b> value, and/or timings be generated as averages related to the <b>Call Count</b> .
packageCoverage	Analyzes the statistical <b>Service Counter</b> tree structure and produces a <b>Service Coverage</b> assessment for the snapshot. The analysis result is produced only for a given list of package names. Optionally, the resulting list may contain Packages that do not hold Services.
browseSnapshot	Analyzes the statistical <b>Service Counter</b> tree structure and produces accumulators per tree node. Optionally, the accumulator list can be filtered to only include those that are equal or above a given <b>Minimum Elapsed Time</b> value.
browseRunningServices	Analyzes the statistical <b>Service Counter</b> tree structure and produces accumulators per tree node but restricted to the top level services that did not yet return from its initial invocation. The analysis is similar to a finished statistical <b>Service Counter</b> tree structure, but rooted at the <b>Service Thread</b> in is running and with information on the currently running path. Depending on the version of the running JVM additional <b>Thread State</b> info and <b>Java Stack Trace</b> info may also be included.

Table 18 - Export Snapshot to file Services

wia.pub.issprofiler.export:	
Service Name	Comments
to_csv	<p>Takes the current Snapshot and exports it to a Comma Separated Values (CSV) formatted file.</p> <p>The Snapshot is automatically taken by this service as part of the service workflow, and it is thus affected by global filtering settings and the snapshot freeze setting.</p> <p>See <a href="#">CSV Format</a>, in page 30, for a reference of the CSV data structure.</p> <p>Even though not available in the UI, since v1.2.3, this service also accepts an optional argument that gives indication to reset service counters when getting the current snapshot.</p>
as_csv_download	<p>Takes the current Snapshot, exports it to a Comma Separated Values (CSV) formatted file and downloads it as an attachment of the request. This service is <u>not</u> used by the tool's infrastructure and is intended to facilitate the remote access of automated scripts and/or browser based users. Its interface is designed to be invoked through an HTTP request, to which a <code>text/csv</code> MIME type response is given.</p> <p>The Snapshot is automatically taken by this service as part of the service workflow, and it is thus affected by global filtering settings and the snapshot freeze setting.</p> <p>Optionally, an indication to reset snapshot counters is also possible.</p> <p>See <a href="#">CSV Format</a>, in page 30, for a reference of the CSV data structure.</p>
to_xml	<p>Takes the current Snapshot to a XML formatted file.</p> <p>The XML dictionary is the one used internally by the tool to provide the current Snapshot.</p> <p>The Snapshot is automatically taken by this service as part of the service workflow, and it is thus affected by global filtering settings and the snapshot freeze setting.</p> <p>See <a href="#">XML Format</a>, in page 31, for a reference of the XML data structure.</p> <p>Even though not available in the UI, since v1.2.3, this service also accepts an optional argument that gives indication to reset service counters when getting the current snapshot.</p>
as_xml_download	<p>Takes the current Snapshot to a XML formatted file and downloads it as an attachment of the request. This service is <u>not</u> used by the tool's infrastructure and is intended to facilitate the remote access of automated scripts and/or browser based users. Its interface is designed to be invoked through an HTTP request, to which a <code>text/xml</code> MIME type response is given.</p> <p>The XML dictionary is the one used internally by the tool to provide the current Snapshot.</p> <p>The Snapshot is automatically taken by this service as part of the service workflow, and it is thus affected by global filtering settings and the snapshot freeze setting.</p> <p>Optionally, an indication to reset snapshot counters is also possible.</p> <p>See <a href="#">CSV Format</a>, in page 30, for a reference of the CSV data structure.</p>

Table 19 - Utility Services

wia.pub.issprofiler.util:	
Service Name	Comments
getPackageNames	<p>Get the current list of package names, based on the complete list of Packages and the current <a href="#">Package Exclusion Patterns</a> settings.</p>

For details on the input and output parameters for the listed Services, please use [Developer](#) to view their signature definitions and the comments filled for the fields. It's bound to stand true that those comments are more complete and up-to-date than any detached documentation.

## 5.3 Satellite Tools

Starting with version 1.3.4 of the [Service Profiler](#), which also corresponds to patch #4 for v1.3, the recognition of [Satellite Tools](#) has been added to the profiler.

[Satellite Tools](#) are tools that depend of a licensed instance of the [Service Profiler](#) up and running at the target [Integration Server](#). The licensing of these tools fall under the umbrella of the [Service Profiler](#), *i.e.*, if you have a licensed profiler you can use these tools free of charge.

[Satellite Tools](#) once installed are automatically recognized by the [Service Profiler](#) main menu and listed there for quick access (*see [Figure 23](#), below*).

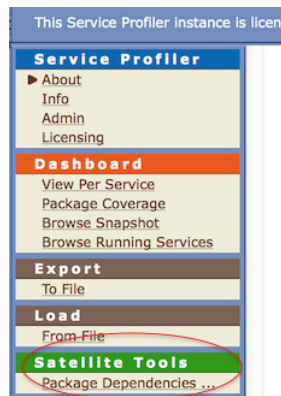


Figure 23 - Satellite Tools on the main menu

**Satellite Tools** are helper tools that, one way or another, are related or complement the work with the **Service Profiler**. These are goodies provided to customers that license the **Service Profiler**. Nevertheless, they can be installed and evaluated with an evaluation **License Key**.

The only readily available **Satellite Tool** is the **Package Dependencies Analyzer**. This tool analyzes the dependencies between packages:

- Declared and undeclared;
- Design-Time and Run-Time (with the help of an *impromptu* **Service Profiler** snapshot);
- Identifies references to undefined nodes, circular references, etc.

The **Package Dependencies Analyzer** is intended to answer the following questions:

- If I want to install this package, what other packages do I need to also install?
- What other packages are referencing this package?
- Which packages are not referenced by any other package (*aka*, top-level packages)?
- The **Package Dependencies Analyzer** is currently only available as direct download links. Please contact us to get the link to the most recent version.

The **Service Profiler Demo** package v1.1 also registers as **Satellite Tool**, but this just for convenience of access. There is no license limitation and it does not really add any useful functionality as a tool.

## 6 TROUBLESHOOTING

Refer to this section prior to asking for support.

This section describes known solutions to known situations caused by common pitfalls.

A listed effect may be triggered by a non-listed cause and thus not having a description of a known sequence of actions to overcome it. When this happens, please report it to support.

### The Licensing menu option does not appear

---

This is not the case where the menu option exists but is disabled... in this case the menu option is not listed at all.

On some installations, this may happen on the first time the **Service Profiler** pages are accessed: refreshing the page usually makes the link appear.

However, only users with **Administrators ACL** can see this link: make sure the user you are using obeys to this definition.

### (Access Denied) Cannot login onto the Service Profiler pages

---

```
Access Denied.

Services necessary to show this page are currently unavailable on this port. This is most
likely due to port security restrictions.

If this is the only port available to access the B2B Server, contact webMethods Support.
```

The reason for this kind of problem is manifold. It can be a problem with the browser itself not being able to let go of the session status. Close all browser windows, reload the **WiaServiceProfiler** package and try again.

Occasionally, it may happen after the installation of an upgrade of the tool. Due to the incomplete or mixed settings of the **ACLs** caused by things being removed while others are added, the loaded settings do not match the runtime environment. Delete the **WiaServiceProfiler ACL** and reload the **WiaServiceProfiler** package. The security settings are reset to the default (*refer to **Security and Controlled Access**, on page 14, for details about which are these default settings*).

### Functionality menu options appear disabled

---

Basically, the menu options are disabled if the core libraries are not loaded.

This can happen in the following known situations:

- The installation is not complete;  
When the **Service Profiler** package is installed just via the **IS Admin** pages, the menu options are found disabled because the installation is not complete (*see **Installation and Setup**, on page 11 for further details*).
- A file named **ISSPROF\_VOID** exists in the `<wm_is>/IntegrationServer` folder;  
Delete the file.  
It's not automatically created. It's a manual mechanism to prevent the **Service Profiler** from loading without the need to uninstall.

- **LIB**rary paths and **CLASSPATH**s are not correctly mounted or do not include the **Service Profiler** directories;

The directories have been removed or the `server.<sh|bat>` script has been replaced or edited.

Check if there was any **webMethods** upgrade or (re)installation performed recently. These are known to sometimes replace the server's launch script.

- The **Operating System** platform is not supported;

The **Service Profiler** libraries depend on native shared libraries which are available only for a limited list of operating systems & **CPU** architectures. If the expected shared library is not found the libraries will not load correctly and the menus are kept disabled.

Depending on the **Service Profiler** version, native code loading errors may not appear on logs and the list of supported platforms may differ.

(See *Environment Requirements & Compatibility*, on page 8, for further details)

- File permissions;

This may happen in **UNIX** environments. However, it will rarely occur because **Service Profiler** shell scripts set the needed execution permissions.

Nevertheless, depending on how the files have been copied, who is the owner and what user is being used to start the **Integration Server**... some glitch may happen and the files do not get the needed execute permissions.

- Errors during launch.

Check the console, server logs and the

`<wm_is>/IntegrationServer/serviceprofiler/logs/issprof.log` file for errors during the launch.

To specifically know if an error occurred while loading the native libraries, you can use the **Developer** tool and run<sup>5</sup> the service `wia.issprofiler.admin:getStatus`. If an exception was raised, a variable named `stackTrace` will be placed on the pipeline containing the stack trace for it. Please pick its contents and report to support.

---

<sup>5</sup> A service can also be run directly from the **Integration Server Administration Pages**, from the Package Management menu option by following the link on the package name, browsing the package services and select a service. There, you will find an option to test/run the desired service.

## Timeout or long-running script alert on browser

With very large snapshots (1000, 2000 or more service nodes), the browser may stop the page loading with an alert/warning that the running JavaScript is taking too long. The message and the conditions for raising this warning vary from browser-to-browser and from environment-to-environment.

On the environment side of the issue, and again depending on the browser, little available CPU and RAM may increase the chance of raising the alarm.

The alert message is a security protection that works differently depending on the browser: for instance, while IE evaluates the number of executed script statements; Firefox evaluates the number of elapsed seconds. If the number of service nodes in the snapshot is border-line with the limits the browser evaluates, tweaking its setting may successfully eliminate the alert. But if the number of service nodes is too high (above 3000) it will have to be a job for the Service Profiler tool itself to overcome the limitation (*please read further*).

Assuming that these are the most used browsers, some tests with around 3000 service nodes have been made with them, and so providing some performance and tweaking info:

Table 20 - Script timeout browser related info

Browser	Result
Internet Explorer 6	<p>The slowest in the test, both on page loading and script execution.</p> <p>With 3000 nodes, the problem happens on every page load. Open All and Close All are very slow but do not raise the alarm. IE has a way of increasing the threshold of when the alarm is raised (<a href="http://support.microsoft.com/kb/175500">http://support.microsoft.com/kb/175500</a>). Only with a setting of 50,000,000 was possible to eliminate the alert. Browser restart is needed.</p> <p>No tests have been done with IE7.</p>
Firefox 2.0	<p>Highest size in memory.</p> <p>With 3000 nodes, the problem happens on every page load and while opening all nodes (but only on the first time). Close All is faster than Open All, but the latter is only slow on the first time it is called.</p> <p>Firefox has a setting to change the script timeout, in seconds: open the page <code>about:config</code> and change the <code>dom.max_script_run_time</code> value. The alert was suppressed by setting its value to 20. No browser restart is needed.</p>
Opera 9	<p>The fastest in the test, both on page loading and script execution.</p> <p>The smallest size in memory.</p> <p>With 3000 nodes, no problem ever arises.</p> <p>With 4000 nodes, no problem ever arises, apart from taking a little longer to load. However, in spite of the increased load, the Open All and Close All operations are quite fast.</p> <p>Opera also has a configuration setting page (<code>opera:config</code>), but no setting related to a script timeout was identified.</p>

However, if the number of service nodes on the snapshot is above 1000, the Browse Snapshot and Browse Running Services dashboards automatically set a paging control for the tree of nodes. The setting of 1000 nodes is the default, but can be set to a different value by entering the `issprof.browsesnapshot.nodesperpage` key on the `issprof.cnf` file:

```
...
issprof.browsesnapshot.nodesperpage = 2000
...
```

The setting change is done by editing the properties file with a text editor and entering a new line as demonstrated above. The properties file is located in the `config` folder of the tool's package.

The first time the value is entered in the properties file, no further action is needed. However, if the setting is afterwards changed, a package reload is required.

A **limitation** exists, however:

- If the **Snapshot** tree in the dashboard is paginated, the scope of the **find** for a node in the tree is limited to the current page.

On the special case of the navigation link from the **View Per Service** to the **Browse Snapshot**, the navigation & find is always (and only) made onto the first page. However, jumping to the next/previous page and repeat the find is always possible, but hits on the entire snapshot are not seen all at once.

### **Profiler not correctly initialized: E01**

---

This error will only happen if the **Java** instances of the tool classes are remove from memory in runtime... for whatever reason... and should never happen.

To correct the situation you can either reload the tool package of just run the `wia.issprofiler.admin:startup` service from the **Developer** tool or using a browser to do an **HTTP** service invoke.

## 7 SUPPORT

### 7.1 Disclaimer

---

Product support is exclusively provided by Wrightia.

webMethods Support Services must not be contacted with issues regarding Service Profiler or somehow related with the tool. Furthermore, when reporting a webMethods Integration Server issue with hard evidence (*e.g.*; when providing a JVM thread dump), it may be required to completely remove the Service Profiler from the equation for full webMethods Support Policy compliance. For this purpose, the tool may be prevented from loading without needing to uninstall it (*see Intentionally preventing the Profiler from loading, on page 12*).

### 7.2 Reporting a Service Profiler issue

---

When reporting Service Profiler issues please use the [product.support@wrightia.com](mailto:product.support@wrightia.com) e-mail address and provide information that identifies:

- You as a Wrightia customer;
- Your Service Profiler version and patch level;  
This can be obtained by clicking the WiaServiceProfiler link on the Packages > Management IS administration page, and print it.
- Your Integration Server version and patch level;  
This can be obtained by printing the Integration Server About Page.
- Your Operating System name, version and CPU architecture;  
This can be obtained by printing the Integration Server About Page.
- Additional e-mail and phone contacts that can be reached to clarify the issue.

And identification of the problem/issue/situation itself:

- A short description of the problem;
- An extended description of the problem and, if possible, a way of having it reproduced;
- Environment;  
Define what kind of setting is the target platform: Development, Test, System Test, Staging, Production, etc.
- Issue classification.  
It is a classification of the issue in terms of whether it is an error/bug, an enhancement request or just a question/doubt.

Issues have to be reported in English language.



## 8 FURTHER READING

You may find reading the [Profiling Whitepaper](#) useful. This document is provided separately and the latest version can be downloaded from [Wrightia](http://www.wrightia.com) website (<http://www.wrightia.com>). Rather than providing additional information about the tool, the whitepaper focuses on profiling in general, the usefulness of a [Profiler](#) and the reasoning for using a tool such as the [Service Profiler](#).

## INDEX

- ACLs, 13, 14, 35, 38
  - Administrators, 14, 17, 38
- Analysis Tools
  - Browse Running Services, 36
  - Browse Running Services, 40
  - Browse Snapshot, 24, 35, 36
  - Browse Snapshot, 40
  - Package Coverage, 21, 35, 36
  - View Per Service, 18, 26, 35, 36
- Calling Service, 31, 32
- Configuration Properties
  - issprof.cnf
  - issprof.browsesnapshot.nodesperpage, 40
- Error
  - WiaServiceProfiler loaded as partial, 11
- Export To File, 29, 35, 37
- False Root, 31, 32
- file
  - ISSPROF\_VOID, 12, 38
  - server.bat, 14, 39
  - server.sh, 14, 39
- Filters
  - Package Exclusion Patterns, 16, 25
- Format
  - Timing fields, 19
- Licensing
  - License Key
    - set, 17
- Scripts
  - server.bat, 14, 39
  - server.sh, 14, 39
- Settings
  - Freeze Snapshot, 16, 36
- Windows NT Service
  - uninstall, 14

## APPENDIX A GLOSSARY

Item	Definition
<b>Aggregated Count</b>	<p>Number of services that being aggregated at the referenced node.</p> <p>It means that the referenced node counts for all found services that fall out of the <i>Licensed Service Volume</i> (see <i>Licensing Pages</i>, on page 12).</p> <p>This value only appears when the <i>Licensed Service Volume</i> is exceeded.</p>
<b>Analysis Tool</b>	A dashboard view of the <i>Snapshot</i> data organized in a way that reveals relations on the collected data, either interactively or in the form of a report.
<b>False Root</b>	<p>A <i>False Root Service</i> is one that is presented at the top of the call tree but because that's when the <i>Service Profiler</i> has first collected it, but in fact, it is being called by another Service... not collected by the <i>Service Profiler</i>.</p> <p>See also <i>Top Level Service</i>.</p>
<b>Faux Root</b>	See <i>False Root</i> .
<b>Own Code</b>	In opposition to <i>Whit Child Code</i> , this set of timings relates <u>solely</u> to the code that, in itself, is the body of the subject Service. This excludes times measured while in any Services the current one calls as part of its designated functionality.
<b>Snapshot</b>	A <i>Snapshot</i> is an instant sample of the data structures internal to the <i>Service Profiler</i> .
<b>Time Elapsed</b>	This is the absolute time that has elapsed as measured by an external chronograph, or even on your wrist watch.
<b>Time Spent</b>	<p>In opposition to <i>Time Elapsed</i>, this value measures only the time the Service was in the CPU doing real work.</p> <p>The CPU is being used by more than one thread. The scheduler slices the availability of the CPU by giving each thread an opportunity to run. This value counts the time the Service spend effectively using the CPU.</p> <p>A Service may have a Time Elapsed much larger that the Time Spent if it is given little opportunity to run. This can be caused by:</p> <ul style="list-style-type: none"> <li>• A thread with much higher priority hogging the CPU availability;</li> <li>• A badly dimensioned system with not enough threads allocated, resulting in an activity of context switching;</li> <li>• Not enough resources (RAM, network bandwidth, etc.) causing the Service to spend most of the time waiting;</li> <li>• Etc.</li> </ul>
<b>Top Level Service</b>	<p>A Service that is not called by another service, thus being at the top of the call tree.</p> <p>This kind of service may also appear referenced as <i>Root Service</i>.</p>
<b>Root Service</b>	See <i>Top Level Service</i> .
<b>With Child Code</b>	This set of timings corresponds to the current <i>Service Own Code</i> <u>plus</u> any times measured while in any Services the current one calls as part of its designated functionality.