



StatInspector - Domino Server Statistic Inspector Version 0.8.0 (Freeware)



Copyright (c) 2012 ABdata, Andy Brunner
All Rights Reserved

ABdata, Andy Brunner
Kappelstrasse 43
CH-8002 Zürich
E-Mail Andy.Brunner@ABdata.CH
Member of NotesNet.CH

Table of Content

1. Introduction.....	3
2. Background.....	3
3. License.....	3
4. Domino 8.5 Addin Prerequisite	3
5. Installation / Upgrade	3
6. Configuration Documents.....	4
7. Probe Documents.....	4
8. Alert Documents.....	6
9. Startup / Termination.....	6
9.1 Manual Startup.....	6
9.2 Domino Notes.Ini	7
9.3 Program Document.....	7
9.4 Terminate the Addin.....	7
10. Some Technical Background	7
11. Feedback.....	7
12. Frequently Asked Questions.....	8
13. Links	8
14. Release History.....	8

1. Introduction

The Domino Server Statistic Inspector (StatInspector) is a small Domino server Java addin, which monitors the server statistics based on user definable probe documents and creates alert documents when needed.

2. Background

During its execution, the Domino server maintains a large number of statistic values, which can be displayed either with the `Show Statistic` console command or thru the Domino Administrator client.

The standard Domino Monitoring Configuration database `events4.nsf` allows some triggering based on these server statistics. But it is not possible to specify all available statistics or to define correlations between individual server statistic values.

Together with the configuration database (default `StatInspector.nsf`), this small Domino server addin allows the administrator to monitor all server statistics and define simple or complex conditions when an alert document should be created.

3. License

"This software shall be used for Good, not Evil."

4. Domino 8.5 Addin Prerequisite

Some of the Java classes used in StatInspector require Java 6 (Java version 1.6) or higher. The Java virtual machine with this functionality was first available with Domino 8.5. Therefore, this tool will run on any Domino version 8.5 and higher (any platform, operating system and processor architecture).

This restriction only applies to the Domino server running the StatInspector addin. All remote servers, for which the addin collects the statistics, may run any Domino version.

5. Installation / Upgrade

The following steps are required to install or upgrade StatInspector:

- Copy the three Java class files (`JAddin.class`, `JAddinThread.class` and `StatInspector.class`) into the Domino program directory (Windows) or into the Domino data directory (Linux). Alternatively you can combine all class files in one JAR container and add this JAR file name to the Domino Notes.Ini parameter. Example:
`JavaUserClasses=C:\Lotus\Domino\traveler.jar;C:\Apps\StatInspector.jar`
- Copy the `StatInspector.ntf` database template file into the Domino data directory.
- Create a new database based on the `StatInspector.ntf` template. The recommended default name is `StatInspector.nsf`. If you upgrade from a previous version you just need to do a design refresh of the database.

- Create or verify the configuration and probe documents for your environment.
- Start the StatInspector addin (see chapter "Startup").

6. Configuration Documents

Each Domino server running the StatInspector addin must be defined with a configuration document.

Configuration for Server1/Test

Collecting server:	Server1/Test
Collect from:	Server1/Test Server2/Test
Statistic names:	DAOS Mail Platform Traveler
Probe interval:	30 Minutes
Alert expiration:	32 days
Mail alerts to:	

In this example, the StatInspector is running on Server1/Test and monitors the statistics for Domino Server1/Test and Server2/Test. The alert documents are automatically deleted after one month. If you add one or more recipients in the field "Mail alerts to:" then an email is sent to these recipients, whenever an alert documents is created.

If you collect statistics from remote Domino servers, you must add the collecting server in the corresponding server documents in the Domino Directory as a trusted server, e.g.

Server Access	Who can -
Access server:	All users can access this server
Not access server:	\$DenyAccess
Create databases & templates:	LocalDomainServers LocalDomainAdmins
Create new replicas:	LocalDomainServers LocalDomainAdmins
Create master templates:	LocalDomainServers LocalDomainAdmins
Allowed to use monitors:	*
Not allowed to use monitors:	
Trusted servers:	Server1/ACME

7. Probe Documents

You define the alert conditions thru the use of probe documents. Each probe document contains an expression field, which is freely programmable. These expressions are evaluated with a built-in JavaScript engine so every expression based on the JavaScript language is allowed. If the result of the expression

evaluates to a Boolean true, the StatInspector addin will create an alert document. If the expression returns anything else (e.g. nothing or false), the alert does not get triggered.

Probe

Status:	<input checked="" type="radio"/> Enabled <input type="radio"/> Disabled
Categories:	DAOS
Limit to servers:	
JavaScript code:	if ([DAOS.Engine.Status] == 'Enabled' && [DAOS.Engine.Catalog] != 'Synchronized') return true
Alert message:	DAOS: The DAOS catalog is not synchronized.
Description:	Alerts if the DAOS catalog is not in a synchronized state. To correct this error, enter "Tell DAOSMgr ReSync".

All statistic names specified between brackets will be replaced with the current value of the corresponding server statistic. If the statistic value does not exist, the expression will not be evaluated and the probe document is skipped.

Example:

```
if ([Mail.Dead] > 0)
return true
```

There are some special variables available for use in your JavaScript code:

- `$Interval` Number of seconds since last statistic inspection
- `$ServerName` Current server name in abbreviated format (e.g. "Server1/ACME")

Some of the statistics represent incrementing values, e.g. "Mail.Delivered". If you need to reference these numeric counters, you must prefix the statistic name with a plus sign to retrieve the difference between the current and the previous probe interval.

Example 1: Test if the DAOS catalog is in a synchronized state:

```
if ([DAOS.Engine.Status] == 'Enabled' && [DAOS.Engine.Catalog] != 'Synchronized')
return true
```

Example 2: Test if any undeliverable mails are in the routers mailbox:

```
if ([Mail.Dead] > 0)
return true
```

Example 3: Test if the number of mailboxes (mailn.box) is not sufficient:

```
if ((([Mail.Mailbox.AccessConflicts] / [Mail.Mailbox.Accesses]) * 100) > 2)
return true
```

Example 4: Test if more than 10 mails/sec were delivered:

```
if ([[+Mail.Delivered] / [$Interval]) > 10)
return true
```

8. Alert Documents

The StatInspector creates an alert document for each server when one or more probe document triggered an event.

Alert on Server/Test

Date and time:	07.07.2012 16:11:32
Server name:	Server/Test
Messages:	 Number of undeliverable mails: 5

In this example, an alert document was created after checking the probe for undelivered mails on Domino server Server/Test. The document link at the beginning of each message opens the corresponding probe document.

9. Startup / Termination

The StatInspector is always started thru Domino RunJava. This built-in task created the Domino Java Virtual Machine (JVM) environment and then loads and executes the users Java class. Because the StatInspector is based on the JAddin framework, the first Java class loaded is "JAddin" which then loads the Java class of the StatInspector tool. Therefore, the first parameter of the RunJava is always JAddin which is followed by StartInspector. The general syntax is:

```
RunJava JAddin StatInspector <DatabaseName>
```

The optional parameter may be used to override the default path and name of the StatInspector.nsf database, e.g.

```
RunJava JAddIn StatInspector Apps\StatInspector.nsf
```

As any other addin, this tool may be terminated with the "Tell StatInspector Quit" command.

9.1 Manual Startup

You may start the StatInspector by entering the command "Load RunJava JAddin StatInspector" at the Domino console, e.g.

```
> Load RunJava JAddin StatInspector
22.07.2012 14:18:35   JVM: Java Virtual Machine initialized.
22.07.2012 14:18:35   RunJava: Started JAddin Java task.
22.07.2012 14:18:35   StatInspector: Domino Server Statistic Inspector Version 1.0.0
(Freeware)
>
```

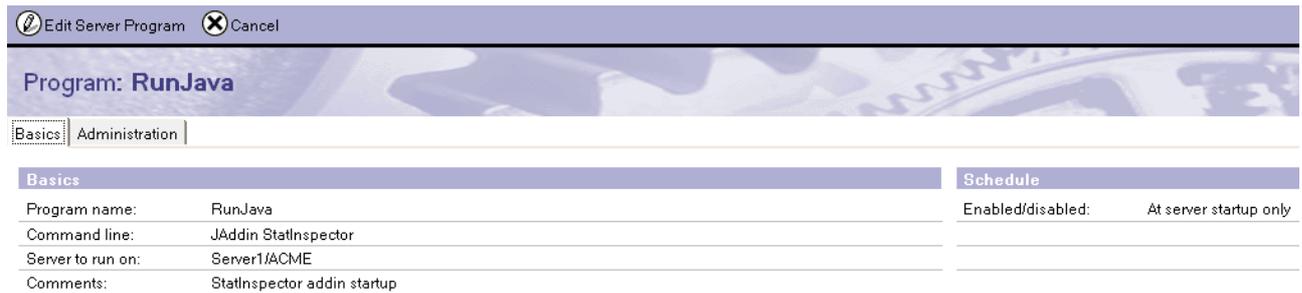
9.2 Domino Notes.Ini

If you want to start the StatInspector addin at Domino server startup, you can directly modify the Domino Notes.Ini file. Simply search for the line starting with "ServerTasks=" and add the command to start the addin, e.g.

```
ServerTasks=Replica, Router, Update, RunJava JAddin StatInspector, AMgr, HTTP, Traveler
```

9.3 Program Document

The preferred way to start the addin is thru the use of a program document in the Domino Directory:



The screenshot shows the 'Edit Server Program' dialog box for a program named 'RunJava'. The 'Basics' tab is selected, and the configuration is as follows:

Basics		Schedule	
Program name:	RunJava	Enabled/disabled:	At server startup only
Command line:	JAddin StatInspector		
Server to run on:	Server1/ACME		
Comments:	StatInspector addin startup		

9.4 Terminate the Addin

The StatInspector can be terminated the same way as any other Domino server addin with the command `Tell StatInspector Quit`.

10. Some Technical Background

- This server addin is written entirely in Java for maximum operating system and platform independency.
- Since there is no Java API to collect the Domino server statistics, this tool needs to execute one or more `Show Statistic` console commands. Due to a 32 KB limit on the data returned by Domino, these console commands must be executed once per category. Therefore these categories must be defined in the configuration database.
- StatInspector is based on the free JAddin framework. If you are interested in writing your own Domino server addin with Java, you can download the JAddin framework via <http://ABData.CH/JAddin.html>.

11. Feedback

Please share your experience, knowledge and best practice of the Domino server statistics by sending your JavaScript code or your feedback. This will make the tool even more valuable. I will be glad to include your probes into the next release. You can contact the author via email at Andy.Brunner@ABData.CH. Thank you!

12. Frequently Asked Questions

- Question: Why do I see one of these messages during startup?
 RunJava: Can't find class Jaddin or lotus/notes/addins/jaddin/Jaddin in the classpath. Class names are case-sensitive
 JAddin: Error: Unable to load the Java class StatInspector: StatInspector (wrong name: StatInspector)

Answer: All Java class names are case sensitive. The class names "JAddin" and "StatInspector" must be entered exactly as shown. Make sure you start the addin with the command "Load RunJava JAddin StatInspector".

- Question: Why do I see several console commands Show Statistic in the Domino log.nsf?

Answer: See explanation in the chapter "Some Technical Background".

13. Links

- JavaScript Reference: <https://developer.mozilla.org/en/JavaScript/Reference>

14. Release History

Future Version	<ul style="list-style-type: none"> • First formal release • Add: More sample probe documents based on feedback from the beta users
Version 0.8.0 19/Aug/2012	<ul style="list-style-type: none"> • First public beta version • Minor documentation changes • Minor code changes
Version 0.7.0 29/Jul/2012	<ul style="list-style-type: none"> • Minor documentation changes • Add surrounding quotes when replacing a string [Variable] • Reformat mail sent to the administrator • Include JAddin 1.2.0
Version 0.6.0 18/Jul/2012	<ul style="list-style-type: none"> • First private beta version
Version 0.5.0 04/May/2012	<ul style="list-style-type: none"> • Proof of concept