
Created: 2002-01-21

Last modified: 2011-08-20

dim_STAT User's Guide

by Dimitri

dimitri.kravtchuk@free.fr

Table of contents

- [Overview...](#)
 - ◆ [General View](#)
- [LICENSE](#)
 - ◆ [GPL v2 License](#)
 - ◆ [Freeware End User License](#)
- [Installation](#)
 - ◆ [INSTALL.sh](#)
 - ◆ [Starting Web and Database servers](#)
 - ◆ [Migration from any old dim_STAT version to the new one](#)
 - ◆ [First-Level Security](#)
- [STAT-service](#)
 - ◆ [Install STAT-service](#)
- [Main Page](#)
 - ◆ [ERROR: No X_ROOT configuration for SERVER](#)
 - ◆ [Web Browsers](#)
- [Preferences](#)
 - ◆ [Example](#)
- [Start On-Line Collecting](#)
 - ◆ [Main Steps](#)
 - ◆ [Few screenshots...](#)
 - ◆ [Load collect from output files](#)
 - ◆ [Standalone configuration](#)
- [EasySTAT](#)
 - ◆ [Example](#)
 - ◆ [EasySTAT Hints](#)
- [BatchLOAD](#)
 - ◆ [BatchLOAD Example](#)
 - ◆ [Special NOTE](#)
 - ◆ [GUDs integration](#)
- [Analyzing](#)
 - ◆ [Welcome Analyze!](#)
 - ◆ [LOG Messages](#)
- [Multi-Host Analyzing](#)
 - ◆ [Select Multi-host](#)
 - ◆ [Choose Collect\(s\) and Time interval](#)
 - ◆ [Choose STATs](#)
 - ◆ [Go!](#)
 - ◆ [Result with Static Log](#)

- ◆ [Result with Dynamic Log](#)
- [Single-Host Analyzing](#)
 - ◆ [Choose Collect and STAT](#)
 - ◆ [Example IOSTAT: Choose Disks criteria](#)
 - ◆ [Example IOSTAT: Choose STAT Variables](#)
 - ◆ [Example IOSTAT: Result Graph](#)
 - ◆ [Save Graph as Bookmark...](#)
- [Bookmarks](#)
 - ◆ [Choose Collect and click on Bookmarks...](#)
 - ◆ [Choose Time interval and Graphics style](#)
 - ◆ [Select all Data you want to see and GO!](#)
 - ◆ [Result Page](#)
 - ◆ [Administration actions](#)
- [Multi-Host Extended Analyze](#)
- [dim_STAT CLI](#)
 - ◆ [Example](#)
- [Administration](#)
 - ◆ [Active/Stopped Collect](#)
 - ◆ [Delete/Recycle Collects](#)
 - ◆ [Export/Import collects](#)
 - ◆ [Modify Collect parameters](#)
 - ◆ [LOG Messages operations](#)
- [Add-On Statistics](#)
 - ◆ [Example of SINGLE-Line command integration](#)
 - ◆ [MULTI-Line Add-On command integration](#)
 - ◆ [REAL LIFE EXAMPLE...](#)
 - ◆ [Pre-Integrated Add-Ons](#)
 - ◆ [Administration tasks](#)
- [Linux Special Notes](#)
 - ◆ [Linux STAT-service](#)
 - ◆ [Lvmstat](#)
 - ◆ [Lmpstat](#)
 - ◆ [LcpuSTAT \(deprecated\)](#)
 - ◆ [LioSTAT](#)
 - ◆ [psSTAT for Linux](#)
 - ◆ [LpsSTAT \(psSTAT\)](#)
 - ◆ [LPrcLOAD \(ProcLOAD\)](#)
 - ◆ [LUsrLOAD \(UserLOAD\)](#)
 - ◆ [LnetLOAD \(netLOAD\)](#)
- [Report Tool](#)
 - ◆ [Overview](#)
 - ◆ [Datatype: Text, HTML, Image, Binary](#)
 - ◆ [Datatype: SysINFO](#)
 - ◆ [Datatype: HTML.tar.Z](#)

- ◆ [Datatype: dim_STAT-Snapshot](#)
- ◆ [Datatype: dim_STAT-Collect](#)
- ◆ [Preview / Generate / Publish](#)
- ◆ [Export / Import](#)
- ◆ [Let's try! New Report](#)
- ◆ [Click on Report Tool](#)
- ◆ [New Report](#)
- ◆ [Edit Report](#)
- ◆ [Edit Actions](#)
- ◆ [Edit Note](#)
- ◆ [Edit Note, continue...](#)
- ◆ [Edit Note, continue2...](#)
- ◆ [Edit Note, continue3...](#)
- ◆ [Edit Report, continue...](#)
- ◆ [Edit Report, continue2...](#)
- ◆ [Add Note](#)
- ◆ [New Note -- SysINFO](#)
- ◆ [New Note -- SysINFO Form](#)
- ◆ [New Note -- SysINFO Result](#)
- ◆ [New Note -- SysINFO Link Contents](#)
- ◆ [Edit Report, continue3...](#)
- ◆ [Edit Report, continue4...](#)
- ◆ [Add New Note -- Image](#)
- ◆ [Add New Note -- Image Inline](#)
- ◆ [Add New Note -- Image Linked](#)
- ◆ [Add New Note -- dim_STAT Collect, Step1](#)
- ◆ [Add New Note -- dim_STAT Collect, Step2](#)
- ◆ [Add New Note -- dim_STAT Collect, Step3](#)
- ◆ [Add New Note -- dim_STAT Collect, Step3 continue](#)
- ◆ [Add New Note -- dim_STAT Collect, Step4](#)
- ◆ [Add New Note -- dim_STAT Collect, Step5](#)
- ◆ [Add New Note -- dim_STAT Collect Result](#)
- ◆ [Add New Note -- dim_STAT Collect Contents, ordered by:Collect](#)
- ◆ [Add New Note -- dim_STAT Collect Result per STATS](#)
- ◆ [Add New Note -- dim_STAT Collect Contents, ordered by:STATS](#)
- ◆ [Edit Report, next...](#)
- ◆ [Edit Report -- Cut](#)
- ◆ [Edit Report -- Paste!](#)
- ◆ [Edit Report -- Pasted...](#)
- ◆ [Edit Report -- Preview](#)
- ◆ [Edit Report -- Preview Output](#)
- ◆ [Edit Report -- Preview Output2](#)
- ◆ [Generate Report](#)
- ◆ [Generated Report documents](#)
- ◆ [Report Tool Home](#)
- [Additional Tools](#)
 - ◆ [Java2GIF Tool](#)
 - ◆ [Java2PNG Tool](#)
 - ◆ [HTMLDOC Tool](#)

- [FAQ](#)
 - ◆ [Sizing of dim_STAT Instance...](#)
 - ◆ [I've started my collects but it seems that nothing gets collected?](#)
 - ◆ [Syntax of text matching pattern](#)
 - ◆ [When will you upgrade to the newer MySQL version?](#)
 - ◆ [With multiple hosts to monitor, is it possible to graph them together?..](#)
 - ◆ [How easy is it to integrate any new stats to monitor, including DTrace stuff?](#)
 - ◆ [Could I get the raw data via dim_STAT-CLI instead of the graphs?...](#)
 - ◆ [I have a Windows machine to monitor remote UNIX boxes.... Any help?..](#)
- [Full Working cycle Example](#)

Overview...

dim_STAT is a tool for both high-level and detailed, monitoring and performance analysis of Solaris, Linux, and other UNIX systems.

The main features of dim_STAT are:

- ◆ A web based user interface
- ◆ All collected data is saved in a database
- ◆ Multiple data views
- ◆ Interactive (Java) or static graphs (PNG)
- ◆ Real Time monitoring
- ◆ Multi-Host monitoring
- ◆ Post analyzing
- ◆ Statistics integration (Add-On)
- ◆ Professional reporting with automated features
- ◆ One-click STAT-Bookmarks
- ◆ etc.

All STAT data is collected from standard UNIX tools like vmstat, iostat, etc. (or some special ones, like psSTAT for monitoring users and processes activity) and saved in the MySQL database. Collected data is accessed via a web interface and can be presented in several manners (interactive or static graphs, text, HTML tables). Since v.8.1 there is also a way to collect data from other UNIX systems (HP/UX, AIX, MacOSX, etc.)

dim_STAT can be used for the on-line monitoring of one or several hosts at the same time. As well, data can be post loaded from output files of stat commands and analyzed in the same manner. At any time data collection from new stat commands can be added to the tool (via Add-On interface) to enlarge your view on application workloads, RDBMS, your personal STAT program, etc.

By default, dim_STAT interfaces with the following Solaris stats (SPARC and x86):

- ◆ vmstat
- ◆ mpstat
- ◆ iostat
- ◆ netstat
- ◆ psSTAT, ProcLOAD, UserLOAD (processes and users)
- ◆ ZoneLOAD, PoolLOAD, ProjLOAD, TaskLOAD (CPU/memory/etc. load per zone/pool/project/task (Solaris 10))
- ◆ netLOAD (extended network stats)
- ◆ UDPstats (UDP traffic)
- ◆ IOpatt (Solaris 10 I/O pattern via DTrace)
- ◆ vxstat (VxVM stats)

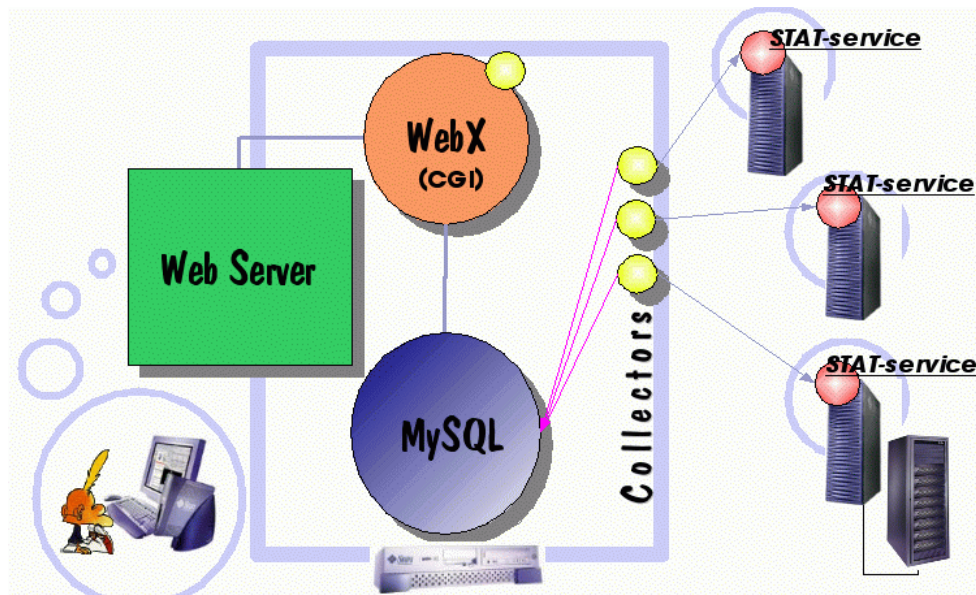
as well as the following Add-On extensions for both Solaris SPARC/x86 and/or Linux/x86:

- ◆ CoreSTAT (Solaris)
- ◆ MEMSTAT (Solaris)

- ◆ HAR v2 (Solaris CPU chip counters for SPARC and x64)
- ◆ jvmSTAT (Java VM GC Activity and Memory Usage stats)
- ◆ oraEXEC, oraIO, oraSLEEP, oraENQ, oraASMIO (Oracle activity stats)
- ◆ mysqlSTAT, mysqlLOAD, innodbSTAT, innodbMUTEX, innodbMETRICS (MySQL & InnoDB activity stats)
- ◆ pgsqlSTAT, pgsqlLOAD (PostgreSQL activity stats)
- ◆ LvmSTAT (Linux vmstat)
- ◆ LcpuSTAT (Linux mpstat)
- ◆ Lmpstat (Linux mpstat v2)
- ◆ LioSTAT (Linux iostat)
- ◆ LnetLOAD (Linux netLOAD)
- ◆ LpsSTAT (Linux psSTAT)
- ◆ LprcLOAD (Linux ProcLOAD)
- ◆ LusrLOAD (Linux UserLOAD)
- ◆ IObench (tool for I/O stress load)
- ◆ dbSTRESS (tool for database stress load)
- ◆ OSXiostat, OSXvmstat, OSXnetstat (experimental MacOSX support was added since v.9.0)
- ◆ and **mostly any** other program you want to add...

The CPU utilization of dim_STAT during collect is very low and even less than standard tools like top or perfbar.

General View



Just to get an idea how dim_STAT works.

Each machine you want to monitor in real-time should run a special STAT-service daemon (client). Via the web browser you start collectors to communicate with clients. All information collected gets saved in a database and may be analyzed as soon as the data is arriving or later on. In general, all analysis, reporting or administration is done from the web browser. The web interface is developed and runs on WebX (my own tool) ...

LICENSE

Since v.8.3 dim_STAT is moving to GPLv2 license!

But all old stuff which I have only as binary or other binaries shipped without sources will stay under freeware license.

[GPL v2 License](#) @

[Freeware End User License](#) @

Installation

The dim_STAT installation package is either delivered as a TAR archive (dim_STAT.tar) or, when on CDs, already "untarred".

Before install: Verify your available disk space - you will need ~60MB for the initial install, mostly to store Web Server and Database Server data. The database volume will grow according to the number of (future) STAT collections and the web directory may grow with your reports. So reserve enough space for your data ...

During installation: a new user "**dim**" and a group "**dim**" will be created. User "dim" is the owner of the dim_STAT database and the web server. In case your system has special rules or restrictions, you may create these manually beforehand, or you may choose other user and group names that are following your system policies. Please, after installation, don't forget to set a password to this user! (otherwise cron is not allowing execution of regular clean-up tasks via 'crontab')...

INSTALL.sh

As the **root** user, unload the tar archive into some directory and start the installation script:

```
# cd /tmp
# tar xvf /path_to_tar/dim_STAT.tar
# cd dim_STAT-INSTALL
#
# INSTALL.sh
```

During installation you will be asked to confirm your host IP address (found automatically), host and domain name, the script verifies if the user "dim" already exists on the system, if not it will be created, and you will be asked about WebX and

home directories (Web Server, Database Server, Administration and Client scripts, etc.) and about port numbers to be used.

Mainly you have to choose 3 application directories:

- ◊ WebX home (default: /opt/WebX)
- ◊ Data home (default: /apps)
- ◊ Temporary space (default: /tmp)

And a user/group name which will be the owner of the dim_STAT data in your system (default: 'dim')

If you are not sure about the meaning of some values, leave them by default.

NOTE: **WebX** is the main interpreter (or execution engine), it interprets all application script files and **absolutely** needs a fixed and trusted root (home) directory. Otherwise, anyone may execute whatever they want on your machine (like /etc/passwd to crack logins, etc.). So, as a first step protection for its root directory: you may choose one of 4 available paths (hey, 4 choices anyway, better then one :)). Also, the WebX engine itself is very small (only a few MB) and not growing.

After install, the dim_STAT software will be distributed on your system in the following way:

```
+ /WebX, /apps/WebX, /opt/WebX or /etc/WebX - WebX main directory (only 4
|
+ /apps - default dim_STAT home directory
|
+-- /ADMIN      - administration scripts (start/stop dim_STAT Server, B
|
+-- /mysql      - MySQL database server main directory
|
+-- /httpd      - Apache Web server directory
|
+-- /client      - client collect script(s)
|
+-- /Java2GIF   - Java applet graph to GIF convertor
|
+-- /htmldoc    - HTML to PDF converting tool
|
+-- ...         - there may be other directories depending on dim_STAT
```

NOTE: To simplify things, the next examples assume that your home directory is '/apps' and owner's user name is 'dim'.

Silent INSTALL

Since version 8.1 there is a silent "auto install" feature integrated in the install script. It may be very useful in case you need to automate the installation of dim_STAT on your servers. To activate it, use the '**-Auto yes**' option.

Then add more options if you need to have any settings different from the default:

- -HOST `hostname`
- -IP ip_address
- -USER dim
- -GROUP dim
- -WebX_DIR /opt/WebX
- -TEMP_DIR /tmp
- -HOME_DIR /apps
- -HTTP_PORT 80
- -DB_PORT 3306
- -STAT_PORT 5000
- -USERADD yes (add user/group)
- -AutoLink yes (make auto-start links in /etc/rc*.d)

Examples :

Default install:

```
# ./INSTALL.sh -Auto yes
```

With customized Home:

```
# ./INSTALL.sh -Auto yes -HOME_DIR /export/home/apps/dim_STAT
```

With existing User:

```
# ./INSTALL.sh -Auto yes -USER stat -GROUP staff -ADDUSER no -HOME_
```

etc...

Starting Web and Database servers

As you saw before, administration scripts are placed in **/apps/ADMIN** :

```
# cd /apps/ADMIN
# dim_STAT-Server start
```

To stop servers:

```
# cd /apps/ADMIN
# dim_STAT-Server stop
```

NOTE: a global **dim_STAT-Server** script is working as the main admin interface and replaces various separate httpd / mysql scripts. This global script also checks before a stop/start action if there are any active collects running and restarts them automatically during the next startup. Also, if the shutdown was not properly done, startup script will print a warning messages about a possible need of index rebuild on

some databases...

At any moment you may look in the database for any active connections.

```
$ su - root
# /apps/mysql/bin/mysql -S /apps/mysql/data/mysql.sock

mysql>
mysql> show processlist;
```

Id	User	Host	db	Command	Time	State	Info
3	dim	localhost	Mind	Sleep	18	NULL	NULL
4	dim	localhost	Mind	Sleep	17	NULL	NULL
5	dim	localhost	Mind	Sleep	2	NULL	NULL
6	dim	localhost	Mind	Sleep	1	NULL	NULL
7	dim	localhost	Mind	Sleep	2	NULL	NULL
8	dim	localhost	Mind	Sleep	16	NULL	NULL
9	dim	localhost	Mind	Sleep	104	NULL	NULL
10	dim	localhost	Mind	Sleep	1	NULL	NULL
11	dim	localhost	Mind	Sleep	0	NULL	NULL
53	dim	localhost	UPC	Sleep	108	NULL	NULL
54	dim	localhost	UPC	Sleep	103	NULL	NULL
56	dim	localhost	UPC	Sleep	115	NULL	NULL
57	dim	localhost	UPC	Sleep	118	NULL	NULL
58	dim	localhost	UPC	Sleep	112	NULL	NULL
59	dim	localhost	UPC	Sleep	105	NULL	NULL

...

and even kill any of them (however, be very careful !!)

```
mysql> kill 57;
mysql> quit
Bye

#
#
```

MySQL Admin Tips

MySQL administration is very easy. However, depending on a user's past experience, here are some tips which may help...

First of all, be aware, dim_STAT is using MySQL MyISAM engine to save data. This engine has no transactions support nor transaction log, etc., but it's very easy to manage, it does all needed stuff quite well, providing a reasonable SQL interface, and keep all saved data fully platform-independent! (you may simply copy your data files from Linux/x86 to Solaris/SPARC station and continue to work with them without any problem!). Of course, without transaction log there is still a risk to loose some data due system crash or power outage... But if you'll put to the list of

priorities all important points you'll see that loosing few minutes of collected data are much less important rather database software cost as well having skills to administrate it.. - you don't need any DBA skills to administrate MySQL for dim_STAT! UNIX admin habits will be enough :-)

As much as you can, use separated databases: it's much more easier for administration, it avoids possible future activity conflicts, etc. Since v.8.3 there is a possibility to add an Admin password while creating a new database - all administration action then will require giving this password (start/ stop/ restart of collects, data drop, etc.)

Limitation in the number of connections: each MySQL connection uses 5 file descriptors (avg). This means that with a maximum of 1024 file descriptors per process (default in some old systems), we can't create more than ~200 connections on a multi-threaded MySQL server (Note: each STAT command in collect uses its own single connection). In case you run dim_STAT server on Solaris and need more connections (several hosts, many stats, etc), first check the values of your /etc/system parameters : *rlim_fd_cur* and *rlim_fd_max*. Next, in the file */apps/mysql/mysql.server* replace the default value of 2000 by a new one (current dim_STAT server is just configured with a limit of 2000 connections, however it depends on the system how much it'll be able to acquire, as well you may always increase again this value)...

Accidental "power off" on your machine: MySQL server within dim_STAT is configured in way to force data flush every 5 minutes. So, if your database was not used for a long time - your data should be safe.. However for active databases it's very possible some of their index files will be corrupted. The dim_STAT-Server script will print a warning message in this case, but you'll need to run manually the data checks..

NOTE: you do NOT need to stop dim_STAT server! :-)

Supposing you discovering some data errors on the database "Demo" (for example):

- First of all you stop all collects on this database (and check via 'Preferences' there is no connections anymore to this database)..
- Wait 15 minutes (MySQL will flush data and close files)
- Start "repair" MySQL command:

```
# cd /apps/mysql/data/Demo
# /apps/mysql/bin/myisamchk -r *.MYI
```

- Restart all your collects you previously stopped

Since v.8.2 auto-repair was removed from dim_STAT-Server script, because:

- Recovery process blocked all users from using database during whole recovery time..

- It's extremely difficult to say which table/database will need or not need a data recovery (even if it was closed properly it doesn't mean yet indexes were not corrupted - during system crash filesystem buffers may still stay dirty and not flushed to disk(s)).
- Finally the only running "myisamchk -r" gives you a true repair in this case and it may take a lot of time.

Since v.8.2:

- Every 5 minutes mysql daemon is forced to flush key buffers and close all table files - it's protecting at least non-active databases, their data normally will still stay stable in case of system crash!
- If system crash happens, MySQL server will still start correctly but with a warning message - probably some of the databases will need a data repair!..
- If you discover your database is broken:
 - stop all active collects on it
 - wait 5 minutes (within 5 minutes all your tables will be closed)
 - start recovery on your database (see above)
- This solution is give a way to recover databases in preferred by user order, as well leave other working (if they don't need to repair) or just create a new database and still continue your work!

Probably with a time for some critical system environment there will be a possibility to upgrade databases to InnoDB engine and not take care anymore about system crashes, but it's just a part of future plan for the moment :-)

No more disk space: just add disks if possible :). The collect part of dim_STAT is done in such a way to "keep the flow", in case of errors nothing will be stopped. Once you have added space, the collects will continue, but you probably will get some holes during this period.

To get a backup/copy of your collects in the fastest way: one of the great features of MySQL is its support of cross-platform data compatibility. As an example, the same database files may be moved from a Solaris machine and successfully reused on a Linux laptop. And most cases, copying the whole database to another machine will be much more faster than exporting and again importing collects via flat files. The exception is if you want to move only a very small amount of data from a large database.
Fine, but can we do this on-line? - Yes!! Like in "repair" steps:

- Stop all collects in your database
- Wait 15 minutes
- Backup the database (ex. "Demo"):

```
# cd /apps/mysql/data /your_backup
# cp -rp Demo /your_backup_path
OR:
# tar cf - Demo | gzip > /your_backup_path/Demo.tgz
```

- Restart all previously stopped collects...
- **NOTE:** since v.8.3 there is a web interface added to safely backup whole database.

Delete the database: there is no way to delete a database via the web interface (generally, I don't like deleting :)). Delete by error is such a common thing ... so, if you really need to delete your database, the only way is:

- Check there is no more connections to your database
- Delete database files (ex. "Demo"):

```
# rm -rf /apps/mysql/data/Demo
```

Running several MySQL instances on the same host: long time ago it was one of the bigger problems to avoid dim_STAT to conflict with already installed and running databases on an existing system. The solution I found is isolating the dim_STAT database completely from existing instances, but the price for it is a few more complexity for simple things. The tool now uses its own parameters for TCP/IP ports and UNIX sockets. For example, to connect locally to your database server, instead of the usual:

```
# /apps/mysql/bin/mysql DatabaseName
```

you should now use:

```
# /apps/mysql/bin/mysql -S/apps/mysql/data/mysql.sock DatabaseName
```

MySQL: datafile corruption

This section is covering a particular case when table is not repaired by "myisamchk", and usually you get a following message:

"table TABLE doesn't have a correct index definition" etc.

The solution is:

- Stop dim_STAT server
- Start only MySQL instance
- Connect to your database
- Execute CHECK, then REPAIR of your TABLE
- Stop MySQL instance

- Start dim_STAT server

The following example is demonstrating a real case with "dim_MPSTAT" table:

```
bash# /apps/ADMIN/dim_STAT-Server stop

bash# /apps/mysql/bin/myisamchk -r -f dim_MPSTAT.MYI

IF IT DID NOT HELP:

bash# /apps/mysql/mysql.server start

bash# mysql -S /apps/mysql/data/mysql.sock Benchmark_TTT
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Didn't find any fields in table 'dim_MPSTAT'
Welcome to the MySQL monitor.  Commands end with ; or g.
Your MySQL connection id is 1 to server version: 3.23.53

Type 'help;' or 'h' for help. Type 'c' to clear the buffer.

mysql> check table dim_MPSTAT;
+-----+-----+-----+-----+
| Table                                | Op      | Msg_type | Msg_text |
+-----+-----+-----+-----+
| Benchmark_TTT.dim_MPSTAT             | check   | warning  | Table is marked as cra |
| Benchmark_TTT.dim_MPSTAT             | check   | warning  | Size of datafile is: 1 |
| Benchmark_TTT.dim_MPSTAT             | check   | error    | Found 16918142 keys of |
| Benchmark_TTT.dim_MPSTAT           | check | error   | Corrupt   |
+-----+-----+-----+-----+
4 rows in set (19.39 sec)

mysql> repair table dim_MPSTAT;
+-----+-----+-----+-----+
| Table                                | Op      | Msg_type | Msg_text |
+-----+-----+-----+-----+
| Benchmark_TTT.dim_MPSTAT           | repair | status  | OK      |
+-----+-----+-----+-----+
1 row in set (7 min 34.16 sec)

mysql>

bash# /apps/mysql/mysql.server stop

bash# /apps/ADMIN/dim_STAT-Server start
```

The doc reference is here (see comments) -

<http://dev.mysql.com/doc/refman/5.0/en/myisamchk-repair-options.html>

(Thanks Google! :-))

Using InnoDB Engine instead of MyISAM

Since **dim_STAT v.9.0** it is possible to use **InnoDB** Storage Engine within MySQL instead of MyISAM. This Engine is a true transactional one and pretty safe against server power-off or system crashes.. You may choose to use this InnoDB instead of MyISAM on Database creation, or at any moment

convert your Database from one Engine to another. The only thing you'll not be able to do with InnoDB is a full "physical" backup of your Database files (in this case you'll need to convert your Database to MyISAM first). However there is no problem with Import or Export.

NOTE: bigger is your database, more it'll take time to convert it from one Engine to another one..

Since v.9.0 to simplify DBA-like tasks there is an admin tool included:
dim_STAT-Admin .

dim_STAT-Admin Tool

dim_STAT-Admin is shipped since v.9.0 to avoid to use a web interface for sometimes heavy DBA tasks.

With dim_STAT-Admin you're able from the command line:

- Create a new Database
- Convert existing Database to another Storage Engine
- Backup a whole Database
- Export STAT Collect(s)
- Import STAT Collect(s)
- Recycle STAT Collect(s)

Command line:

```
$ ./dim_STAT-Admin
```

dim_STAT-Admin CLI (dim) v.1.0 > Usage:

dim_STAT-Admin [options] Options: -CMD Command

Commands: CREATE, BACKUP, CONVERT, EXPORT, IMPORT, RECYCLE -Base DBname Database Name (if empty: prints database name list) ...

Additional options: (depending on Command) **CREATE** :

-Engine Name MyISAM (default) or InnoDB -Passwd

PASSWORD optional password setting for Admin actions

BACKUP : -Passwd PASSWORD if password was assigned for Admin actions -File Filename full path output file name for tar.Z backup file

CONVERT : -Engine Name MyISAM or InnoDB -Passwd

PASSWORD optional password setting for Admin actions

EXPORT : -ID id1[,id2,...] Collect ID(s) to export (if empty: prints available Collect list) -Begin YYYYMMDDhhmiss optional begin date+time -End YYYYMMDDhhmiss optional end date+time -File filename full path output file name for tar.Z export file

IMPORT : -ID id1[,id2,...] optional Collect ID(s) to import (if known) -File filename full path file name for input tar.Z import file

RECYCLE : -Days N keep data collected during last N days

-ID CollectID optional collect ids (ex: id1,id2,id3 or "ALL"
for any ID) (if empty: All active collects only)

Migration from any old dim_STAT version to the new one

The migration procedure is quite easy:

- ◇ Stop all activity on your current dim_STAT installation
- ◇ dim_STAT-Server stop
- ◇ Backup all your databases from '/apps/mysql/data/' (see below) except:
dim_00, mysql and dim
 - mysql: system database, don't play with it !!
 - dim_00: is a reference database and changing with every release
 - **dim**: is a "Default" database, and if you really need it, rename it before backup
- ◇ Install the new dim_STAT distribution
- ◇ Restore your backup-ed data into '/apps/mysql/data'
- ◇ Start dim_STAT-Server

Enjoy :))

NOTE: The old database should be seen as before and work correctly, but if you want to get an advantage of the all new features coming within new version, then create a new database and start new collects.

First-Level Security

The main point: **ANY SECURE SYSTEM IS NEVER SECURE ENOUGH...** The question is only, what will you consider secure ENOUGH for you :))

Anyway, during discussions with our engineers and customers, the security issue was so often raised that I cannot leave it without attention.

For paranoia-users: there is a Solaris X86 or Linux version of dim_STAT and if you really need maximum protection, spend some money on a small dedicated PC, run dim_STAT on it and protect any access with firewalls, etc.

In my experience, I suggest to protect access to the web server, to prevent somebody from just by error stopping or suspending active collects. For this kind of first-level access protection, a good candidate is Apache's ".htaccess". For a more detailed information, please refer to the Apache documentation. But in short, just to make it work with dim_STAT:

- ◇ 1) via `/apps/httpd/bin/htaccess` create `/apps/httpd/etc/.htpasswd` file and add any pairs of user/password you need
- ◇ 2) create `".htaccess"` file with context:

```
AuthName "Welcome to dim_STAT Host"
AuthType Basic
AuthUserFile /apps/httpd/etc/.htpasswd
```

```
require valid-user
```

- ◇ 3) copy `".htaccess"` file into `/apps/httpd/home/docs` and `/apps/httpd/home/cgi-bin`
- ◇ 4) try to connect to your web server now and check the access user/password - that's all! ;-)

Example:

```
$ /apps/httpd/bin/htpasswd

Usage: htpasswd [-c] passwordfile username
The -c flag creates a new file.

$ /apps/httpd/bin/htpasswd -c /apps/httpd/etc/.htpasswd login1
Password:
...

$ vi /tmp/.htaccess
$ cat /tmp/.htaccess
AuthName "Welcome to dim_STAT Host"
AuthType Basic
AuthUserFile /apps/httpd/etc/.htpasswd

require valid-user
$
$ cp /tmp/.htaccess /apps/httpd/home/cgi-bin
$ cp /tmp/.htaccess /apps/httpd/home/docs
```

STAT-service

STAT-service was introduced in dim_STAT since version 3.0 and provides a simple, stable and secure way for on-line collecting of STAT data from Solaris/SPARC, Solaris/x86 and Linux/x86 servers. Since v. 8.1 it's distributed under GPL with source code, so you may compile it now yourself on other platforms to collect data from other UNIX platforms. As a pilot example, a package for HP/UX is provided. And any newly ported kits are of course welcome! Since Jun.2009 there is also available a version of STAT-service daemon rewritten in Perl by Marc KODERER: http://search.cpan.org/~mkoderer/stat_agent-0.09/stat_agent.pl - feel free to try this version too and don't forget to send your comments and RFE to Marc! :-)

Install STAT-service

The STAT-service module is shipped as part of the dim_STAT distribution (*dim_STAT-INSTALL/STAT-service* directory), in form of Solaris packages or as tar archives for manual integration. STAT-service has to be installed on every machine that needs to be monitored. The install is to be done as "root" user.

Package install (".pkg" file) :

```
# pkgadd -d STATsrv.pkg
```

Manual install (".tar" file) :

```
# cd /etc
# tar xvf /path_to/STATsrv.tar
# ln -s /etc/STATsrv/STAT-service /etc/rc2.d/S99STATsrv
# ln -s /etc/STATsrv/STAT-service /etc/rc1.d/K99STATsrv
# ln -s /etc/STATsrv/STAT-service /etc/rc0.d/K99STATsrv
# ln -s /etc/STATsrv/STAT-service /etc/rcS.d/K99STATsrv
```

The software needs to be installed into a special **/etc/STATsrv** directory, which is the home directory of STAT-service. The contents of this directory is:

```
/etc/STATsrv/
  STAT-service      -- script to start/stop service daemon, also defines port
  access            -- access control file
  /bin              -- contains extended STAT programs/scripts
  /log              -- contains all logged information about service demands
```

Next step, start the service daemon:

```
# /etc/STATsrv/STAT-service start
```

The way dim_STAT and STAT-service are communicating with each other is very simple:

- ◇ 1) dim_STAT connects to the STAT-service daemon of the monitored server
- ◇ 2) if the service is not available, then wait a time-out and go to 1) or exit if the STAT collect is stopped during this period
- ◇ 3) dim_STAT will ask about the stat command that it needs
- ◇ 4) if there are no permissions for this command or the command is not found, the "command" connection will be closed with an error message
- ◇ 5) dim_STAT collects the data, maintaining any time-shift due to previous time-outs
- ◇ 6) if the TCP connection is broken: go back to 1)
- ◇ 7) if STAT is stopped, then close the connection and exit
- ◇ 8) if there was no activity during the "auto-eject" timeout, close the connection and goto 1)

As you see, this schema is quite robust and will work after cluster switching, network corruptions, reboots, etc. Collections can be started once and then left running for a long period. In case you need to collect only during specific time intervals, you may just start and stop the STAT-service through a "cron" job or a similar tool.

Note: it appears that during a **halt** of the system (a power-off of a running machine), the TCP/IP connections can stay and don't receive an error code. When this happens, the collect should be broken via a "auto-eject" timeout. However, auto-eject can also happen due to a mini-hang on the system or simply of the stat program. In this case you'll see holes in your collects, so take care when interpreting the results.

STAT-service Access control file

Here is an example of STAT-service access control file. As you see, you may limit the number of stat commands accessible for each machine. This task may be done by host administrator and may be completely independent.

IMPORTANT :

- access file all the time checked by STAT-service daemon, so you never need to restart service to activate your modifications.
- since v.8.0 only stat commands working for sure on a given system are **enabled** by default. It's up to you to enable other commands which may need some additional configuration (like jvmSTAT, oraEXEC, etc.) or simple software presence (like VxVM for vxstat) - "enable" means just uncomment them within your /etc/STATsrv/access file :-)
- since v.8.5 you may add a port number for a command! - it gives a way to collect several similar stats from the same host but from the different sources :-)

For example, if you're running say 3 Oracle database instances on the same server and still wanting to monitor each one in details, but there is only one oraEXEC possible per system because (as it) it may accept only one Oracle SID... So you may just make several copies of the same oraEXEC.sh wrapper and assign them to the different ports like that:

```
command  oraEXEC      /etc/STATsrv/bin/oraEXEC_sid0.sh
command  oraEXEC:5001  /etc/STATsrv/bin/oraEXEC_sid1.sh
command  oraEXEC:5002  /etc/STATsrv/bin/oraEXEC_sid2.sh
```

then you start several STAT-service processes (on port 5000, 5001 and 5002) and collect data from your servers like it was 3 different hosts :-) (and from port 5000 you'll collect data about SID#0, from 5001 - SID#1, 5002 - SID#2)... - it's a straight forward way in a such situation as well for MySQL and PostgreSQL too as it's still more simple solution rather to rewrite whole the stuff to accept several databases on the same time...

```
#
# STAT-service access file
#
# Format:
#
#   ...      command      name[:port]    fullpath
#   ...
#   access   IP-address
```



```

# Oracle
#command oraEXEC      /etc/STATsrv/bin/oraEXEC.sh
#command oraIO         /etc/STATsrv/bin/oraIO.sh
#command oraENQ        /etc/STATsrv/bin/oraENQ.sh
#command oraLATCH      /etc/STATsrv/bin/oraLATCH.sh
#command oraSLEEP      /etc/STATsrv/bin/oraSLEEP.sh

# MySQL
#command innodbSTAT    /etc/STATsrv/bin/innodbSTAT.sh
#command mysqlSTAT     /etc/STATsrv/bin/mysqlSTAT.sh
#command mysqlLOAD     /etc/STATsrv/bin/mysqlLOAD.sh

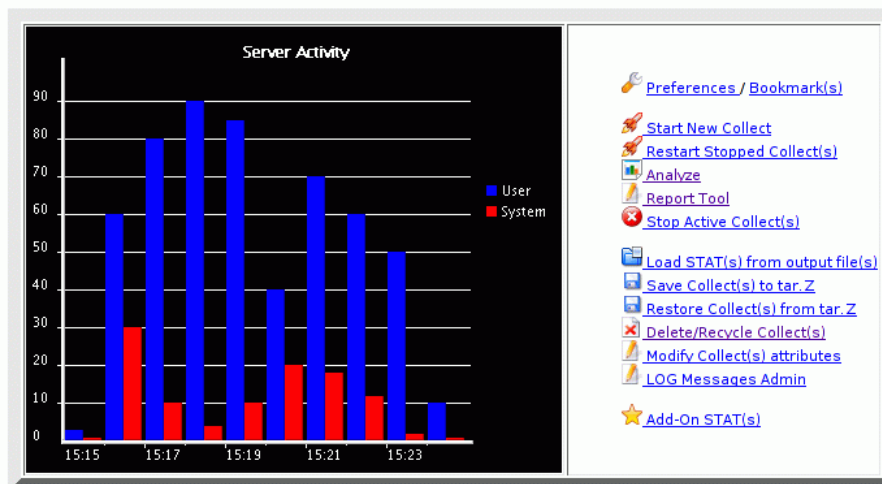
# PostgreSQL
#command pgsqLSTAT     /etc/STATsrv/bin/pgsqLSTAT.sh
#command pgsqLOAD      /etc/STATsrv/bin/pgsqLOAD.sh
#

```

Main Page

Database: [Default]

dim_STAT v.8.0



dimitri.kravtchuk@france.sun.com

Now, the installation is finished, the database and the web servers are running. Be sure that the STAT-service is installed and running on **all** servers you want to monitor. You'll be surprised, but when people are having trouble, in 90% of cases it is just forgetting to start the STAT-service.

Once it's done, you are ready to open a web browser (doesn't matter if it is Java enabled or not) and connect to the dim_STAT web server. The first page contains some links to documentation, presentation, tool history, etc., but the link you'll need to click is "**Main**"

Page".

As you already supposed, the Main Page will group all main actions ... and you're right!

I will not present this action by action, but rather functionality by functionality, in order of operation. However, the shortest working cycle is probably still:

- ◆ Starting STAT collect
- ◆ Analyze/Monitor collecting data
- ◆ Stop STAT collect

A few words about the User Interface. Don't be surprised if you will not find any "Back" button once you leave the Main Page. There isn't one! You have to use your browser's navigation back button for it. And it's not because I'm just lazy :)) The reason is simple: dim_STAT uses Java applets to present data in graphical mode, but it seems for every Java applet instance the web browser instantiates a dedicated JVM. And all JVMs will stay in the browser's memory until it will crash with an "out of memory" error. To prevent that, I unfortunately have to force you to use your browser's button.

Since version 7.0 you'll see a small toolbar at the top of your page representing:

- Currently used Database Name
- Short links into Home/ Preferences/ Log Admin

ERROR: No X_ROOT configuration for SERVER

Sometimes, instead of the Main Page, you see this error message.

Don't worry, nothing wrong!! What is happening is that your DNS translation simply did not match the configuration settings. Go to the **WebX** home directory (ex: /opt/WebX) and open the "**x.config**" file in a text editor. Find the line containing your host name in the first column. Duplicate this line and replace in it your hostname:port pair as given by the string in the error message after "SERVER:". Save the file and try to connect again. It should work immediately!

Example: Error Message: "No X_ROOT configuration for SERVER:
harms.France.Sun.COM:88"

- ◇ vi /opt/WebX/x.config
- ◇ duplicate the line with "harms:88"
- ◇ in the new line, replace "harms:88" with "harms.France.Sun.COM:88"
- ◇ save the file
- ◇ reload the Main Page in your browser

Note: X_ROOT is a one of WebX's configuration parameters. As WebX is an interpreter, there should be a way to protect it from "interpreting" something else than application pages (ex: /etc/passwd). X_ROOT gives WebX its main "root" directory, so that only pages in this specific directory tree can be executed, and nothing else.

Note: Since v.9.0 the pattern "*" is provided to accept any host name with any port number in case such a level of security is not required..

Web Browsers

Since version 7.0 you may use any web browser as long as it supports the PNG image format (true for nearly all available browsers).

However, if you prefer the interactive graphs from dim_STAT's Java Applet, you must have a Java plug-in configured. Here are a few notes about specific browser programs:

- ◇ **Firefox** - most stable web browser for today, works perfectly with Java applets and may be the best choice. Specially useful as it's able to keep all checkboxes remaining pre-selected even if you're reloading an active page ;-)
- ◇ **Opera** - seems to work fine since v.5 (and I'm using it a lot as an excellent alternative ;-))
- ◇ **Konqueror** - generally working out of the box, probably the best choice for KDE-lovers :-))
- ◇ **Safari** - works just fine out of the box, probably the best choice for Mac-lovers ;-))
- ◇ **Mozilla** - you should upgrade to at least to version 1.7. In previous versions there was a bug starting an applet before receiving all given parameters. Also, 1.7 and later is much faster compared to previous versions.
- ◇ **IE** - never used it myself, but it seems to work for customers, etc.

There are some other browsers out there, but as a general rule, if you see instead of the graphics an error message "Browser BUG", then you should either upgrade your browser or move on to another one. As well if you use only PNG graphs you will usually never meet any problem...

Preferences

The preferences page contains a set of key options used by different parts of the application. The most critical of them are grouped here. All other options (if supported) are "auto-keeping" their last value. If you used dim_STAT before you will notice that there are no graph settings anymore, all graph values are auto-saved each time when you use the graph view.

Note: your browser must accept cookies to make some of the following features working!!

There isn't a global "settings" button, and I didn't want to create too many links. So, each option has its own validation button, don't forget to click it to apply your modifications.

Database - Without any special settings, all collected data is stored in the "Default" database (the real MySQL name is "dim"). However, to avoid possible contention and simplify further administration, it's highly recommended to use different databases for different projects/ users/ centers/ etc. Within the Database section you can choose the name of the existing database you want to use or you can create a new one and use it instead. Since v.8.3 there is a possibility to add an Admin password while creating a new database - all administration action then will require giving this password (start/ stop/ restart of collects, data drop, etc.). As a reminder, the current database name is shown in the browser's title and the toolbar of every dim_STAT window.

Free and Used disk space - Showed for the current database. (Note: MySQL has a quite small storage footprint, so disk space usage will be most reasonable, but it's a good habit to check from time to time if you still have a disk space! (since v.8.2 datafiles are configured to be able to reach 2TB in size (seems enough, no? ;-))...

Host Name List - Here you can specify a pre-defined list of the servers you usually monitor. This list is saved within a database, so every person using the same database may reuse it; as well if you switch databases time to time in your browser your host list will be changed automatically! Since v.8.2 the host "aliasing" is added: the complete syntax for the host name is [alias/]hostname[:port]

Example:

- ◆ you want to collect data from a host known in the LAN as **abz45060** , IP address **10.1.1.15** , and running STAT-service on the port **5050** (because 5000 was already used by another application)...
- ◆ if you like the name "abz45060" - you may just enter **abz45060:5050** into the host list
- ◆ but if you prefer another name (ex. reflecting a server role, etc.) - for example "oradb" - you may just enter **oradb/abz45060:5050** and in every graph this host will be named as **oradb**
- ◆ NOTE: you may also replace abz45060 by IP address: **oradb/10.1.1.15:5050** (according to your taste :-))

Bookmark Term - If you have never used dim_STAT before, just leave it as it is. For others, this option was created to satisfy everyone who prefers a different name for "Bookmark" functionality. Bookmarks were introduced in version 4.0, but after long discussions we still have no agreement on the right name. So, now you're free to name it as you like! :))

LOG Messages option - Gives you a way to set:

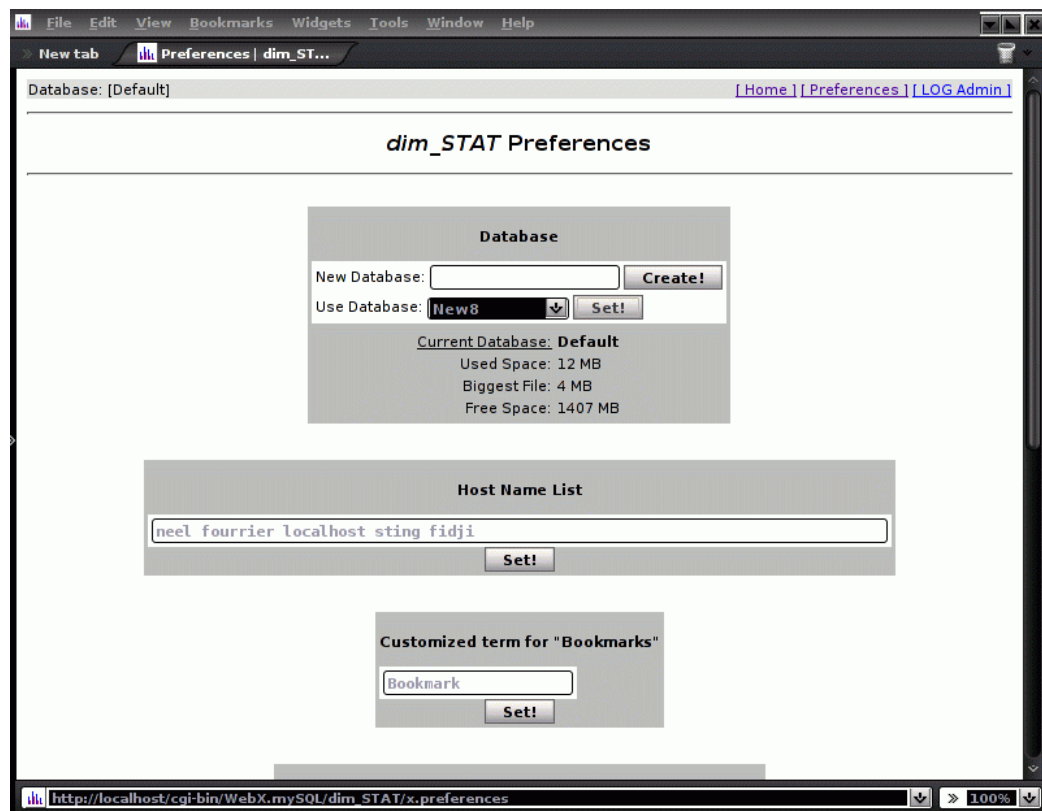
- ◆ enable/disable auto-generated time slice messages for easier time interval selection
- ◆ message list size setting (in lines)

- ◆ max message visible length (in characters)

Page Colors - You're free to play with page colors if you're not happy with the default settings or simply prefer to change it from time to time.

Check Java support - A simple way to check if the dim_STAT applet is working correctly with your browser.

Example



Start On-Line Collecting

Before starting any STAT collect, first check if the STAT-service is running on every server you want to monitor. This is the most common error!!

Another point, if you want to monitor a Linux server, be sure you've installed the Linux STAT Add-Ons, before starting any collect (see the special Linux section in this document).

Now, from the dim_STAT Main Page you may just follow the [Start New Collect](#) link. (Note: since version 8.0 there is no distinction anymore between single and multi host collect).

IMPORTANT:

- ◆ A STAT collect for a host is independent of any other, so it can be stopped and/or restarted at any time, independent of other collects.
- ◆ Your collect options saved into special script files with a name based on the "Collect Base Name". Using customized names you may pre-load a different set of options, according to your needs.
- ◆ You may start a collect on-line from your browser, or you can make a start script, to be run by hand, via cron, as a batch job, etc.

Main Steps

There are 4 main points in starting a STAT Collect:

1. choose a host name(s)
2. set collect attributes (title, id, etc.)
3. choose collected statistics
4. start now, or prepare a script for manual/delayed execution

1. Host name(s)

Since version 8.0 you choose your host(s) first. You may setup a list of frequently used host names on the 'Preferences' page. This list as well all other used host names are kept via browser cookies. Before you start any STAT collect, for each given host name, dim_STAT will indicate the status of your host's STAT-service by LED color. I hope it avoids potential misconfiguration issues for both new and experienced users. For now there are 3 LED colors:

- Red: the host is not running STAT-service on the default port, or the host is inaccessible from the network, or the host is down.
- Orange: the host is running STAT-service but an older version.
- Green: ok! STAT-service is running and has all required features.

NOTE: since v.8.0, STAT-service has a new 'stat publish' feature. Using this, the application knows exactly what kind of STATs you can or can't collect from any given host. It protects you from choosing the wrong or unavailable data.

2. Set Collect Attributes

Collect BaseName - all selected options are saved in a special start script. The name of this script is composed of BaseName + some context extensions. When you start a new collect, the next time you may pre-load previously selected options by giving the previous

BaseName and clicking on "Preload" (by default the last given BaseName is stored using a cookie).

ID - all data in the database referenced to this ID. The ID is not assigned automatically, to give you a choice to use personalized range numbers (your project id; etc.).

Title - the title description you give for starting the collect.

Time Interval - how frequent (in secs.) you want dim_STAT to collect data (the default is 30 seconds, which is right in most cases)

Client Log File - the name of a file on the "host" that you want to watch. All text lines appended to this file will be automatically be copied into the STAT database and timestamped. While analyzing the collected STATs you can visualize the log messages that correspond with the analyzed interval. This may be very useful to trace auto-starting jobs, night batches, etc. They also give you a simple and fast way to find the correct time position during data analysis, like "show N minutes before/after/around a selected message".

STAT-service Port - the port number on which STAT-service is running. By default the tool will use the port number given during installation and it's a good practice to use the same port on every host.

3. Choose Statistics

Simply select all statistics you want to collect. Help bullets show a full description of each STAT (if you have JavaScript enabled in your web browser). Better be selective, probably you don't need everything.

A good set of STATs to start with:

- VMSTAT
- MPSTAT
- IOSTAT
- netLOAD (avoid using 'netstat')
- ProcLOAD

These STATs will give you a good overview of the resource utilization on your hosts. Once you have analyzed them, you may go more in-depth and fine-tune the selected STATs.

NOTE: all "official" Add-Ons are installed by default in each dim_STAT database, BUT! not enabled by default in the STAT-service! On the host side only surely working stats are enabled! Be sure to check [/etc/STATsrv/access](#) file on each server before you're starting any collect! :-)

For example: if you're needing to collect "vxstat" data, and you know a VxVM is installed and running on this host - just uncomment the VXSTAT line in your /etc/STATsrv/access file and things will work!...

4. Start Mode

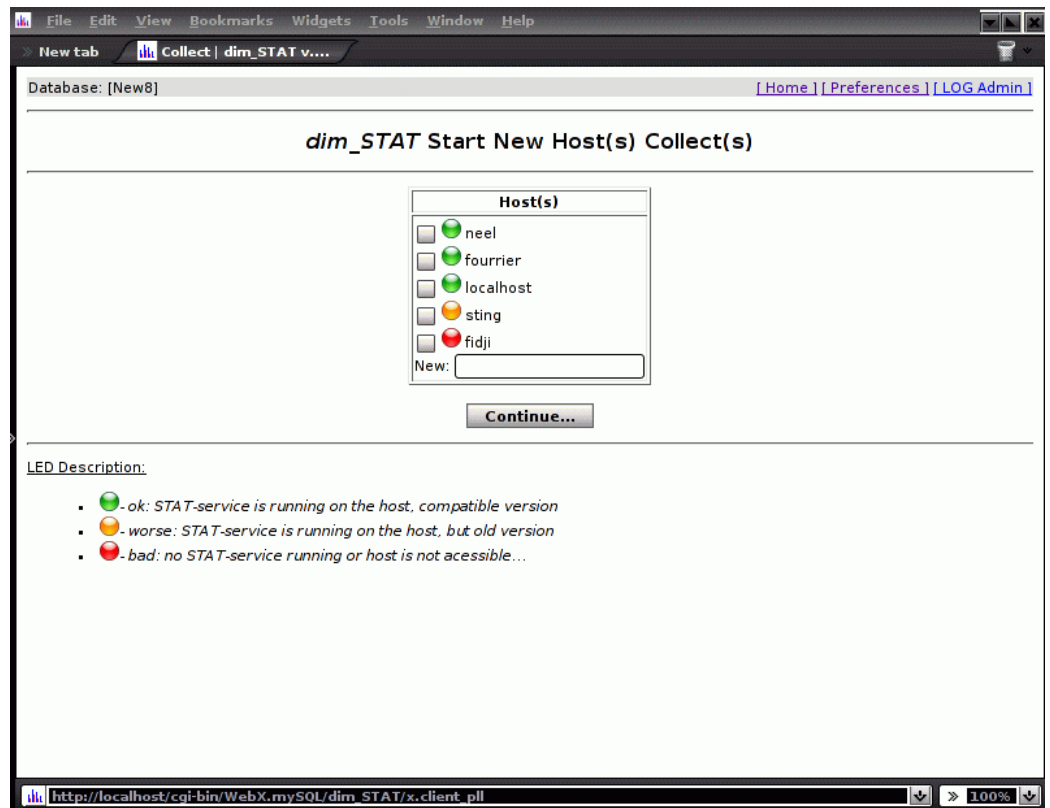
[**Make Start script only**] - don't start the collect, just create a script

[**Start Now!**] - start the new STAT collect right now

[*] **Show Debug output** - in case you want to see debug messages about the collect startup

Few screenshots...

Select Host(s)



You may see here several servers:

- **neel, fourrier** - Solaris hosts running upgraded STAT-service
- **localhost** - Linux box, upgraded STAT-service
- **sting** - Solaris host, old STAT-service
- **fudji** - Solaris host, powered off

I select neel, fourrier, localhost and sting and click on [Continue] button...

Choose STATs

Database: [New8] [\[Home \]](#) [\[Preferences \]](#) [\[LOG Admin \]](#)

dim_STAT Start New Host(s) Collect(s)

Collect BaseName:

Stat Title:

Time Interval: sec.

STAT-service port:

Host	Stat ID	VMSTAT	MPSTAT	IOSTAT	VXSTAT	NETSTAT	psSTAT	harSTAT
neel	<input type="text" value="7"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Active-ONLY		<input type="checkbox"/>	<input type="checkbox"/> Active-ONLY	
fourrier	<input type="text" value="8"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Active-ONLY		<input type="checkbox"/>	<input type="checkbox"/> Active-ONLY	
localhost	<input type="text" value="9"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	
sting	<input type="text" value="10"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Active-ONLY	<input type="checkbox"/> Disk Group(s): <input type="text" value="rootdg"/>	<input type="checkbox"/>	<input type="checkbox"/> Active-ONLY	<input type="checkbox"/>

☐ Show Debug Output

WARNING: you have to be sure STAT-service is running on all machines you want to monitor!

LED Description:

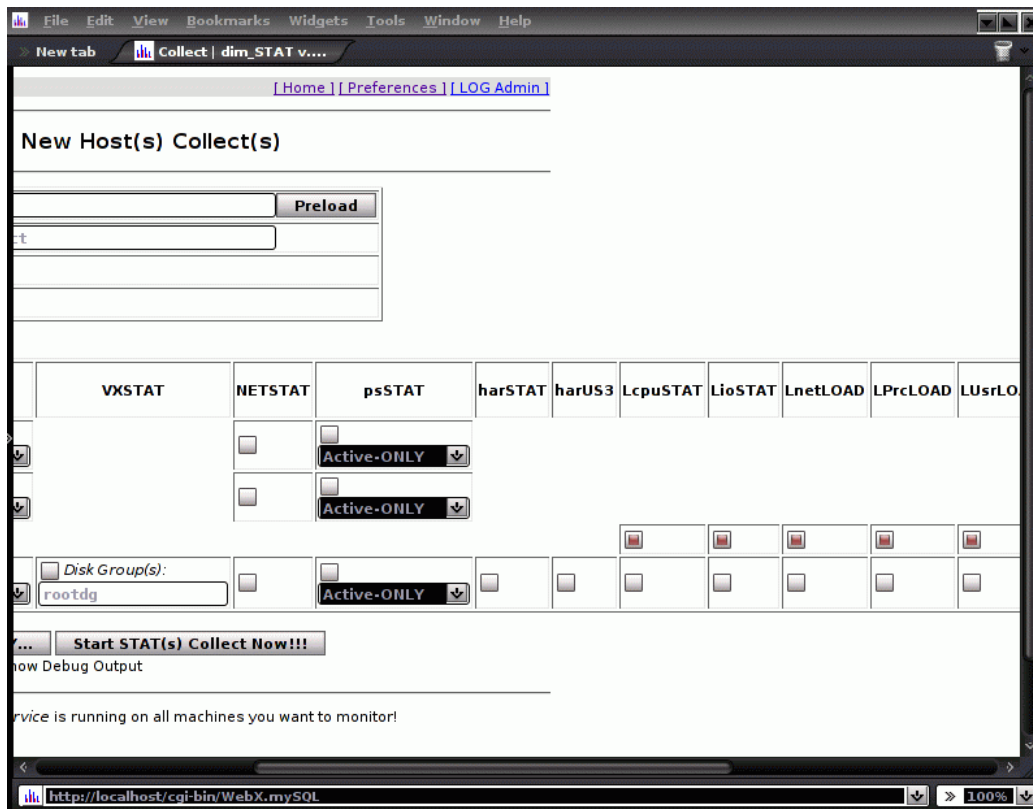
<http://localhost/cgi-bin/WebX.mysql>

The hosts are chosen, let's select the STATs to collect.

Some remarks about these hosts:

- ◆ Linux stats are not proposed for 'green' Solaris hosts
- ◆ Solaris stats are absent for 'green' Linux hosts
- ◆ for any 'green' host, not configured or disabled stats are absent
- ◆ the 'orange' host `sting` has all its stats present, but as it was from before v.8.0, it's up to you to remember which commands will run on it or not

Choose STATS, next



Load collect from output files

If you cannot collect data directly from your hosts and all you have is a set of statistics output files, then you may still download them via the Web interface as a STAT collect and analyze later. Just fill the required parameters and of you go.

However, if your output files are representing a big volume, it may take much more time to load, and your browser may simply timeout and loose the connection. And you'll never see the final result.

In such cases, a better solution is to use **EasySTAT** (simplified) or **BatchLOAD** (for more experimented users). See the following sections for more details.

Standalone configuration

Before you think about collecting your stats via some kind of scripts, don't forget about the possibility of a "standalone" dim_STAT. There is absolutely no restriction to:

- ◆ install dim_STAT on a host
- ◆ start STAT-service on the same host
- ◆ collect data from that host into dim-STAT on the same host
- ◆ and **be aware**, on a 4 CPU machine (which is relatively small server) a collect with a 20 second interval (vmstat + mpstat + iostat + psSTAT + netLOAD), will generate only **0.2% CPU usage**. (Yes !!)

The CPU usage of dim_STAT for collecting data is very low. However, during data analysis or when doing export/import/etc. actions, CPU utilization is very high.

So, don't forget about this simple solution: install dim_STAT on the same host you want to collect from, collect locally all the data you need, and then backup the whole database, copy it onto another machine and analyze it there. Alternatively, in the case of a benchmark, keep the data on the same server, but take care that you're not doing any analysis at the same time as you're running your testruns.

EasySTAT

Since dim_STAT version 7.0, the **EasySTAT** script makes part of the STAT-service for Solaris. EasySTAT is designed to simplify the combination of collecting STATs on "very remote" or "highly secured" hosts with BatchLOAD.

In a few words all you need to do is:

- ◆ install STAT-service on the host
- ◆ run EasySTAT
- ◆ backup the output directory
- ◆ restore the directory on your dim_STAT server
- ◆ execute the "LoadDATA.sh" script (from the directory)

EasySTAT Usage (v.1.9)

```
$ /etc/STATsrv/bin/EasySTAT.sh OutDIR IntervalSec NbHours [Title [Hostname [DBName]
```

options:

```
OutDIR      - Output directory for stat collects (def: /var/tmp)
Interval    - measurement interval for stat commands in sec. (default: 15)
NbHours     - execution duration in hours (default: 8 hours)
Title       - title to use during BatchLOAD processing
Hostname    - hostname to use during BatchLOAD processing
DBname      - database name to use during BatchLOAD processing
Batch       - full path to BatchLOAD binary on your server (default: /apps/ADM
Log         - log file name (if given, all processing output is forwarded into
              NOTE: may also be enabled via LOG environment variable (see EasyS
```

EasySTAT Config

By default script collects 5 main stats:

- ◆ VMSTAT (runqueue, memory, CPU)
- ◆ MPSTAT (per CPU usage, interrupts, mutex, etc.)
- ◆ IOSTAT-xn (per disk I/O stats)
- ◆ netLOAD (network per interface stats +nocanput)
- ◆ ProcLOAD (processes stats summarized by process name)
- ◆ you may add any other Add-On commands by editing /etc/STATsrv/bin/EasySTAT.sh

Additional Options

- ◆ To reduce disk space usage, since v.8.3 if environment variable **COMPRESS** is set, EasySTAT will automatically call it to compress every finished output file:

```
# COMPRESS=gzip /etc/STATsrv/bin/EasySTAT.sh
...
```

Don't forget to "uncompress" output files before start any load process! :-)

- ◆ Since v.8.3 if **TIMER** environment variable is set to "yes", EasySTAT will automatically timestamp all collecting data within its output files:

```
# TIMER=yes /etc/STATsrv/bin/EasySTAT.sh
...
```

All timestamp tags are transparent for BatchLOAD and serving only to simplify human reading. Also, if during collecting there were some output freezes due high system load or other - Timer will automatically take care about it and add a special time sync tag to synchronize data when loading to the database..

NOTE : since v.8.3-1 both COMPRESS and TIMER options are included within EasySTAT.sh script **by default** !!! - it's preferable to have compressing and timestamps out of the box to avoid any space overflow as well a faster text file analyzing. However be aware you have to edit EasySTAT.sh file to disable them (but at least you know what you're doing :-))

Example

Collect STATS :

On the 'Very Remote' Host :

```
==> copy STATsrv.pkg somewhere (ex: /tmp) and install: # pkgadd
-d /tmp/STATsrv.pkg
==> create data dir # mkdir /var/tmp/Easy # cd /var/tmp/Easy
==> collect data every 30sec. for 24 hours # nohup
/etc/STATsrv/bin/EasySTAT.sh /var/tmp/Easy 30 24 & ...
```

```

==> archive+compress collected data # cd /var/tmp # tar cf - Easy |
compress > /tmp/Easy.tar.Z
==> copy /tmp/Easy.tar.Z into your laptop/flash/CD/etc. # ...up to
you :)
==> remove all stuff if no more need # rm /tmp/Easy.tar.Z; rm -rf
/var/tmp/Easy; pkgrm STATsrv

```

Load Collect then Analyze :

On your local dim_STAT server:

```

==> restore Easy.tar.Z somewhere (ex: /home/tmp): # cd /home/tmp
# uncompress < Easy.tar.Z | tar xvf - # cd Easy/* # gunzip *.gz ## (if
compressions was used)
==> edit if you need to modify default settings (db name for ex.) # vi
LoadDATA.sh
==> load all data into your database (don't forget to create this
database before!!!) # sh LoadDATA.sh
==> Analyze data via web interface & enjoy :)

```

EasySTAT Hints

Few notes about EasySTAT hints (some were introduced with 8.3-1 version):

- ◇ **Per hour files** -- to avoid having collected data out of sync with a real time, EasySTAT is restarting each stat program every hour; so every hour you have a new file for all stats, and it's by default, and was designed from the beginning
- ◇ **Run forever** -- to run EasySTAT for undetermined period just give a "0" for a number of hours - also, in this case EasySTAT will not create a new working directory for incoming stats, but (re)use the given directory name
- ◇ **Inittab** -- you may use */etc/inittab* to make your EasySTAT collects permanent - if for any reasons collects were stopped (or killed) - init process will restart them automatically! - all you need is just to add a such kind of line at into your */etc/inittab*:

```
dim:3:respawn:/etc/STATsrv/bin/EasySTAT.sh /var/tmp/stats 15 0 2>&1 >>
```

it'll collect stats forever with 15sec time interval, and keep data in "/var/tmp/stats" directory; to force rescan of modified */etc/inittab*:

```
# init q
```

to disable collecting just replace "respawn" by "off" and then "init q" again :-)

the advantage of a such solution is that it'll work on **any UNIX** platform ;-)

◇ **PID file** -- EasySTAT always creating a pid file within its working directory:
.EasySTAT.pid

◇ **Stopping** -- at any time you need to stop EasySTAT gracefully - just send a TERM or INT signal to its PID:

```
# kill `cat .EasySTAT.pid`
```

◇ **LoadDATA.sh file(s)** -- on USR1 signal EasySTAT backing up its current LoadDATA.sh file into a LoadDATA.sh-saved-... and then creating a new LoadDATA.sh for all next incoming collects (until next SIGUSR1 ;-)) - it may be helpful if you're collecting your stats permanently but want to be able to upload them into your dim_STAT database by time periods, etc...

BatchLOAD

The idea for **BatchLOAD** came (as many things) from day to day needs. Sometimes you are facing customers/users who want to know what happens on their machines, but then they don't allow the installation of any additional software (a very constructive approach :-)).

All you can do now is to ask them to run some stat commands on their systems and send you the output files. While loading their files every day via the Web interface, you start to think harder and harder if there isn't a way to do this automatically. Are you ready for BatchLOAD??

I decided to add a new component to dim_STAT, but I kept in mind that other tools already exist that are collecting output from stat commands. All these tools are keeping data in their own format, so I've tried to design the input format for BatchLOAD to be easily adaptable. Of course, I didn't think to create something universal :)), but I hope it shouldn't be too hard to write a script that can convert from an existing format into BatchLOAD.

Some words about the internals of BatchLOAD. There is no dependency on the name of loaded files. All needed information is given by command options and in the contents of the loaded file. The loaded file must have special TAGs. At least two: to give the STAT name and to confirm the END.

USAGE:

```
Usage: /apps/ADMIN/BatchLOAD -cmd NEW/ADD options
```

Options [NEW]:	-- force new collect creation
-base DBname	-- database name
-ID id	-- Collect ID, if 0 use max+1 id automatically
-title Title	-- Collect Title
-host Hostname	-- Collect Host Name
-isec sec	-- Collect STATs Interval (sec)
-start datetime	-- Collect Start DateTime in format YYYYMMDDHHMISS
-skip1 yes/no	-- Yes/No skip first STAT measurement (often wrong values)
-file Filename	-- Full path to file with STATs outputs
-verbose on/off	-- verbose output on/off

```
Options [ADD]:      -- add to existing collect whenever possible
                   -base DBname      -- database name
                   -host Hostname     -- Collect Host Name (optional)
                   -ID id             -- Collect ID, if 0 :
                                     -- if host is given - use max id used by host
                                     -- otherwise, use max (last) id automatically
                   -skip1 yes/no      -- Yes/No skip first STAT measurement (often wrong values)
                   -file Filename     -- File with STATs outputs
                   -verbose on/off    -- verbose output on/off
```

Example :

```
$ /apps/ADMIN/BatchLOAD -cmd NEW -ID 0 -base ANT -file `pwd`/vmstat.out -skip1 no -t
$ /apps/ADMIN/BatchLOAD -cmd ADD -ID 0 -base ANT -file `pwd`/iostat.out -skip1 no
$ /apps/ADMIN/BatchLOAD -cmd ADD -ID 0 -base ANT -file `pwd`/mpstat.out -skip1 no -v
```

In this example the first line will create a new STAT Collect using an automatic new ID (max+1), with the title "Test BatchLOAD" and it will load the first file: "vmstat.out" The second and third lines load into the new Collect the next data, "iostat.out" and "mpstat.out". Once it is finished, we can connect to the dim_STAT web server and start to analyze.

Note : multiple "-file" options can be used at the same time. For example:

```
$ /apps/ADMIN/BatchLOAD -cmd NEW -ID 0 -base ANT -skip1 no -title "Test BatchLOAD"
    -host V880 -isec 20 -start 20031024100000 -file `pwd`/vmstat.out
    -file `pwd`/mpstat.out -file `pwd`/iostat.out
```

File Format of STAT output

The file format is designed in such a way as to give maximum flexibility on data grouping and processing.

The main TAGs are STAT and END:

```
==> STAT StatName -- after this point all following data corresponds to
given STAT command (StatName) Supported STAT names: VMSTAT
MPSTAT IOSTAT (iostat -x) IOSTAT-xn (iostat -xn) VXSTAT (vxstat -v)
psSTAT
And all other Add-On STAT you are able to create, like some already
shipped:
netLOAD T3stat oraEXEC oraIO ...
```

```
==> END -- end of STAT data
```

At any time the following TAGs may also be inserted:

```
==> DTSET yyyy-mm-dd hh:mi:ss      -- set date+time point for next STAT data
```

==> **LOGMSG** message -- add log message into database corresponding to the currently loading data

Outside of the "STAT" - "END" blocks, any other lines are ignored.

Note : TAGs are exactly as it shown: "==> STAT", "==> END", "==> DTSET", "==> LOGMSG". Don't miss any characters!

BatchLOAD Example

A small example, let's say you have three vmstat and three iostat files corresponding to let's say "morning", "day" and "night" activity for some special tasks. Therefore you can make six load files, each one containing its own "STAT", "DTSET", "END" TAGs, or put all in one.

```
...
==> DTSET 2004-01-19 10:30:00           -- set "morning" point
==> LOGMSG Morning workload
==> STAT VMSTAT                         -- load vmstat
... output of vmstat.out1
==> LOGMSG Strange CPU activity         -- marking time period to anal
... continue ...
==> END                                -- end of first vmstat

==> STAT IOSTAT-xn ... output of iostat.out1 ==> END

==> DTSET 2004-01-19 14:30:00 -- set "day" point ==> LOGMSG
Day workload ==> STAT VMSTAT ... output of vmstat.out2 ==>
END

==> STAT IOSTAT-xn ... output of iostat.out2 ==> END

==> DTSET 2004-01-19 23:30:00 -- set "night" point ==> LOGMSG
Night workload ==> STAT VMSTAT ... output of vmstat.out3 ==>
END

==> STAT IOSTAT-xn ... output of iostat.out3 ==> END
```

All information is placed in one single file ready to load:

```
$ /apps/ADMIN/BatchLOAD -cmd NEW -ID 0 -base ANT -skip1 no -title "Customer W
                        -host V880 -isec 20 -start 20040119100000 -file `pwd`/all_stat.out
```

In the same way, you can group all data of the same STAT command in a single file. Or all outputs corresponding to the same collecting time period.

NOTE : don't forget to create your database before starting any load!! In this example the database name is 'ANT'.

Special NOTE

Please, take care - there is no option to give a name of loaded stat command! That's why "STAT" and "END" tags are **mandatory!**. Even you want to load just one vmstat file, tool have no idea about your file contents till it'll meet a "STAT" tag inside!

GUDs integration

If you already worked with Sun support or you're Sun employee - you may know or already used GUDs (shell script collecting various system information + stats and saving them into special case archive). GUDs was created by Sun France engineer, and another French engineer made an integration script to load GUDs data into dim_STAT via BatchLOAD - 'guds2dim.sh'. This script is shipped now with dim_STAT and may be found in /apps/ADMIN directory. To obtain GUDs script - please, contact directly Sun support.

Analyzing

Analyzing your STAT data is quite intuitive, but let's just give some screen shots and few words of comment.

Once you click on the "Analyze" link you have 3 options:

- ◆ Single-Host Analyze
- ◆ Multi-Host Analyze
- ◆ Multi-Host Extended Analyze

Let's take for now the Multi-Host option, as it's the easier one :-)

There are some other additional options:

- ◆ Active ONLY - show only currently running collects
- ◆ STATs Status - in Single Host mode this option shows high numbers of already collected stats (very **important** to see if something is really collecting)
- ◆ Title matching - to filter collects on title pattern
- ◆ LOG matching - to filter LOG messages with a text pattern

Welcome Analyze!

Database: [V8_5_00]

[\[Home \]](#) [\[Prefs \]](#) [\[LOG Admin \]](#)**dim_STAT Analyze**

☒ Single-Host Analyze
☐ Multi-Host Analyze
☐ Multi-Host Extended Analyze

- ☐ Active ONLY
- ☐ Preselect Multi-Hosts
- ☐ Show STAT(s) Status
- Titles Matching: *
- Hostname Matching: *
- LOG Messages Matching: *

LOG Messages

A few words about LOG Messages. As we saw already during the start of a new collect, you can use an optional parameter, Client Log File, to catch during the collect time any new text messages in this logfile. All messages are saved with a time-stamp in the same database as where the collect data is stored. Alternatively, at any moment you may add these kind of messages manually using the web interface. There is a special link "LOG Messages Admin" and under every graph view there is a link to add a new message.

But, when can this be helpful?

Firstly, it'll help you to choose the correct time intervals for analyzing data, without having to remember the exact time slices when something particular happened on this machine.

Secondly, when analyzing the activity on your machine, you'll be able to get a list of every registered event, corresponding to the same time interval.

Example 1

Let's say you DBA in vacations and you're acting for a few days. The user claims that time-to-time something happens on the machine and slows down his work. You're starting to monitor the system, and yes sometimes you observe strange activity on the Oracle side. So, instead to write down the times corresponding to the problem, you simply add two messages: "Something strange" and "Ok now" while you're analyzing activity graphs. Once your DBA comes back, you may just point him to your messages. Also, if somebody else will analyze the time slices, entering the same perimeter, he or she will also be warned by your messages!

Example 2

Every night you're starting some batch jobs while nobody else is working on the system. There are several important parts and you're trying to optimize them or simply check nothing goes wrong.

Let's assume your main batch script is looking like:

```
#bin/sh
start_batch01
start_batch02
start_batch03
start_batch04
...
start_batch20
exit
```

Now, simply add log messages:

```
#bin/sh
echo "Start Night Batch" >> /var/tmp/log
echo "start batch01" >> /var/tmp/log
start_batch01
echo "start batch02" >> /var/tmp/log
start_batch02
echo "start batch03" >> /var/tmp/log
start_batch03
echo "start batch04" >> /var/tmp/log
start_batch04
...
echo "start batch20" >> /var/tmp/log
start_batch20
echo "End Night Batch" >> /var/tmp/log
exit
```

After that, every time you start a new STAT collect to monitor this machine, you give "/var/tmp/log" as Client Log File name. This way, every time you start your main batch script, every message written into /var/tmp/log will be saved and timestamped in the dim_STAT database. To select the correct time interval for analyzing the workload during for example batch04, you only need to simply click between the messages: "start batch04" and "start batch05".

Tasks

There are two special "Task" tags that may be used with log messages:

====> TASK_BEGIN: Unique_Task_Name --Marking begin of task execution

====> TASK_END: Unique_Task_Name --Marking the end

The Unique_Task_Name should be **one word** of up to 40 characters and unique within the current collect. For example, for 4 batches started in parallel we can add to the script:

```
( echo "====> TASK_BEGIN: batch1" >> /tmp/log; batch1.sh; echo "====> TA
( echo "====> TASK_BEGIN: batch2" >> /tmp/log; batch2.sh; echo "====> TA
```

```
( echo "====> TASK_BEGIN: batch3" >> /tmp/log; batch3.sh; echo "====> TA
( echo "====> TASK_BEGIN: batch4" >> /tmp/log; batch4.sh; echo "====> TA
```

When you analyze activity graphs later, you can use the "Show Tasks" button to get a short summary about all the executed tasks during the observed period and with their total execution time (if they are finished). This can be useful in case you're starting big long jobs in parallel. And they are all executed by the same process, so there is no way to know which one is running which job.

Multi-Host Analyzing

Multi-Host analyzing is simpler than Single-Host analyzing and a good point to start.

NOTE: some screenshots may not be 100% up to date and don't match exactly the latest dim_STAT version.

Main point: as we want to see several **hosts** at the same time and on the same graph, we cannot show more than one single stat-value per graph, however there can be several graphs viewed on the same page.

In general:

- ◆ Choose STAT collects
- ◆ Choose the time interval you are interesting in
- ◆ Choose Graph size/mode attributes
- ◆ Choose STAT data you want to analyze
- ◆ Go!! :-)

Select Multi-host

Database: [Bench_PP2] [\[Home \]](#) [\[Preferences \]](#) [\[LOG Admin \]](#)

dim_STAT Analyze

☐ Single-Host Analyze
☒ Multi-Host Analyze

- ☐ Active ONLY
- ☐ Show STAT(s) Status
- Titles Matching:
- LOG Messages Matching:

Choose Collect(s) and Time interval

Database: [Bench_PP2] [\[Home \]](#) [\[Preferences \]](#) [\[LOG Admin \]](#)

dim_STAT Multi-Host Analyzer

	ID	Host	Title	Started	Timeout	State
<input type="checkbox"/>	5	samoa	Test 25K US4+	2006-01-31 22:15:25	30	Finished
<input type="checkbox"/>	6	tahiti	Test 25K US4+	2006-01-31 22:15:28	30	Finished
<input type="checkbox"/>	7	java	Test 25K US4+	2006-01-31 22:15:31	30	Finished

Interval

☐ All data
☐ Last minutes (current time)
☐ Between and
☐ After LOG Message minutes
☒ Between LOG Messages

☐ AVG value(s) by every measurement(s)

Interval

```

[16:43 2006.02.15] samoa: End: 492 CAPS
[12:27 2006.02.21] samoa: End
[16:02 2006.02.21] tahiti: Final 3 x 10 threads
[16:15 2006.02.21] tahiti: Final 3 x 20 threads
[16:30 2006.02.21] tahiti: Final 3 x 30 threads
[16:44 2006.02.21] tahiti: Final 3 x 40 threads
[17:01 2006.02.21] tahiti: Final 3 x 50 threads
[17:14 2006.02.21] tahiti: End
[17:14 2006.02.21] tahiti: Final 1 x 50 threads
[17:25 2006.02.21] tahiti: Final 1 x 60 threads

```

Graphics

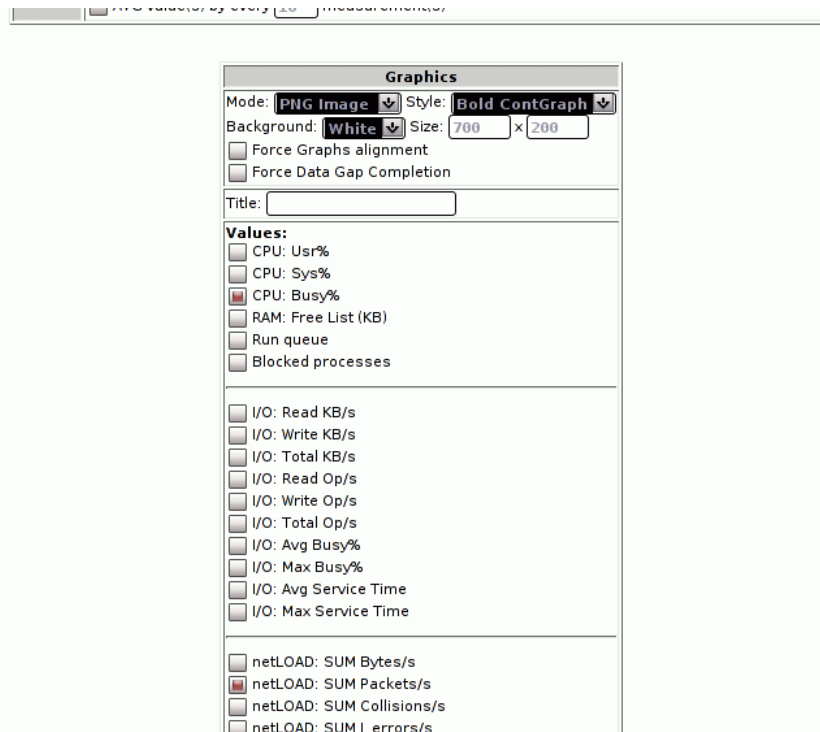
Mode: Style:

Collects - Let's assume there are three hosts I want to see together. OK, these collects are only used as examples and not to give demo data.

Time Interval - I described before the advantages of using LOG messages. Here is one of the better examples. I've simply selected the begin and the end of the time slice I'm interested in for my production workload.

NOTE: you may select **several** intervals and compare them all together on the same graph. For example, to compare today's and last week's activity during a similar workload.

Choose STATs



Graphics - This is a quite intuitive section, isn't it?

You simply choose the style of your graphical presentation:

- ◇ **Java Applet/ PNG Image** - graph output format
- ◇ **Histogram** - one comment: histograms are only supported with Java output.
- ◇ **Real Graph** - in case there was no data during any time period for some stat components, the graph line will be stopped for this period and will continue once this component came back. Example: while collecting, one user was disconnected for a while and re-connected again. So, the graph will represent both "real" activity and "inactive" periods will be represented by holes. The only problem occurs when the observed component switched too often from/to "live"- "dead"- "live" states. In this case, instead of a graph, you may see a set of dots, which isn't much less fun.
- ◇ **Continuous Graph** - as opposed to Real Graph, ContGraph will replace the "holes" by zero. So there will never be "dots" on your graphics and each graph line will stay perfectly continuous. However, there is no more a visual difference between an "inactive" and a "dead" component.
- ◇ **Force Graph alignment** - this is useful only for Java graphs and done automatically for static PNG images.
- ◇ **Force Data Gap completion** - this may help you to see continuity in time scale graphs, when you have short periods of data missing (host reboot, etc.). If you don't use this option, a data hole is made visible by a red vertical

bar in the graph. Be careful with this option, because if your time gap is large (days, weeks, etc.), you may wait for a few hours to get your graph. In the meantime the tool will try to refill all missing data with zero, and you will just see a big hole in the middle of graph.

- **Auto-Sync:** with version 8.1 a new auto-time-sync feature was implemented to avoid the problem of time shift with some Solaris and Linux commands. This is done by automatically re-syncing every hour the collected data with the current time. But the red bar may still be present on your graphs even when there was no stop time on the service, etc.

Finally, to accommodate your preference, there is an option to choose between Normal and Bold lines for drawing your graphs.

Note: all Graphics parameters are saved and kept with cookies. They will be used again the next time you use this function.

Next, you just choose the STAT values you want to see on your graph (example: CPU and Net packets/sec)...

Go!

The screenshot shows a configuration window with the following sections:

- I/O:**
 - ☐ Avg Busy%
 - ☐ Max Busy%
 - ☐ Avg Service Time
 - ☐ Max Service Time
- netLOAD:**
 - ☐ SUM Bytes/s
 - ☒ SUM Packets/s
 - ☐ SUM Collisions/s
 - ☐ SUM I_errors/s
 - ☐ SUM Nocarput
 - ☐ SUM O_errors/s
- oraEXEC:**
 - ☐ Commit/sec
 - ☒ Exec/sec
 - ☐ Sessions

Below these sections are controls for the log display:

- Show **ALL** Host's LOG Messages Matching: Mode: **Static**
- Show Task(s) from **ALL** Host(s)
- Refresh every **60** sec.

At the bottom are **Start** and **Clean** buttons.

Once you set "content" and "presentation", you can also set some other parameters:

Show LOG: In case you want to see LOG messages at the same time as graphs, so that you can analyze better the events that happened. There are also two modes to view logs: Static and Dynamic. In Static Mode all messages are presented inside of a simple HTML table. In Dynamic Mode they are all inserted into a small scrollable window and if you click on any message in that window you will set/unset a red bar

crossing all graphs that correspond to the message timestamp...

Show Tasks: print a table of all running/finished tasks corresponding to the current time period

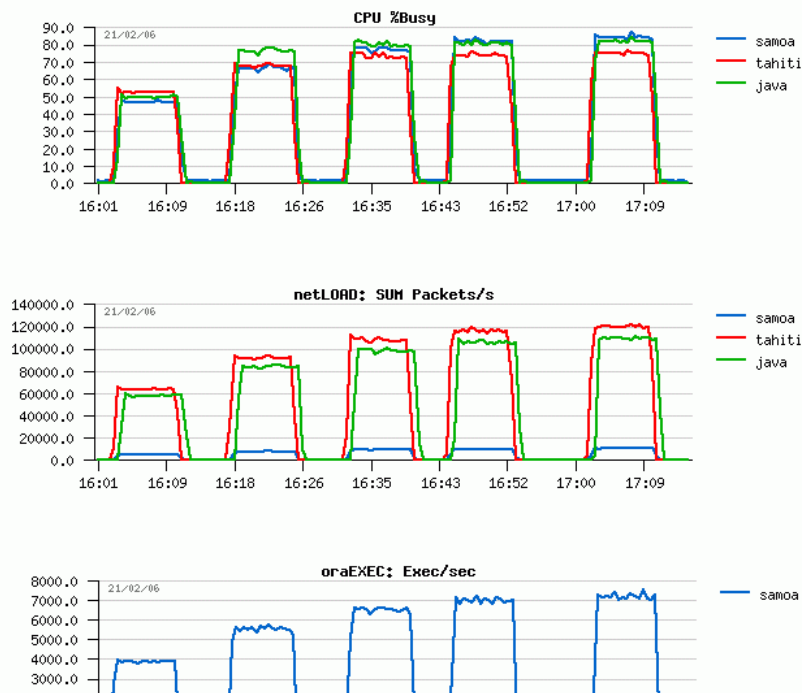
Refresh: this will refresh the result page every number of seconds. A function, very useful for on-line monitoring. You can do the same through browser options in Opera or Firefox)

Let's START!!

Result with Static Log

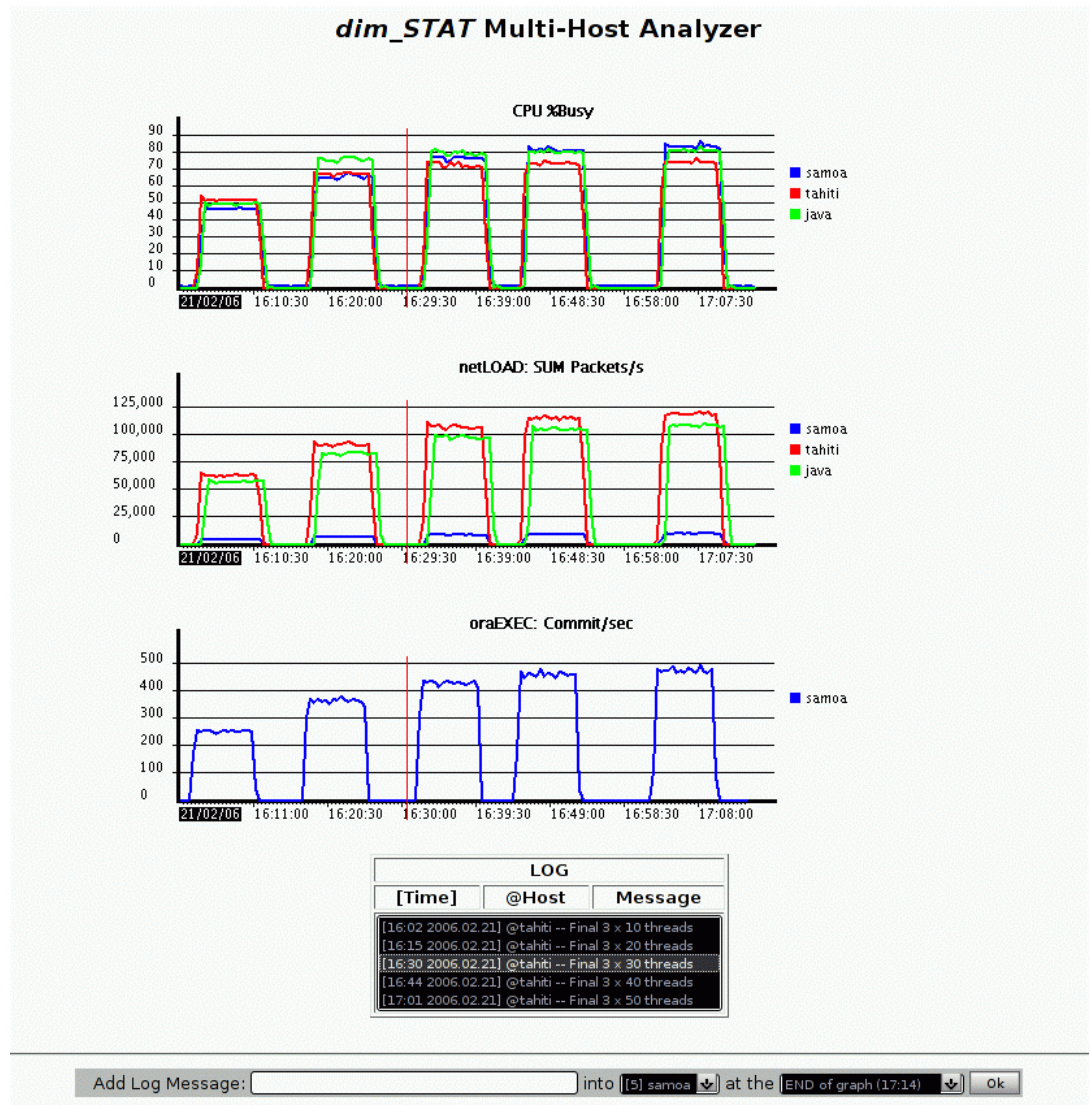
Database: [Bench_PP2] [\[Home \]](#) [\[Preferences \]](#) [\[LOG Admin \]](#)

dim_STAT Multi-Host Analyzer



(Sorry, there was no more place on screen for the LOG :))

Result with Dynamic Log



If you use dynamic logs and applet output, single clicking on a message line will set on / off a vertical red bar on the graph. This bar shows you exactly the place that corresponds with the message timestamps.

As you see, at any moment you may add another Log message.

Single-Host Analyzing

Single-Host Analyzing is very similar to Multi-Host, but gives a wider variety of parameters as it is working only with one particular STAT collect. Let's use as an example the Demo collect, which is provided with the dim_STAT database and let's analyze IOSTAT data.

Open your browser and follow step by step how we're connecting to the dim_STAT server.

Choose Collect and STAT

Database: [Bench_PP2] [\[Home \]](#) [\[Preferences \]](#) [\[LOG Admin \]](#)

dim_STAT Analyzer

	ID	Host	Title	Started	Timeout	State
<input checked="" type="radio"/>	1	goldgate	Demo collect, just to see it's ok!	1998-12-18 16:28:27	15	Finished
<input type="radio"/>	2	samoa	Probe PP2	2006-01-25 15:29:57	30	Finished
<input type="radio"/>	3	tahiti	Probe PP2	2006-01-25 15:30:00	30	Finished
<input type="radio"/>	4	java	Probe PP2	2006-01-25 15:30:03	30	Finished

Use LogFile Messages from: **the same host** Matching: *

Example IOSTAT: Choose Disks criteria

Database: [Bench_PP2] [\[Home \]](#) [\[Preferences \]](#) [\[LOG Admin \]](#)

dim_STAT Analyzer

Demo collect, just to see it's ok! (goldgate)

I/O Statistics Started: 1998-12-18 16:28:27 Timeout: 15 sec.

Disk(s)

☐ c0t0d0
☐ c0t1d0
☐ c0t2d0
☐ fd0

☐ Inversed Selection

☐ Select TOP DISK(s) with MAX-Total Read/s
☐ Accept ONLY data with Read/s =
☐ Use Select Pattern * for DISK

Interval

☐ All data
☒ Last measurements
☐ Between and Measurements (total: 133)
☐ After LOG Message minutes
☐ Between LOG Messages

FIRST collected data
 [16:29 1998.12.18] Oracle query...
 [16:32 1998.12.18] Informix query...
 [16:38 1998.12.18] MySQL query...
LAST collected data

☐ AVG value(s) by every measurement(s)

Values

☒ Per disk
☐ AVG all disk(s)
☐ SUM all disk(s)
☐ Grouped AVG by first/last letters in disk name
☐ Grouped SUM by first/last letters in disk name

Your choice of options is much broader in Single-Host mode. You can analyze your collected data in fine detail, adapting them to your needs...

Disks - several possible combinations, but quite similar to other multi-line STATs

- ◇ nothing selected means using all data without refining your select
- ◇ you may refine your criteria by selecting only certain disk(s)
- ◇ you may exclude your selected disk(s) by clicking the 'Inversed Selection' checkbox
- ◇ you may use value-oriented selection (ex. Top-10 Busy% disks)
- ◇ you may exclude disks with unwanted data values
- ◇ or finally, give a select pattern (very useful if you want to avoid SDS metadevices, etc.)

Interval is similar to Multi-Host analysis. To simplify, let's look at the last 100 measured data per disk (there are only a few).

Values Special Operations - You can analyze on a per disk basis, or SUM/AVG all of them, or group values by the first N characters of the disk name (very useful if you want to analyze I/O activity per controller), or when N is a negative number by the last N characters.

Example IOSTAT: Choose STAT Variables

The screenshot shows the IOSTAT configuration window with the following sections:

- Values:**
 - ☒ Per disk
 - ☐ AVG all disk(s)
 - ☐ SUM all disk(s)
 - ☐ Grouped AVG by first/last letters in disk name
 - ☐ Grouped SUM by first/last letters in disk name
- Top 10 value(s):**
 - ☐ Read/s
 - ☐ Read KB/s
 - ☐ Write/s
 - ☐ Write KB/s
 - ☐ Wait
 - ☐ Actv
 - ☐ Service time
 - ☐ %Wait
 - ☐ %Busy
- Search for:**
- Table of results:**
 - ☐ ID
 - ☐ Sno#
 - ☐ DISK
 - ☐ Read/s
 - ☐ Read KB/s
 - ☐ Write/s
 - ☐ Write KB/s
 - ☐ Wait
 - ☐ Actv
 - ☐ Service time
 - ☐ %Wait
 - ☐ %Busy
- Format:**
- Output:**
- Graphics:**
 - Mode: Style:
 - Background: Size: x
 - ☐ Force Graphs alignment
 - ☐ Force Data Gap Completion
 - Title:
 - Values:**
 - ☐ Read/s
 - ☐ Read KB/s
 - ☐ Write/s
 - ☐ Write KB/s
 - ☐ Wait
 - ☐ Actv
 - ☐ Service time
 - ☐ %Wait
 - ☒ %Busy
 - ☐ **Show Value Distribution:**
 - Value: Scale:
- Show Bookmark-links in the:**
- Show LOG Messages Matching:** Mode:
- ☐ Show TASK(S)
- ☐ Refresh every sec.

At the bottom, there are **Start** and **Clean** buttons.

The data can be presented in three different forms:

- ◇ **Graphics** - graphical representation (as we saw already before)
- ◇ **Table of Results** - the raw data is presented as HTML or Text output (table format) and printed on screen or into a temporary file
- ◇ **Top-N values** - in a few clicks check the MAX/MIN values of any STAT variables during the given time period. For example: if there were no disks busier than 30%, you even don't need to look at graphs, or if there are any, you know at once the time slices you need to analyze for a possible jump in activity.

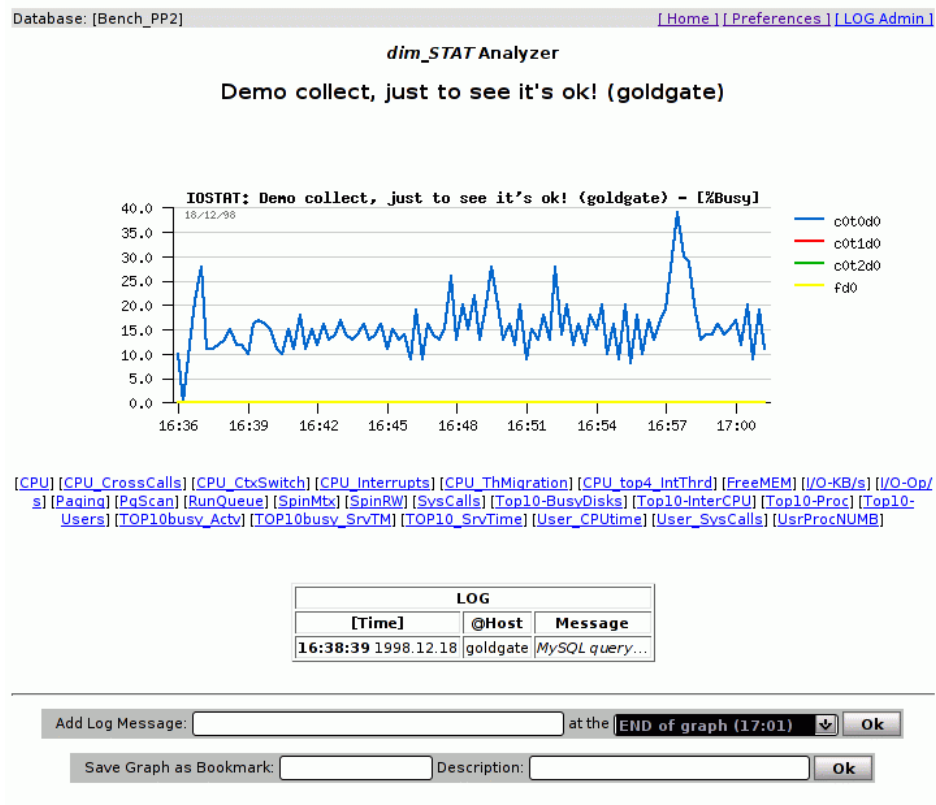
Fine, here I want to see:

- ◇ Graph
- ◇ with disk Busy%
- ◇ and Bookmark Links

A **Bookmark Links** may be inserted at the bottom of every viewed graph. Clicking on one of these links will show you another statistics view for exactly the same time period.

Click "Start" !!

Example IOSTAT: Result Graph



Some new things here.

Under the graph you'll see a list of Bookmark links. If you click on "CPU" (for ex.), a new graph will appear with the CPU activity during **the same** time period you're observing now. This is useful, because even 3 days later will still point to the same time slice.

You'll also find an "Add LOG-Massage" field, the same as with Multi-Host.

And a new one: **Save Graph as Bookmark.**

Save Graph as Bookmark...

This is a really cool feature that will save you time. Right now, you can simply give short and long names for your graph view and save it as a new "Bookmark". Once this is done, all the options you selected will be saved (booked) under the name given. And instead of having to click again on all those checkboxes, to get similar data but for another time period or another STAT collect, all you will need to do is just click on the one button with your "Bookmark Name"!

NOTE : Since v.9.0 there is a possibility to create Bookmarks for Multi-Host Analyzing too! And all Multi-Host stats are Bookmarks since then ;-) -- To be able to create a "Multi-Host Bookmark" just keep in mind that when you're comparing several hosts you cannot bring on the same graph more than one statistic value on the same time! (for ex: you cannot see both Sys% and Usr% CPU usage on the same time without creating a mess in the graph legend, while you're using only Sys% or Usr% you'll need to show only host names within a legend) - so as far as you're generating a graph with a single statistic value and using only generic data filter conditions in the Bookmark form under your graph the choice will be automatically extended by "Multi-Host" option within a select box!

There is a huge benefit to use Bookmarks when you're analyzing many hosts on the same time and on the same graph, for ex:

- ◊ you can follow a fixed list of disk controllers on all servers rather to see a sum of all disks..
- ◊ you can follow CPU usage by selected users/processes on all servers rather a whole CPU usage..
- ◊ and many others ;-)

As well, don't forget to share with others if you're creating new Bookmarks ;-))

Bookmarks

Most of the bookmarks are pre-defined to save your time. Their number may vary from release to release, but never forget, you can always create your own and keep them as your specific kit. And you can easily move them from one base to another.

People very quickly are starting to use only bookmarks and then sometimes they are lost: *"Oh, there is no way to see per network interface activity!"* or, *"no way to see a single process, only top-10!"* But don't forget, all data is there, just go directly to the STAT interface and you'll find them. Then create new bookmarks covering other needs and you're all set.

Choose Collect and click on Bookmarks...

Database: [Bench_PP2] [\[Home \]](#) [\[Preferences \]](#) [\[LOG Admin \]](#)

dim_STAT Analyzer

	ID	Host	Title	Started	Timeout	State
	1	goldgate	Demo collect. just to see it's ok!	1998-12-18 16:28:27	15	Finished
	2	samoa	Probe PP2	2006-01-25 15:29:57	30	Finished
	3	tahiti	Probe PP2	2006-01-25 15:30:00	30	Finished
	4	java	Probe PP2	2006-01-25 15:30:03	30	Finished

Use LogFile Messages from: Matching:

Choose Time interval and Graphics style

dim_STAT Analyzer

Demo collect, just to see it's ok! (goldgate)

Started: 1998-12-18 16:28:27 Timeout: 15 sec.

Interval

☐ All data

☒ Last measurements

☐ Between and

☐ After LOG Message minutes

☐ Between LOG Messages

FIRST collected data

[16:29 1998.12.18] Oracle query...

[16:32 1998.12.18] Informix query...

[16:38 1998.12.18] MySQL query...

LAST collected data

☐ AVG value(s) by every measurement(s)

Measurements (total: 139)

Graphics Style

Mode: Style:

Background: Size: x

☐ Force Graphs alignment

☐ Force Data Gap Completion

Title:

Bookmark(s)

<input checked="" type="checkbox"/> CPU	- CPU %Busy
<input type="checkbox"/> CPU_CrossCalls	- CPU Cross-Calls
<input type="checkbox"/> CPU_CtxSwitch	- CPU Context Switch
<input type="checkbox"/> CPU_Interrupts	- CPU Interrupts
<input type="checkbox"/> CPU_ThMigration	- CPU Thread Migration
<input type="checkbox"/> CPU_top4_IntThrd	- Top-4 CPU on Interrupts
<input checked="" type="checkbox"/> FreeMEM	- Memory Free List(KB)

Select all Data you want to see and GO!

<input checked="" type="checkbox"/>	FreeMEM	- Memory Free List(KB)
<input type="checkbox"/>	I/O-KB/s	- I/O Throughput KB/sec
<input checked="" type="checkbox"/>	I/O-Op/s	- I/O Operations/sec
<input type="checkbox"/>	Paging	- Page In/Out (KB)
<input type="checkbox"/>	PgScan	- Page Scanner Rate (Pg/sec)
<input type="checkbox"/>	RunQueue	- Queued, Blocked, Swapped runnable processes
<input type="checkbox"/>	SpinMtx	- Spin Mutex/sec
<input type="checkbox"/>	SpinRW	- Spin/sec Read/Write
<input checked="" type="checkbox"/>	SysCalls	- System Calls/sec
<input type="checkbox"/>	Top10-BusyDisks	- Top-10 Busy% Disks
<input type="checkbox"/>	Top10-InterCPU	- Top-10 Interrupts CPU
<input type="checkbox"/>	Top10-Proc	- Top-10 Processes using CPU%
<input type="checkbox"/>	Top10-Users	- Top-10 Users using CPU%
<input type="checkbox"/>	TOP10busy_Actv	- TOP-10 Busy% Disks Actv
<input type="checkbox"/>	TOP10busy_SrvTM	- TOP-10 Busy% Disks Service Time
<input type="checkbox"/>	TOP10_SrvTime	- TOP-10 SrvTime Disks
<input type="checkbox"/>	User_CPUtime	- User CPU Time Usr/Sys
<input type="checkbox"/>	User_SysCalls	- User System Calls
<input type="checkbox"/>	UsrProcNUMB	- Proc Number per User

☒ Show Bookmark-links in the **New window**

☒ Show LOG Messages Matching: Mode: **Static**

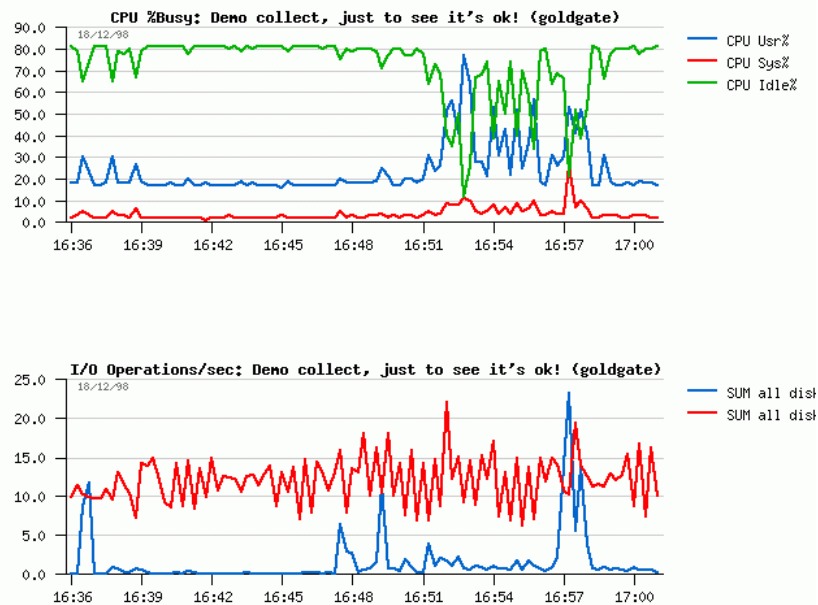
☐ Show TASK(S)

☐ Refresh every sec.

Go!

Result Page

Database: [Bench_PP2]

[\[Home \]](#) [\[Preferences \]](#) [\[LOG Admin \]](#)**dim_STAT Analyzer****Demo collect, just to see it's ok! (goldgate)**

Note: There were a lot of discussions about "Bookmark" as the name for this feature. And I'm quite agreeing that the term is not the best fit to describe the functionality, but the problem is I never received a new name that seemed to please everybody.

So, I've simply decided to put this term on the preferences page. This way, everybody is free to rename "Bookmark" to something else, even to "X-Files". :))

Administration actions

From the "Main Page" you may go directly to the "Bookmarks" management page and

- ◇ Rename
- ◇ Export
- ◇ Import
- ◇ Delete

any Bookmark, as well as Restore the "Standard Kit". This is if you lost your bookmarks for any reason. The standard kit contains some of the more popular data views.

Multi-Host Extended Analyze

Since v.8.5 the Extended Multi-Host Analyze was introduced - it combines the traditional Multi-Host options with per host Bookmarks. Probably the most sophisticated way now to analyze a server performance :-) but it gives you all the needed information grouped on the one single page :-) As well the Bookmarks links are also present now on demand - so at any time you may get a more detailed graphs while analyzing on the Multi-Host :-)

dim_STAT CLI

I was really surprised by the strong demand by users for a dim_STAT CLI solution! It seems a Web interface is not making everybody happy :))

And here we are, with version 8.1 there is a CLI module in dim_STAT :)

```
# /apps/ADMIN/dim_STAT-CLI

dim_STAT CLI v.1.7
Usage: dim_STAT-CLI [options]
Options:
  -Base DBname                (if empty: prints available Collect list)
  -ID CollectID                (if empty: prints available Stat list)
  -Stat Name                   (if empty: prints available Stat list)
  -Begin YYYYMMDDhhmiss
  -End YYYYMMDDhhmiss
  -Out fname

optional:
  -Title graphtitle           (if empty: uses Collect title)
  -Width size                  (if empty: uses default graph width)
  -Height size                 (if empty: uses default graph height)
  -AVG number                  (use average for too wide graphs)
  -Data filename               save also raw stat data into file
```

For the moment it gives you a way to generate a single graph in PNG format for a given Database, CollectID and Time interval. Stat names are corresponding directly to your Bookmarks in your Database, so the more Bookmarks you have, the more graphs you may generate.

Since v.9.0 if you're using several Collect IDs on the same time (ID1,ID2,ID3,...) dim_STAT-CLI will propose you to use Multi-Host stats and draw Multi-Host graphs! ;-))

Example

Check the STAT-collects in database 'EasyLux':

```
$ /apps/ADMIN/dim_STAT-CLI -Base EasyLux
```

```
== Available Collect(s):
ID Host Started Title
```

```
----- 1
goldgate 1998-12-18 16:28:27 Demo collect, just to see it's ok! 2
x4100 2007-03-28 17:01:37 EasySTAT_TMG 4 galaxy3 2007-04-05
```

13:28:41 EasySTAT_CacheON

```
-----
dim_STAT CLI v.1.4 Usage: dim_STAT-CLI [options] Options:
-Base DBname -ID CollectID (if empty: prints available Collect list)
-Stat Name (if empty: prints available Stat list) -Begin
YYYYMMDDhhmiss -End YYYYMMDDhhmiss -Out fname
optional: -Title graphtitle (if empty: uses Collect title) -Width size (if
empty: uses default graph width) -Height size (if empty: uses default
graph height)
```

ERROR: ## Not filled dim_STAT ID!

Get the available Stats for Collect #4:

```
$ /apps/ADMIN/dim_STAT-CLI -Base EasyLux -ID 4
```

== Available Stat(s):

```
CPU -- CPU %Busy CPU_CrossCalls -- CPU Cross-Calls
CPU_CtxSwitch -- CPU Context Switch CPU_ThMigration -- CPU
Thread Migration FreeMEM -- Memory Free List(KB) I/O-KB/s --
I/O Activity KB/sec I/O-Op/s -- I/O Activity Operations/sec
Net_Byte/s -- Network Bytes/sec Net_ByteALL/s -- Network SUM
ALL Bytes/sec Net_Collis/s -- Network Collisions/sec Net_Error/s --
Network Errors/sec Net_Nocanput -- Network Nocanput Net_Pack/s
-- Network Packets/sec Net_PackALL/s -- Network SUM ALL
Packets/sec Paging -- Page In/Out (KB) PgScan -- Page Scanner Rate
(Pg/sec) RunQueue -- Queued, Blocked, Swapped runnable
processes SpinMtx -- Mutex Lock Spin/sec SpinRW -- Read/Write
Lock Spin/sec SysCalls -- System Calls/sec Top10-BusyDisks --
Top-10 Busy% Disks Top10Busy_Actv -- Active Queue @Top-10
Busy% Disks Top10Busy_SrvTM -- Service Time @Top-10 Busy%
Disks Top10Busy_Wait -- Wait Queue @Top-10 Busy% Disks
Top10_ProcCPU -- Top-10 CPU% Usage @Process
Top10_ProcNUMB -- Top-10 Active Processes Top10_ProcSysTM
-- Top-10 CPU SysTime @Process Top10_ProcUsrTM -- Top-10
CPU UsrTime @Process Top10_SrvTime -- Top-10 High Service
Time Disks
```

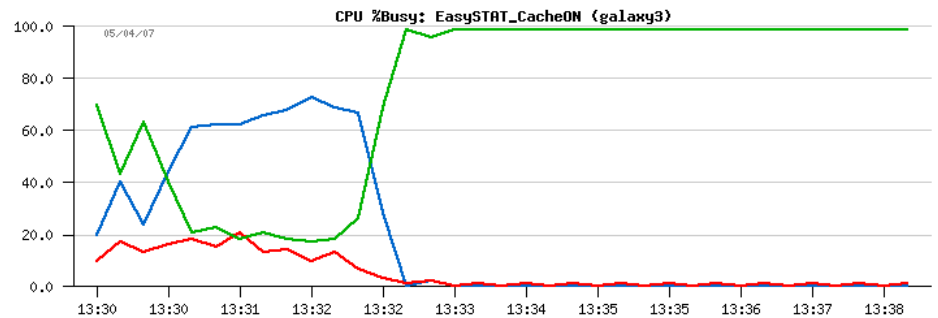
```
dim_STAT CLI v.1.4 Usage: dim_STAT-CLI [options] Options:
-Base DBname -ID CollectID (if empty: prints available Collect list)
-Stat Name (if empty: prints available Stat list) -Begin
YYYYMMDDhhmiss -End YYYYMMDDhhmiss -Out fname
optional: -Title graphtitle (if empty: uses Collect title) -Width size (if
empty: uses default graph width) -Height size (if empty: uses default
graph height)
```

ERROR: ## ## Empty Stat!

Get a CPU Usage graph from Collect #4 between 13:30 and 14:00.

```
$ /apps/ADMIN/dim_STAT-CLI -Base EasyLux -ID 4 -Stat CPU -Begin 20070405133000
[ ]=> CPU %Busy: EasySTAT_CacheON (galaxy3)
$
```

CPU.png



Administration

Several administration points were already covered in previous sections. Let's speak about some other, more oriented on day to day management...

Active/Stopped Collect

Each STAT-collect may be only in 2 states: Active or Stopped.

The state a collector is in is stored in the database. When the state of the collect is changed from the Web interface, the only action is an update of the corresponding database record, that's all. From time to time each collector checks its own record for changes, and if so, it takes corresponding action.

Since v.7.0 at any time any stopped collect may be restarted again.

Active : a collector gets data from the server via the STAT-service, and while the service is up, it continues to insert data into your database. If the STAT-service is down, it will try to reconnect every 20 secs.

Stopped : the collect is stopped as well all the corresponding stat commands on the monitored server. No more data is inserted into the database.

Delete/Recycle Collects

Finished collects can be completely removed from the database, or recycled. You may remove, for example, all data previously collected during the last N days. Actually, only manual recycling is possible.

Note: a delete operation frees space in the database index/data files, but it doesn't reduce the actual file size! Freed-up space will simply be reused for next collects.

Deleting a database was covered previously in "MySQL Admin Tips"...

Auto-Recycle

Since v.8.1 an Auto-Recycle module is integrated into dim_STAT. Well, it still needs to be run from a cron job or another execution planner, but at least, once it's configured, it gives you a simple way to recycle your collected data automatically.

In your '/apps/ADMIN' directory you find the 'dim_STAT-Recycle' command:

```
# /apps/ADMIN/dim_STAT-Recycle
```

```
Usage: dim_STAT-Recycle -Days N [-Base DBname] [-ID CollectID]
      -Days N           -- keep data collected during last N days
      -Base DBname      -- database name(s) (def: Default)
      -ID CollectID     -- collect ids (ex: id1,id2,id3 or "ALL" for any
                          (def: All active collects only)
```

So, to recycle every 24 hours and to maintain in your database 'Prod' only data collected during the last 3 weeks, all you need to do is to add the following to the crontab on your dim_STAT server:

```
0 0 * * * /apps/ADMIN/dim_STAT-Recycle -Days 21 -Base Prod
```

NOTE :

- Days delay is purely by calendar! Recycle will delete all your data from the last collected day to N calendar days back, independent of possible inactivity holes in the collected data
- if no ID is given, only currently **active** collects will be recycled
- if a list of ID is given, all these collects will be recycled independently if they are active or not

- if ID is equal to ALL - all collects will be recycled independently if they are active or not

Export/Import collects

Collect Export and Import is an easy way to save/copy/restore small amounts of data in a compressed form. In case you need to copy a large amount of data, it is much faster to copy the whole database! (This was extensively covered in "MySQL Admin Tips".)

Modify Collect parameters

You should be VERY CAREFUL with these actions!

Changing the Title and Hostname are just for decoration. :))

Changing Collect-ID, which is a global operation, will lock all corresponding tables, while making modifications.

Changing Time Interval makes only sense with wrongly loaded data from output files. Be aware that you're changing your time scale and will loose synchronization with real world events.

Changing Start Time can be used when you want to compare similar workloads, that were collected on different periods. You can bring them onto the same time scale and then analyze via Multi-Host mode. However, if you have any LOG messages corresponding to the same collect, then don't forget to move them also in time to keep timestamp synchronization.

LOG Messages operations

This can be used in case there are too many messages, or that you want to share them with other collects, or when you want to move them slightly in time, etc. You can do all of that and much more via "LOG Messages Admin".

Add-On Statistics

Database: [Default] [\[Home \]](#) [\[Preferences \]](#) [\[LOG Admin \]](#)

dim_STAT - Add-On STAT(s)

Current database: [Default]

[★ Integrate New Add-On-STAT](#)
[📄 Save Add-On STAT\(s\) Description](#)
[📄 Restore Add-On STAT\(s\) Description](#)
[✖ Delete Add-On STAT\(s\)](#)

Add-On STAT(s) -- Standard Kit		
ForkExec	-- ForkExec Statistic(s)	<input type="button" value="Install"/>
LPrclLOAD	-- Linux ProcLOAD	<input type="button" value="Install"/>
LUsrLOAD	-- Linux UserLOAD	<input type="button" value="Install"/>
LcpuSTAT	-- Linux MPSTAT (per CPU) Statistic(s)	<input type="button" value="Install"/>
LioSTAT	-- Linux I/O Statistic(s)	<input type="button" value="Install"/>
LnetLOAD	-- Linux netLOAD	<input type="button" value="-[Ok]-"/>
LpsSTAT	-- Linux psSTAT	<input type="button" value="Install"/>
Lvmstat	-- Linux VMSTAT Statistic(s)	<input type="button" value="Install"/>
MEMSTAT	-- Solaris Memory Related Statistics (vmstat -p)	<input type="button" value="Install"/>
ProcLOAD	-- ProcLOAD per Process Name Load Statistic(s)	<input type="button" value="Install"/>

One of the most powerful features of dim_STAT is the ability to integrate your own statistic programs with the tool. Once added, they will be considered by dim_STAT as being the same as the standard set of STAT(s) and give you the same kind of service: Online Monitoring, Up-Loading, Analyzing, Reporting, etc.

However, the choice of external stat programs is so wide that it's quite impossible to design a wrapper for each and every format. Therefore, I've decided to limit the input recognizer to just 2 formats (which covers maybe 95% of needs) and leave it to you to write, if necessary, your own wrapper and modify the output to one of the supported formats.

Formats supported by dim_STAT:

- **SINGLE-Line:** with one output line per measurement (ex: vmstat)
- **MULTI-Line:** with several output lines per measurement (ex: iostat)

To be correctly interpreted, your stat program should produce a stable output. This means the same format for data lines, at least one line in case of MULTI, keep the time-out interval constant, etc. Lines not containing data have to be declared, so that they can be ignored by dim_STAT.

NOTE: lines shorter than 4 characters are considered as "spam" and will be ignored!

Let's look at some examples...

Example of SINGLE-Line command integration

Let's assume we want to monitor a read/write cache hit on the system. This information can be retrieved using "sar":

```
$ sar -b 1 1000000000000000
```

```
SunOS sting 5.9 Generic_112233-05 sun4u 07/09/2004
```

```
18:10:13 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s
pwrit/s 18:10:14 0 1 100 0 0 100 0 0 18:10:15 0 14 100 0 0 100 0 0
18:10:16 0 7 100 0 0 100 0 0 18:10:17 0 0 100 0 0 100 0 0 18:10:18
0 0 100 0 0 100 0 0 18:10:19 0 135 100 0 0 100 0 0 18:10:20 0 0 100
0 0 100 0 0 18:10:21 0 69 100 0 2 100 0 0 18:10:22 0 86 100 0 2 100
0 0 18:10:23 0 0 100 0 0 100 0 0 18:10:24 0 0 100 0 0 100 0 0
18:10:25 0 0 100 0 0 100 0 0 ...
```

What we are interested in are the "4"-th and "7"-th columns from the sar output, and ignoring any lines containing "*SunOS*" or "*read*".

Folowing the "Integrate New Add-On-STAT" link:

Step 1: FIRST INFO

dim_STAT Integrate New Add-On STAT

STAT Name	CacheHIT (One Word)
Columns per Line	2
Type of output	Single-Line <input type="checkbox"/> per measurement

Let's give the new Add-On the name **CacheHIT**.

We need only **2** columns from the output line (4th and 7th value). This is a "Single-Line" output...

Click on "New"...

Step 2: INTEGRATION

dim_STAT *Integrate New Add-On STAT*

Add-On STAT Name	CacheHIT
Description	Cache %HIT via SAR
Shell Command	sar -b %i 1000000000000000 %h - hostname, %i - interval(sec.), %p - parameter(s)
Ignore Line(s)	Any line from pattern(s): "SunOS" "read"

DB ColumnName	Data Type	Column# in input Line	Short Name	Full Name	Use in Multi-Host Analyze
id	int	-	ID	Stat ID	-
sno	int	-	Sno	#Serie STAT	-
column3	Float	4	%rcache	Read %HIT Cache	Yes
column4	Float	7	%wcache	Write %HIT Cache	No

During this step we need to explain what we want to run and which information we'll need:

Description: CacheHIT via SAR

Shell Command: sar -b %i 1000000000000000

- During execution of sar %i will be replaced with the time interval in seconds.
- The command name doesn't matter here because it is only used as an alias for STAT-service. Have a look at the "access" file section, it's possible to name the shell command "toto" and put in it /usr/bin/sar as an alias.

Ignore Lines: we should ignore any lines containing "*SunOS*" or "*read*"

Data Descriptions:

- **ColumnName** - leave it as it is, if you don't need to access the database directly. Note: there are 2 reserved columns for Collect-ID and measurement No.
- **Data Type** - if you're not sure, set it to "Float", otherwise it will be "Int"
- **Column# on input** - in our case we need columns 4 and 7
- **Short Name** - single word descriptions, here %rcache and %wcache
- **Full Name** - description to be used where detailed information is needed
- **Use in Multi-Host** - if you choose "Yes" the corresponding value will be automatically enabled in Multi-Host mode for analyzing of several hosts at once.

You should be VERY CAREFUL with these actions!

Create!!

Created!

What's Next? Will it work now?

Yes! **IF YOU DID NOT FORGET** to give your STAT-service access to this new command! This is a very common error.

If you want to collect "CacheHIT" data from server "S" be sure that the STAT-service on "S" is given execution permissions for the "sar" command. Add the following lines to your /etc/STATsrv/access file:

```
# CacheHIT Add-On
command sar /usr/sbin/sar
#
```

And **now** it'll work! :-))

NOTE: for security reasons and for a cleaner "stat to command" relationship, it is preferable to create for our new add-on a specific script 'CacheHIT.sh', and then use that instead of the direct access to the 'sar' command.

Example:

```
$ cat /etc/STATsrv/bin/CacheHIT.sh
#bin/ksh
exec /usr/sbin/sar -b $1 10000000000000000
```

```
$ CacheHIT.sh 5 ...
```

```
$ tail -3 /etc/STATsrv/access # CacheHIT Add-On command
CacheHIT /etc/STATsrv/bin/CacheHIT.sh #
```

And the Add-On shell command needs to be changed to: "CacheHIT %i"

Anti-Spam Filter

IMPORTANT: There is an anti-spam filter feature, that is always active during data collecting. It rejects any input line having shorter than 4 characters in length. If your newly made stat command prints only one small column of numbers, you need to add leading spaces to take care that the data is accepted by dim_STAT.

MULTI-Line Add-On command integration

Multi-Line integration is quite similar to Single-Line, except few additional things:

You should be VERY CAREFUL with these actions!

- ◇ **Line Separator pattern:** this is by default "new-line", but in some cases it can be a header (like iostat)
- ◇ **Attribute Column:** very important! As you have several lines per measure you need to distinct these by something (like the "diskname" column in iostat).
- ◇ **Use In Multi-Host:** is more than simply Yes/No, you should use SUM and/or AVG for collected values.

REAL LIFE EXAMPLE...

To probably even better feel a new Add-On integration process in dim_STAT, let me tell you a one real life story happened this year with one of our customers..

So well, once understood with dim_STAT what goes on the system and storage, customer also decided to bring more light on what is going wrong (or well) on their application too (finally)..

Initially they wrote a lot of debug messages into their log files, but nothing useful really to understand what's going wrong.. Also, more data they wrote to the log files - more slower worked application :-) normal, no? So, as the first step they simplified logging and got a single file: **/var/tmp/appstats.log**. Every N seconds a new line was added into this file and containing just 3 numbers, and the las one (we're interested in) is an avg TPS during the last time period (M seconds (bigger vs N)):

```
# tail -5 /var/tmp/appstats.log
10:17 5 20
10:20 7 30
10:23 2 50
10:26 8 30
10:30 1 10
#
```

And then customer is creating a simple monitoring script **AppStats.sh**:

```
# AppStats.sh 5
10
50
40
20
30
^C
#
```

In few minutes customer integrated this new stat command as dim_STAT Add-On, but... 15 minutes later it still did not collect any data...

WHY?...

Common Error #1

The first problem: the output **line is very short!** and lines shorter than 4 characters are ignored by anti-spam filter (as mentioned before)! All we need is just to add 3 blank characters in the begin of the line.

Let' get a look on the script source:

```
#bin/bash
#=====
# AppStats
#=====
while true
do v
  tail -1 /var/tmp/appstats.log
  sleep $1
done | awk '{ printf( "%d\n", $3 ) }'
```

Just add 4 spaces into {printf("%d\n", \$3)} before %d and it'll be ok!

```
#bin/bash
#=====
# AppStats
#=====
while true
do
  tail -1 /var/tmp/appstats.log
  sleep $1
done | awk '{ printf( "    %d\n", $3 ) }'
```

The script output now is:

```
# AppStats.sh 5
  10
  50
  40
  20
  30
^C
#
```

Common Error #2

But that's not all! It'll still not work!...

Why?.. - the output of this script is not regular yet!...

To check it (as well with any other script) just execute it in the same way but piped to the 'more':

```
# AppStats.sh 5 | more
```

... 10 minutes later there will be still no any output!... - and it exactly what's happening when STAT-service is trying to send data to the dim_STAT server

You should be VERY CAREFUL with these actions!

via process pipe...

What is wrong here?.. - the problem is inside of the script its output is self-piped into 'awk' program, and 'awk' itself is not flushing its output - data will stay buffered until the whole 'awk' buffer is not filled.. and only then data will be flushed to the pipe...

How to fix it?.. - add fflush instruction into the script (depending on 'awk' version) - change the script in way to have 'awk' call inside of the loop

Updated script :

```
#bin/bash
#=====
# AppStats
#=====
while true
do
    tail -1 /var/tmp/appstats.log | awk '{ printf( "      %d\n", $3 ) }'
    sleep $1
done
#=====
```

As 'awk' is finished on each loop passing, data will be always flushed and entered into the pipe with each iteration.

Continue improvement...

So well, customer copied the new script into /etc/STATsrv/bin on all needed servers and added into the end their /etc/STATsrv/access files:

```
# AppStats add-on
command AppStats /etc/STATsrv/bin/AppStats.sh
```

On the dim_STAT the Add-On was integrated as:

- Single-Line
- name: AppStats
- 1 column
- shell command: "AppStats %i"
- value: integer, 1st position, name: TPS

And we started to collect some first data...

Within first 40 minutes, once customer fully enjoyed to graph their application TPS levels, one of the developers said it will be fine to see on the same time an avg response time!.. And within one hour they extended their log file line with additional value showing avg RespTM.

The new script showing one value more:

```
#bin/bash
```

```
#=====
# AppStats
#=====
while true
do
    tail -1 /var/tmp/appstats.log | awk '{ printf( "      %d  %d\n", $3, $4
    sleep $1
done
#=====
```

And we reintegrated again the same script but describing now **2 columns** from output. And it worked just fine!..

Should I say during the next few hours they already wanted to add 3 other new columns! :-))

And finally...

Finally it was hard for developers to decide how many stat values they will need on each server, because it depends on application deployment as well on server role.. So, they understood how to extend their script with any other values, but preferred to avoid Add-On integration step every time they added a new value into their log file..

Well.. nothing impossible :-)

The only way to have "dynamic" stat list is to improve AppStats script in way it working like a Multi-Line stat command (like 'iostat' may show more or less disks according your server configuration)..

The idea is simple, this output:

```
# AppStats.sh 5
      TPS  AvgTM  Users  Active
      30    20    200    40
      40    20    200    50
^C
#
```

into multi-line:

```
# AppStats.sh 5

Name      Value
TPS        30
AvgTM      20
Users      200
Active     40

Name      Value
TPS        40
AvgTM      20
Users      200
Active     50
^C
#
```

And according to needs, log file may contain on the same time the value names, as well values itself:

```
# tail -2 /var/tmp/appstats.log
11:12 33 TPS 30 AvgTM 20 Users 200 Active 40
11:22 33 TPS 40 AvgTM 20 Users 200 Active 50
```

The new script version:

```
#bin/bash
#=====
# AppStats
#=====
while true
do
    echo " Name          Value"
    tail -1 /var/tmp/appstats.log | awk '{ printf( " %-8s  %3d\n %-8s  %3d\n",
        $3, $4, $5, $6, $7, $8 ) }'
    sleep $1
done
#=====
```

This scripts may be integrated now as Multi-Line Add-On, having 2 columns on the output... And even if script will be extended again with other values - they will just extend a list of lines with names and values.

Pre-Integrated Add-Ons

To make your life easier, there are several additional already pre-integrated stat programs (Oracle, Java, Linux, etc).

They are all **already installed by default** in your dim_STAT server, **BUT!** not all of them enabled in your STAT-service by default - only commands not needing any additional checking are enabled!...

As a rule, check first if the add-on works correctly, by starting it directly from the STAT-service bin-directory on the client side (/etc/STATsrv/bin), and only then enable it via **access** file (usually a simple uncomment in /etc/STATsrv/access)...

ProcLOAD / UserLOAD

There are 2 additional psSTAT wrappers:

- ProcLOAD: all output information on-the-fly summarized by process name
- UserLOAD: all output information on-the-fly summarized by user name

These stats are very useful when you have hundreds or thousands of running processes and you want to study groups of processes or users, instead of the activity of a single process.

Example of output :

```
# /etc/STATsrv/bin/ProcLOAD.sh 5
```

PNAME	NTOT	NACT	UsrTM	SysTM	%CPU	VSZ	SYSC	NLWP	VC
STATcmd	312	58	0.00	0.00	0.0	594112	1472	312	
WebX.mySQL	312	58	0.70	0.04	3.4	1142968	8307	312	
fsflush	1	1	0.00	0.03	0.4	0	0	1	
httpd	7	1	0.00	0.00	0.0	18008	10	7	
in.rlogind	1	0	0.00	0.00	0.0	2240	0	1	
inetd	1	1	0.00	0.00	0.0	5304	1	4	
init	1	0	0.00	0.00	0.0	2400	0	1	
java	2	2	0.00	0.00	0.1	455448	255	50	
mysqld	1	1	0.24	0.12	2.0	62216	21258	315	1
nfs4cbd	1	0	0.00	0.00	0.0	2360	0	2	
picld	1	1	0.00	0.00	0.0	4632	33	6	
psSTAT64	1	1	0.02	0.08	0.3	5856	5006	1	
rpcbind	1	0	0.00	0.00	0.0	2880	0	1	
sendmail	2	1	0.00	0.00	0.0	15456	10	2	
svc.startd	1	1	0.00	0.00	0.0	10200	9	13	
syseventd	1	0	0.00	0.00	0.0	2552	0	14	
ttymon	2	0	0.00	0.00	0.0	4648	0	2	
utmpd	1	1	0.00	0.00	0.0	1280	0	1	
vold	1	0	0.00	0.00	0.0	2912	0	6	
wrapper-solari	1	1	0.00	0.00	0.1	3040	237	2	1
xntpd	1	1	0.00	0.00	0.0	2320	25	1	
ypbind	1	0	0.00	0.00	0.0	2360	0	1	
^C									

Special Solaris 10: ZoneLOAD / PoolLOAD/ TaskLOAD/ ProjLOAD

Four psSTAT_10 wrappers were added, that are specific to Solaris 10 and later:

- **ZoneLOAD** : all output information on-the-fly grouped by zone id
- **ProjLOAD** : the same, but grouped by project id
- **TaskLOAD** : the same, but grouped by task id
- **PoolLOAD** : the same, but grouped by pool id

These stats give you more extended information comparing to the standard 'prstat'.

Following some more details about output columns (given for ZoneLOAD, but valid for others too :-))

ZoneLOAD.sh - a shell script wrapper for psSTAT command to collect all data pre-grouped per Solaris Zone (psSTAT option: -M zone). Description of

values printed per zone (each value is printed per a given time period):

- N_total -- current number of all processes running within a zone
- N_activ -- current number of processes being *active within a zone per a given time period
- UsrCPU -- total User CPU *time consumed within a zone per a given time period
- SysCPU -- total System CPU *time consumed within a zone per a given time period
- CPU% -- percent of CPU Busy% within a zone - this value will depend on were or not some CPU assigned to the zone, so it's still better to monitor a CPU% usage within a zone via "vmstat" command!
- VSize -- total "virtual memory size" in KB of all processes running within a zone (be aware each process within its VSZ value may already include several shared libraries or shared memory segments (SHM), and these *same* shared objects may be accounted several times within a total VSize...
Currently there is no any "simple" way to say you how much memory is used by a group of processes (for ex. Oracle processes, etc.) - even there is still possible to write a script which will account each shared object only once, such script will use a significant amount of CPU time..
So, nobody is perfect, but there is a room for improvement! :-))
- SysCalls -- total number of all system calls/sec within a zone
- N_lwp -- current number of LWP (kernel threads) running within a zone
- Vol_CTX -- total number of all voluntary context switch/sec within a zone
- InVol_CTX -- total number of all involuntary context switch/sec within a zone
- Sigs -- total number of all signals/sec within a zone
- I_Blks -- total number of all input I/O blocks/sec within a zone
- O_Blks -- total number of all output I/O blocks/sec within a zone
- IO_Chrc -- total number of all I/O character operations/sec within a zone

The last 3 values are very curious :-)) because on time I've needed it I did not find any document describing what they are meaning, so I've based my naming on the description given within a /proc structure header files - these values are helping in some cases without involving any DTrace script to understand which process (or Zone in the current case) is doing more I/O operations than others...

netLOAD

The netLOAD wrapper is to monitor Solaris network activity. This tool is already for a long time included into dim_STAT's STAT-service. And since v.8.0, netLOAD monitors all network interfaces present in the system (including virtual and loopback). If some indicators are not populated by device drivers, a '-1' value is presented instead. Also, a new '-I' option is added: You may give a fixed list of network interfaces you want to monitor (run '/etc/STATsrv/bin/netLOAD' for more details). In STAT-service, netLOAD is integrated via a 'netLOAD.sh' script, to provide an easy way to change an option.

Example of output :

```
# /etc/STATsrv/bin/netLOAD.sh 5
Name          IBytes/s      OBytes/s      Ipack/s      Opack/s      Ierr/s      Oerr/s
lo0            -1.0          -1.0          0.4           0.4           0.0
ce0           26300.6       3840.0        105.2         64.0           0.0
ce1             0.0           0.0           0.0           0.0           0.0

Name          IBytes/s      OBytes/s      Ipack/s      Opack/s      Ierr/s      Oerr/s
lo0            -1.0          -1.0          0.8           0.8           0.0
ce0           27624.4       2688.0        77.2          44.8           0.0
ce1             0.0           0.0           0.0           0.0           0.0
```

UDPstat

The UDPstat is a wrapper around of "netstat -s" command on Solaris, and made to monitor a UDP traffic on the system. While it's printing all main counters (In/Out traffic, In/Out errors), it's particularly interesting to analyze Input Overflows (and Input Checksums as well). option.

Example of output :

```
# /etc/STATsrv/bin/UDPstat.sh 5

UDP-stat          Tot#      Delta      Val/s
udpInDatagrams    65700      0          0.00
udpInErrors        0           0          0.00
udpOutDatagrams   68321      0          0.00
udpOutErrors       0           0          0.00
udpNoPorts        3514281    0          0.00
udpInCksumErrs     0           0          0.00
```

```

udpInOverflows          0          0          0.00
none  0 0 0

UDP-stat                Tot#    Delta    Val/s
udpInDatagrams          65900      200      40.00
udpInErrors              0         0       0.00
udpOutDatagrams         68321         0       0.00
udpOutErrors            0         0       0.00
udpNoPorts              3514281        0       0.00
udpInCksumErrs          0         0       0.00
udpInOverflows          0         0       0.00
none  0 0 0

```

HAR

HAR - is the Hardware Activity Reporter tool for Solaris 8 and up. Starting with Solaris 8, Sun had begun to deliver public interfaces for the SPARC and x86 hardware performance counters --libcpc, to access CPU counters and libpctx, to track a process. HAR differs from other tools in the fact that it combines the low-level counts into higher-level metrics more useful to application programmers. Application programmers are typically interested in the following metrics: CPI, FLOPS, MIPS, address bus percentage utilization, cache miss rates, branch and branch miss rates, and stall rates. These metrics help in assessing the fair usage of available processing units, locating bottlenecks and guiding tuning efforts, when needed...

Check [this valuable article](#) to discover everything about this powerful tool!..

· **NOTE** : by default HAR add-on is disabled within a Solaris STAT-service, why? - to get a CPU counters data Solaris library functions requiring an exclusive access to the chip - for a very short time, but exclusive anyway - so any other process running on the requesting CPU will be moved to another CPU and get some unwanted side effects.. That's why I'm not suggesting to run HAR for a long period on your production system until you're not fully understanding how it works..

Oracle Add-Ons

NOTE : Originally all these scripts were made as examples to show how easily we may collect data even from Oracle. But with a time people started to use them more and more (while I still expected, inspired by examples, they'll add something more optimal :-)). For example, current scripts all the time connecting/disconnecting to/from the database, and collector keeping connection opened will be more optimal, etc... But well - it's still better then nothing! :-))

Anyway, all following wrappers are needing a correctly setting of Oracle environment for the "Oracle" user. By default the user's name is **oracle** , but

it may be changed inside of the scripts.

It means that:

```
# su - oracle -c "sqlplus /nolog"
```

should work correctly and give you a **SQL>** prompt for the right database instance.

Then you may check that:

```
# /etc/STATsrv/oraEXEC.sh 5
```

prints you the current number of Oracle sessions and current exec/commit activity.

If it doesn't work - fix it before to go further :-)) (BTW, there is a dim_STAT user group where you may always ask questions - <http://groups.google.com/group/dimstat>)

Oracle Add-Ons:

- **oraIO** : Oracle I/O stats for data/temp files
- **oraEXEC** : Oracle SQL QueryExecutions/sec, Commits/sec, Number of Sessions
- **oraLATCH** : Oracle latch stats
- **oraSLEEP** : Oracle latch sleeps stats
- **oraENQ** : Oracle enqueue stats

By default all these Add-Ons are already enabled within dim_STAT database, and all you need is just to uncomment them within a STAT-service access file (/etc/STATsrv/access) and start a new collect including Oracle stats :-))

And of course you may add any other one. Some people even collect statspack reports directly into dim_STAT!

MySQL Add-Ons

mysqlSTAT - is monitoring a "show status" output. Each output variable is presented with 3 values:

- current value of a variable
- delta between current and previous value
- value of delta/sec

And it's up to you to choose from the list of variables what kind of information you're interesting in :-)) To work properly this add-on needs to be configured - edit your /etc/STATsrv/bin/mysqlSTAT.sh file to setup user/password and host/port information.

mysqlLOAD - is oriented multi-host monitoring and presenting a compact list of data from "show status" output:

- On -- MySQL Server On-Line flag (0 or 1)
- Sessions -- number of currently connected user sessions (threads)
- InnDirty -- amount of dirty pages in InnoDB
- InnFree -- amount of free pages in InnoDB
- KeyDirty -- amount of dirty pages in MyISAM Key buffer
- OpFiles -- number of currently open files
- OpTables -- number of currently open tables
- ByteRx/s -- received bytes/sec via network
- ByteTx/s -- sent bytes/sec via network
- Commit/s -- number of COMMIT requests/sec
- Delete/s -- number of DELETE requests/sec
- Insert/s -- number of INSERT requests/sec
- Select/s -- number of SELECT requests/sec
- Update/s -- number of UPDATE requests/sec
- InnDsy/s -- InnoDB Data Sync/sec
- InnDrd/s -- InnoDB Data Read/sec
- InnDwr/s -- InnoDB Data Write/sec
- InnLwr/s -- InnoDB Log Write/sec
- InnLsy/s -- InnoDB Log Sync/sec
- Key_Rd/s -- MyISAM Key Read/sec
- Key_Wr/s -- MyISAM Key Write/sec
- Query/s -- Query/sec execution
- AbrtClnt -- aborted clients (delta)
- AbrtConn -- aborted connections (delta)
- Connects -- number of recent connects (delta)
- SlowReqs -- number of slow requests (delta)
- TabLckWt -- table lock waits (delta)
- Rollback -- called rollbacks (delta)

This add-on also needs to be configured to work properly - edit your /etc/STATsrv/bin/mysqlSTAT.sh file to setup user/password and host/port information.

innodbSTAT - is monitoring a "show innodb status" output (or "show engine innodb status" since MySQL 5.5). Working similar to "mysqlSTAT", but list of variables is based on InnoDB status only. To work properly this add-on needs to be configured - edit your /etc/STATsrv/bin/innodbSTAT.sh file to setup user/password and host/port information.

innodbMUTEX - is monitoring a "show mutex status" output (or "show engine innodb mutex" since MySQL 5.5). Printing the InnoDB MUTEX related stats, already ready to print not only "waits" (as a standard), but also

more detailed data (available via compiling of InnoDB with debug options or just hacking (like counters, spins, real waited time on each mutex, etc.)). To work properly this add-on needs to be configured - edit your `/etc/STATsrv/bin/innodbMUTEX.sh` file to setup user/password and host/port information.

Example of output :

```
# /etc/STATsrv/bin/innodbMUTEX.sh 5
```

MUTEX	count	count/s	spin_waits	spin_waits
db-server-online	1	1	1	1
buf/buf0buf.c:1122	-1	-1	-1	-1
fil/fil0fil.c:1535	-1	-1	-1	-1
srv/srv0srv.c:973	-1	-1	-1	-1
combined_buf/buf0buf.c:818	-1	-1	-1	-1
log/log0log.c:830	-1	-1	-1	-1
btr/btr0sea.c:181	-1	-1	-1	-1
combined_buf/buf0buf.c:820	-1	-1	-1	-1

MUTEX	count	count/s	spin_waits	spin_waits
db-server-online	1	1	1	1
buf/buf0buf.c:1122	-1	-1	-1	-1
fil/fil0fil.c:1535	-1	-1	-1	-1
srv/srv0srv.c:973	-1	-1	-1	-1
combined_buf/buf0buf.c:818	-1	-1	-1	-1
log/log0log.c:830	-1	-1	-1	-1
btr/btr0sea.c:181	-1	-1	-1	-1
combined_buf/buf0buf.c:820	-1	-1	-1	-1

^C

NOTE: the -1 is printed if information is not available.

innodbIOSTAT (deprecated, works only with old InnoDB) - is an adoption of DTrace script [published by Neel](#) but with one additional feature: it detects automatically if mysqld is not running anymore or started/restarted again. And of course you may run it only on the system supporting DTrace :-)

PostgreSQL Add-Ons

pgsqlSTAT is monitoring a "pg_stat_bgwriter" and "pg_stat_database" output. Each output variable is presented with 3 values:

- current value of a variable
- delta between current and previous value
- value of delta/sec
- some values are also presented per database name

And it's up to you to choose from the list of variables what kind of information you're interesting in. To work properly this add-on need to be configured - edit `/etc/STATsrv/bin/pgsqlSTAT.sh` file to setup user/password and host/port information.

pgsqlLOAD is oriented multi-host monitoring and presenting a compact summary (single line) from "pg_stat_bgwriter" and "pg_stat_database" output:

- On -- Server On-Line flag (1/0)
- Sessions -- number currently connected user sessions (backends)
- Commit/s -- number of executed COMMITs/sec
- Rollback -- number of executed rollbacks (delta)
- B_Read/s -- Block reads/sec
- B_hit/s -- Block read hit/sec
- RowSnd/s -- Rows sent/sec
- RowFch/s -- Rows fetched/sec
- RowIns/s -- Rows inserted/sec
- RowUpd/s -- Rows updated/sec
- RowDel/s -- Rows deleted/sec
- ChpTimed -- Checkpoints involved by timeout (delta)
- ChptReqs -- Checkpoints involved by request (delta)
- probably out of checkpoint segments
- BuffChpt -- Buffers written by checkpoint (delta)
- BufClean -- Buffers cleaned by background writer (delta)
- MxWClean -- number of times Max Written level was reached by background writer (delta)
- BufBkend -- Buffers written by backends (delta)
- BufAlloc -- Allocated buffers (delta)

Please, read an excellent howto written by Greg Smith to see how analyze this data -

<http://www.westnet.com/~gsmith/content/postgresql/chkp-bgw-83.htm>

To work properly this add-on also need to be configured - edit /etc/STATsrv/bin/pgsqlLOAD.sh file to setup user/password and host/port information.

jvmSTAT

This is a wrapper to bring information from the "jvmstat" package. This jvmstat is now officially integrated with the JVM 1.5 distribution or later (and called "jstat" now). The jvmSTAT wrapper is giving a way to monitor **ALL** running JVMs on your server on the same time!

To run jvmSTAT properly you need first of all to have jdk 1.5 (or later) installed on your host and check it works correctly on your server:

```
# cd /usr/jdk15/bin
```

```
# jps
...
#
```

If you don't see your running JVM(s) within "jps" output - try to fix it first before continue on next steps :-) - normally it should work with any JVM since Java version 1.4.2.

To get the 'jvmSTAT.sh' wrapper working:

- edit the /etc/STATsrv/bin/jvmSTAT.sh file (from STAT-service) on each client machine, to set the right path environment for **JAVA_HOME** pointed to the jdk 1.5 home. (ex: JAVA_HOME=/usr/jdk15)
- enable jvmSTAT in STAT-service on each client (uncomment jvmSTAT in /etc/STATsrv/access file)
- before starting any new collect, including jvmSTAT, be sure that the jvmSTAT Add-On is already installed (Add-On interface from Main Page)

Then start to collect JvmSTAT data :-)

jvmGC

This one still exists, but I don't see any reason why anyone would still use it, jvmSTAT is the better solution for any kind of "GC" collection.

This wrapper collects on-the-fly information about GC (garbage collector) activity of any JVM running with the "-verbose:gc" option. Before JVM 1.4.2 the only possible way to get information on the GC activity of the standard JVM was dump of the log output, so this wrapper is simply based on log file scanning.

Usage: If you want to see GC activity of one of your JVMs, running on server "J".

0) Install "jvmGC" via the Add-Ons page.

1.) jvmGC uses the \$LOG file for data input (you may change name and permissions according to your needs (default filename: /var/tmp/jvm.log), modify if needs on the server "J" STAT-service side (/etc/STATsrv/bin).

2) use the web interface to start the collect including "jvmGC"

3) on server "J" add the "-verbose:gc" option to java in your starting application script and redirect output into the application log file (for ex. app.log)

- 4) once you want to monitor your JVM:
`$ tail -f app.log | /etc/STATsrv/bin/grepX GC >> /var/tmp/jvm.log`
- 5) observe jvmGC output data and have fun!

LINUX specific STATS

Linux Add-Ons:

- **LvmSTAT** (Linux vmstat)
- **LcpuSTAT** (Linux mpstat)
- **LioSTAT** (Linux iostat)
- **LnetLOAD** (Linux netLOAD)
- **LpsSTAT** (Linux psSTAT)
- **LprcLOAD** (Linux ProcLOAD)
- **LusrLOAD** (Linux UserLOAD)

For details, see the following special Linux note...

Administration tasks

At any moment you can:

Edit Add-On Description - in case you make a mistake in any value name, or in a shell command corresponding to your Add-On you may quickly repair it via Edit interface (however you cannot change anymore MySQL table column names or datatypes - if the error was here, you're better to recreate this Add-On one again ;-))

Save Add-On Description - this will give you an ASCII text file which may be reused for another database. This way you may share with others any new findings and any new tools you found useful!

Restore Add-On Description - from information on a given Description file, re-create all Add-On required database structures and fill all information required for it to function correctly. **WARNING:** if you're already using the same Add-On in the current database, all previous data will be destroyed!

Delete Add-On - removes the Add-On and all corresponding data from the current database...

Linux Special Notes

You should be VERY CAREFUL with these actions!

I don't know if it will surprise you that all dim_STAT binaries for Solaris SPARC until now were compiled on the same old and legendary SPARCstation-5, which runs Solaris 2.6 and that they still work on every next generation Sun SPARC machines. This includes the last generation, and Solaris 10. Some unchanged binaries are still here and are even 10 years old! This is calling a TRUE binary compatibility! :))

Now, can I say the same thing about Linux??? Sometimes, even the same vendor breaks binary compatibility between previous and next distributions!

Because the main problem lies with the different implementations of shared libraries, I've recompiled all main dim_STAT programs as static binaries to be sure they will run on every distribution. Over time, things got worse: static binaries are core dumping on some distros. Therefore, the current dim_STAT Linux version ships with both dynamic and several static versions of the same binary generated on the different distros.

dim_STAT reported to work out-of-the-box on MEPIS 3.3.1-1, MEPIS 6.0/7.0, Debian 3/4, RHEL 4.x/5.x, CentOS 4.x/5.x, OEL 5.x/6.x, SuSE 9/10/11/12, Fedora Core. Anyway, if you encounter any problems during installation or execution of dim_STAT, please, contact me directly and we'll try to fix the issue together. Last years many Linux vendors have stopped even to ship system libraries to run 32bit programs on their 64bit distributions.. - keep it in mind if you're planning to install dim_STAT on a 64bit Linux, you may will need to add 32bit packages then like: glibc.i686 / libc6-i386, libzip.i686/ lib32z1, libX11, libssl, libcrypto, libpng12, libjpeg, .. (check for some discussions on the dim_STAT Users Group @Google: <http://groups.google.com/group/dimstat>)

NOTE: PC boxes are quite cheap nowadays. So rather than trying to fix issue after issue, ask yourself if buying a \$300 PC, installing MEPIS-6.0 or openSUSE-11.2 32bit on it (10 minutes), installing dim_STAT (5 minutes) and starting the collection of stats from all your servers, will not be a cheaper, easier and simpler solution.

And Again: why you simply don't use Solaris/OpenSolaris and just avoid all such kind of problems?... :-) There is even Pocket Solaris available (<http://milax.org>) - 300MB full install + 60MB dim_STAT = all other disk space to use securely with ZFS and collect data from your servers!... Seriously...

Linux STAT-service

While there is in general no problem with the stat programs for Solaris, there are always a lot of questions about Linux stats integration.

Keep in mind: The most important part of collecting stats from a Linux box is a working STAT-service! If it starts on your box, you may integrate any existing or new stat commands (there are many, many available on the internet).

Pre-integrated stats are already coming with the STATsrv-Lux.tgz package. It doesn't mean it will work on your system at once (linux distribution compatibility is always an issue). Some of them I got from the 'sysstat' kit and were recompiled on MEPIS 6.0. If required, you may recompile them yourself, these stat programs are coming from sysstat (<http://perso.wanadoo.fr/sebastien.godard/>). And some I developed myself, as I was tired of seeing different outputs on different distros, even

with standard commands like 'vmstat'! Therefore, the STAT-service is shipping with its own vmstat, netLOAD and psSTAT!

Wrappers may be needed for some stat commands to skip unused information or just transform input data into the form expected. The following commands already have wrappers and are pre-integrated into the packaged STAT-service.

NOTE: sometimes the same command gives a different output on different Linux distribution! Be ready to create in this case new Add-Ons or to create common wrappers to adapt command output.

Lvmstat

Source: the Linux "vmstat", as shipped with STAT-service since v.8.0

Output example :

```
dim$ /etc/STATsrv/bin/vmstat 1
procs -----memory----- ---swap-- -----io----- --system-- ----cpu---
 r  b   swpd   free   buff  cache   si   so    bi    bo    in    cs  us  sy  id  w
 0  0 434384 691948   9708 220592    3    4    32    28    36    47   3   1  95
 0  0 434384 691948   9708 220592    0    0     0     0   347   913   2   0  98
 0  0 434384 691948   9708 220592    0    0     0     0   396  1083   2   1  97

dim$
```

A wrapper is not needed anymore. On all systems, the same output is guaranteed (if it runs ;-)).

Lmpstat

Per CPU detailed usage statistics.

Source: the Linux "mpstat" v2 (improved) from Sysstat, and shipped with STAT-service since v.8.3

Output example :

```
# /etc/STATsrv/bin/Lmpstat.sh 5

09:44:12      CPU   %user   %nice   %sys %iowait   %irq   %soft   %steal   %i
09:44:17      all    4.57    0.00    1.12    1.52    0.10    0.00    0.00    92
09:44:17        0    3.81    0.00    1.20    2.00    0.00    0.00    0.00    92
09:44:17        1    5.59    0.00    0.62    0.83    0.00    0.00    0.00    92

09:44:17      CPU   %user   %nice   %sys %iowait   %irq   %soft   %steal   %i
09:44:22      all    1.65    0.00    0.68    0.00    0.00    0.00    0.00    97
09:44:22        0    1.80    0.00    1.00    0.00    0.00    0.00    0.00    97
09:44:22        1    1.32    0.00    0.38    0.00    0.00    0.00    0.00    98
^C
```

LcpuSTAT (deprecated)

The source: "mpstat" from Sysstat

Output example :

```
# /etc/STATsrv/bin/cpuSTAT.sh 1
Linux 2.6.15-26-386 (dimitri) 11/16/06

16:45:15 CPU %user %nice %system %idle intr/s
16:45:16 all 0.00 0.00 0.00 100.00 115.00
16:45:16 0 0.00 0.00 0.00 100.00 115.00
16:45:17 all 1.00 0.00 0.00 99.00 147.00
16:45:17 0 1.00 0.00 0.00 99.00 147.00
16:45:18 all 0.00 0.00 0.00 100.00 162.00
16:45:18 0 0.00 0.00 0.00 100.00 162.00
^C
#
```

A wrapper is not really needed, but simplifies usage. Just ignore the
 "**Linux**||CPU*||" lines and use "**all*" as a separator.

Deprecated (on some systems may show over 100% values :-) - better to use Lmpstat
 now).

LioSTAT

Source: "iostat" from Sysstat

Output example :

```
# /etc/STATsrv/bin/ioSTAT.sh 5

Device: rrqm/s wrqm/s r/s w/s op/s rsec/s wsec/s rkB/s kB/s
sdb 0.00 515.90 17.81 88.83 106.65 1286.49 9897.66 643.24 4948.83
sdb1 0.00 515.90 17.81 88.60 106.42 1286.49 9897.66 643.24 4948.83
sda 0.02 10.39 0.15 0.66 0.81 29.14 87.50 14.57 43.75
sda1 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
sda2 0.02 10.39 0.15 0.53 0.68 29.14 87.50 14.57 43.75
dm-0 0.00 0.00 0.03 8.02 8.05 1.09 64.15 0.54 32.07
dm-1 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
dm-2 0.00 0.00 0.00 0.00 0.00 0.03 0.00 0.01 0.00
dm-3 0.00 0.00 0.00 0.00 0.00 0.03 0.00 0.01 0.00
dm-4 0.00 0.00 0.00 0.00 0.00 0.02 0.00 0.01 0.00
dm-5 0.00 0.00 0.12 2.92 3.04 27.97 23.35 13.98 11.68
dm-6 0.00 0.00 0.00 0.00 0.00 0.01 0.00 0.01 0.00

Device: rrqm/s wrqm/s r/s w/s op/s rsec/s wsec/s rkB/s kB/s
sdb 0.00 1.79 0.00 5.78 5.78 0.00 70.12 0.00 35.06
sdb1 0.00 1.79 0.00 5.78 5.78 0.00 70.12 0.00 35.06
sda 0.00 0.20 0.00 1.39 1.39 0.00 12.75 0.00 6.37
sda1 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
sda2 0.00 0.20 0.00 1.39 1.39 0.00 12.75 0.00 6.37
dm-0 0.00 0.00 0.00 1.59 1.59 0.00 12.75 0.00 6.37
dm-1 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
```

You should be VERY CAREFUL with these actions!

dm-2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
dm-3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
dm-4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
dm-5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
dm-6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

^C
#

Wrapper: ioSTAT.sh - to ignore the CPU-related part, the devices and partition list may vary from system to system.

psSTAT for Linux

I was tired by strange/wrong 'top' output which in many cases just not showing or ignoring low loaded processes, and finally give you a wrong vision about your system. So I adapted my Solaris psSTAT idea to the Linux /proc structures...

So well, there are few similar options:

```
psSTAT (dim) v.2.0 Nov.2006
Usage: psSTAT [options]
  -l                Long output
  -O                active Only processes/users
  -T sec           Timeout sec seconds between outputs
  -N name[,name2[,...]] only proc Name containing name, or name2, or ...
  -M mode          Use Special Mode output:
                    proc      - output is grouped by process name
                    user      - output is grouped user name
                    ref        - reference: process name combined with pid
```

dim\$

Output example :

```
dim$ /etc/STATsrv/bin/psSTAT -O -T 1
```

PID	PNAME	UsrTM	SysTM	CPU%	MinF	MajF	PRI	NI	Thr	VmSIZE
1	init	0.00	0.00	0.0	0	0	16	0	1	1568
3153	dbus-daemon	0.02	0.00	2.0	0	0	17	0	1	2324
3166	hald	0.01	0.00	1.0	0	0	16	0	1	6916
3761	Xorg	0.01	0.00	1.0	0	0	5	-10	1	100680
3879	konsole	0.02	0.00	2.0	2	0	16	0	1	29416
24904	kpowersave	0.01	0.00	1.0	0	0	16	0	1	32720
28035	psSTAT	0.02	0.00	2.0	336	0	16	0	1	1812

PID	PNAME	UsrTM	SysTM	CPU%	MinF	MajF	PRI	NI	Thr	VmSIZE
1	init	0.00	0.00	0.0	0	0	16	0	1	1568
28035	psSTAT	0.03	0.00	3.0	336	0	17	0	1	1812

PID	PNAME	UsrTM	SysTM	CPU%	MinF	MajF	PRI	NI	Thr	VmSIZE
1	init	0.00	0.00	0.0	0	0	16	0	1	1568
3761	Xorg	0.03	0.00	3.0	0	0	5	-10	1	100680
22726	java_vm	0.01	0.00	1.0	0	0	16	0	21	231760
28035	psSTAT	0.03	0.00	3.0	336	0	17	0	1	1812

PID	PNAME	UsrTM	SysTM	CPU%	MinF	MajF	PRI	NI	Thr	VmSIZE
1	init	0.00	0.00	0.0	0	0	16	0	1	1568

You should be VERY CAREFUL with these actions!

```

3761 Xorg          0.02  0.00  2.0    0    0    5 -10    1 100680
3879 konsole      0.01  0.00  1.0    0    0   15  0    1 29416
28035 psSTAT      0.03  0.00  3.0   336    0   16  0    1 1812

```

```

^C
dim$

```

There are 3 Linux add-ons based on psSTAT:

- ◇ LpsSTAT - process stat using 'ProcName-PID' pair as unique process reference (mode: ref)
- ◇ LPrcLOAD - grouped by process name activity stats (mode: proc)
- ◇ LUsrLOAD - grouped by user name activity stats (mode: user)

NOTE: data are collected in live from '/proc' data but by given time interval, so be aware - if during this interval some processes are forked and dead very quickly - they're simply not seen by tool as there will be no trace about them in any '/proc' data...

LpsSTAT (psSTAT)

Source: psSTAT for Linux, mode: ref

Output example :

```
dim$ /etc/STATsrv/bin/psSTAT.sh 1
```

PNAME-PID	UstrTM	SysTM	CPU%	MinF	MajF	PRI	NI	Thr	VmSIZE	VmLCK
init-00001	0.00	0.00	0.0	0	0	16	0	1	1568	
dbus-daemon-03153	0.03	0.00	3.0	0	0	17	0	1	2324	
hald-03166	0.01	0.00	1.0	0	0	16	0	1	6916	
Xorg-03761	0.02	0.00	2.0	0	0	5	-10	1	100680	
konsole-03879	0.01	0.00	1.0	0	0	16	0	1	29416	
opera-13455	0.01	0.00	1.0	0	0	15	0	1	84380	
java_vm-22726	0.01	0.00	1.0	0	0	16	0	21	231760	
psSTAT-27995	0.01	0.00	1.0	336	0	16	0	1	1816	

```
^C $dim
```

This STAT should be used if you're looking for a single process activity and go in detail for PID, etc.

LPrcLOAD (ProcLOAD)

Source: psSTAT for Linux, mode: proc

Output example :

```
dim$ /etc/STATsrv/bin/ProcLOAD.sh 1
```

PNAME	UstrTM	SysTM	CPU%	MinF	MajF	Nmb	Act	Thr	VmSIZE	VmLCK
NetworkManager	0.00	0.00	0.0	0	0	1	0	1	3928	0

You should be VERY CAREFUL with these actions!

```

Xorg          0.01  0.00  1.0    0    0    1    1    1    100680    0
konsole       0.01  0.00  1.0    0    0    5    1    5    148032    0
psSTAT        0.03  0.00  3.0   338    0    1    1    1     1816    0

PNAME          UstrTM  SysTM  CPU%  MinF  MajF  Nmb  Act  Thr   VmSIZE  VmLCK
NetworkManager 0.00   0.00   0.0    0    0    1    0    1     3928    0
Xorg            0.01   0.00   1.0    0    0    1    1    1    100680    0
konsole         0.01   0.00   1.0    0    0    5    1    5    148032    0
psSTAT          0.01   0.00   1.0   338    0    1    1    1     1816    0

^C
$dim

```

This STAT should be used if you're looking for global per 'process name' activity and don't really need to go in detail - specially when you have a lot of processes running (!)

LUsrLOAD (UserLOAD)

Source: psSTAT for Linux, mode: user

Output example :

```

dim$ /etc/STATsrv/bin/UserLOAD.sh 1

UNAME          UstrTM  SysTM  CPU%  MinF  MajF  Nmb  Act  Thr   VmSIZE  VmLCK
root           0.01   0.00   1.0   420    0   62    1   62   256312   3576
dim            0.03   0.00   3.0    46    0   92    2  124  1774180    0

UNAME          UstrTM  SysTM  CPU%  MinF  MajF  Nmb  Act  Thr   VmSIZE  VmLCK
root           0.02   0.00   2.0   338    0   62    1   62   256312   3576
dim            0.02   0.00   2.0    46    0   92    2  124  1774180    0

^C
$dim

```

This STAT should be used if you're looking for global per 'user' activity and don't really need to go in detail - specially when your tasks are grouped per user or you have a lot of users using the system (!)

LnetLOAD (netLOAD)

Source: my netLOAD script for Linux

Output example :

```

/etc/STATsrv/bin/netLOAD.sh 1

Name          IBytes/s  OBytes/s  IPack/s  OPack/s  IErr  OErr  IDrp  ODrp  Bytes/s
none           0           0           0           0         0     0     0     0         0
lo          66070356  66070356   130181   130181         0     0     0     0  132140712
eth0         32074500  19059001   236433   218784         0     0     0     0   51133501
eth1          3766140   1544506    93950    56325        60     0    60     0   5310646

Name          IBytes/s  OBytes/s  IPack/s  OPack/s  IErr  OErr  IDrp  ODrp  Bytes/s
none           0           0           0           0         0     0     0     0         0

```

lo	0	0	0	0	0	0	0	0	0
eth0	0	0	0	0	0	0	0	0	0
eth1	0	0	2	3	0	0	0	0	0

Name	IBytes/s	OBytes/s	IPack/s	OPack/s	IErr	OErr	IDrp	ODrp	Bytes/s
none	0	0	0	0	0	0	0	0	0
lo	0	0	0	0	0	0	0	0	0
eth0	0	0	0	0	0	0	0	0	0
eth1	0	0	2	3	0	0	0	0	0

^C

For the STAT-service Wrapper, no need, it should work as on any Linux system.

Report Tool

This User's Guide is completely written using Report Tool!! And as so often, this tool was mainly created to cover my own day to day needs.

Quite often I have to write reports to show performance findings, to present the observed system / application activity, etc., etc. Yes, etc. because sometime we have to write too much to make things work or simply to protect people from doing stupid things. :))

OK, you've started to write your document for a French customer, so you write it in French, and then it appears that the majority of the development team only speaks English. You start to keep two copies in parallel for the same document: FR/EN. Then you discover something very important but you can not say it yet your customer, but you absolutely need to communicate it internally. So you split the document once again: FR/EN and Customer/Internal, which means four different documents. The next split will give you eight version of the document. But it is still based on the same source of information. The result is a lot of hours spent doing copy-paste of activity graphs from the browser, teamquest, best1, patrol, etc. into your wordprocessor. It makes me cry... :))

I was really tired of this situation and tried to imagine something **different**.

Overview

The first issue was the choice of format: At least everybody on any platform is able to read HTML. So that's an easy one. If needed you can easily convert HTML into other formats, like PDF, etc.

The next problem is harder to solve. It was my idea to find a solution for generating different kinds of documents from the same main data source. When you take a look at any document, how is its content organized? You'll see:

◇ Document = N x Chapters

You should be VERY CAREFUL with these actions!

- ◇ Chapter = M x Sections
- ◇ Section = P x Paragraphs
- ◇ and so on ...
- ◇ Smallest part = Smallest part :-)

It all depends on what is your smallest part. So, I've named my smallest part a **Note** and a Document or Report is presented simply as an ordered tree of Notes.

The main points :

- ◇ the position of each Note in a Report is decided by its parent-ID (level + 1) and order number (same level)
- ◇ Note : each Note has/contains:
 - a Data Type
 - a Title
 - text comments
 - possibly an attachment (depends on Data Type)
 - a list of attributes
- ◇ Attributes : any Note may have zero, one or several attributes on:
 - Language (French, English, ...)
 - Confidentiality (Personal, Customer, ...)
 - ... (any other can be easily added into the system)
- ◇ Data Type : the list of Data Types is fixed (but may be extended):
 - Text
 - HTML
 - Image
 - Binary
 - dim_STAT collect
 - SysINFO
 - HTML.tar.Z archive

Any Note can be created/edited/deleted at any time. During Report generation you only need to choose the right criteria for your requirements to create a valid document with all parts corresponding to the criteria.

Datatype: Text, HTML, Image, Binary

These data types are quite similar, you can create any note with any text, html, image or binary file in an attachment, with or without your comments. Except binary, any other file may be presented "In-Line" or "Linked".

In-Line means your file will be part of the main document page and the visible contents, ex: text directly included, image showed, etc.

Linked means linked :)), meaning that the main document page will only include a link to your attachment. However, this attachment will be always included with

document.

Note: the same idea is applied to other types of Notes as well.

Datatype: SysINFO

This is a special type, with the purpose to get on-line system information from any host on the network that runs STAT-service. Of course only if you have permission to access this service and SysINFO.

Datatype: HTML.tar.Z

A special type in case you want to integrate into your Report any other documents, already written, that are converted to HTML and archived into a single tar.Z file. As you may have several files in your archive, the tool will ask you for the name of the 'main' file, which will maintain references to all other files.

Datatype: dim_STAT-Snapshot

A type for when you've saved graph pages based on Java applets during dim_STAT analysis. You may integrate them 'as is', the tool will extract the applet data and insert them as Note contents.

Probably this should be deprecated, as any graph can be saved in PNG format, or you could simply convert it to PNG or GIF.

Datatype: dim_STAT-Collect

This is a very special type, it helps you to generate all STAT graphs automatically and it will save you a lot of time. Follow the example below.

Preview / Generate / Publish

At any moment you can 'Preview' your Report or 'Generate' a current/final version to be accessed on-line, or saved and shared as tar.Z archive, or as a single PDF file. Also, your document may be published on another site (actually, this part is limited to the same physical host).

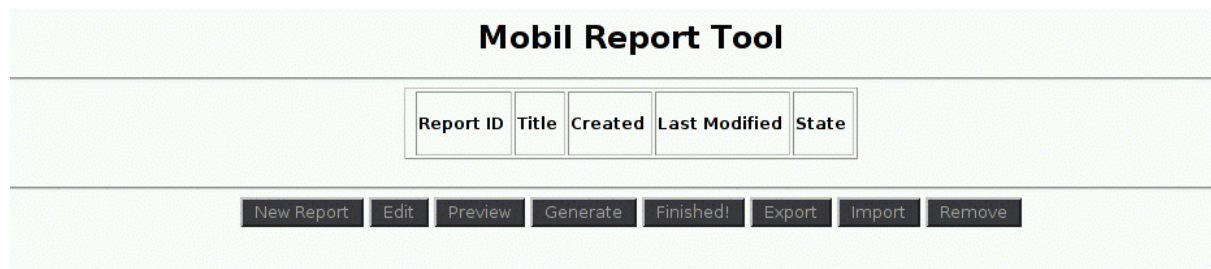
Export / Import

These features explain why Report Tool is called 'Mobile'. At any time you can export your Report and import it into any other dim_STAT server. This means: you edit/prepare everything on your laptop, and from time to time you synchronize your work with a central repository. Also, it gives you a simple way to prepare your own templates! Instead of starting a new report every time, just import your template (old report) and continue.

Let's try! New Report

Now relax, take your coffee, be sure you've 20 minutes of free time (while nobody is stressing you), your GSM is off, you're ready to listen ... go to the dim_STAT Main page and click on 'Report Tool'.

Click on Report Tool



The screenshot shows the 'Mobil Report Tool' interface. At the top, the title 'Mobil Report Tool' is centered. Below it is a table with five columns: 'Report ID', 'Title', 'Created', 'Last Modified', and 'State'. The table is currently empty. Below the table is a row of action buttons: 'New Report', 'Edit', 'Preview', 'Generate', 'Finished!', 'Export', 'Import', and 'Remove'.

As you could have expected, nothing yet here for the moment.

Let's click on the "New Report" button.

New Report

Start a New Report

Report ID:	15
Title:	Demonstration Report
Owner:	Dimitri
Chart Comments:	WCB Team, 2006

Use: General Template

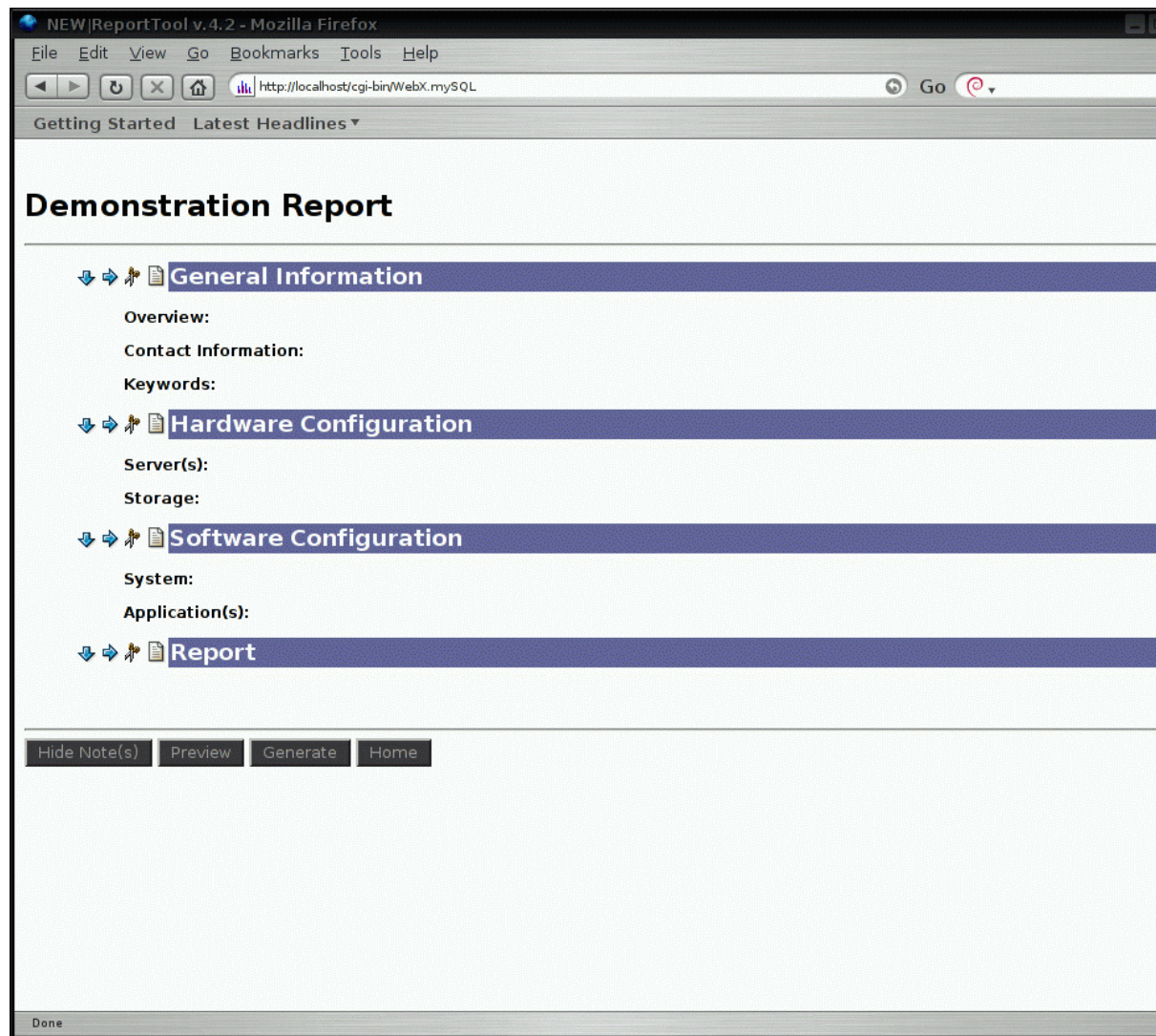
Create Clear

All you need to do here is just to fill in the new report form:

- ◇ ID: unique digital number
- ◇ Title: the main title
- ◇ Owner: owner information
- ◇ Chart: any additional comments to be present on the cover page
- ◇ Use: choose a pre-configured Report template

and click on "Create"

Edit Report



Wow! It works! :))

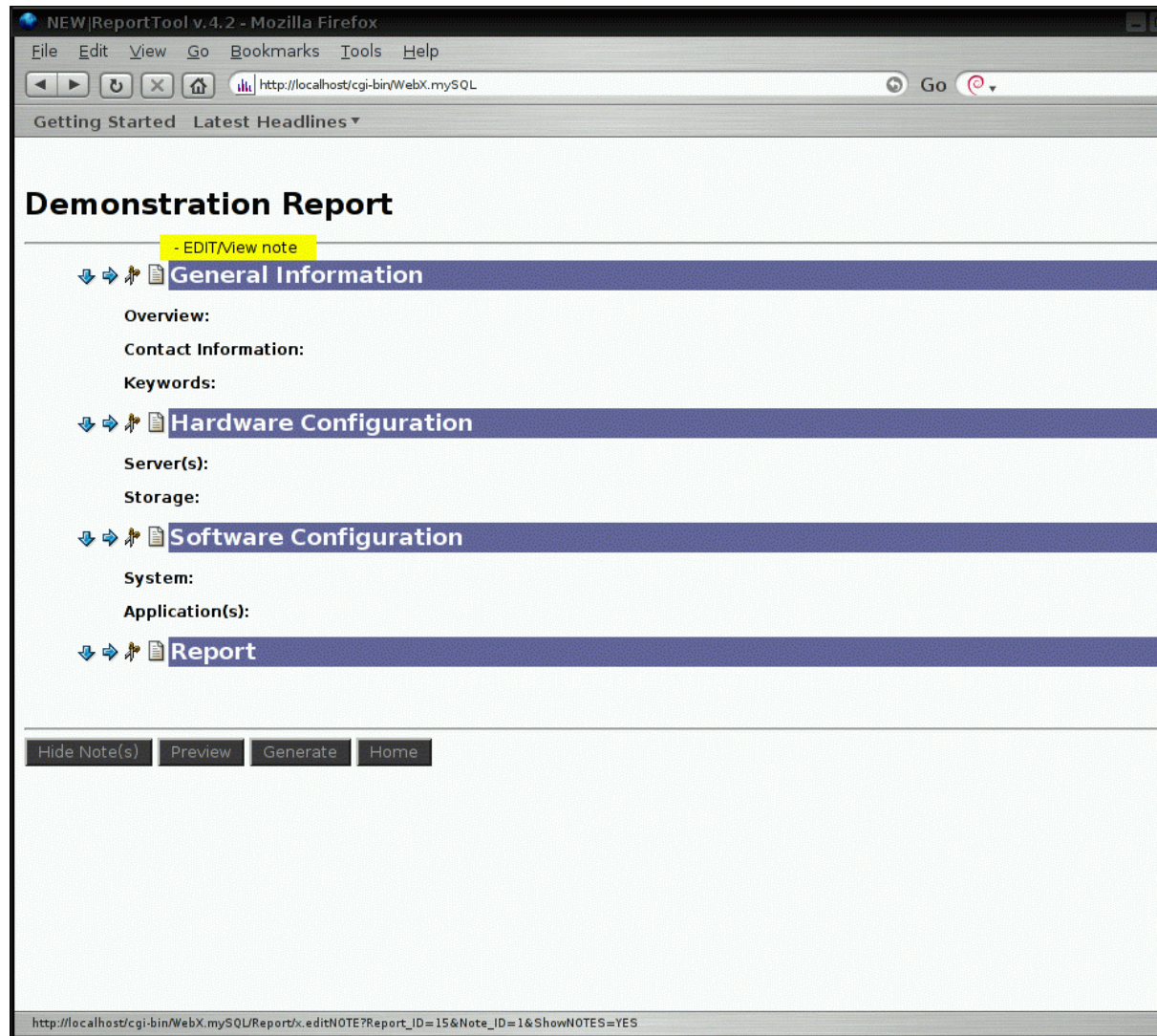
With the 'big' buttons, you may now:

- ◇ Hide/Show Note comments
- ◇ Preview your report
- ◇ Generate the report
- ◇ go Home (back to the main Report page)

But if you'll hover your mouse over the pre-generated notes you'll see pop-ups explaining each action.

Edit Actions

You should be VERY CAREFUL with these actions!

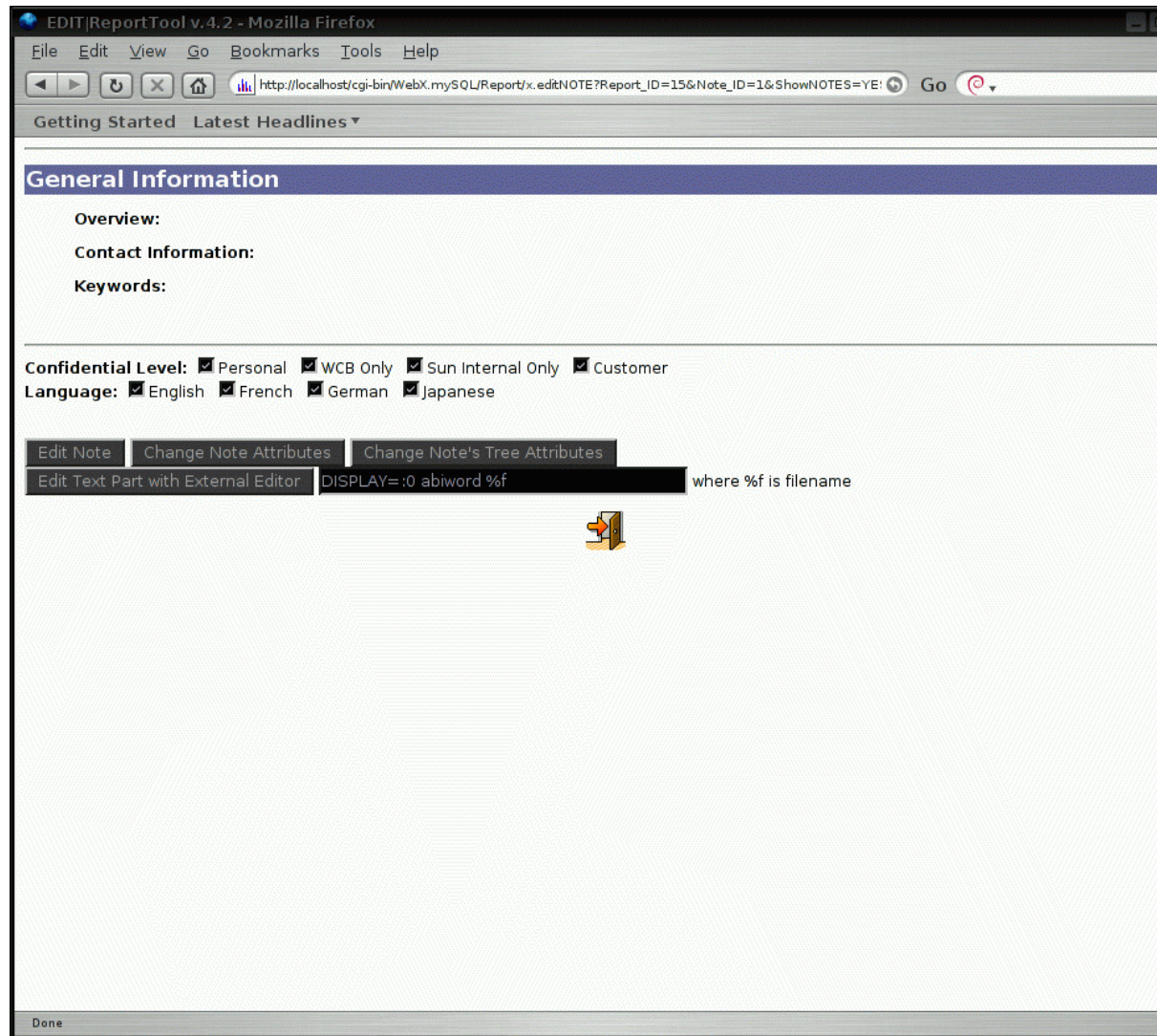


And now:

- ◇ click on the 'down' icon to create a new note 'after' the current one (same parent level)
- ◇ click on the 'right' icon to create new 'child' note 'under' the current one (parent level+1)
- ◇ click on the 'cut' icon to cut and then paste (may go to 'trash' if need to be deleted (end of screen))
- ◇ click on the 'data' to edit/view the Note

Let's edit 'General Information' (click on 'data' icon).

Edit Note



From here you may see the current Note preview and edit the Note comments or attributes. If you change only attributes, then click on the corresponding button to apply the changes. If you want to modify the Note comments, click on 'Edit Note'. BTW, you can also do that with any external editor.

Edit Note, continue...

EDIT|ReportTool v.4.2 - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://localhost/cgi-bin/WebX.mysql

Getting Started Latest Headlines ▾

General Information

Overview:

Contact Information:

Keywords:

Title: General Information

Info:

Note: Text-Format Use javaEditor (Ekit)

Overview:

Contact Information:

Keywords:

Data: Include New TEXT file In-Line in Document : Browse...

Confidential Level: ☒ Personal ☒ WCB Only ☒ Sun Internal Only ☒ Customer

Language: ☒ English ☒ French ☒ German ☒ Japanese

Save Note

Done

Add what you want in the text fields (you may use any HTML tags, etc.)

Edit Note, continue2...

EDIT|ReportTool v.4.2 - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://localhost/cgi-bin/WebX.mysql

Getting Started Latest Headlines ▾

General Information

Overview:

Contact Information:

Keywords:

Title: General Information

Info:

Note: Text-Format ☐ Use javaEditor (Ekit)

Overview: This is a general demonstration project to show how Report Tool works. Any comments are welcome as well any RFE, suggestions, etc...

Contact Information:

- Dimitri
- dim_STAT@sun.com

Keywords: dim_STAT, Report

Data: Include New In-Line

Confidential Level: ☒ Personal ☒ WCB Only ☒ Sun Internal Only ☒ Customer

Language: ☒ English ☒ French ☒ German ☒ Japanese

Done

Note: if you choose Text-format option your text is auto-formatted.

- ◇ an **empty line** is seen as a 'new paragraph'
- ◇ **three spaces** at the start of the line are replaced by a "blanked-tabulation"
- ◇ some kind of limited **wiki-like** syntax is supported (see below example of input text containing wiki-like tags and its output result)..

Save the Note.

Wiki-Like syntax: INPUT

```
Here is a !=Big BOLD Header!=
Here is just a text +!with INCREASED font size!+
```

You should be VERY CAREFUL with these actions!

```

*!Here!* or **Here** will be a bold text

/!Here!/ is text in italic

_!Here!_ or __Here__ will by underlined test

__Simple TEXT List__ :
- one
- two
- three

__Simple HTML List__ :
* one
* two
* three

__Simple code or text formatted__ :
[code]
$ ls -l /usr/sfw/bin/*
...
$ ps -ef
...
$ pkill -9 oracle
[/code]

__Simple Table__ :

| **System/Performance** | **TPS** | **Resp.Time(ms)** |
| M5000 | 4.500 | 10.0 |
| M8000 | 8.000 | 10.0 |
| M9000 | 15.000 | 9.2 |

```

Wiki-Like syntax: OUTPUT

Here is a

Big BOLD Header

Here is just a text with INCREASED font size

Here or **Here** will be a bold text

Here is text in italic

Here or Here will by underlined test

Simple TEXT List :

- one
- two
- three

Simple HTML List :

- one
- two
- three

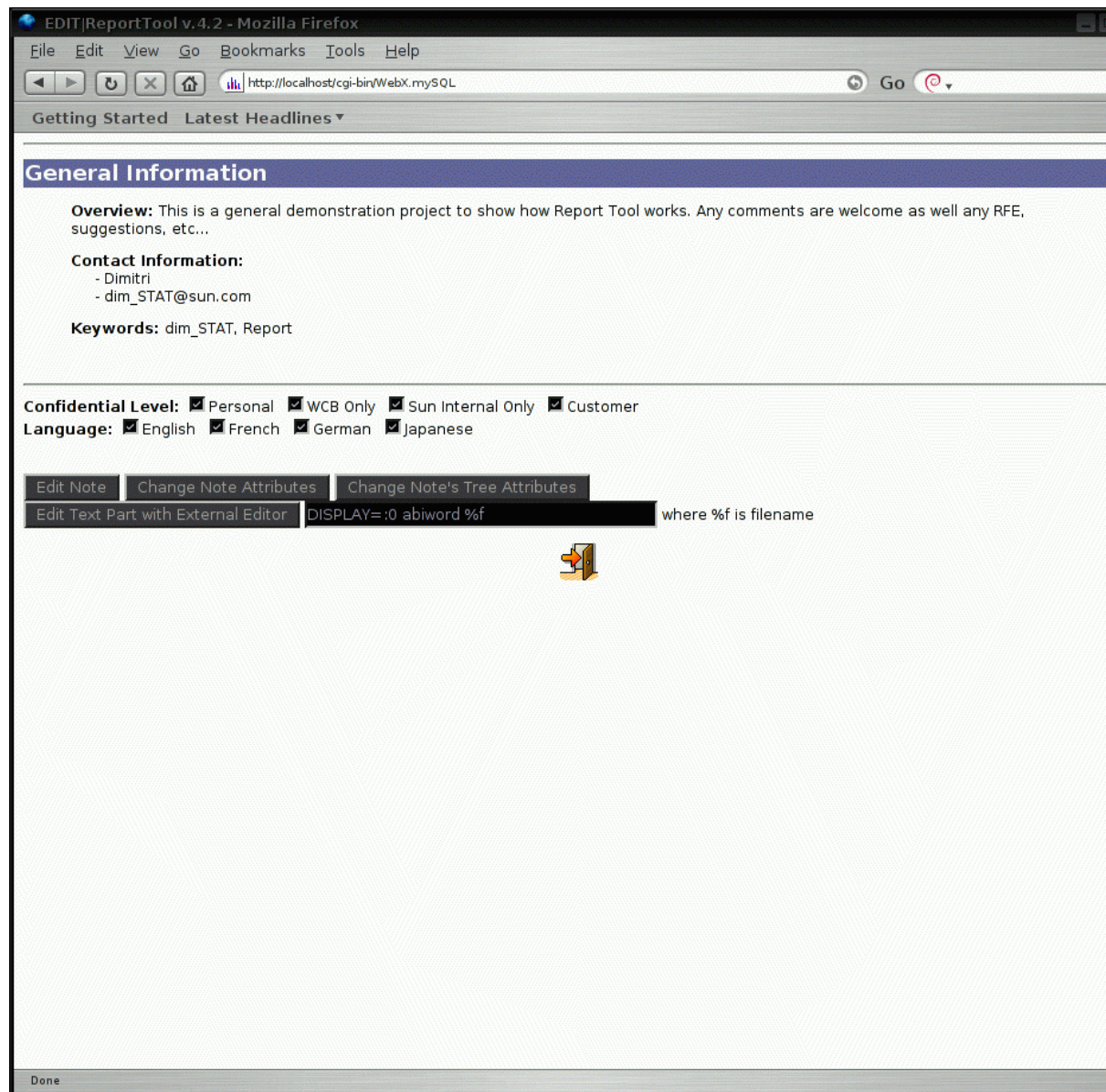
Simple code or text formatted :

```
$ ls -l /usr/sfw/bin/*  
...  
$ ps -ef  
...  
$ pkill -9 oracle
```

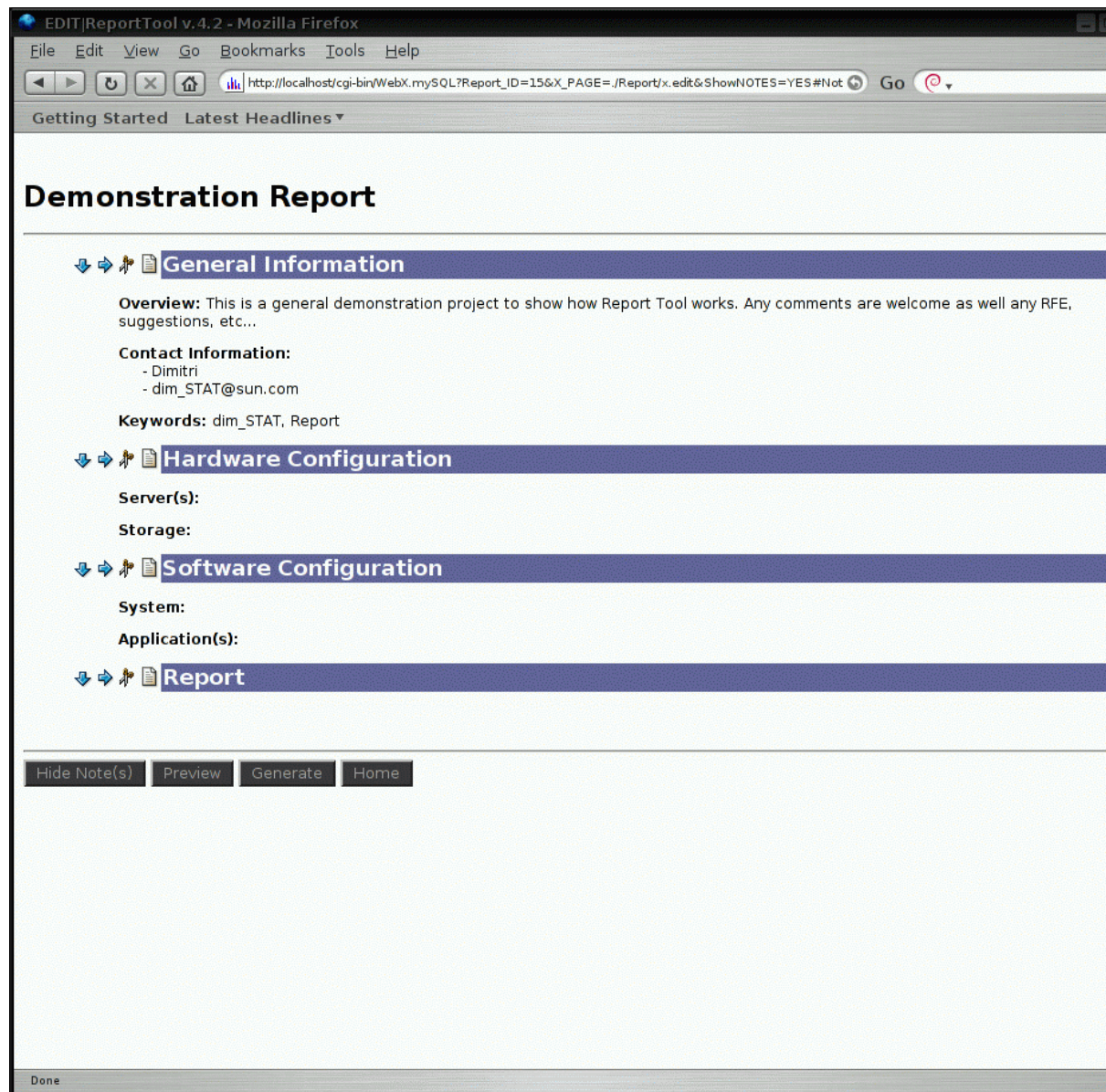
Simple Table :

System/Performance	TPS	Resp.Time(ms)
M5000	4.500	10.0
M8000	8.000	10.0
M9000	15.000	9.2

Edit Note, continue3...

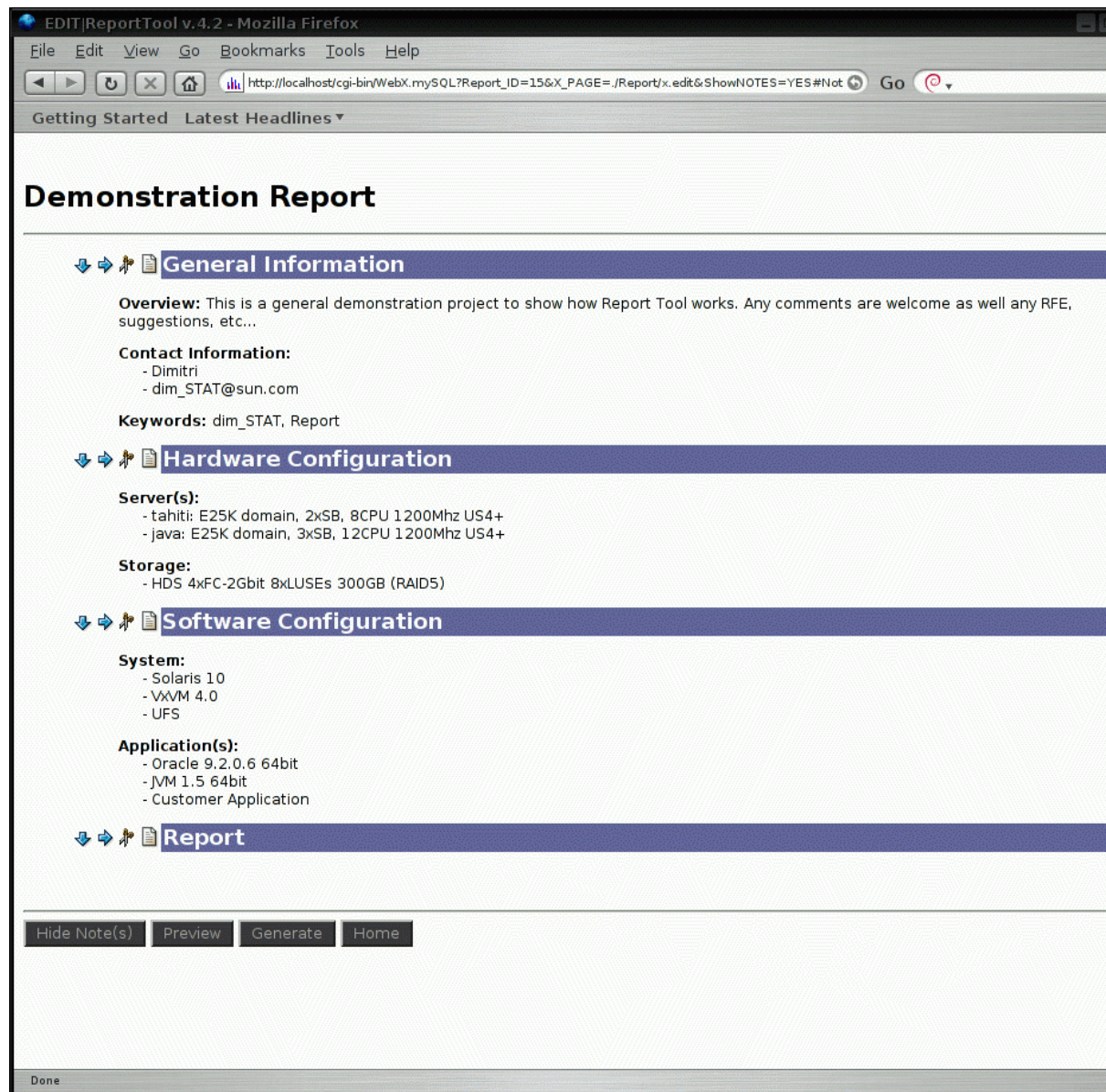


Edit Report, continue...



Let's fill other notes in the same way...

Edit Report, continue2...



So far so good :))

Now, I want to add a SysINFO Note for both hosts 'tahiti' and 'java'. SysINFO data is collected on-line the moment you're asking for and it's an easy way to keep your document updated at the moment you're writing. BTW, look into the STAT-service package to know how it is configured on the host side. You may extend it with any other information you need.

So, a new SysINFO note under 'Software Configuration'... (right icon)

Add Note

EDIT|ReportTool v.4.2 - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://localhost/cgi-bin/WebX.mysql?Report_ID=15&X_PAGE=../Report/x.edit&ShowNOTES=YES#Not

Getting Started Latest Headlines ▾

Demonstration Report

General Information

Overview: This is a general demonstration project to show how Report Tool works. Any comments are welcome as well any RFE, suggestions, etc...

Contact Information:

- Dimitri
- dim_STAT@sun.com

Keywords: dim_STAT, Report

Hardware Configuration

Server(s):

- tahiti: E25K domain, 2xSB, 8CPU 1200Mhz US4+
- java: E25K domain, 3xSB, 12CPU 1200Mhz US4+

Storage:

- HDS 4xFC-2Gbit 8xLUSEs 300GB (RAID5)

- ADD New note under this one, level+1

System:

- Solaris 10
- VxVM 4.0
- UFS

Application(s):

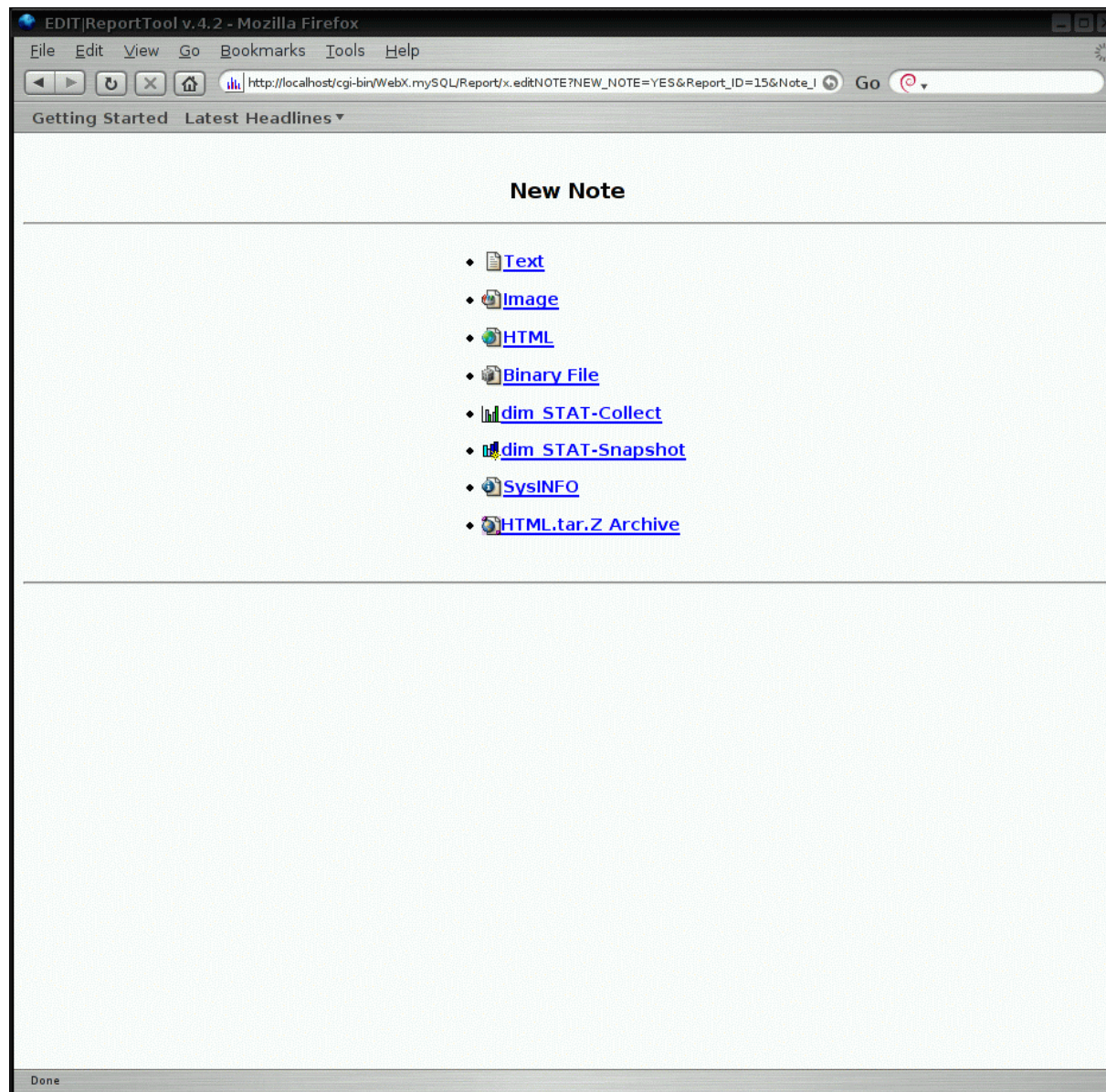
- Oracle 9.2.0.6 64bit
- JVM 1.5 64bit
- Customer Application

Report

Hide Note(s) Preview Generate Home

http://localhost/cgi-bin/WebX.mysql/Report/x.editNOTE?NEW_NOTE=YES&Report_ID=15&Note_ID=3&ShowNOTES=YES&Direction=RIGHT

New Note -- SysINFO



As the tool has no idea what kind of Note you want to add, it will ask you to choose one before it can continue. Also, I did not want to add too much complexity to the interface.

So, just click on 'SysINFO' here...

New Note -- SysINFO Form

The screenshot shows a web browser window titled "EDIT|ReportTool v.4.2 - Mozilla Firefox". The address bar shows a URL: `http://localhost/cgi-bin/WebX.mysql/Report/x.editNOTE?NEW_NOTE=YES&Report_ID=15&Note_I`. The page has a navigation bar with "Getting Started" and "Latest Headlines". The main heading is "New SysINFO note...".

The form contains the following fields and options:

- Title:** SysINFO: tahiti
- Info:** 12:13 2006-04-04 (Dimitri)
- Note:**
 - Text-Format (dropdown menu)
 - Copy text from neighbor note (button)
 - Use javaEditor (Ekit) (button)
 - A large text area for the note content.
- Hostname:** tahiti
- STAT-service Port:** 5000
- Data:** as External Link (checkbox) in Document

Below the form, there are two rows of checkboxes:

- Confidential Level:** ☒ Personal ☒ WCB Only ☒ Sun Internal Only ☒ Customer
- Language:** ☒ English ☒ French ☒ German ☒ Japanese

At the bottom of the form is a "Save Note" button. The footer of the browser window shows "Done".

Here you will need to fill in the SysINFO form: the usual data (title/comments/attributes) and SysINFO specific ones:

- ◇ the host name
- ◇ the host's STAT-service port

As SysINFO output is usually quite wide, it's preferred to keep it as an 'External Link'.

Save the Note. If you gave the right hostname, port and the STAT-service is up and running on this host, You'll receive your data in a few seconds, in our example from the 'tahiti' domain :))

New Note -- SysINFO Result

[SysINFO: tahiti](#) @

Confidential Level: ☒ Personal ☒ WCB Only ☒ Sun Internal Only ☒ Customer
Language: ☒ English ☒ French ☒ German ☒ Japanese

Edit Note

Change Note Attributes

Change Note's Tree Attributes



Because I asked for 'Linked' contents, there is only a link to SysINFO data from 'tahiti'. Let's click on it to see if it works correctly.

New Note -- SysINFO Link Contents

2005-09-04 12:22

SysINFO: tahiti

SunOS tahiti 5.10 Generic_118822-25 sun4u sparc SUNW,Sun-Fire-15000

- [System View](#)
- [prtdiag](#)
- [/etc/system](#)
- [/etc/vfstab](#)
- [mount](#)
- [df -k](#)
- [vxprint](#)

System View

```
tahiti    09/04/05   12:22:23
Solaris 10 3/05 s10_74L2a SPARC Generic_118822-25
sun4u    1500/150 MHz
Sun Fire 15000  16 CPUs   64.00 GB memory
```

prtdiag

```
System Configuration: Sun Microsystems sun4u Sun Fire 15000
System clock frequency: 150 MHz
Memory size: 65536 Megabytes

===== CPUs =====
```

Slot ID	CPU ID	Run MHz	E\$ MB	CPU Impl.	CPU Mask
/SB02/P0	64, 68	1500	32.0	US-IV+	2.1
/SB02/P1	65, 69	1500	32.0	US-IV+	2.1
/SB02/P2	66, 70	1500	32.0	US-IV+	2.1
/SB02/P3	67, 71	1500	32.0	US-IV+	2.1
/SB13/P0	416, 420	1500	32.0	US-IV+	2.1
/SB13/P1	417, 421	1500	32.0	US-IV+	2.1
/SB13/P2	418, 422	1500	32.0	US-IV+	2.1
/SB13/P3	419, 423	1500	32.0	US-IV+	2.1

Done

Edit Report, continue3...

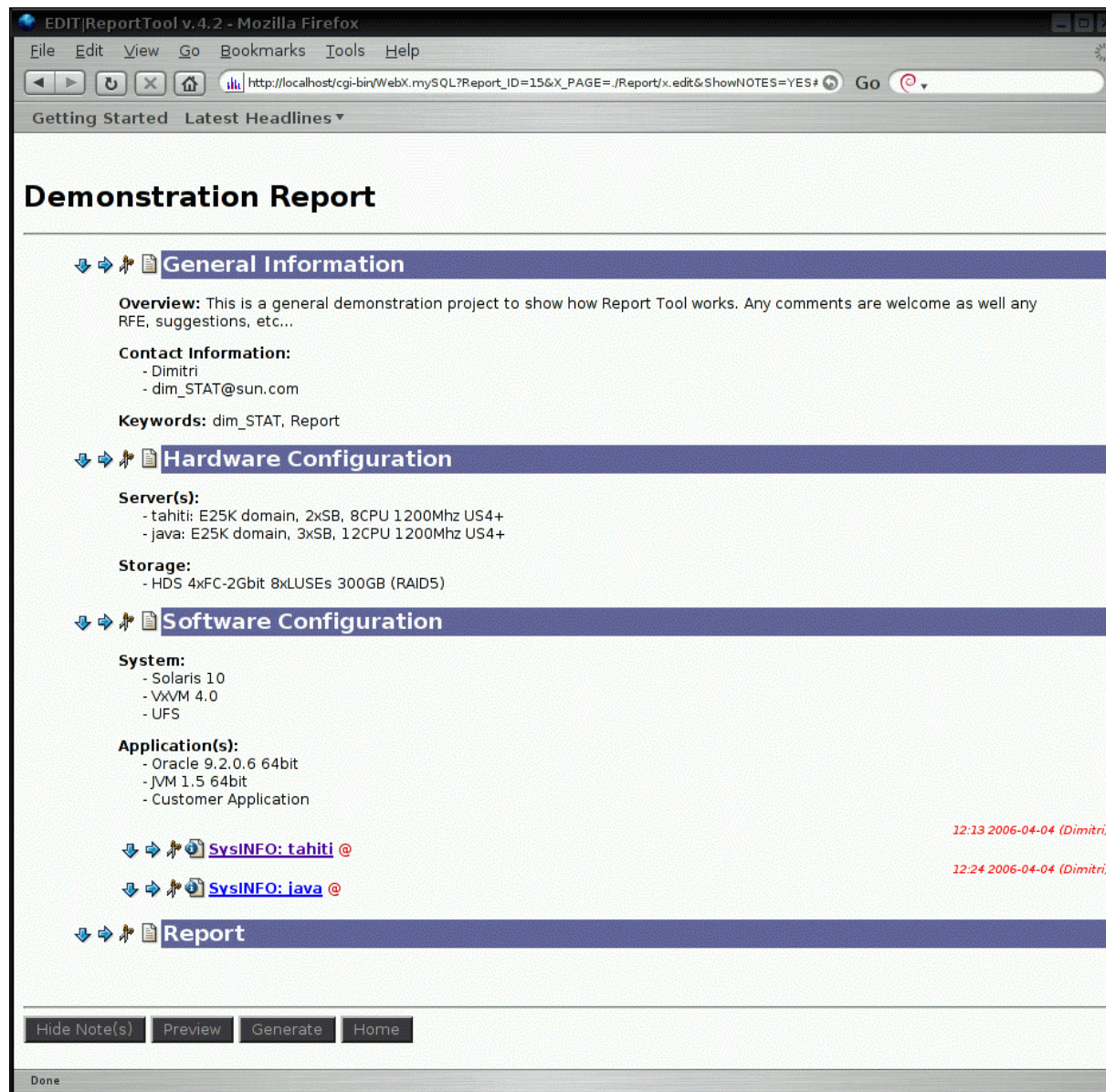
The screenshot shows a web browser window titled "EDIT|ReportTool v.4.2 - Mozilla Firefox". The address bar displays the URL: `http://localhost/cgi-bin/WebX.mysql?Report_ID=15&X_PAGE=/Report/x.edit&ShowNOTES=YES#`. The page content is titled "Demonstration Report" and is organized into several sections:

- General Information**:
 - Overview**: This is a general demonstration project to show how Report Tool works. Any comments are welcome as well any RFE, suggestions, etc...
 - Contact Information**:
 - Dimitri
 - dim_STAT@sun.com
 - Keywords**: dim_STAT, Report
- Hardware Configuration**:
 - Server(s)**:
 - tahiti: E25K domain, 2xSB, 8CPU 1200Mhz US4+
 - java: E25K domain, 3xSB, 12CPU 1200Mhz US4+
 - Storage**:
 - HDS 4xFC-2Gbit 8xLUSEs 300GB (RAID5)
- Software Configuration**:
 - System**:
 - Solaris 10
 - VxVM 4.0
 - UFS
 - Application(s)**:
 - Oracle 9.2.0.6 64bit
 - JVM 1.5 64bit
 - Customer Application
- SysINFO: tahiti @**: A new SysINFO note is added under this section, with a timestamp "12:13 2006-04-04 (Dimitri)".
- Report**: The final section of the report.

At the bottom of the report, there are buttons for "Hide Note(s)", "Preview", "Generate", and "Home". The status bar at the very bottom shows "Done".

As you see, I've my new SysINFO note under 'Software Configuration'. Let's get SysINFO from 'java' host now and place it 'under' current tahiti SysINFO...

Edit Report, continue4...



Now, under 'Hardware Configuration' I want to add an image representing my platform diagram (a very simple image, just for those who are not able to imagine two hosts with one storage device :)), but "a picture says more than a thousand words". :))

So: 'Hardware Configuration' -> Image -> ...

Add New Note -- Image

The screenshot shows a web browser window titled "EDIT|ReportTool v.4.2 - Mozilla Firefox". The address bar shows a URL: `http://localhost/cgi-bin/WebX.mysql/Report/x.editNOTE?NEW_NOTE=YES&Report_ID=15&Note_I`. The page has a navigation bar with "Getting Started" and "Latest Headlines". The main heading is "New IMAGE note...".

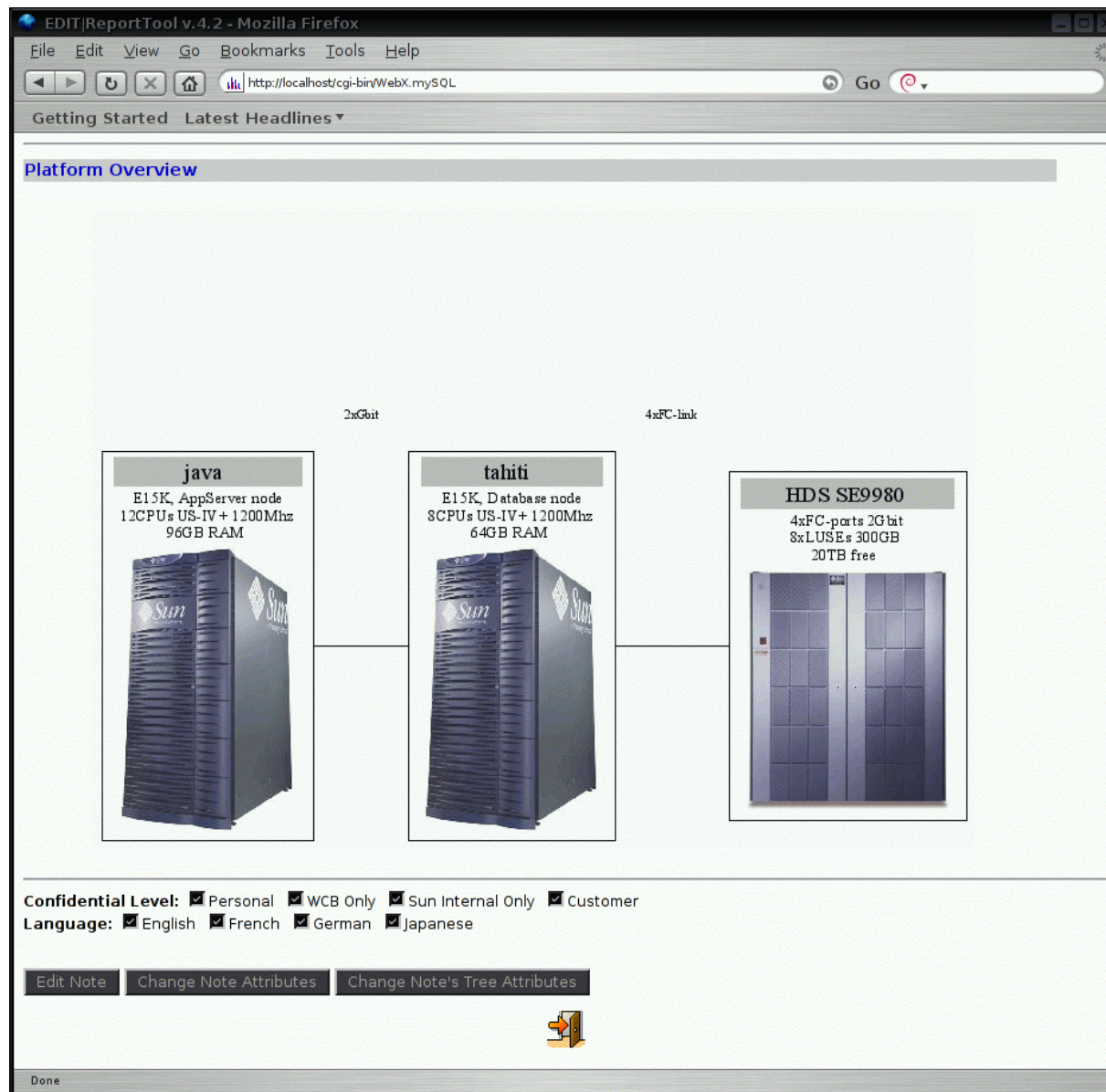
The form contains the following fields and options:

- Title:** Platform Overview
- Info:** 13:05 2006-04-04 (Dimitri)
- Note:** A large text area for the note content.
- Data:** Includes a dropdown for "Include New" (set to "IMAGE file"), a dropdown for "In-Line" (set to "In-Line"), and a text field for "in Document :" containing the path `/home/apps/dim_STAT/DOC/VER_7_0/snaps/demo_conf.gif`. There is a "Browse..." button next to the path field.
- Confidential Level:** Checkboxes for ☒ Personal, ☒ WCB Only, ☒ Sun Internal Only, and ☒ Customer.
- Language:** Checkboxes for ☒ English, ☒ French, ☒ German, and ☒ Japanese.
- A "Save Note" button is located below the form.

The status bar at the bottom of the browser window shows "Done".

Once again, similar info to fill, except you may give a name of your image file to upload [Browse]. Let's fill it and save as 'In-Line' attachment.

Add New Note -- Image Inline



Oops, it's TOO BIG! And that's not so you can see it better!! I prefer to keep all big images 'linked'.

So, [Edit Note] -> 'As External Link' (no more need to give image file again) -> [Save Note]

Add New Note -- Image Linked

[Platform Overview](#) @

Confidential Level: ☒ Personal ☒ WCB Only ☒ Sun Internal Only ☒ Customer
Language: ☒ English ☒ French ☒ German ☒ Japanese

Edit Note

Change Note Attributes

Change Note's Tree Attributes



That's better!!

Now, let's add a 'dim_STAT Collect' note!

Leave this page [Door], go to the end of Report and click on the [Right] icon on 'Report' note, then choose 'dim_STAT Collect'.

Add New Note -- dim_STAT Collect, Step1

EDIT|ReportTool v.4.2 - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://localhost/cgi-bin/WebX.mysql/Report/x.editNOTE?NEW_NOTE=YES&Report_ID=15&Note_I Go

Getting Started Latest Headlines ▾

New dim_STAT-Collect note...

Title:	Demo Graphs
Info:	13:12 2006-04-04 (Dimitri)
Note:	Text-Format ▾ Copy text from neighbor note Use JavaEditor (Ekit) <div style="background-color: black; height: 150px; width: 100%;"></div>
Data:	In-Line ▾ in Document
dim_STAT	Database Server: sting Database TCP/IP Port: Default Database Name: EasySTAT STAT(s) Title pattern: *

Confidential Level: ☒ Personal ☒ WCB Only ☒ Sun Internal Only ☒ Customer
Language: ☒ English ☒ French ☒ German ☒ Japanese

Next

Done

The dim_STAT Collect Note needs several steps to be created:

- ◇ 1. setup dim_STAT server database parameters, [Next]
- ◇ 2. select STAT collect you want to use, [Next]
- ◇ 3. select STATs you want to see and time interval, [Next]
- ◇ 4. [Finish] or select STATs you want to see and time interval, [Next] (goto 4)
- ◇ 5. graph titles, choose graph parameters, [Save]

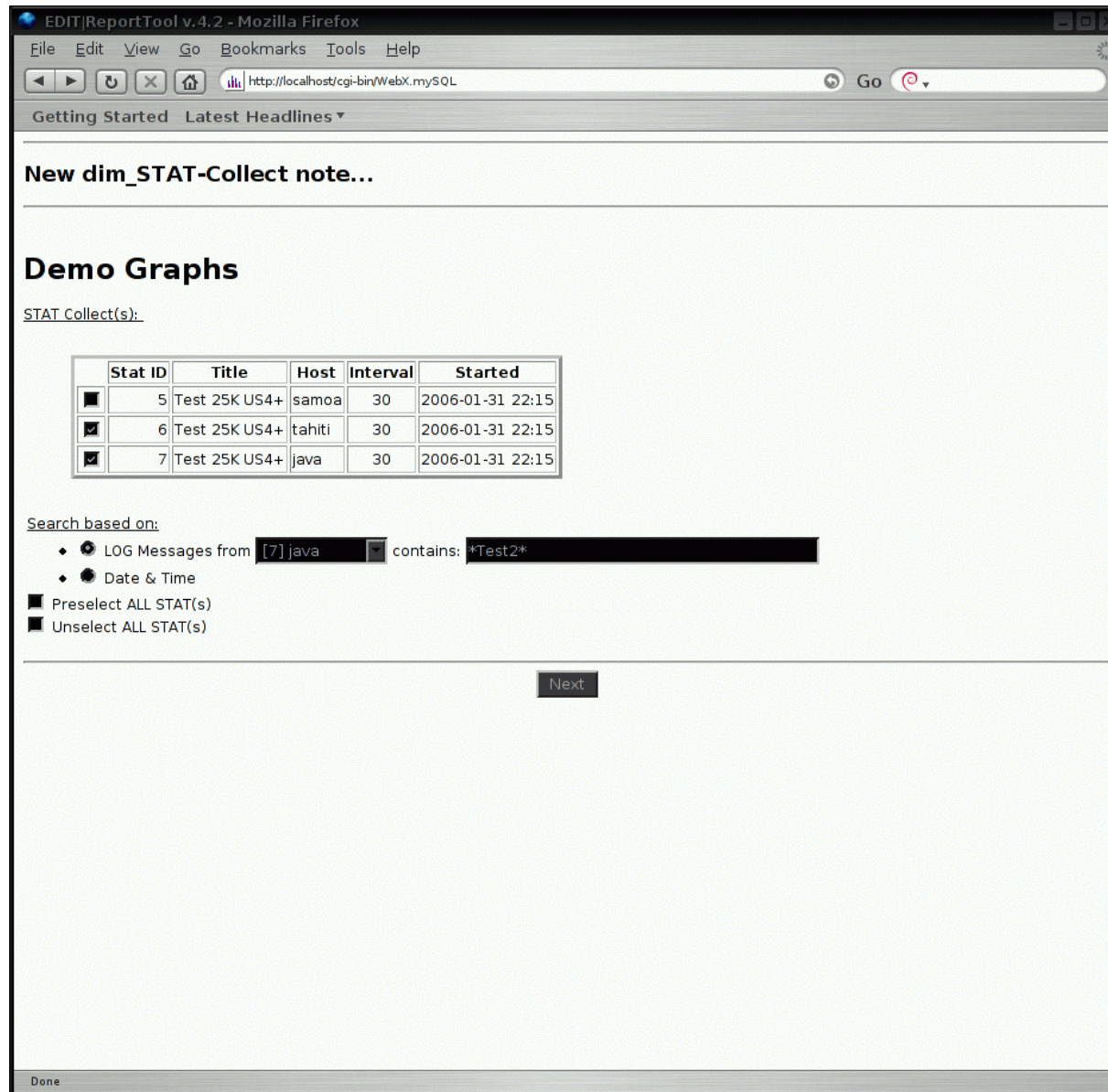
We are on the Step-1 here, and if you don't have any data collected, you may get them from the 'Default' demo collect:

- ◇ **Server** : localhost
- ◇ **Port** : Default
- ◇ **Database** : Default

[Next]...

NOTE: the interface becomes more optimized and more extended with each new release, so screen shots are probably not everywhere up to date.

Add New Note -- dim_STAT Collect, Step2



Choose STAT collect here and Search mode. We have already the log messages from the 'java' host, each message was added before any of the tests started, so it's quite easy to find them, corresponding to the time interval for each test. Otherwise we can always do a 'Date and Time' search, but you'll quickly understand that that is much more painful compared to LOG messages.

NOTE: with version 8.0, more options added to simplify reporting:

- ◊ replay the same time slices for N days (in Date and Time)
- ◊ auto include time/date into generated graph titles

◇ replace on-the-fly some part (max 5) of the LOG messages

Add New Note -- dim_STAT Collect, Step3

EDIT/ReportTool v.4.2 - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://localhost/cgi-bin/WebX.mySQL

Getting Started Latest Headlines ▾

Demo Graphs

Stat ID	Collect Title	Host	Started	LOG
6	Test 25K US4+	tahiti	2006-01-31 22:15	
7	Test 25K US4+	java	2006-01-31 22:15	*

Begin End STAT(s) Graph Title

STAT(s): Per-Host View

- ☐ CPU %Busy
- ☐ Top-4 CPU on Interrupts
- ☐ Net Packet/sec
- ☐ Queued, Blocked, Swapped runnable processes
- ☐ Top-10 Processes using CPU%
- ☐ Proc Number per User
- ☐ CPU Cross-Calls
- ☐ Memory Free List(KB)
- ☐ NIC Pakets/sec
- ☐ Spin Mutex/sec
- ☐ Top-10 Users using CPU%
- ☐ CPU Context Switch
- ☐ Net Bytes/sec
- ☐ Page In/Out (KB)
- ☐ Spin/sec Read/Write
- ☐ User CPU Time
- ☐ User System Calls
- ☐ CPU Thread Migration
- ☐ Net Errors
- ☐ Page Scanner Rate (Pg/sec)
- ☐ System Calls/sec

STAT(s): Multi-Host View

- ☐ CPU: Ustr%
- ☐ CPU: Sys%
- ☐ CPU: Busy%
- ☐ RAM: Free List (KB)
- ☐ Run queue
- ☐ Blocked processes
- ☐ netLOAD: SUM Bytes/s
- ☐ netLOAD: SUM Packets/s
- ☐ netLOAD: SUM Collisions/s
- ☐ netLOAD: SUM I_errors/s
- ☐ netLOAD: SUM Nocarput
- ☐ netLOAD: SUM O_errors/s

Between Messages

15 min. Around of Messages

- [2006-01-31 22:15] *FIRST data in Collect*
- [2006-02-06 16:38] *Test2 25K US4+ Conf3 3x40threads*
- [2006-02-06 16:54] *Test2 25K US4+ Conf3 3x40threads R2*
- [2006-02-06 17:08] *Test2 25K US4+ Conf3 3x40threads R3*
- [2006-02-06 17:22] *Test2 25K US4+ Conf3 3x40threads R4 +pset*
- [2006-02-06 17:39] *Test2 25K US4+ Conf3 3x50threads*

Done

Now we need to choose the type of graphs we want to see and the time interval of them.

NOTE:

- ◇ All the Per-Host STATs are Bookmarks. The more you created Bookmarks during analyzing, the more data you can generate in for report.
- ◇ When I selected the two hosts, the tool gave me also Multi-Host STATs, depending on stat commands to be present or not. Each STAT (like in Multi-Host Analyze) will put all requested hosts onto single graph.

Add New Note -- dim_STAT Collect, Step3 continue

EDIT|ReportTool v.4.2 - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://localhost/cgi-bin/WebX.mysql

Getting Started Latest Headlines ▾

6	Test 25K US4+	tahiti	2006-01-31 22:15
7	Test 25K US4+	java	2006-01-31 22:15 *

Begin End STAT(s) Graph Title

STAT(s): Per-Host View

<input checked="" type="checkbox"/> CPU %Busy	<input type="checkbox"/> CPU Cross-Calls	<input type="checkbox"/> CPU Context Switch	<input type="checkbox"/> CPU Thread Migration
<input type="checkbox"/> Top-4 CPU on Interrupts	<input type="checkbox"/> Memory Free List(KB)	<input type="checkbox"/> Net Bytes/sec	<input type="checkbox"/> Net Errors
<input type="checkbox"/> Net Packet/sec	<input type="checkbox"/> NIC Pakets/sec	<input type="checkbox"/> Page In/Out (KB)	<input type="checkbox"/> Page Scanner Rate (Pg/sec)
<input checked="" type="checkbox"/> Queued, Blocked, Swapped runnable processes	<input checked="" type="checkbox"/> Spin Mutex/sec	<input type="checkbox"/> Spin/sec Read/Write	<input checked="" type="checkbox"/> System Calls/sec
<input type="checkbox"/> Top-10 Processes using CPU%	<input type="checkbox"/> Top-10 Users using CPU%	<input type="checkbox"/> User CPU Time Ustr/Sys	<input type="checkbox"/> User System Calls
<input type="checkbox"/> Proc Number per User			

STAT(s): Multi-Host View

<input type="checkbox"/> CPU: Ustr%
<input type="checkbox"/> CPU: Sys%
<input checked="" type="checkbox"/> CPU: Busy%
<input type="checkbox"/> RAM: Free List (KB)
<input type="checkbox"/> Run queue
<input type="checkbox"/> Blocked processes
<input checked="" type="checkbox"/> netLOAD: SUM Bytes/s
<input checked="" type="checkbox"/> netLOAD: SUM Packets/s
<input type="checkbox"/> netLOAD: SUM Collisions/s
<input type="checkbox"/> netLOAD: SUM I_errors/s
<input type="checkbox"/> netLOAD: SUM Nocomput
<input type="checkbox"/> netLOAD: SUM O_errors/s

Between Messages

15 min. After Messages

- [2006-01-31 22:15] FIRST data in Collect
- ☒ [2006-02-06 16:38] Test2 25K US4+ Conf3 3x40threads
- ☒ [2006-02-06 16:54] Test2 25K US4+ Conf3 3x40threads R2
- ☒ [2006-02-06 17:08] Test2 25K US4+ Conf3 3x40threads R3
- ☒ [2006-02-06 17:22] Test2 25K US4+ Conf3 3x40threads R4 +pset
- [2006-02-06 17:39] Test2 25K US4+ Conf3 3x50threads
- [2006-03-28 20:40] LAST data in Collect

Next

Done

Here we're choosing:

- ◇ per host : CPU busy%, Run queue, Mutex spin, System calls/s
- ◇ multi-host : CPU busy%, Network load bytes/s and packets/s

Time interval: as we know each test run for ~15min, we can choose a time interval of '15 min. After each LOG message.

[Next]...

Add New Note -- dim_STAT Collect, Step4

New dim_STAT-Collect note...

Demo Graphs

Stat ID	Collect Title	Host	Started	LOG
6	Test 25K US4+	tahiti	2006-01-31 22:15	
7	Test 25K US4+	java	2006-01-31 22:15	*

Begin	End	STAT(s)	Graph Title
16:38 2006-02-06	16:53 2006-02-06	[CPU] [RunQueue] [SpinMtx] [SysCalls] [CPU_US] [netLOAD:1:allbyt] [netLOAD:1:allpac]	Test2 25K US4+ Conf3 3x40threads
16:53 2006-02-06	17:09 2006-02-06	[CPU] [RunQueue] [SpinMtx] [SysCalls] [CPU_US] [netLOAD:1:allbyt] [netLOAD:1:allpac]	Test2 25K US4+ Conf3 3x40threads R2
17:07 2006-02-06	17:23 2006-02-06	[CPU] [RunQueue] [SpinMtx] [SysCalls] [CPU_US] [netLOAD:1:allbyt] [netLOAD:1:allpac]	Test2 25K US4+ Conf3 3x40threads R3
17:22 2006-02-06	17:37 2006-02-06	[CPU] [RunQueue] [SpinMtx] [SysCalls] [CPU_US] [netLOAD:1:allbyt] [netLOAD:1:allpac]	Test2 25K US4+ Conf3 3x40threads R4 +pset

STAT(s): Per-Host View

<input checked="" type="checkbox"/> CPU %Busy	<input type="checkbox"/> CPU Cross-Calls	<input type="checkbox"/> CPU Context Switch	<input type="checkbox"/> CPU Thread Migration
<input type="checkbox"/> Top-4 CPU on Interrupts	<input type="checkbox"/> Memory Free List(KB)	<input type="checkbox"/> Net Bytes/sec	<input type="checkbox"/> Net Errors
<input type="checkbox"/> Net Packet/sec	<input type="checkbox"/> NIC Pakets/sec	<input type="checkbox"/> Page In/Out (KB)	<input type="checkbox"/> Page Scanner Rate (Pg/sec)
<input checked="" type="checkbox"/> Queued, Blocked, Swapped runnable processes	<input checked="" type="checkbox"/> Spin Mutex/sec	<input type="checkbox"/> Spin/sec Read/Write	<input checked="" type="checkbox"/> System Calls/sec
<input type="checkbox"/> Top-10 Processes using CPU%	<input type="checkbox"/> Top-10 Users using CPU%	<input type="checkbox"/> User CPU Time	<input type="checkbox"/> User System Calls
<input type="checkbox"/> Proc Number per User			

STAT(s): Multi-Host View

<input type="checkbox"/> CPU: Ustr%
<input type="checkbox"/> CPU: Sys%
<input checked="" type="checkbox"/> CPU: Busy%
<input type="checkbox"/> RAM: Free List (KB)
<input type="checkbox"/> Run queue
<input type="checkbox"/> Blocked processes
<input checked="" type="checkbox"/> netLOAD: SUM Bytes/s
<input checked="" type="checkbox"/> netLOAD: SUM Packets/s
<input type="checkbox"/> netLOAD: SUM Collisions/s
<input type="checkbox"/> netLOAD: SUM I_errors/s
<input type="checkbox"/> netLOAD: SUM Ncanput

So, this looks OK. I've got my STATs selected with a pre-populated graph title (from the LOG message). BTW, you may see that all your previously selected STATs are pre-selected here (the selection is saved via cookies and specific to each database name).

[Finish] ...

Add New Note -- dim_STAT Collect, Step5

EDIT|ReportTool v.4.2 - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://localhost/cgi-bin/WebX.mysql

Getting Started Latest Headlines ▾

New dim_STAT-Collect note...

Stat ID	Collect Title	Host	Started
6	Test 25K US4+	tahiti	2006-01-31 22:15
7	Test 25K US4+	java	2006-01-31 22:15

Title:

Begin	End	STAT(s)	Graph Title
16:38 2006-02-06	16:53 2006-02-06	[CPU] [RunQueue] [SpinMtx] [SysCalls] [CPU_US] [netLOAD:1:allbyt] [netLOAD:1:allpac]	Test2 25K US4+ Conf3 3x40threads
16:53 2006-02-06	17:09 2006-02-06	[CPU] [RunQueue] [SpinMtx] [SysCalls] [CPU_US] [netLOAD:1:allbyt] [netLOAD:1:allpac]	Test2 25K US4+ Conf3 3x40threads R2
17:07 2006-02-06	17:23 2006-02-06	[CPU] [RunQueue] [SpinMtx] [SysCalls] [CPU_US] [netLOAD:1:allbyt] [netLOAD:1:allpac]	Test2 25K US4+ Conf3 3x40threads R3
17:22 2006-02-06	17:37 2006-02-06	[CPU] [RunQueue] [SpinMtx] [SysCalls] [CPU_US] [netLOAD:1:allbyt] [netLOAD:1:allpac]	Test2 25K US4+ Conf3 3x40threads R4 +pset

Order by:

Graphics

Mode: Style:

Background: Size: x

☐ Force Graphs alignment

☒ Force Data Gap Completion

☐ Auto-AVG too large Graphs

☐ Show LOG Messages as matching:

☐ Show TASK(s)

☐ Show Processing

Done

Here you have to specify the graphs parameters:

- ◇ Main title
- ◇ per graph title
- ◇ order generation
- ◇ graph mode, style, size, etc.
- ◇ Auto-AVG: good to select if you have too large time intervals and your graph become too dense
- ◇ Show LOG/TASK (as during analyze)
- ◇ Show processing - get generation output on the browser. Not all browsers work correctly with this feature, some are waiting for an EOF before they show something. If you don't choose this option, processing output is always printed into a /tmp/.report.log file on the Report Tool server side.

[Save]...

Now you're free to start doing something else, because your machine is working for you and all you have to do is sit back and relax. Once you get use to the report tool, you'll ask it to generate A LOT OF graphs at the same time and you've time on your hands to do something else.

Add New Note -- dim_STAT Collect Result

Demo Graphs

- [Test2 25K US4+ Conf3 3x40threads](#)
- [Test2 25K US4+ Conf3 3x40threads R2](#)
- [Test2 25K US4+ Conf3 3x40threads R3](#)
- [Test2 25K US4+ Conf3 3x40threads R4 +pset](#)

Confidential Level: ☒ Personal ☒ WCB Only ☒ Sun Internal Only ☒ Customer
Language: ☒ English ☒ French ☒ German ☒ Japanese

Edit Note
 Change Note Attributes
 Change Note's Tree Attributes

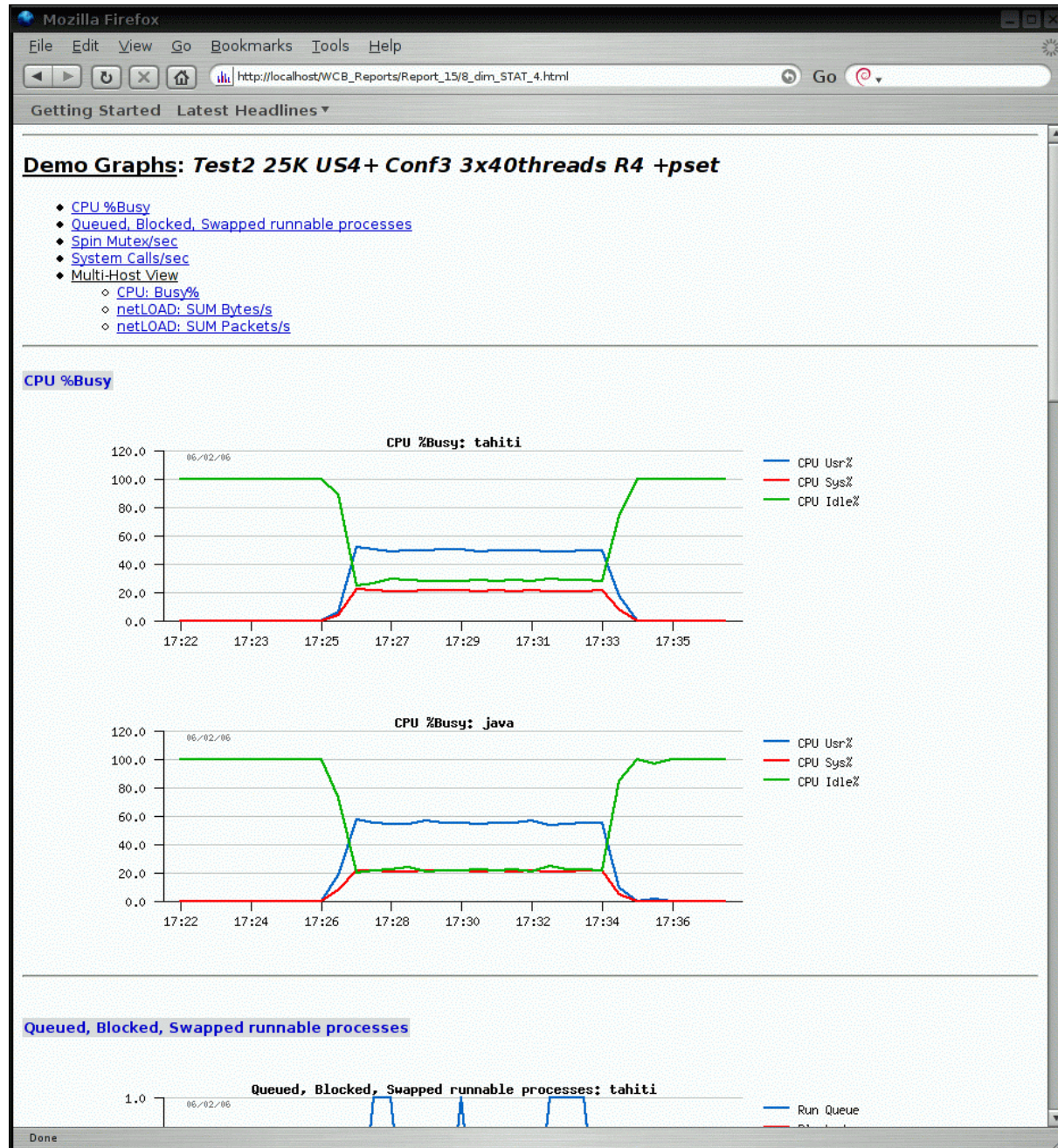
Here is the final result after all the graphs are generated!

Click on a link to see the graph results.

NOTE: If you remember, I've selected generating order by Collect , and what I see now is a list of collects first, and each collect link will show me all selected STAT graphs for the same given STAT collect.

Now, if I select the by STATs order generation - I'll see here a STAT list, and each link will show me the same STAT metric for different collects on single page...

Add New Note -- dim_STAT Collect Contents, ordered by:Collect



Add New Note -- dim_STAT Collect Result per STATs

Demo Graphs

- [CPU %Busy](#)
- [Queued, Blocked, Swapped runnable processes](#)
- [Spin Mutex/sec](#)
- [System Calls/sec](#)
- [CPU: Busy%](#)
- [netLOAD: SUM Bytes/s](#)
- [netLOAD: SUM Packets/s](#)

Confidential Level: ☒ Personal ☒ WCB Only ☒ Sun Internal Only ☒ Customer

Language: ☒ English ☒ French ☒ German ☒ Japanese

[Edit Note](#)[Change Note Attributes](#)[Change Note's Tree Attributes](#)

As you see here, the single STAT link contains all given collects, so if you want to compare the network usage in different cases, just click on either the bytes/sec or the packets/sec link.

Add New Note -- dim_STAT Collect Contents, ordered by:STATS



Edit Report, next...

EDIT|ReportTool v.4.2 - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://localhost/cgi-bin/WebX.mysql?Report_ID=15&X_PAGE=/Report/x.edit&ShowNOTES=YES Go

Getting Started Latest Headlines ▾

General Information

Overview: This is a general demonstration project to show how Report Tool works. Any comments are welcome as well any RFE, suggestions, etc...

Contact Information:

- Dimitri
- dim_STAT@sun.com

Keywords: dim_STAT, Report

Hardware Configuration

Server(s):

- tahiti: E25K domain, 2xSB, 8CPU 1200Mhz US4+
- java: E25K domain, 3xSB, 12CPU 1200Mhz US4+

Storage:

- HDS 4xFC-2Gbit 8xLUSEs 300GB (RAID5)

13:05 2006-04-04 (Dimitri)

Platform Overview @

Software Configuration

System:

- Solaris 10
- VxVM 4.0
- UFS

Application(s):

- Oracle 9.2.0.6 64bit
- JVM 1.5 64bit
- Customer Application

12:13 2006-04-04 (Dimitri)

SysINFO: tahiti @

12:24 2006-04-04 (Dimitri)

SysINFO: java @

Report

Just several graphs here to get an idea about observed system activity.

13:12 2006-04-04 (Dimitri)

Demo Graphs per STAT

13:12 2006-04-04 (Dimitri)

Demo Graphs

Hide Note(s) Preview Generate Home

Done

Edit Report -- Cut

EDIT|ReportTool v.4.2 - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://localhost/cgi-bin/WebX.mysql?Report_ID=15&X_PAGE=/Report/x.edit&ShowNOTES=YES Go

Getting Started Latest Headlines ▾

General Information

Overview: This is a general demonstration project to show how Report Tool works. Any comments are welcome as well any RFE, suggestions, etc...

Contact Information:

- Dimitri
- dim_STAT@sun.com

Keywords: dim_STAT, Report

Hardware Configuration

Server(s):

- tahiti: E25K domain, 2xSB, 8CPU 1200Mhz US4+
- java: E25K domain, 3xSB, 12CPU 1200Mhz US4+

Storage:

- HDS 4xFC-2Gbit 8xLUSEs 300GB (RAID5)

[Platform Overview @](#) 13:05 2006-04-04 (Dimitri)

Software Configuration

System:

- Solaris 10
- VxVM 4.0
- UFS

Application(s):

- Oracle 9.2.0.6 64bit
- JVM 1.5 64bit
- Customer Application

[SysINFO: tahiti @](#) 12:13 2006-04-04 (Dimitri)

[SysINFO: java @](#) 12:24 2006-04-04 (Dimitri)

Report

Just several graphs here to get an idea about observed system activity.

- CUT this note into clipboard 13:12 2006-04-04 (Dimitri)

[Demo Graphs per STAT](#) 13:12 2006-04-04 (Dimitri)

[Demo Graphs](#)

Hide Note(s) Preview Generate Home

http://localhost/cgi-bin/WebX.mysql?Report_ID=15&CUT_Note_ID=9&ShowNOTES=YES&PrevID=4#Note_4

Last thing now: I don't want to see my 'per STAT' first in Report section, just let's move it at the end...

Click on [Cut] icon, then [Paste] where you want ([Trash] icon does delete operation!)

Edit Report -- Paste!

EDIT|ReportTool v.4.2 - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://localhost/cgi-bin/WebX.mysql/Report/x.edit?Report_ID=15&CUT_Note_ID=9&ShowNOTES Go

Getting Started Latest Headlines

Demonstration Report

General Information

Overview: This is a general demonstration project to show how Report Tool works. Any comments are welcome as well any RFE, suggestions, etc...

Contact Information:

- Dimitri
- dim_STAT@sun.com

Keywords: dim_STAT, Report

Hardware Configuration

Server(s):

- tahiti: E25K domain, 2xSB, 8CPU 1200Mhz US4+
- java: E25K domain, 3xSB, 12CPU 1200Mhz US4+

Storage:

- HDS 4xFC-2Gbit 8xLUSEs 300GB (RAID5)

[Platform Overview](#) @ 13:05 2006-04-04 (Dimitri)

Software Configuration

System:

- Solaris 10
- VxVM 4.0
- UFS

Application(s):

- Oracle 9.2.0.6 64bit
- JVM 1.5 64bit
- Customer Application

[SysINFO: tahiti](#) @ 12:13 2006-04-04 (Dimitri)

[SysINFO: java](#) @ 12:24 2006-04-04 (Dimitri)

Report

Just several graphs here to get an idea about observed system activity.

- PASTE note from clipboard after this one, on the same level tree

[Demo Graphs](#) 13:12 2006-04-04 (Dimitri)

Hide Note(s) Preview Generate Home

http://localhost/cgi-bin/WebX.mysql/Report/x.edit?Report_ID=15&CUT_Note_ID=9&PASTE_Note_ID=8&PASTE_Direction=DOWN&ShowNOTES=YES#Note_8

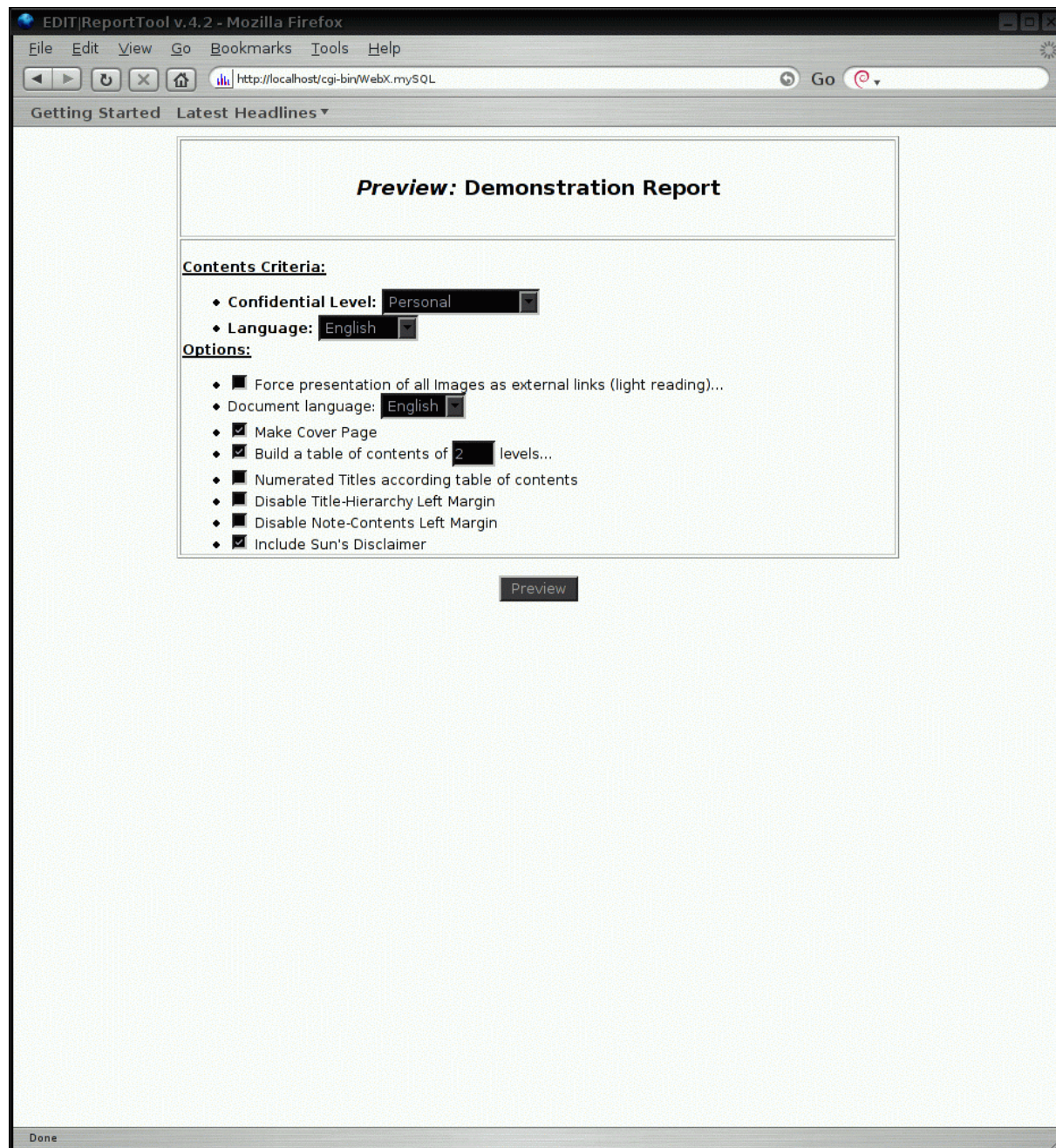
Edit Report -- Pasted...

The screenshot shows a Mozilla Firefox browser window titled "EDIT|ReportTool v.4.2 - Mozilla Firefox". The address bar displays the URL: `http://localhost/cgi-bin/WebX.mysql/Report/x.edit?Report_ID=15&CUT_Note_ID=9&PASTE_Note`. The browser window shows a web application with a navigation menu at the top: "Getting Started" and "Latest Headlines". The main content area is divided into several sections, each with a blue header bar and a list of links (down arrow, right arrow, and a document icon).

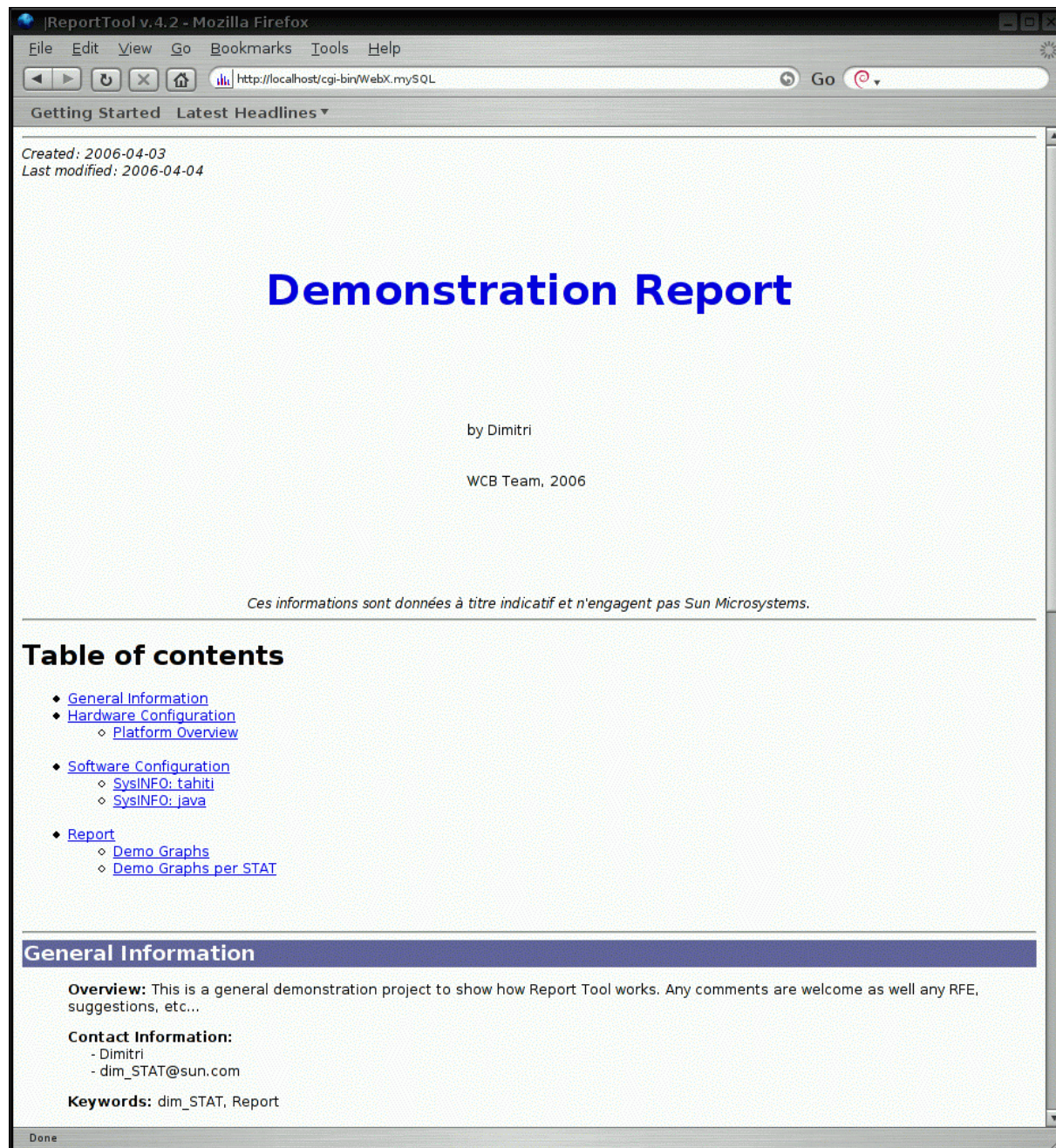
- General Information**
 - Overview:** This is a general demonstration project to show how Report Tool works. Any comments are welcome as well any RFE, suggestions, etc...
 - Contact Information:**
 - Dimitri
 - dim_STAT@sun.com
 - Keywords:** dim_STAT, Report
- Hardware Configuration**
 - Server(s):**
 - tahiti: E25K domain, 2xSB, 8CPU 1200Mhz US4+
 - java: E25K domain, 3xSB, 12CPU 1200Mhz US4+
 - Storage:**
 - HDS 4xFC-2Gbit 8xLUSEs 300GB (RAID5)
- Platform Overview** @ 13:05 2006-04-04 (Dimitri)
- Software Configuration**
 - System:**
 - Solaris 10
 - VxVM 4.0
 - UFS
 - Application(s):**
 - Oracle 9.2.0.6 64bit
 - JVM 1.5 64bit
 - Customer Application
- SysINFO: tahiti** @ 12:13 2006-04-04 (Dimitri)
- SysINFO: java** @ 12:24 2006-04-04 (Dimitri)
- Report**
 - Just several graphs here to get an idea about observed system activity.
- Demo Graphs** 13:12 2006-04-04 (Dimitri)
- Demo Graphs per STAT** 13:12 2006-04-04 (Dimitri)

At the bottom of the page, there is a navigation bar with buttons: "Hide Note(s)", "Preview", "Generate", and "Home". The status bar at the very bottom shows "Done".

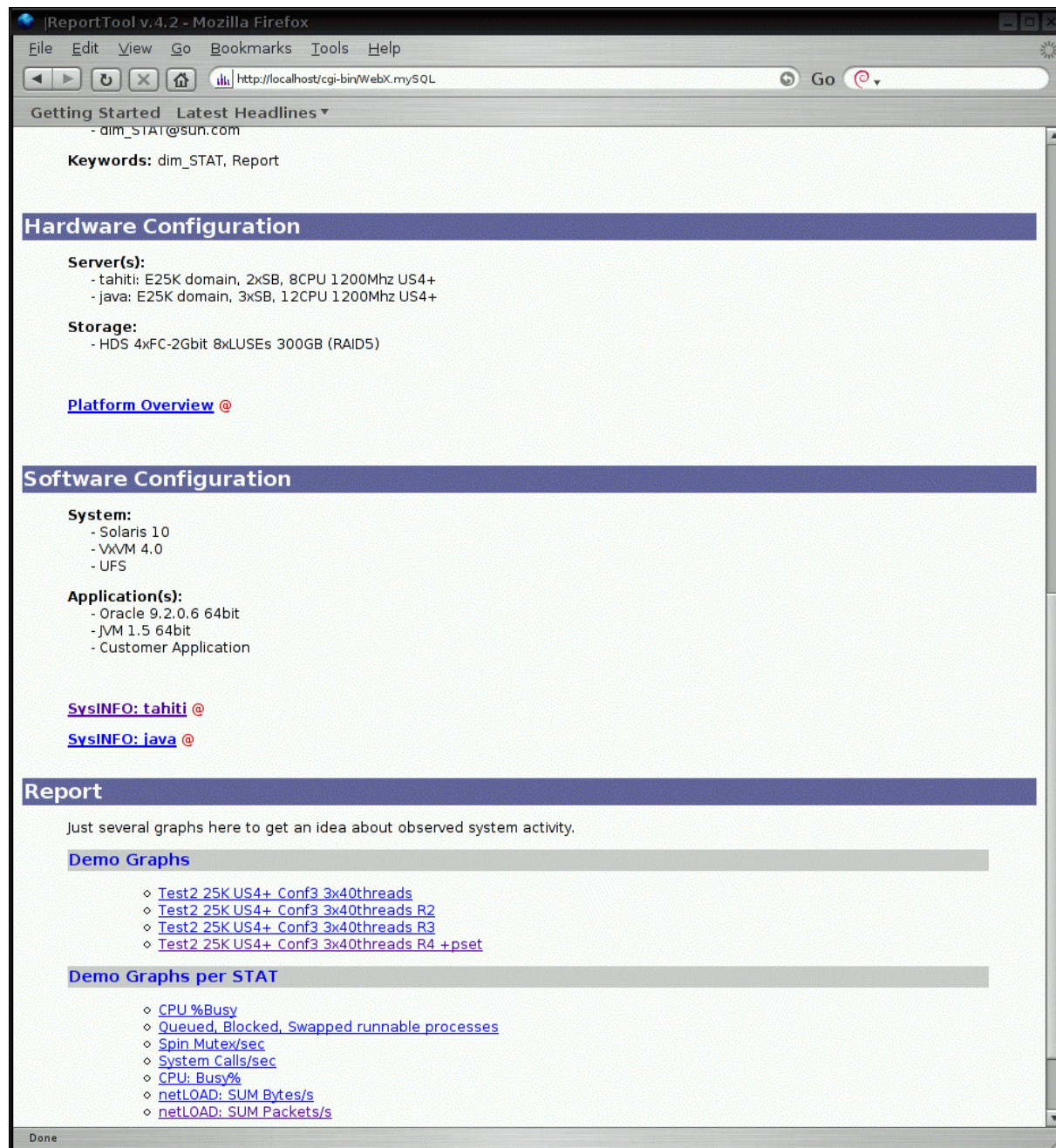
Edit Report -- Preview



Edit Report -- Preview Output



Edit Report -- Preview Output2



Generate Report

The screenshot shows a web browser window titled "EDIT|ReportTool v.4.2 - Mozilla Firefox". The address bar shows "http://localhost/cgi-bin/WebX.mysql". The page has a navigation bar with "Getting Started" and "Latest Headlines". The main content area is titled "Generate: Demonstration Report".

Contents Criteria:

- ◆ Confidential Level: Personal
- ◆ Language: English

Options:

- ◆ ☐ Force presentation of all images as external links (light reading)...
- ◆ Document language: English
- ◆ ☒ Make Cover Page
- ◆ ☒ Build a table of contents of 2 levels...
- ◆ ☐ Numerated Titles according table of contents
- ◆ ☐ Disable Title-Hierarchy Left Margin
- ◆ ☐ Disable Note-Contents Left Margin
- ◆ ☒ Include Sun's Disclaimer

Additionally:

- ◆ ☐ PNG -- Convert all (if any) Java Applets into PNG images...
- ◆ ☒ PDF -- Make a single PDF document for local download/viewing...
- ◆ ☒ TAR -- Make a "tar.Z" archive for local download/viewing...
- ◆ Publish Generated Report into: Public

Done

Generated Report documents

Build PDF...

PAGES: 46

BYTES: 764050

Done!


You may access now your Report:

- ♦ on-line: /Reports/Report_15.html
- ♦ PDF: [Report_15.pdf](#)
- ♦ TAR.Z: [Report_15.tar.Z](#)



Report Tool Home

Mobil Report Tool

	Report ID	Title	Created	Last Modified	State
	15	Demonstration Report	2006-04-03 by Dimitri	2006-04-04 13:41	In Progress...

[New Report](#)
[Edit](#)
[Preview](#)
[Generate](#)
[Finished!](#)
[Export](#)
[Import](#)
[Remove](#)

THAT'S ALL, folks! :))

The export file of this demonstration report may be found within dim_STAT distribution as 'ExpReport_15.tar.Z'. You may import and play with it as long as you want! :))

Also, for good first exercise you may try to generate your first graphs from 'Demo collect' giving by default in your dim_STAT database!...

Additional Tools

Since version 5, additional tools are shipping with the package, but it seems I forgot to mention them explicitly and a lot of users didn't know about it.

Java2GIF Tool

This tool converts HTML pages containing dim_STAT graphs as **Java applets** to HTML pages with GIF images. This is very useful for reporting, printing, etc. (of course you don't need it if you used PNG :-))

Installed in : /apps/Java2GIF

Requirements :

- ◇ JRE or JDK installed on the system
- ◇ X11 DISPLAY positioned for image output

Configuration : edit the "j2gif.sh" script to point to the right PATH for your "java" binary

Usage :

```
$ j2gif.sh /full/path/to/dir/with/your/html/files
```

Example :

- ◇ Analyzing dim_STAT Java applet graphs time to time you "Save As" your pages into /Report/J
- ◇ Once finished, make a backup first of your files
- ◇ Execute:

```
$ /apps/Java2GIF/j2gif.sh /Report/J
```

- ◇ That's all :-)

Java2PNG Tool

Similar to Java2GIF, but with few differences:

- ◇ doesn't need the X11 server for output
- ◇ processing execution is much faster compared to Java2GIF
- ◇ uses PNG image format
- ◇ doesn't support histogram mode

Installed in : /apps/ADMIN

Requirement : -

Configuration : -

Usage :

```
$ cd /apps/ADMIN
$ Java2PNG /full/path/to/dir/with/your/html/files
```

HTMLDOC Tool

Installed in : /apps/htmldoc

Usage : (RTFM first! :-))

```
$ cat /apps/htmldoc/README
$ /apps/htmldoc/bin/htmldoc --webpage --header t.D -f Report.pdf *.html Repo
```

README

This is a short README about "htmldoc" program.

This program is free and I've found it very useful for making printable and well presented HTML ==> PDF documents. Of course, HTML is great for screen viewing, but when you should bring a printed version - it's not so simple to obtain something presentable in easy way... Also, I like to send PDF documents, they are small and very portable :))

The home page of "htmldoc" tool is:

<http://www.easysw.com/htmldoc>

You may download and compile the last version from this site. But as people are lazy by defenition :)) , I've pre-installed not last, but well working binary of this great tool...

For detailed description you may start to read the htmldoc manual, but if you are lazy as me :)), you may just start:

```
/apps/htmldoc/bin/htmldoc --webpage --header t.D -f Report.pdf *.html Report/*.
```

to get PDF document (Report.pdf) from collection of HTML files...

That's all! :))

-Dimitri

FAQ

Sizing of dim_STAT Instance...

This problem is simple: there are no sizing rules. :))

Disk space: it depends only on the size of the information collected. On the Preferences page you can see the space used by the current database and the size of

your biggest file. You cannot reduce the file sizes by data recycle, however it's possible now with a Convert Engine operation (as the table will be fully recreated) - keep in mind anyway that InnoDB is using much more disk space than MyISAM.

CPU: for a collect your CPU is hardly used at all. However, once you start a query via the Web interface you will access a big amount of data! Your query may use all of CPU. Normally query execution time is relatively short, but depends directly on amount of data demanded.

Separated databases are fine when you need different administrative tasks regarding the data collected. For example, it may be annoying when somebody is loading a large amount of data at the same time you're trying to analyze something. This will create additional locks and slow down the performance for others. MySQL (in the version used by dim_STAT) uses "table locking", so there can be only a single writer at the same time, and write operations are exclusive (no reads at the same time). If you use your own database you have less reasons to blame others.

A desktop running dim_STAT server could be very heavily used, or not used at all. It all depends only on what you're doing with it.

I've started my collects but it seems that nothing gets collected?

First of all be sure that:

- ◊ you've installed the STAT-service package on this host and **started** it.
- ◊ be sure your server is seen with "Green LED" by dim_STAT Server

If everything seems to be correct in that sense, check the output of your '/etc/STATsrv/log/access.log' file.

Syntax of text matching pattern

Quite often in the dim_STAT interface you may see an input text field that filters values or attributes matching a specified pattern. By default they are filled with '*' (means all), but what kind of syntax does it accept?

Pattern by example:

- ◊ * - any character or none
- ◊ ? - any single character
- ◊ [amp] - single character and one from 'a', 'p', or 'm'
- ◊ [a-z] - any single character between 'a' and 'z' (both included)

- ◇ [^a-z] - any single character NOT between 'a' and 'z' (both included)
- ◇ !Pattern - apply NOT condition on the whole pattern
- ◇ Pattern || Pattern - apply OR condition between two patterns (or more)
- ◇ Pattern && Pattern - apply AND condition between two patterns (or more), has higher priority vs OR

Examples matching LOG messages:

- ◇ *Test??* - match all messages having TestNN in title
- ◇ *Test??* && *End* - match all TestNN messages containing End
- ◇ *Test??* && *End* || *Begin* - match all TestNN messages containing End or Begin
- ◇ !*Test??* && *End* || *Begin* - match any messages except TestNN and containing End or Begin

When will you upgrade to the newer MySQL version?

But why?... :-))

Should we change a good old working horse just because it's old?? It worked fine for over 10 years now, and does exactly what it needs to do. And MyISAM is not working better in MySQL4 or MySQL5.

MyISAM is really great for its binary compatibility between all platforms - it's simplifying so many things! :-)

In some cases it make sense to move some critical tables from MyISAM to InnoDB engine and get advantage of a data protection against crashes...

As well should be interesting to ship dim_STAT in parallel with a version of PostgreSQL!! But that's another story...

UPDATE : since version 9.0 - dim_STAT is based on MySQL 5.5 (GA) and include both MyISAM and InnoDB engines, and you're free at any time to convert your database to the best situated engine for your activity! :-)

With multiple hosts to monitor, is it possible to graph them together?..

It's exactly what do you have with a Multi-Host Analyze feature. As well when you have hundreds of hosts you may even group stats by N first/last letters in the hostname, etc.. Data are here, and you just play with them.. :-)

How easy is it to integrate any new stats to monitor, including DTrace stuff?

Usually it's quite straight forward to add new stat commands into dim_STAT. But at any time feel free to ask for help from the [dim_STAT Users Group](#) - as well there are already several debug hints were discussed:

- ◇ [add-on @Linux](#)
- ◇ [Disk space usage add-on](#)

Regarding DTrace, once you have a working script with regular and well formatted output - usually it takes 5 minute to integrate it as a new dim_STAT Add-On. Solaris STAT-service already contains some DTrace scripts (for example, see: IOpatt Add-On)...

Could I get the raw data via dim_STAT-CLI instead of the graphs?...

Yes, of course!

See "-Data" option within dim_STAT-CLI.

I have a Windows machine to monitor remote UNIX boxes.... Any help?..

Sorry, there is no dim_STAT distribution for Windoze :))

But(!) if you absolutely want to work under Win, you may install VirtualBox for free (from [VirtualBox](#)) on it, and then within VirtualBox install Linux or Solaris (there are several mini distros available across Internet (Pocket Solaris: [Milax](#))), and monitor your servers from Windoze, but via VirtualBox... (as well for \$200 you may buy a new PC and setup it in native with Solaris/x86 or Linux :-))

Full Working cycle Example

TBD...

Freeware End User License

LICENSE

This software is released as "freeware". You are encouraged to redistribute unmodified copies of this software, as long as no fee is charged for the software, directly or indirectly, separately or as part of ("bundled with") another software product, without the express permission of the author. You may not attempt to reverse compile, modify or disassemble the software in whole or in part.

SUPPORT, BUG REPORTS, SUGGESTIONS

You are encouraged to send bug reports and suggestions. This software is not supported. Hence, your technical questions may or may not be answered. Questions, bug reports, comments and suggestions should all be sent to: Dimitri KRAVTCHUK (dimitri.kravtchuk@free.fr) or to dim_STAT Users Group @Google (<http://groups.google.com/group/dimstat>).

DISCLAIMER

ANY USE BY YOU OF THE SOFTWARE IS AT YOUR OWN RISK. THE SOFTWARE ARE PROVIDED FOR USE "AS IS" WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED BY LAW, THE AUTHOR (**) DISCLAIMS ALL WARRANTIES OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. THE AUTHOR (**) IS NOT OBLIGATED TO PROVIDE ANY UPDATES TO THE SOFTWARE.

**Dimitri KRAVTCHUK

GPL v2 License

GNU General Public License

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble
=====

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:
 - a. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
 - b. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
 - c. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:
 - a. Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
 - b. Accompany it with a written offer, valid for at least three years, to give any third-party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
 - c. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.
5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.
6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation

excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.
12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

=====

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to

attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
ONE LINE TO GIVE THE PROGRAM'S NAME AND A BRIEF IDEA OF WHAT IT DOES.
Copyright (C) YYYY NAME OF AUTHOR
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) 19YY NAME OF AUTHOR
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
SIGNATURE OF TY COON, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.
