
TraffAcct

A General Purpose Network Traffic Accountant

Users Guide and Reference

Version 1.3

June 2002



HUGHES
TECHNOLOGIES

Table of Contents

Introduction	1
Installation	2
Ember.....	2
SNMP Module	2
Web Server	2
Crontab	3
Configuration.....	4
Devices	4
Circuits	4
Users.....	5
Circuit Identifier	6
traffacct.conf file	7
Reports.....	8
Format Keys.....	8
Format Strings.....	9
Example	9
Collectors.....	10
Ethernet.....	10
Serial.....	10
ISDN.....	10
IP.....	11

Introduction

There are many software packages available, both free and commercial, that provide some level of traffic monitoring or traffic logging functionality. TraffAcct is another such application with a very specific focus. Unlike most packages, TraffAcct is aimed at the age-old problem of billing for network traffic rather than just providing a graphical representation of network utilisation. Quite simply, TraffAcct gathers network circuit traffic statistics and stores them. The reporting tools included with the software then allow you to generate reports on that stored data. The data is never compressed or “rolled” so you never lose accounting information.

Each aspect of TraffAcct has been designed to be as simple and as flexible as possible. The data acquisition framework can be extended to gather information from new types of data sources very rapidly. The report generator produces reports based on formats provided in user specified report definitions. The web interface is template driven allowing the “look’n’feel” of the web pages to be changed at will without impacting upon the operation of the application. Customising the software to suit your specific requirements can be done very simply and without modifying the source code of the core TraffAcct distribution.

Although TraffAcct provides a client oriented web interface, it does not generate the visually appealing graphs that are expected by some users. All output from the software is generated by the report generator and shows the accounting information in numbers rather than pictures. If you require a graphical representation of network link utilisation you should look into the MRTG package. If you require more sophisticated reporting or real-time interpretation of traffic statistics you should look at a commercial application.

Installation

The entire TraffAcct package has been implemented in the Ember language. Ember is a freely available scripting language developed by Hughes Technologies and distributed under the GNU public license. Ember is a perfect language for implementing a mixed environment application (i.e. console and web applications) like TraffAcct as it provides support for both stand-alone and HTML embedded scripting. You will need to install Ember on your system prior to installing TraffAcct. Once Ember has been installed, installing TraffAcct should be as simple as typing the following in the TraffAcct source directory

```
./configure  
make install
```

Ember

Before TraffAcct can be installed, Ember must be installed and operation on your machine. The current Ember distribution is always available over the Internet from www.Hughes.com.au. Simply download and install the software following the instructions provided within the Ember distribution. Installing Ember should not require anything more complicated than running the following on your system

```
./configure  
make  
make install
```

SNMP Module

The primary method used by TraffAcct to acquire the network traffic information is SNMP. For TraffAcct to be able to collect this data, Ember's SNMP module must be installed. An Ember module is a dynamic loading "plug-in" that extends the languages functionality. The SNMP module is freely available from www.Hughes.com.au and must be installed prior to the installation of TraffAcct. Please remember, however, that you cannot install the SNMP Module until Ember has been installed on your system.

Web Server

A web based interface providing access to traffic reports is included in the distribution. To utilise the web interface you will naturally need a web server installed on your system. Your web server must also know how to execute Ember enhanced web pages (.ehtml files). Details for configuring the Apache web server to handle Ember pages is included in the Ember distribution. If you are running the Apache server then please ensure the required modifications are made to your httpd.conf file. If you are running a web server other than Apache then you will need to research this yourself. For apache user, usually you only need to add the following to your httpd.conf file

```
AddType application/ember .ehtml  
Action application/ember /cgi-bin/w3e
```

If you experience problems relating to "syntax errors" in your config file after adding the above then also check that you have the following:

```
LoadModule action_module /usr/lib/apache/1.3/mod_actions.so  
LoadModule access_module /usr/lib/apache/1.3/mod_access.so
```

Crontab

The TraffAcct poller is a simple script that needs to be executed regularly to ensure collection of the statistics information. The usual way to ensure that the poller runs regularly is to add an entry for it in your systems crontab file. It is recommended that you run the poller every 5 minutes. On most UNIX systems, the crontab entry required would look like the one below

```
#  
# TraffAcct  
#  
*/5 * * * * root /usr/local/traffacct/poller
```

Configuration

Once the TraffAcct software has been installed it must be configured to suit your traffic monitoring needs. Configuration is based on the concept of a Device (the router, switch etc), an Access Method (Ethernet, ISDN, Serial etc), and a Circuit ID (the ID of the Ethernet interface, the username of an ISDN connected dial user etc). The combination of all three elements will uniquely identify a connection that can be monitored.

All configuration information is located in the config directory under the TraffAcct installation directory. For our examples below we will assume that the configuration is stored in /usr/local/traffacct/config

Devices

To add a device to the TraffAcct configuration, simply create a directory in the config directory where the name of the directory is the hostname of the device. For example, to add the device core.hughes.com.au to the TraffAcct configuration you would simply create the following directory

```
/usr/local/traffacct/config/core.hughes.com.au
```

Because SNMP will be used to access the device and collect statistics, we must provide an SNMP Community string for that device. Simply create a file called community in the device's directory. The file should contain nothing but the SNMP community string (i.e. no blank lines etc).

Most of the data collectors provided with TraffAcct will only require read-only access to the SNMP objects on your device. However, some collectors (IP Accounting in particular) will require read-write access. Please see the documentation for the specific collectors you wish to use to ensure the correct community string is provided.

Circuits

Circuits are defined based on the device they are connect to and their access method. To configure an Ethernet based circuit you would add the details of the circuit in a file called ethernet in the device's configuration directory. Similarly, if you wanted to monitor an ISDN based connection the details of the circuit would be added to a file called isdn in the device's configuration directory. Quite simply, the name of the file is the same as the name of the collector being used to acquire the data. It should be noted that all configuration file names use lower case only.

The basic format of a circuit definition is the same among all the different access methods. The format is

```
Connction ID : Connection Name : Interface ID
```

A "Connection ID" is an identifier that you specify to uniquely identify this circuit for administrative purposes. If you intend to use the reports from TraffAcct for billing purposes then the ID may be a customer code. Otherwise it could be an abbreviation of the name of the organisation using the circuit. Please note that this ID is not intended to be a textual description of the circuit and it should not contain any spaces. For example, if the circuit belonged to a customer called Hughes Technologies, the Circuit ID may be *hughes* or *hughes-tech* or *hugh01*

A "Connection Name" is a general purpose, descriptive name for the circuit. In the example mentioned in the paragraph above, the Circuit Name might be *Hughes Technologies Office Link*

The "Interface ID" of the circuit is an identifier used by the collector script to gather the traffic data from the device. It is often an SNMP Interface number or a username. You must read the specific details of the collector you wish to use to determine what information it requires as the Interface ID. All the collectors are covered in detail later in this manual.

Some example circuit definitions are shown below. The first is for an Ethernet connected customer, the second is an ISDN user, and the third is for monitoring traffic to an IP address block

```
abc01:ABC Accounting Colo Link:17  
smith:Smith and Sons:smith-gw  
infotech:Info Tech Services Network:10.1.1.0/24
```

Users

Controlling access to the TraffAcct web interface is achieved using user profiles. A user profile consists of a username, a password, a specification of what circuits the user may view, and a few other optional details. User information is stored in the users directory under the installation directory. The details of each user are contained in a separate file where the name of that file is the username. For example, details of Fred's access details might be in a file called fred in /usr/local/traffacct/users.

Each user's file consists of a number of lines each containing a tag and a value separated by a colon. Blank lines and lines starting with the # character are ignored. A sample user definition is shown below

```
auth_type: internal
password: foobaa
allow_pw_change: true
email: foo@baa.com
circuit: *:*:*
```

A description of the available configuration items is given below

Item	Description
auth_type	The type of authentication can be set to either internal or external. If internal authentication is used then the user's password is expected to be stored in clear text in the password field of the user's definition file. If external authentication is used then the task of validating the user is handed off to an external program. See the auth_prog entry below for more details of external authentication.
auth_prog	If authentication of a user is to be handled by an external program (i.e. the auth_type configuration entry is set to external) then the value of this field will indicate the path and parameters of the external program. Before the program is executed, the value provided is scanned and any occurrences of %u and %p are replaced by the current user's username and password respectively. If the exit value of the program is 0 then the user is authenticated and access is allowed. An example entry is shown below auth_prog: /usr/local/bin/ldap_check_password %u %p
password	If internal authentication is being used then the value of this entry is the user's password in clear text.
allow_pw_change	If internal authentication is being used then setting this entry to true will allow the user to change their password. If set to false, the Change Password button on the web based user interface is not displayed.
email	A contact email address for this user.
circuit	The circuit definition specifies which circuits this particular user can see. There is no restriction on the number of circuit entries that a user's definition may contain. But, in general practice, most users will only need one circuit definition. The value specified in this field is a circuit identifier. Details of the format of a circuit identifier are given below.

Circuit Identifier

Circuit identifiers are used in several parts of TraffAcct to identify a circuit of interest. In terms of a user definition file, it identifies one or more circuit that the user can view. The circuit identifier is a three part value formed using the device name, the circuit's access method, and the circuit's Connection ID. An example circuit identifier might be

```
colo-switch.hughes.com.au:ethernet:hugh01
```

The above circuit identifier tells us that the circuit is an Ethernet connection terminated on a device called colo-switch.hughes.com.au, and that the Circuit is referred to as hugh01 within TraffAcct. The circuit identifier for an ISDN based connection for a user called fred might look like

```
nas.hughes.com.au:isdn:fred
```

If a user is allowed to view more than one circuit, you have two options. You can either list each of the individual Circuit IDs within the user's definition file, or you can specify a wildcard circuit ID specification. A wildcard specification is used to match multiple related circuits and is specified by using a * in one or more of the sections of the Circuit ID. As an example, if you wished to view all Ethernet connected customers on your co-location switch, you may use a specification like

```
colo-switch.hughes.com.au:ethernet:*
```

To specify that a user can view all circuits monitored by TraffAcct, simply specify a Circuit ID containing three wildcards, e.g. *:*:*. Some example circuit IDs are shown below. The first would identify all ISDN connected customers. The second would identify all Ethernet circuits that are terminated on the two co-location switches. The third specification defines just a single serial circuit while the final specification identifies an IP network block known to TraffAcct as "servers" that is monitored using Cisco IP accounting from the core.hughes.com.au router.

Example 1	*:isdn:*
Example 2	colo-switch1.hughes.com.au:ethernet:* colo-switch2.hughes.com.au:ethernet:*
Example 3	cust-gw.hughes.com.au:serial:benco
Example 4	core.hughes.com.au:ip:servers

traffacct.conf file

Several system wide configuration parameters can be set using the traffacct.conf file. The file is located in the installation directory, usually /usr/local/traffacct. The format of the file is quite simple: All blank lines and lines starting with the # character are ignored. All other lines are expected to contain a tag and value pair separated by a colon, as shown in the sample file below.

```
# Log file location
log_file: /usr/local/traffacct/traffacct.log

# user audit trail directory
audit_dir: /usr/local/traffacct/audit

# username of the user under which cgi-scripts run
web_user: nobody

# Do you want informational logs generated in the log file?
log_info: true

# Do you think a megabyte is 1024 or 1000 bytes
megabyte: 1000
```

A description of the configuration items is given below:

Item	Description
log_file	The full path to a file in which both the collector scripts and the web based User Interface can write warning, error and informational log entries
audit_dir	A directory in which the web based user interface can log details of a user's activities. A file per user will be created.
web_user	The username of the user used by your web server when it executes CGI scripts (quite often the 'nobody' user). The software will try to ensure that log files are owned by this user at all times to ensure that the web based user interface can write to them.
log_info	A boolean value indicating whether or not the collector scripts should generate information entries in the log file. If set to yes, an Info line will be added to the log file each time a collector runs. It will usually contain some rough statistics about the execution of the script (e.g. the number of circuits monitored on that particular host etc).
megabyte	Some service providers tend to redefine the term megabyte to imply 1000 kb rather than 1024 kb. It is common in Australia at least for telco styled providers to operate in this way. If you wish to operate using the more correct but "harder to explain to users" value of 1024 then simply change this entry. Note, providing a value other than 1000 or 1024 will generate and error.

Reports

Included in the software distribution is a general purpose report generator called `export`. You can find it in the `tools` directory under the installation directory. The report generator uses format files to define what a report will look like. Using the format files you can specify any report definition you like. Included with the software are sample reports in text, HTML and Microsoft Comma Separated Value (CSV) format.

The `export` utility takes the following command line arguments:

```
format file      : Name of report definition file
start date      : First report date in dd-Mon-Year format (eg 10-Nov-2001)
stop date       : Last report date
circuit         : The circuit(s) to report
```

The circuit is identified using the following syntax

```
Device Name : Circuit Type : Circuit ID
```

If any segment of the circuit identifier is not provided then it is wildcarded. E.g using `'router1:isdn:'` will match all ISDN clients on `reouter1`, using `':ethernet:'` will match all ethernet connections.

Some samples are shown below. The first would generate a report for the month of November 2001, formatted based on the `web_cust_daily_html` format definition, for a single Ethernet circuit connected to the `core.hughes.com.au` router. The second example produces an annual report for the 2001 year for all ISDN connections that terminate on the `cust.hughes.com.au` device

```
export web_cust_daily_html 1-Nov-2001 30-Nov-2001 core.hughes.com.au:ethernet:telco
export cust_bill 1-Jan-2001 31-Dec-2001 cust.hughes.com.au:isdn:
```

Format Keys

The report definitions are stored in individual files within the `formats` directory under the `TrafficAcct` installation directory. The name of the file defines the name of the format, which is passed as an argument to the `export` utility when you wish to generate that report. The file is structured as a series of keys and values separated by an `=` sign. The value definition terminates at the end of the line. The available keys and their meanings are shown below.

Key	Description	Valid values
<code>report_data_head</code>	start of the data table	any format string
<code>report_data_line</code>	a line of the data table	any format string
<code>report_data_tail</code>	end of the data table	any format string
<code>report_head</code>	start of the report	any format string
<code>report_tail</code>	end of the report	any format string
<code>report_period</code>	length of each report interval	<i>daily</i> or <i>monthly</i>

Format Strings

For the report “format” to be of any use, there must be some method by which you can define the data you want generated as output. There are several Format Keys listed in the table above that are used for this purpose. Each of those keys takes something called a Format String as its value. A format string is the actual definition of the text that must be generated.

A format string is a combination of normal text and some embedded variables containing the report data. When the format string is about to be processed, the variables it contains are expanded and their current values are included in the text output. The list of available variables is shown in the table on the included below.

You may use any number of variables in a Format String. You can also use any of the standard escape characters provided by the C and Ember programming languages. The most commonly used escape characters are `\n` to signify a new line, and `\t` to signify a tab.

Variable	Definition
<code>\$circuitNode</code>	Name of the device the circuit is connected to (e.g. router hostname)
<code>\$circuitType</code>	Access method of this circuit (e.g. ethernet)
<code>\$circuitID</code>	The unique ID associated with this circuit
<code>\$startDate</code>	First day of the report
<code>\$stopDate</code>	Last day of the report
<code>\$periodTag</code>	The label used to identify the current report line period (day or month depending on report mode)
<code>\$periodInKbytes</code>	Total input kilo bytes for the current time period
<code>\$periodOutKbytes</code>	Total output kilo bytes for the current time period
<code>\$periodInMegs</code>	Total input mega bytes for the current time period
<code>\$periodOutMegs</code>	Total output mega bytes for the current time period
<code>\$totalInKbytes</code>	Total input kilo bytes for the entire report duration
<code>\$totalOutKbytes</code>	Total output kilo bytes for the entire report duration
<code>\$totalInMegs</code>	Total input mega bytes for the entire report duration
<code>\$totalOutMegs</code>	Total output mega bytes for the entire report duration

Example

```
report_head = Device : $circuitNode\nLink Type : $circuitType\nLink Name :  
$circuitID\nPeriod : $startDate to $stopDate\n\nreport_data_head = Date      Input KBytes   Output KBytes\nreport_data_line = $periodTag $periodInKbytes   $periodOutKbytes\nreport_data_tail =  
report_tail = Total      $totalInKbytes Kb   $totalOutKbytes Kb\nreport_period = daily
```

Collectors

TrafficAcct uses data acquisition scripts called “collectors” to gather the statistical information. A collector is simply a script designed to acquire data for a specific type of connection. Included in this package are collectors for Ethernet links, Serial links, ISDN connections, and even TCP/IP network blocks (i.e. address based accounting).

As mentioned in the Configuration section of this document, each collector will have specific requirements in terms of how it identifies a “data circuit”. For simple SNMP based interfaces, like Ethernet or Serial, an SNMP interface number may be used. For IP Accounting based collection, an IP network definition would be required. The following section describes each of the available collectors and discusses how they are configured.

Because some of the collectors require an SNMP Interface Number as the Interface ID element of the circuit definition, a utility that will help you determine that information is included in the software distribution. The sysinfo utility will gather a range of useful pieces of information about a device via SNMP and report the details to you. Included in the report is a complete list of all the interfaces available in the device’s interface table, along with the interface number of each device. The sysinfo utility is included in the tools directory of the software installation.

Ethernet

Overview The Ethernet collector uses SNMP to gather the input and output byte counts of the specified network interface. The configuration details of Ethernet connections are located in a file called ethernet in the device’s configuration directory.

Config To configure monitoring of a particular interface, the SNMP Interface Number of that interface must be provided as the Interface ID element of the Circuits configuration entry.

Example abc:ABC Partners Accountants:17

Serial

Overview The Serial collector is based on the Ethernet collector and provides the same level of functionality. To configure monitoring for a serial interface, the details must be in a file called serial in the device’s configuration directory.

Config To configure monitoring of a particular interface, the SNMP Interface Number of that interface must be provided as the Interface ID element of the Circuits configuration entry.

Example jones:Jones and Sons:12

ISDN

Overview The ISDN collector uses SNMP to query a proprietary Cisco MIB called the ISDN Call History MIB. If the device to be monitored is not a Cisco then this collector will fail to gather any statistics. The collector can determine if multiple ISDN calls are active and will aggregate the traffic of multi-link connections. Configuration details for this collector must be stored in a file called isdn in the device’s configuration directory.

Config ISDN connections are based on calls rather than circuits. To identify a call, the username of the authenticated user for that call is used. The username must be provided as the Interface ID element of the configuration entry.

Example smith01:Smith Brothers Office Link:smith-gw

IP

Overview The IP Accounting collector gathers statistics about the amount of traffic to and from a certain IP network block (CIDR block). It is unlike the other collectors in that it does not directly monitor a specific interface. Rather, it uses the IP Accounting facility of Cisco routers and grabs statistics from a shared statistics table. The router used does not have to be directly connected to the monitored network, in fact routers close to the core of the network are best of this purpose.

Config This collector uses a comma separated list of CIDR address definitions as it's Interface ID (i.e. a list of Address / Netmask Length definitions). The CIDR block can be of any length, including a /32 (i.e. a single IP Address). When providing a list of CIDR blocks, no spaces should be used anywhere in the list.

Example office:Office Network:10.1.1.0/24,10.2.4.0/28
www:Company Web Server:10.1.1.1/32

Note The Cisco IP Accounting functionality uses a checkpoint table for collection of the traffic statistics. The collection process involves notify the device to create the checkpoint table and then reading the checkpoint data. To perform this operation, the collector script must have SNMP Read/Write access to the device. You must specify the SNMP Write Community string as the community for any devices that will use the IP collector.

Another point to note is that IP Accounting is only performed on outbound packets. If you only enable it on the interface "pointing towards" the specified address block then you will only be tracking outbound statistics. To gather statistic for both outbound and inbound data you may have to enable IP Accounting on all the Cisco's interfaces (or at least the client side and core side interfaces).