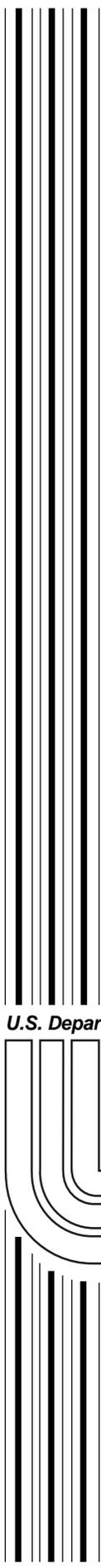


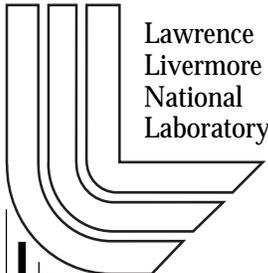
# Accounting Data to Web Interface Using PERL

*Charlie Hargreaves*  
*Lawrence Livermore National Laboratory (LLNL)*  
*Livermore Computing – Data Storage Group*  
*Livermore, CA 94550*

August 13, 2001



*U.S. Department of Energy*



Lawrence  
Livermore  
National  
Laboratory

## DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This work was performed under the auspices of the U. S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.

This report has been reproduced  
directly from the best available copy.

Available to DOE and DOE contractors from the  
Office of Scientific and Technical

Lawrence Livermore National Laboratory  
Technical Information Department's Digital Library  
<http://www.llnl.gov/tid/Library.html>

# Accounting Data to Web Interface Using PERL

By Charlie Hargreaves  
Computer Science Major

Lawrence Livermore National Laboratory (LLNL)  
Livermore Computing – Data Storage Group  
7000 East Avenue  
Livermore, CA 94550

August 13, 2001

## Table of Contents

Overview.....	2
HPSS Accounting Report .....	3
The Master Script (run_acct_html).....	3
Database Scripts.....	4
HTML Scripts.....	5
CGI Scripts.....	8
The acctinfo.cgi Script.....	8
The Graphing CGI Scripts .....	16
Additional Scripts .....	17
Conclusion .....	17
Glossary .....	18
Bibliography .....	19

## Overview

This document will explain the process to create a web interface for the accounting information generated by the High Performance Storage Systems (HPSS) accounting report feature. The accounting report contains useful data but it is not easily accessed in a meaningful way. The accounting report is the only way to see summarized storage usage information. The first step is to take the accounting data, make it meaningful and store the modified data in persistent databases. The second step is to generate the various user interfaces, HTML pages, that will be used to access the data. The third step is to transfer all required files to the web server. The web pages pass parameters to Common Gateway Interface (CGI) scripts that generate dynamic web pages and graphs. The end result is a web page with specific information presented in text with or without graphs.

The accounting report has a specific format that allows the use of regular expressions to verify if a line is storage data. Each storage data line is stored in a detailed database file with a name that includes the run date. The detailed database is used to create a summarized database file that also uses run date in its name. The summarized database is used to create the group.html web page that includes a list of all storage users. Scripts that query the database folder to build a list of available databases generate two additional web pages. A master script that is run monthly as part of a cron job, after the accounting report has completed, manages all of these individual scripts.

All scripts are written in the PERL programming language. Whenever possible data manipulation scripts are written as filters. All scripts are written to be single source, which means they will function properly on both the open and closed networks at LLNL. The master script handles the command line inputs for all scripts, file transfers to the web server and records run information in a log file. The rest of the scripts manipulate the accounting data or use the files created to generate HTML pages. Each script will be described in detail herein.

The following is a brief description of HPSS taken directly from an HPSS web site. "HPSS is a major development project, which began in 1993 as a Cooperative Research and Development Agreement (CRADA) between government and industry. The primary objective of HPSS is to move very large data objects between high performance computers, workstation clusters, and storage libraries at speeds many times faster than is possible with today's software systems. For example, HPSS can manage parallel data transfers from multiple network-connected disk arrays at rates greater than 1 Gbyte per second, making it possible to access high definition digitized video in real time."<sup>1</sup> The HPSS accounting report is a canned report whose format is controlled by the HPSS developers.

---

<sup>1</sup> Reference web site <http://www.sdsc.edu/hpss/hpss1.html>

## HPSS Accounting Report

The HPSS accounting report file name is in the following format `acct.yyyymmdd.ttttt`<sup>2</sup> with time military style. The report output has comments that are identified by a # character at the beginning of the line. Every line that does not start with the # character is accounting data. The data line columns are all numbers and there are two types of lines. If the third column contains a zero (0) the line contains summary information in the following order: Account Identification Number (AcctID), Class of Service (COS), (0), Number of Accesses (# Accesses), Number of Files (#Files) and Total File Length (Length COS) in a COS. This information is a snapshot taken at the time the accounting report is executed. If the third column contains a non-zero value the line contains storage usage information in the following order: AcctID, COS, SClass, # Accesses and Amount of Data Transferred (Transferred). The Number of Accesses and Amount of Data Transferred are reset each time the accounting report is executed.<sup>3</sup> LLNL has set the report variable "Storage Unit Size" to megabyte, which means that all numbers output on the report are in megabytes.

There is a change in report format between HPSS revision 4.1 and revision 4.2. Revision 4.2 has an additional column on the left that contain Cross-Cell<sup>4</sup> information. The Cross-Cell number identifies the location of the account. The account could be a local registered user or a non-local registered user that connects from one of the sites allowed storage access. This change affects only the script that generates the detailed database. To handle this I have written one script for revision 4.1 and one for revision 4.2. I could have made one script for both versions but for efficiency reasons and the assumption that Cross-Cell will continue to be used in future HPSS releases I decided to make two separate scripts.

The accounting report has been run for the past year so historical data will be included in the databases. It is currently run monthly but in the past has been run daily. To incorporate the data that was collected daily I wrote a script that summed the daily runs into the equivalent of a monthly run. This script is described in the Additional Scripts section.

### The Master Script (run\_acct\_html)

The `run_acct_html` script manages the monthly run of all individual scripts. First it finds out what host it is on, Raven and Toofast are the only allowed hosts. Directory paths are set according to the host. It then locates the accounting report that has just been generated by HPSS accounting. A variable is initialized with the run date that is parsed from the accounting report name. The current date and time is sent to the log file `acct.html.t` that is used records run status.

---

<sup>2</sup> yyyy represents year, mm represents month, dd represents day and ttttt represents time.

<sup>3</sup> Reference HPSS System Administration Guide Release 4.1.1, Revision 1 (June 1999).

<sup>4</sup> Cross-Cell is the concept of accepting authentication credentials from non-local authentication domains.

The createdetailacctdb script is run with an overloaded standard input of the accounting report, an overloaded standard output of acctdetaildb.yyyymmdd and an overloaded standard error concatenating file acct.html.t. The createacctdb script is run with an overloaded standard input of the acctdetaildb.yyyymmdd, an overloaded standard output of acctdb.yyyymmdd and an overloaded standard error concatenating file acct.html.t. The two databases, acctdetaildb.yyyymmdd and acctdb.yyyymmdd, that are queried by the CGI script are now created locally.

There are three scripts called to generate the HTML pages. The createallaccthtml script is run with an overloaded standard input of the acctdb.yyyymmdd, an overloaded standard output of group.html and an overloaded standard error concatenating file acct.html.t. The createaccthtml script is run with an overloaded standard output of individual.html and an overloaded standard error concatenating file acct.html.t. The createtopusershtml script is run with an overloaded standard output of topusers.html and an overloaded standard error concatenating file acct.html.t. The HTML files are now created locally. Web page format and contents are described in the HTML Scripts section.

All database files and HTML files are moved to a directory on the host that can be accessed from a node with access to the web server. This directory is /home/host name/admin/acct/ with host name being Raven or Toofast. Using Distributed Computing Environment (DCE) keytab credentials a secure shell DCE login is made on a node that has access to the web server file directory, Quail and Halffast are the allowed hosts. The files are now moved into the web server accounting directory, /dfs/www/dsg/accounting/. The path is the same from Quail and Halffast. An additional DCE login is used to set the group to dsgweb on all files transferred. For all of these transactions standard error output is concatenated to file acct.html.t.

## **Database Scripts**

Databases for each month are stored on the web server for access by a CGI script. There are two databases required to completely and expeditiously display the accounting data. The first is a detailed database, file name acctdetaildb.yyyymmdd, that holds the same data as the accounting report with the addition of user login and user full name. All comments and white space are removed and columns are separated by the + character. The + character was chosen because it is very unlikely to be used in a name or login. The second database is the summary database, file name acctdb.yyyymmdd, that holds the summarized storage information for each user. The summarized database is sorted by storage space used in descending order to optimize the top users query. Although all of this data could be obtained dynamically at run time it is stored in a file for efficiency purposes.

The createdetailacctdb can only be run on Toofast09 or Reven17 nodes. The first step is to build an array that holds user information for everyone with a DCE login. Although this is a slow process it is the most complete and efficient solution. Next an associative array is built with AcctID as the index and login and full name as the data members. The

accounting report is at standard in so now lines start to be read. The end-of-line character is removed and the line is split into an array. All comments are ignored and any non-comment line is checked for proper format using this regular expression:

`/^\s*\d+\s+\d+\s+\d+\s+\d+\s+\d+\s*\d*\s*$/o5`. Notice that this accepts both types of lines. If the format is not correct the script dies with an error message. The first numbers on a line are AcctID, which is used to access the login and full name in the associative array. If no entry is found a warning message is sent to standard error, the field for login is set to uid:AcctID\_Unknown\_login and the full name field is set to Unknown\_name. For summary line type the line is sent to standard out in the following format: login+full name+AcctID+COS+0+# Accesses+# Files+Length COS. For storage usage lines the format is: login+full name+AcctID+COS+S Class+# Accesses+Transferred. This continues until the end-of-file is reached. The HPSS version 4.2 compatible script ignores the cross-cell because all users have a local login. On the closed side this is not the case. When running with HPSS version 4.2 on the closed side the cross-cell and AcctID will be used to identify users.

The following is a sample of the data stored in the detailed database for one user:

```
chargrea+Charles M. Hargreaves+7155+1+0+8355+85381+13956126
chargrea+Charles M. Hargreaves+7155+1+1+8355+1289447
chargrea+Charles M. Hargreaves+7155+10+0+0+16436+12899
chargrea+Charles M. Hargreaves+7155+10+10+0+0
chargrea+Charles M. Hargreaves+7155+20+0+6518+90261+94825
chargrea+Charles M. Hargreaves+7155+20+10+6518+6231
```

The createacctdb can only be run on Toofast09 or Reven17 nodes. The detailed accounting database is at standard input and lines start to be read. All comments are ignored. The # files, transferred and length data is summed for each login over all COS and stored in an associative array. This associative array is sent to standard output with a descending sort on the length data member. The output format is as follows: login+full name+# Files+Length COS+ Transferred. This continues until the end-of-file is reached.

The following is a sample of the data stored in the summary database for one user:

```
chargrea+Charles M. Hargreaves+192078+14063850+1295678
```

## HTML Scripts

The HTML scripts are run monthly to update the information stored statically as web pages. All HTML code is generated within these filter scripts. Each the web pages have identical headers and footers that follow the same format as other Data Storage Group web pages. The overall design of the pages maintain a similar look and user interface. There is a static index page, index.html, that accesses the three pages created by the HTML scripts. The three options for querying the databases are request storage information for a single user, request storage information for a group of users and see a

---

<sup>5</sup> ^\$ are anchors, s is white space, d is digit and the o tells the compiler to compile the pattern only once

list of the top storage users. All of these web pages supply a CGI script, acctinfo.cgi, with the proper information to generate a selection of data specified by the user.

The createaccthtml script creates the single user storage information query web page. The web page has an HTML form that contains three sections that are arranged in a table with action set to the acctinfo.cgi file<sup>6</sup>. The createaccthtml script builds a multiple select list of dates from the databases available in the database directory with “All” selected. The numerical date is converted to a test date, “mmm dd, yyyy” format with mmm in letters. This is done in all HTML scripts. Radio buttons are built to select report type. The two report types allowed are “Summary” and “Summary with Graphs”. Submit and reset buttons are created along with some text and a text box. The text box accepts the login of the user whose data is to be displayed. When submit is pushed the date(s), report type and text box are passed to the CGI script with the proper parameter labels. (See figure 1)

**Data Storage Group**

---

**HPSS Accounting Report Selection - Individual User**

---

<b>Select Date(s)</b>	<b>Select Report Type</b>
<ul style="list-style-type: none"><li>All</li><li>Aug 01, 2001</li><li>Jul 01, 2001</li><li>Jun 01, 2001</li><li>May 01, 2001</li><li>Apr 01, 2001</li><li>Mar 01, 2001</li><li>Feb 01, 2001</li></ul>	<input checked="" type="radio"/> Summary <input type="radio"/> Summary with Graphs
<b>Enter Login and Click</b>	<input type="text"/> <input type="button" value="Submit"/> <input type="button" value="Reset"/>

---

[HPSS Account Home](#)

If you have questions or comments about this page, contact [Roger Skowlund](#)



and [LLNL Disclaimers](#)

Figure 1 - Individual User Accounting Report Selection

<sup>6</sup> This assumes you know HTML and CGI.

The createallaccthtml script creates the group of users storage information query HTML page. The web page has an HTML form that contains many sections that are arranged in a table with action set to the acctinfo.cgi file. The createallaccthtml script builds a multiple select list of dates from the databases available in the database directory with "All" selected. Radio buttons are built to select report type. The three report types allowed are "Summary", "Detailed" and "Summary with Graphs". Submit and reset buttons are created along with some text. A table that holds all AcctID logins and full names is build along with checkboxes for each entry. This table is populated with data from the most recent summary database that is at the overloaded standard input. When submit is pushed the date(s), report type and login(s) checked are passed to the CGI script with the proper parameter labels. (See figure 2)

**Data Storage Group**

### HPSS Accounting Report Selection - Group of Users

---

**Select Date(s)**

**Select Report Type**

Aug 01, 2001  
 Jul 01, 2001  
 Jun 01, 2001  
 May 01, 2001  
 Apr 01, 2001  
 Mar 01, 2001  
 Feb 01, 2001

Summary  
 Detailed  
 Summary with Graphs

Submit

Reset

**Current Users**

Select as many as you want

Hint: Your browser "Find in Page" feature may be helpful

<input type="checkbox"/> abc "Amy Pezzio"	<input type="checkbox"/> bug "Irma A. Bug"	<input type="checkbox"/> dan "Dan F. Smith"
<input type="checkbox"/> frank "Frank A. France"	<input type="checkbox"/> greg "Greg Herr"	<input type="checkbox"/> has "Al Shasta"

Submit | Reset

---

[HPSS Account Home](#)

If you have questions or comments about this page, contact [Roger Skowland](#)

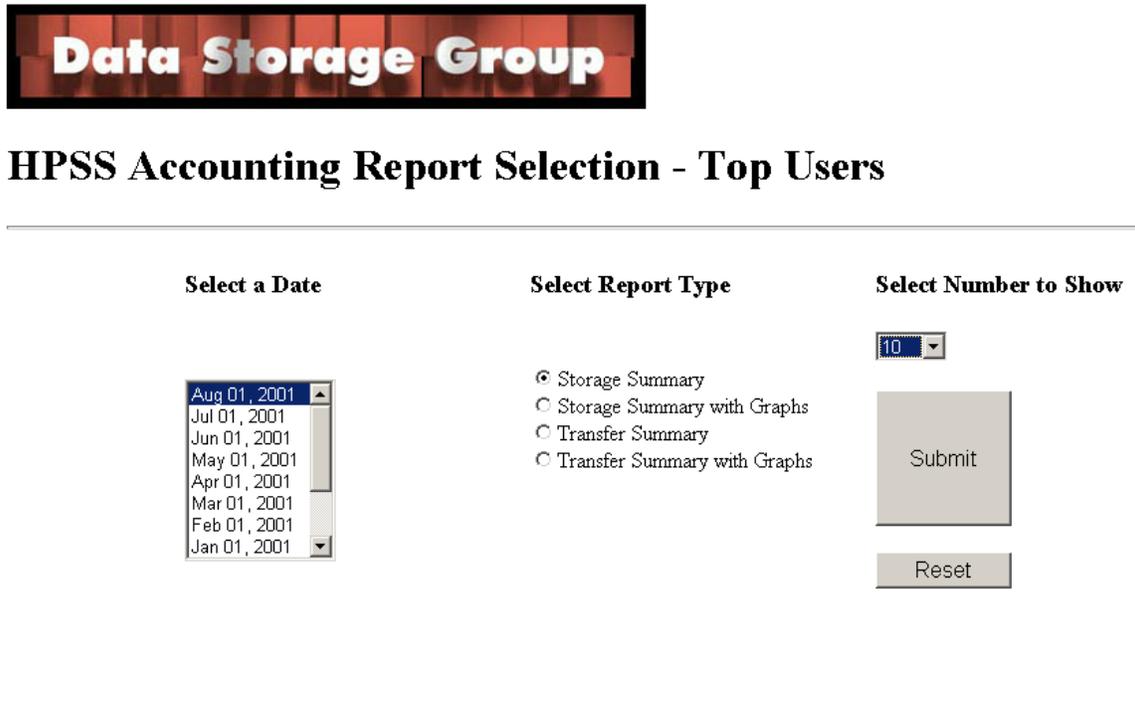
**GRAPEVINE**

and [LLNL Disclaimers](#)

**Figure 2 - Group of Users Accounting Report Selection**

The createtopusershtml creates an HTML page that shows the top 10, 25, 50 or 100 users in regards to storage space used or data transferred. The web page has an HTML form that contains three sections that are arranged in a table with action set to the acctinfo.cgi

file. The creatopusershtml script builds a multiple select list of dates from the databases available in the database directory with the most recent date selected. Only one date is allowed to be selected because a summary of multiple months is not desired. Radio buttons are built to select report type. The four report types allowed are “Storage Summary”, “Storage Summary with Graphs”, “Transfer Summary” and “Transfer Summary with Graphs”. Submit and reset buttons are created along with some text. A pop-up select list is created with the number of top users to show. When submit is pushed the date, report type and number of top users to show are passed to the CGI script with the proper parameter labels. (See figure 3)



The screenshot shows a web form titled "Data Storage Group" and "HPSS Accounting Report Selection - Top Users". The form is divided into three sections: "Select a Date", "Select Report Type", and "Select Number to Show".

- Select a Date:** A dropdown menu showing dates from Jan 01, 2001 to Aug 01, 2001. Aug 01, 2001 is selected.
- Select Report Type:** Four radio button options: "Storage Summary" (selected), "Storage Summary with Graphs", "Transfer Summary", and "Transfer Summary with Graphs".
- Select Number to Show:** A dropdown menu showing the number "10".

Below the form are "Submit" and "Reset" buttons.

[HPSS Account Home](#)

If you have questions or comments about this page, contact [Roger Skowlund](#)



and [LLNL Disclaimers](#)

**Figure 3 - Top Users Accounting Report Selection**

## CGI Scripts

### The acctinfo.cgi Script

The acctinfo.cgi script is on the web server in directory /dfs/www/dsg/cgi-bin/. This script is over 600 lines so I will only give a high level description of its functionality. Detailed comments are in the script. It accepts parameters from any of the three web

pages, uses the parameters to access the specific information in the databases and sends HTML code with the selected information to the requesters browser.

Acceptable scalar<sup>7</sup> parameters passed in are login for single user information, namedetail for login contact information, pageformat for type of report to run and topten for the top users list. Acceptable array parameters passed in are usernames to see more than one users information and dates to see information for more than one date.

Some special cases must be handled before any data is collected. If dates are equal to “All” then the dates array is set to all databases dates available. The date format is converted to all numbers so it can be used as an extension to access the accounting databases. If only a login is passed, like from the individual user page, the full name for that user is found in a database and both are added to the usernames array.

When pageformat is equal to “Detailed” the data from the detailed account database(s) is collected for all dates specified in the dates array. This is done by opening each database file using the date’s array and collecting all records in an associative array. The key for the associative array is AcctID, COS and S Class. The process of opening each file once is used to minimize file access delays. The values in the usernames array are used to index the associative array and access the correct information. This information is separated into two associative arrays, one for summary data and one for storage usage data. Also added to both associative array data is HTML code for web presentation. The key for both arrays is login, date, AcctID, COS and S Class. The name and date inclusions in the key are used to sort the data when displayed. Both associative array data collections are sent to standard output in name then descending date order (See Figure 4)

---

<sup>7</sup> A scalar is a PERL variable data type.



## HPSS Account Information

The following table contains totals for Data Stored, Number of Files and Number of File Accesses.

Totals are snapshots at a particular date and may not represent current account status.

Summary information is for accounts in a particular Class Of Service (COS)

Transfer COS Tables are after all storage tables

Name	Date	COS	SClass	No. of Accesses	No. of Files in Archive	Total Archive Space Used (MB)
<a href="#">chargrea "Charles M. Hargreaves"</a>	Aug 01, 2001	1	0	0	0	0
<a href="#">chargrea "Charles M. Hargreaves"</a>	Aug 01, 2001	20	0	141	58	3

The following table contains totals for Number of File Accesses and Amount of Data Transferred.

Totals are snapshots at a particular date and may not represent current account status.

Summary information is for accounts in a particular Class Of Service (COS)

Name	Date	COS	SClass	No. of Accesses	Transferred This Month (MB)
<a href="#">chargrea "Charles M. Hargreaves"</a>	Aug 01, 2001	20	10	141	8

[HPSS Account Home](#)

If you have questions or comments about this page, contact

[Roger Skowhund](#)



and [LLNL Disclaimers](#)

**Figure 4 – Detailed Account Information**

When pageformat is equal to “Summary” the data from the summary account database(s) is collected for all dates specified in the date’s array. This is done by opening each summary database file using the date’s array and collecting the information in an associative array. The key for the associative array is login. The values in the usernames array are used to index the associative array and access the correct information. The information is entered into a two-dimensional associative array indexed by login and date with the second dimension containing the name of the data column pointing to the login full name, number of files, space used or data transferred. Here is one line of the code that is for full name: `$userinfo{$uname[0]}'".$date}{fullname} = $temp[0];`. `$uname[0]` holds the login and `$temp[0]` holds the full name for the login. If there is no record in the database all values are set to ‘0’. The username array and dates array are used, sorted, as indexes to add HTML code around the values from the two-dimensional associative array.



## HPSS Account Information

Note: Totals are snapshots at a particular date and may not represent current account status.

Name	Date	Total No. of Files in Archive	Total Archive Space Used (MB)	Transferred This Month (MB)
<a href="#">chargrea "Charles M. Hargreaves"</a>	Aug 01, 2001	58	3	8
<a href="#">chargrea "Charles M. Hargreaves"</a>	Jul 01, 2001	47	2	49
<a href="#">chargrea "Charles M. Hargreaves"</a>	Jun 01, 2001	2	0	0

[HPSS Account Home](#)

If you have questions or comments about this page, contact [Roger Skowhund](#)



and [LLNL Disclaimers](#)

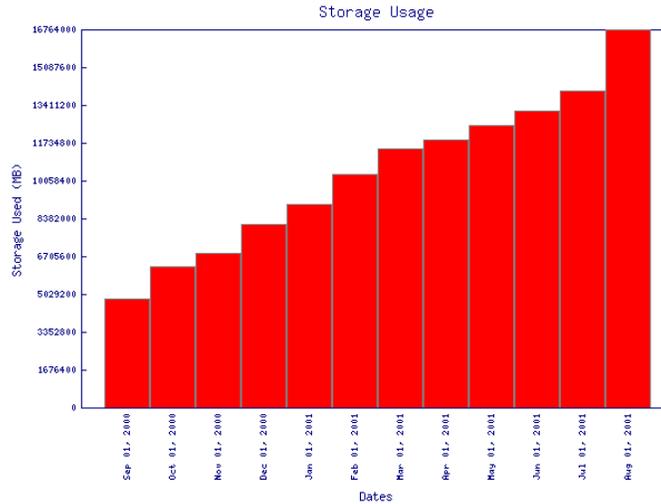
**Figure 5 – Summary Account Information**

When pageformat is equal to “SummaryGraphs” all the summary data is collected as described in the previous paragraph. Before the data is sent to the browser a graph is generated and sent. There are two types of summary graphs with both showing storage used. If there is only one login to display the graph will plot all dates for that one user (See figure 6). If more than one login is requested then the graph will display each users data for the most recent date (See figure 7). I chose to stream the graph directly to the browser instead of saving the image to disk and referencing it in HTML code. This eliminates having to manage image files and deal with the images in the web server cache. To do this an additional CGI script, plot\_graph.cgi, is used. This is so the HTML code can reference it as it would an image. All data to be displayed must be passed to plot\_graph.cgi. To pass the data it must be one continuous string of characters excluding spaces. To do this I used the ‘\_’ character to separate data and the ‘+’ to represent spaces. Here is the HTML code: “<IMG SRC = "plot\_graph.cgi? graphlist = \$graphlist">” with \$graphlist pointing to the continuous string. I’ll explain plot\_graph.cgi later but for now trust me that it displays the data. On return from plotting the graph the summary data text is displayed as in the previous paragraph.

# Data Storage Group

## HPSS Account Information

Graph of Storage Used By Charles M. Hargreaves



Note: Totals are snapshots at a particular date and may not represent current account status.

Name	Date	Total No. of Files in Archive	Total Archive Space Used (MB)	Transferred This Month (MB)
<a href="#">chargrea "Charles M. Hargreaves"</a>	Aug 01, 2001	211,983	16,763,903	1,264,290
<a href="#">chargrea "Charles M. Hargreaves"</a>	Jul 01, 2001	192,078	14,063,850	1,295,678
<a href="#">chargrea "Charles M. Hargreaves"</a>	Jun 01, 2001	178,817	13,186,238	684,031
<a href="#">chargrea "Charles M. Hargreaves"</a>	May 01, 2001	171,191	12,511,708	661,509
<a href="#">chargrea "Charles M. Hargreaves"</a>	Apr 01, 2001	163,561	11,885,765	410,737
<a href="#">chargrea "Charles M. Hargreaves"</a>	Mar 01, 2001	155,113	11,510,008	1,189,024
<a href="#">chargrea "Charles M. Hargreaves"</a>	Feb 01, 2001	143,862	10,333,999	1,487,009
<a href="#">chargrea "Charles M. Hargreaves"</a>	Jan 01, 2001	127,783	9,021,232	927,516
<a href="#">chargrea "Charles M. Hargreaves"</a>	Dec 01, 2000	113,935	8,149,516	1,510,065
<a href="#">chargrea "Charles M. Hargreaves"</a>	Nov 01, 2000	95,179	6,834,578	1,962,719
<a href="#">chargrea "Charles M. Hargreaves"</a>	Oct 01, 2000	89,191	6,244,842	2,224,658
<a href="#">chargrea "Charles M. Hargreaves"</a>	Sep 01, 2000	73,970	4,850,312	588,134

[HPSS Account Home](#)

If you have questions or comments about this page, contact [Roger Skowhund](#)



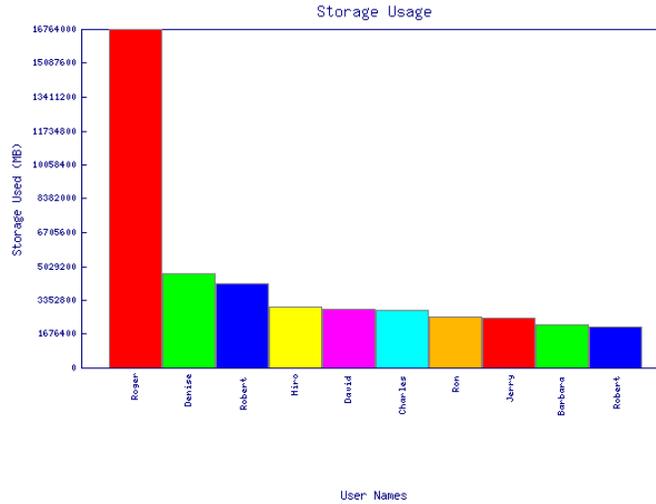
and [LLNL Disclaimers](#)

Figure 6 – Single User Summary Account Information with Graphs

# Data Storage Group

## HPSS Account Information

Graph of Storage Used On Aug 01, 2001



Note: Totals are snapshots at a particular date and may not represent current account status.

No.	Name	Date	Total No. of Files in Archive	Total Archive Space Used (MB)	Transferred This Month (MB)
1	<a href="#">Roger "Roger"</a>	Aug 01, 2001	211,983	16,763,903	1,264,290
2	<a href="#">Denise "Denise"</a>	Aug 01, 2001	122,863	4,693,757	0
3	<a href="#">Robert "Robert"</a>	Aug 01, 2001	19,853	4,174,675	142,782
4	<a href="#">Hiro "Hiro"</a>	Aug 01, 2001	46,529	3,031,067	1,155
5	<a href="#">David "David"</a>	Aug 01, 2001	38,221	2,891,826	0
6	<a href="#">Charles "Charles"</a>	Aug 01, 2001	82,064	2,845,445	0
7	<a href="#">Ron "Ron"</a>	Aug 01, 2001	11,258	2,554,097	0
8	<a href="#">Jerry "Jerry"</a>	Aug 01, 2001	32,293	2,478,134	0
9	<a href="#">Barbara "Barbara"</a>	Aug 01, 2001	22,967	2,162,132	0
10	<a href="#">Robert "Robert"</a>	Aug 01, 2001	1,038,161	2,055,763	14,875

[HPSS Account Home](#)

If you have questions or comments about this page, contact

[Roger Skowhnd](#)



and [LLNL Disclaimers](#)

**Figure 7 – Multiple User Summary Account Information with Graphs**

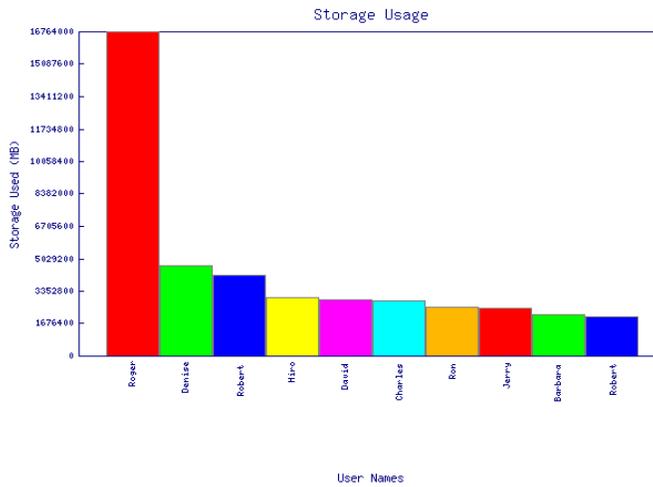
If the topten parameter has a number the list of top users, up to the number in topten, is displayed. The two sorts, both in descending order, available are by storage used or data transferred. Since the summary database is ordered in descending order with regards to data stored the list of top storage users is a simple sequential read from the database file. In the data transferred sort the entire database file is entered in a hash and sorted by data

transferred. If pageformat is equal to "Summary" only the text storage usage data is sent with HTML code. If pageformat is equal to "TopSummaryGraphs" a graph is included before the text data (See figure 8). If pageformat is equal to "TransSummary" only the text transfer data is sent with HTML code. If pageformat is equal to "TransSummaryGraphs" a graph is included before the text data (See figure 9).



## HPSS Account Information

Graph of Storage Used On Aug 01, 2001



This list shows top users of archive space in descending order.

Note: Totals are snapshots at a particular date and may not represent current account status.

No.	Name	Date	Total No. of Files in Archive	Total Archive Space Used (MB)	Transferred This Month (MB)
1	<a href="#">Roger "Roger"</a>	Aug 01, 2001	211,983	16,763,903	1,264,290
2	<a href="#">Denise "Denise"</a>	Aug 01, 2001	122,863	4,693,757	0
3	<a href="#">Robert "Robert"</a>	Aug 01, 2001	19,853	4,174,675	142,782
4	<a href="#">Hiro "Hiro"</a>	Aug 01, 2001	46,529	3,031,067	1,155
5	<a href="#">David "David"</a>	Aug 01, 2001	38,221	2,891,826	0
6	<a href="#">Charles "Charles"</a>	Aug 01, 2001	82,064	2,845,445	0
7	<a href="#">Ron "Ron"</a>	Aug 01, 2001	11,258	2,554,097	0
8	<a href="#">Jerry "Jerry"</a>	Aug 01, 2001	32,293	2,478,134	0
9	<a href="#">Barbara "Barbara"</a>	Aug 01, 2001	22,967	2,162,132	0
10	<a href="#">Robert "Robert"</a>	Aug 01, 2001	1,038,161	2,055,763	14,875

[HPSS Account Home](#)

If you have questions or comments about this page, contact

[Roger Skowlund](#)



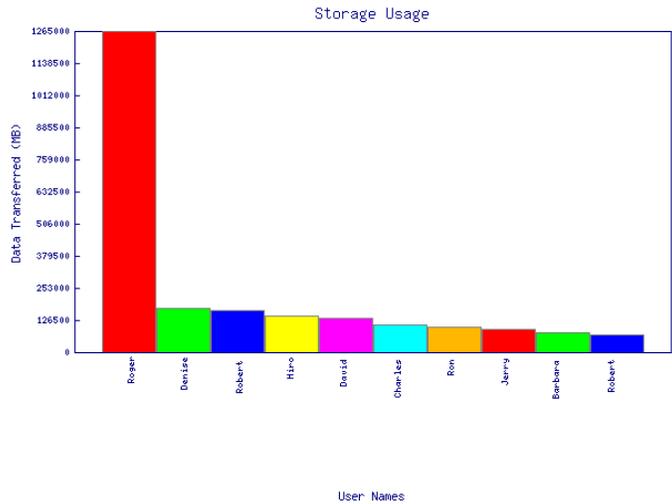
and [LLNL Disclaimers](#)

Figure 8 – Top Storage User Account Information with Graphs

# Data Storage Group

## HPSS Account Information

Graph of Transfers Made the Month Prior to Aug 01, 2001



This list shows top users of archive space in descending order.

Note: Totals are snapshots at a particular date and may not represent current account status.

No.	Name	Date	Total No. of Files in Archive	Total Archive Space Used (MB)	Transferred This Month (MB)
1	<a href="#">Roger "Roger"</a>	Aug 01, 2001	211,983	16,763,903	1,264,290
2	<a href="#">Denise "Denise"</a>	Aug 01, 2001	2,728	1,924,335	172,393
3	<a href="#">Robert "Robert"</a>	Aug 01, 2001	29,194	1,474,763	166,970
4	<a href="#">Hiro "Hiro"</a>	Aug 01, 2001	19,853	4,174,675	142,782
5	<a href="#">David "David"</a>	Aug 01, 2001	242,207	1,666,605	135,246
6	<a href="#">Charles "Charles"</a>	Aug 01, 2001	634	69,112	110,912
7	<a href="#">Ron "Ron"</a>	Aug 01, 2001	454,139	440,960	99,866
8	<a href="#">Jerry "Jerry"</a>	Aug 01, 2001	233,138	975,069	90,264
9	<a href="#">Barbara "Barbara"</a>	Aug 01, 2001	271	161,432	78,407
10	<a href="#">Robert "Robert"</a>	Aug 01, 2001	2,139	76,608	69,202

[HPSS Account Home](#)

If you have questions or comments about this page, contact

[Roger Skowhnd](#)



and [LLNL Disclaimers](#)

**Figure 9 – Top Transfer User Account Information with Graphs**

If the namesdetail parameter is defined the user has asked for contact information for a login. The login is used with the generisdce executable to load an array. The contact

information is pulled from the array and sent, with HTML code, to the standard output (See figure 10)



## HPSS Account Information

---

Name: Charles M. Hargreaves

Login: chargrea

UID: 7155

Phone Number: 925-422-4032

Email Address: hargreaves1@llnl.gov

Home Directory: /users/u05/chargrea

---

[HPSS Account Home](#)

If you have questions or comments about this page, contact

[Roger Skowlund](#)



and [LLNL Disclaimers](#)

**Figure 10 – Account Contact Information**

### The Graphing CGI Scripts

There are two scripts used to send the graph to the browser. The two scripts are virtually identical with the exception of the X and Y labels. A single scalar is accepted which consists of a continuous word, no blanks. The ‘\_’ character separates data and the ‘+’ to represent spaces. Using the PERL split function data is parsed into an array. All ‘+’ characters are switched back to spaces. Arrays are created for the X and Y axis data and they are loaded with the appropriate data. There is a variable for cycling the graph data colors that is set to not cycle for single user data and to cycle for multiple user data. This can be observed in the previous figures with graphs. The maximum Y data value is acquired for scaling the Y axis. This maximum number is rounded up to the next 1000 or gigabyte. This makes the Y axis scale look better. The graph parameters are set and the graph is plotted. The image is sent to standard output, in this case the browser.

## Additional Scripts

The `sumdailyrpts` script is used to convert daily runs of the HPSS accounting report to an equivalent monthly report. This script is a filter. The following command will create a summarized report for the individual reports in the directory: `cat ./directory_with_files/* | sumdailyrpts>acct.yyyymmdd.000000`. All of the reports I have converted have the time extension set to 000000. The script checks the input line format to decide if the line is accounting data. An accounting data line is parsed with data to be summed. The most recent file is used for the snapshot values and the file header. After all files are read the sorted lines are sent to the standard output.

The `rmfirstclmn` script is used to remove the left-most column. This is used to convert an HPSS version 4.2 accounting report to a version 4.1 report. Please note that this is only usable when cross-cell is not being used. This script removes the cross-cell column.

The `pmdesc.cgi` script is an HTML wrapper for a PERL script that shows what PERL modules is accessible. This identifies what modules are available to be used from a web page. The script came from the [www.perl.com](http://www.perl.com) web site.

## Conclusion

The intent of this document was to give a high level definition of the script written. There are additional comments in the scripts that provide a lower level description. If there is need for any additional description contact me at [hargreaves1@hotmail.com](mailto:hargreaves1@hotmail.com) or [hargreaves1@llnl.gov](mailto:hargreaves1@llnl.gov).

## Glossary

Associative Array - Also called a hash is like the array except the index values are not small nonnegative integers, but instead are arbitrary scalars. These scalars (called keys) are used to retrieve the values from the array.

Class of Service (COS) - Class of Service allows HPSS users to select a particular service level based on the space and performance requirements of a file. Each HPSS file will reside in a particular COS that is selected when the file is first created.

Comments - Identified by the '#' character are comment text that the compiler ignores.

Common Gateway Interface (GCI) - Is an interface PERL uses to generate dynamic web pages. The CGI scripts I have written query databases, send HTML and stream graphics.

Cron job - A script that resides on a cron server with a predetermined run time. The cron server executes the cron job when the current time is equal to the run time.

Cross-Cell - The concept of accepting authentication credentials from non-local authentication domains. Shows what site an account is connecting from.

Database - In this document it refers to a text file holding formatted data.

Filter - Here it refers to an executable program that accepts input data through the standard input and sends the filtered data to the standard output.

HTML - HyperText Markup Language is the universal browser language.

PERL - A programming language optimized for scanning arbitrary text files, extracting information from those text files, and printing reports based on that information. It is a practical language.

Regular expression - A pattern or template to be matched against a string.

Scalar - The simplest kind of data that Perl manipulates. Either a number, like 4 or 3.25e20, or a string of characters.

Script - An executable file holding a series of commands or operations.

Storage Class (SClass) - HPSS storage classes define attributes for various virtual volumes.

Storage Data - Accounting information for users.

## Bibliography

### Books:

Perl in a Nutshell  
ISBN 1-56592-286-7  
Copyright © 1999 O'Reilly & Associates

Learning Perl  
ISBN 1-56592-284-0  
Copyright © 1999 O'Reilly & Associates

Learning Perl on Win32 Systems  
ISBN 1-56592-324-3  
Copyright © 1999 O'Reilly & Associates

Programming Perl  
ISBN 1-56592-149-6  
Copyright © 1999 O'Reilly & Associates

Advanced Perl Programming  
ISBN 1-56592-220-4  
Copyright © 1999 O'Reilly & Associates

Perl Cookbook  
ISBN 1-56592-243-3  
Copyright © 1999 O'Reilly & Associates

HTML for Dummies  
ISBN 0-7645-0248-4  
Copyright © 1997 IDG Books Worldwide, Inc.

### Web Sites:

<http://www.sdsc.edu/hpss/hpss1.html>

[http://groups.google.com/googlegroups/deja\\_announcement.html](http://groups.google.com/googlegroups/deja_announcement.html)

<http://www.sdsc.edu/hpss/>

<http://stein.cshl.org/~lstein/>

<http://www.perldoc.com/>

[http://hpcf.nersc.gov/cgi-bin/storcharge/FY2001/repo\\_login?repo=mpccc/ccc](http://hpcf.nersc.gov/cgi-bin/storcharge/FY2001/repo_login?repo=mpccc/ccc)