

Moab Accounting Manager

Administrator Guide 8.1.2

November 2015



© 2015 Adaptive Computing Enterprises, Inc. All rights reserved.

Distribution of this document for commercial purposes in either hard or soft copy form is strictly prohibited without prior written consent from Adaptive Computing Enterprises, Inc.

Adaptive Computing, Cluster Resources, Moab, Moab Workload Manager, Moab Viewpoint, Moab Cluster Manager, Moab Cluster Suite, Moab Grid Scheduler, Moab Grid Suite, Moab Access Portal, and other Adaptive Computing products are either registered trademarks or trademarks of Adaptive Computing Enterprises, Inc. The Adaptive Computing logo and the Cluster Resources logo are trademarks of Adaptive Computing Enterprises, Inc. All other company and product names may be trademarks of their respective companies.

Adaptive Computing Enterprises, Inc.

1712 S. East Bay Blvd., Suite 300

Provo, UT 84606

+1 (801) 717-3700

www.adaptivecomputing.com



Scan to open online help

Welcome To Moab Accounting Manager	1
Chapter 1 Overview	3
Background	3
Conceptual Overview	3
Features	4
Interfaces	6
Documentation	8
License	8
Chapter 2 Initial Setup	11
Integration With Moab Workload Manager And Moab Web Services	11
Select An Appropriate Accounting Mode	11
Follow The Setup Guide For Your Selected Accounting Mode	12
Chapter 3 Strict Allocation Setup Guide	13
Set The Strict Allocation Accounting Mode	13
Decide On A Currency And Set The Currency Precision	14
Define Charge Rates	15
Define Accounts	15
Create Funds	16
Make Deposits	17
Check The Balance	18
Automate Allocation Renewal	18
Run A Job	19
The Usage Lien	19
The Usage Charge	20
Usage Refund	21
List Transactions	22
Examine Fund Statement	22
Chapter 4 Fast Allocation Setup Guide	25
Set The Fast Allocation Accounting Mode	26
Additional Moab Configuration	26
Decide On A Currency And Set The Currency Precision	27
Define Charge Rates	28
Define Accounts	28
Create Funds	28
Make Deposits	29
Check The Balance	30
Automate Allocation Renewal	31
Run A Job	31
The Usage Charge	31
Usage Refund	32

List Transactions	33
Examine Fund Statement	34
Chapter 5 Notional Charging Setup Guide	35
Set The Notional Charging Accounting Mode	35
Decide On A Currency And Set The Currency Precision	36
Define Charge Rates	37
Run A Job	37
The Usage Charge	37
Usage Refund	38
List Transactions	38
Chapter 6 Usage Tracking Setup Guide	41
Set The Usage Tracking Accounting Mode	41
Run A Job	42
Query Job Usage Information	42
Chapter 7 Managing Users	43
Creating Users	43
Querying Users	43
Modifying Users	44
Deleting Users	45
User Auto-Generation	45
Default User	46
Chapter 8 Managing Accounts	47
Creating Accounts	47
Querying Accounts	48
Modifying Accounts	49
Deleting Accounts	50
Account Auto-Generation	50
Default Account	51
Chapter 9 Managing Organization	53
Creating Organizations	53
Querying Organizations	53
Modifying Organizations	54
Deleting Organizations	54
Organization Auto-Generation	54
Default Organization	55
Chapter 10 Managing Funds	57
Creating Funds	58
Querying Funds	59

Modifying Funds	60
Making Deposits	60
Querying The Balance	62
Personal Balance	62
Making Withdrawals	63
Making Transfers	64
Obtaining A Fund Statement	64
Deleting Funds	65
Fund Auto-generation	66
Hierarchical Funds	66
Fund Priority	67
Chapter 11 Managing Allocations	69
Creating Allocations	71
Querying Allocations	71
Modifying Allocations	72
Deleting Allocations	72
Allocation Auto-generation	73
Allocation Precedence	73
Chapter 12 Managing Liens	75
Creating Liens	76
Querying Liens	76
Modifying Liens	77
Deleting Liens	77
Chapter 13 Managing Quotes	79
Creating Quotes	80
Creating Quote Templates	80
Querying Quotes	81
Modifying Quotes	82
Deleting Quotes	82
Chapter 14 Managing Usage Records	83
Creating A Usage Record	83
Querying Usage Records	84
Modifying A Usage Record	84
Deleting A Usage Record	85
Obtaining Usage Quotes	85
Making A Usage Lien	86
Charging For Usage	87
Issuing Usage Refunds	88
Customizing The Usage Record Object	89
Usage Record Property Verification	90

Usage Record Property Defaults	91
Usage Record Property Auto-Generation	91
Usage Record Property Instantiators	92
Chapter 15 Managing Itemized Charges	95
Querying Itemized Charges	95
Displaying Itemized Charges For A Transaction	96
Chapter 16 Managing Charge Rates	97
Creating Charge Rates	99
Querying Charge Rates	101
Modifying Charge Rates	101
Deleting Charge Rates	102
Chapter 17 Managing Transactions	103
Querying Transactions	103
Customizing The Transaction Object	103
Chapter 18 Managing Events	105
Creating Events	106
Querying Events	106
Modifying Events	107
Deleting Events	107
Chapter 19 Managing Notifications	109
Querying Notifications	109
Deleting Notifications	110
Chapter 20 Managing Roles	113
Creating Roles	113
Querying Roles	114
Modifying Roles	114
Deleting Roles	115
Chapter 21 Managing Passwords	117
Setting Passwords	117
Querying Passwords	117
Deleting Passwords	118
Chapter 22 Using The Gold Shell (goldsh)	119
Usage	119
Command Syntax	120
Valid Objects	121
Valid Actions For An Object	122
Valid Predicates For An Object And Action	122

Common Options	123
Common Actions Available For Most Objects	123
Multi-Object Queries	129
Chapter 23 Customizing Objects	131
Managing Objects	131
Creating A Custom Object	132
Querying Objects	132
Modifying An Object	132
Deleting An Object	133
Object Auto-Generation	133
Global Object-Based Defaults	134
Managing Attributes	134
Adding An Attribute To An Object	136
Querying Attributes	136
Modifying An Attribute	137
Removing An Attribute From An Object	137
Local Attribute-Based Defaults	138
Managing Actions	139
Adding An Action To An Object	139
Querying Actions	139
Modifying An Action	140
Removing An Action From An Object	140
Examples Creating Custom Objects	140
Chapter 24 Integration	143
Integrating With Moab Workload Manager	143
Integrating With Moab Web Services	145
Methods Of Interacting With Moab Accounting Manager	146
Chapter 25 Configuration Files	149
Site Configuration	149
Server Configuration	150
Client Configuration	152
GUI Configuration	155
Appendix A Commands Reference	159
Gbalance	162
Gchaccount	168
Gchalloc	172
Gcharge	176
Gchevent	185
Gchfund	189
Gchlien	195

Gchorg	199
Gchpasswd	201
Gchquote	204
Gchrate	207
Gchrole	209
Gchusage	213
Gchuser	220
Gdeposit	223
Glsaccount	230
Glsalloc	235
Glscharge	242
Glsconfig	246
Glsevent	248
Glsfund	253
Glslien	260
Glsnot	267
Glsorg	272
Glsquote	275
Glsrate	281
Glsrole	284
Glstrans	288
Glsusage	295
Glsuser	302
Gmkaccount	306
Gmkevent	310
Gmkfund	314
Gmklien	319
Gmkorg	323
Gmkquote	326
Gmkrate	330
Gmkrole	333
Gmkusage	336
Gmkuser	342
Gquote	346
Grefund	355
Greserve	359
Grmaccount	368
Grmalloc	370
Grmevent	373
Grmfund	375
Grmlien	377
Grmnot	379
Grmorg	381

Grmquote	383
Grmrate	386
Grmrole	388
Grmusage	390
Grmuser	393
Gstatement	395
Gtransfer	400
Gwithdraw	405
Goldsh	410
Goldd	413
Mybalance	415

Welcome to Moab Accounting Manager

Welcome to *Moab Accounting Manager Administrator Guide 8.1.2.*

This guide is intended as a reference for system administrators.

Related Topics

- [Overview on page 3](#)

Chapter 1 Overview

Moab Accounting Manager is an accounting management system that allows for usage tracking, charge accounting, and allocation enforcements for resource usage in technical computing environments. It acts somewhat like a bank in which credits are deposited into funds with constraints designating which entities may access the funds. As resources or services are utilized, funds are charged and usage recorded. It supports familiar operations such as deposits, withdrawals, transfers, and refunds. It provides balance and usage feedback to users, managers, and system administrators.

Since the accounting and billing models vary widely from organization to organization, Moab Accounting Manager has been designed to be extremely flexible, featuring customizable usage and fund configurations, and supporting a variety of tracking, charging and allocation models. Attention has been given to scalability, security, and fault tolerance.

Background

Moab Accounting Manager was originally developed as open source software called the Gold Allocation Manager at Pacific Northwest National Laboratory (PNNL) under the Department of Energy (DOE) Scalable Systems Software (SSS) SciDAC project. It has been extended and enhanced by Adaptive Computing Enterprises, Inc. (formerly Cluster Resources, Inc.) and is in production use at many commercial, government and educational sites.

Conceptual Overview

Moab Accounting Manager was designed to be used in technical computing environments for usage tracking, charge accounting and allocation enforcement. Usage tracking involves [resource usage in customizable usage records](#). Charge accounting involves calculating and recording charges for usage for invoicing or cost tracking. Allocation enforcement involves establishing limits on the use of system resources by defining separate funds having limited debit or credit balances.

In this overview, we will assume that you want to track or charge for workload resource usage. The use of resources by a job or reservation may result in a usage record. The usage record tracks the resources that were used, whom they were used by, and (optionally) how much the usage cost.

With Moab Accounting Manager, it is possible to allocate resource credits to various parties. This is done by associating a cost for the usage by deciding on a currency unit (generically referred to as credits), whether based on a real

currency such as dollars, or a reference currency such as billing units or processor seconds. Next you will define charge rates in this currency for the components of your usage (consumable resource costs, multipliers, fees, etc.). You may create pools of funds called allocations via deposits that can be debit- or credit-based, finite or infinite, and limited to a time frame in which they can be used. These allocations are deposited into logical containers called funds which have constraints that distinguish the conditions under which the funds can be used.

Moab Workload Manager interacts with Moab Accounting Manager to ensure sufficient funds and to track and charge for usage. A typical usage pattern might be as follows. When a job is submitted, a quote is obtained to see how much it will cost and to verify that you have sufficient funds. When it is time for the job to start, a lien (or hold) is placed against your funds for the amount of the requested resources. When the job ends, the appropriate fund is debited and the lien is removed. A usage record is updated with the charge amount and job usage details. The actual composition of the interactions is very flexible and will be defined by the accounting mode and interaction methods.

Features

Feature	Description
Dynamic Charging	Rather than post-processing resource usage records on a periodic basis to rectify fund balances, charging can occur incrementally throughout usage or at usage completion.
Liens	A hold (called a lien) is placed against the funds for the estimated amount of credits before the usage begins, followed by appropriate charges during and/or at the end of the usage, thereby preventing accounts from using more resources or services than were allocated to them.
Customizable Usage Records	Usage record fields can be configured by the site to track custom usage properties.
Flexible Fund Allocation	A uniquely flexible design allows resource or service credits to be allocated to arbitrary entities and purposes.
Expiring Allocations	Credits may be restricted for use within a designated time period allowing sites to implement a use-it-or-lose-it policy to prevent year-end resource exhaustion and establishing an allocation cycle.
Flexible Charging	The billing system can track and charge for composite time-based or non-time-based resource or service usage, and apply flexible charge multipliers and fees.

Feature	Description
Guaranteed Quotes	Users and resource brokers can determine ahead of time the cost of using resources or services.
Credit and Debit Allocations	Allocations feature an optional credit limit allowing support for both debit and credit models. This feature can also be used to enable overdraft protection for specific funds.
Infinite Allocations	Deposits can be made with infinite amounts or infinite credit limits when used with a supporting database.
Powerful Querying	A powerful querying and update mechanism (based on SQL queries) that facilitates flexible reporting and streamlines administrative tasks.
Nonintrusiveness	Object-level, attribute-level and correlated defaults may be established for arbitrary objects such as users, accounts and organizations. Additionally, these objects may be configured to be automatically created the first time they are seen by the resource management system. These features allow the accounting system to be used with less impact and involvement from users and administrators.
Consistency	Moab Accounting Manager has been engineered for robustness, consistency and resiliency. Complex operations are atomic and are automatically rolled back on failure.
Security	Multiple security mechanisms for strong authentication and encryption.
Role-Based Authorization	Fine-grained (instance-level) Role Based Access Controls are provided for all operations which allows users to view and manipulate only those objects permitted to them.
Dynamic Customization	Sites can create or modify record types on the fly enabling them to meet their custom accounting needs. Dynamic object creation allows sites to customize the types of accounting data they collect without modifying the code. This capability turns this system into a generalized information service. This capability is extremely powerful and can be used to manage all varieties of custom configuration data, or to function as a persistence interface for other components.
Web Interface	A powerful dynamic web-based GUI is provided for easy remote access for users, managers and administrators which displays only the actions allowed by their role.
Journaling	A journaling mechanism preserves the indefinite historical state of all objects and records. This powerful mechanism allows historical bank statements to be generated, provides an undo/redo capability and allows commands to be run as if it were any arbitrary time in the past.
Event Scheduler	An event engine can be used to schedule arbitrary Moab Accounting Manager commands to run periodically or at a designated time in the future.

Interfaces

Moab Accounting Manager provides a variety of means of interaction, including command-line interfaces, graphical user interfaces, application programming interfaces, and communication protocols.

Command-Line Clients

The command-line clients provided feature rich argument sets and built-in documentation. These commands allow scripting and are the preferred way to interact with Moab Accounting Manager for basic usage and administration. Use the `--help` option for usage information or the `--man` option for a manual page on any command.

Example 1-1: Listing Users Using a Command-Line Client

```
glsuser
```

Interactive Control Program

The `goldsh` command uses a control language to issue object-oriented requests to the server and display the results. The commands may be included directly as command-line arguments or read from stdin. Use the `ShowUsage :=True` option after a valid Object Action combination for usage information on the command.

Example 1-2: Listing Users Using the goldsh Control Program

```
goldsh User Query
```



The `goldsh` control program allows you to make powerful and sweeping modifications to many objects with a single command. Do not use this command unless you understand the syntax and the potential for unintended results.

Web-based Graphical User Interface

A powerful and easy-to-use web-based GUI permits browser access by users, managers and administrators according to their role definitions.

Example 1-3: Listing Users via the Web GUI

Click on **Manage Users -> List Users**

Perl API

You can access the full functionality via the Perl API. Use `perldoc` to obtain usage information for the Moab Accounting Manager Perl Gold modules.

Example 1-4: Listing Users Using the Perl API

```
use Gold;
my $request = new Gold::Request(object => "User", action => "Query");
my $response = $request->getResponse();
foreach my $datum ($response->getData())
{
    print $datum->toString(), "\n";
}
```

Java API

Although deprecated, the Java API may still be usable to interact with Moab Accounting Manager. The `javadoc` command can be run on the `contrib/java/gold` directory to generate documentation for the Gold java classes.

Example 1-5: Listing Users Using the Java API

```
import java.util.*;
import gold.*;
public class Test
{
    public static void main (String [] args) throws Exception
    {
        Gold.initialize();
        Request request = new Request("User", "Query");
        Response response = request.getResponse();
        Iterator dataItr = response.getData().iterator();
        while (dataItr.hasNext())
        {
            System.out.println((Datum) dataItr.next().toString());
        }
    }
}
```

SSSRMAP Wire Protocol

It is also possible to interact with Moab Accounting Manager by directly using the SSSRMAP Wire Protocol and Message Format over the network. Documentation for these protocols can be found at [SSS Resource Management and Accounting Documentation](#).

Example 1-6: Listing Users via the SSSRMAP Wire Protocol

```
POST /SSSRMAP HTTP/1.1
Content-Type: text/xml; charset="utf-8"
Transfer-Encoding: chunked
190
<?xml version="1.0" encoding="UTF-8"?>
<Envelope>
<Body actor="scottmo" chunking="True">
<Request action="Query" object="User"></Request>
</Body>
<Signature>
<DigestValue>azu4obZswzBt89OgATukBeLy6Y=</DigestValue>
<SignatureValue>YXE/C08XX3RX4PMU1bWju+5/E5M=</SignatureValue>
<SecurityToken type="Symmetric" name="scottmo"></SecurityToken>
</Signature>
</Envelope>
0
```

Documentation

The documentation for Moab Accounting Manager includes this Administrator Guide, release notes, built-in man pages, module documentation and online documentation.

- **Moab Accounting Manager Administrator Guide** — The Moab Accounting Manager Administrator Guide is a comprehensive manual for users and administrators of Moab Accounting Manager and includes information about features, interfaces, installation, getting started, usage, configuration and customization. The Administrator Guide can be found under the `$PREFIX/doc` directory in `.pdf` and `.html` formats. These documents are also available online.
- *Release Notes* — The Release Notes describe the primary features and fixes included in the release, along with notes to aid in migration from previous versions and can be found under the doc directory in the distribution tarball.
- *Command Line built-in Man Pages and Usage Synopsis* — All command-line clients support a `--man` option that provides full documentation of the command options and a `--help` option that provides a brief usage synopsis.
- *Module Perl Pod Documentation* — Documentation for Moab Accounting Manager Perl modules can be viewed by changing directory to the `$PREFIX/lib` directory and running `perldoc <modulename>`, e.g. `perldoc Gold::Request`.
- *Online Documentation* — The Moab Accounting Manager Administrator Guide can be found online at <http://www.adaptivecomputing.com/documentation>. The Gold project web page at <http://www.adaptivecomputing.com/resources/docs/gold/files/index.php> and includes the original Gold project documentation.

License

The Moab Accounting Manager software and associated documentation, data and information include parts which are copyrighted by Adaptive Computing Enterprises, Inc., and parts which are copyrighted by Battelle Memorial Institute. The terms and conditions for the use and redistribution of these parts are governed by the Moab Accounting Manager License and the BSD License respectively. Refer to the LICENSE file for details.

Moab Accounting Manager License

Copyright (C) 2006 - 2015 Pacific Northwest National Laboratory, Battelle Memorial Institute. All rights reserved.

The Moab Accounting Manager License specifies the terms and conditions for use and redistribution.

The Moab Accounting Manager License applies to the Moab Accounting Manager software offered by Adaptive Computing Enterprises, Inc. By installing or using this software, Licensee accepts a non-exclusive license from Adaptive Computing Enterprises, Inc. and is bound to accept acknowledgement of and abide by the notices and conditions of the Moab Accounting Manager License.

BSD License

Copyright (C) 2003 - 2005 Pacific Northwest National Laboratory, Battelle Memorial Institute. All rights reserved.

The BSD license specifies the terms and conditions for use and redistribution.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Battelle nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Chapter 2 Initial Setup

After installation, you will need to perform certain steps to prepare Moab Accounting Manager to fulfill its desired role in your environment.

Moab Accounting Manager can be configured in a myriad of use cases. It can be used in different accounting modes such as for usage tracking, notional charging, or allocation enforcement. This chapter will walk you through some of the steps you need to take to integrate and initialize the accounting manager.

Related Topics

[Integration with Moab Workload Manager and Moab Web Services on page 11](#)

[Select an Appropriate Accounting Mode on page 11](#)

[Follow the Setup Guide for Your Selected Accounting Mode on page 12](#)

Integration with Moab Workload Manager and Moab Web Services

If you have not already done so, you will need to integrate with the resource management system (Moab Workload Manager) and Moab Web Services. See [Integration on page 143](#).

Select an Appropriate Accounting Mode

Moab Accounting Manager can be configured to be used in a variety of different accounting modes. Some sites may wish to create and enforce resource usage limits through allocations. Others may want to impute a charge amount to their workload, but never deny workload based on availability of funds. Still others may not wish to calculate a charge at all, but simply record the usage details of the workload. Select the accounting mode from the following options that best matches your requirements.

- **strict-allocation** - Use this mode if you wish to strictly enforce allocation limits. Under this mode, you can prevent workload from running if the end-users do not have sufficient funds. Funds, allocations, quotes, liens, charge rates, and usage records support this mode. Before a job runs, MAM places a lien (or hold) against the user's funds to prevent overcommitment of their allocation. When a job completes, MAM removes the lien, debits the user's allocation, and records the workload usage details and charge in a usage record. This is the normal default.

- **fast-allocation** - Use this mode if you wish to debit allocations but need higher throughput by eliminating the lien and quote of strict-allocation mode. If you implement it properly through scripts, you can replace the lien and quote of strict-allocation mode with an asynchronous balance check, causing MAM to disable the accounts from further use after the first job that causes the fund to become negative. Funds, allocations, balance checks, charge rates and usage records support this mode.
- **notional-charging** - Use this mode if you wish to calculate and record charges for workload usage but not keep track of fund balances or allocation limits. Charge rates and usage records support this mode. The workload usage details and charge are recorded in a usage record.
- **usage-tracking** - Use this mode if you wish to simply record workload usage details but not to calculate a charge nor keep track of fund balances or allocation limits. Usage records support this mode.

Once you have determined the accounting mode that best matches your requirements, refer to the corresponding setup guide to prepare Moab Accounting Manager and Moab Workload Manager for your selected accounting mode. See [Follow the Setup Guide for Your Selected Accounting Mode on page 12](#).

Follow the Setup Guide for Your Selected Accounting Mode

Once you have determined the accounting mode that best matches your requirements (see [Select an Appropriate Accounting Mode on page 11](#)), refer to the corresponding setup guide to prepare Moab Accounting Manager and Moab Workload Manager for your selected accounting mode.

- strict-allocation - See [Strict Allocation Setup Guide on page 13](#)
- fast-allocation - See [Fast Allocation Setup Guide on page 25](#)
- notional-charging - See [Notional Charging Setup Guide on page 35](#)
- usage-tracking - See [Usage Tracking Setup Guide on page 41](#)

Chapter 3 Strict Allocation Setup Guide

This chapter will walk you through the typical steps needed to set up Moab Workload Manager and Moab Accounting Manager to use the strict allocation accounting mode.

With the strict allocation accounting mode, you can establish rigorous limits on the use of compute resources by your various parties. This is done by associating a cost for the usage by deciding on a currency unit, generically referred to as credits, whether based on a real currency such as dollars, or a reference currency such as billing units or processor-seconds, and then creating charge rates based on this currency. Funds are established to contain credit allocations attributed to specific accounts. Users are designated as members of the accounts. Deposits are made into funds associated with the accounts creating allocations. An allocation cycle may be established whereby allocations are considered for renewal on a regular periodic basis (such as yearly, quarterly or monthly).

Before a job is started, Moab Workload Manager will verify that the user has sufficient credits to run the job by attempting to place a hold against their funds (referred to as a lien). When a job completes, the user's funds will be debited via a charge, usage information will be recorded for the job and the lien will be removed. Users or managers can query the status of their allocations or details of their job charges and resource utilization.



You will need to be a Moab Accounting Manager System Administrator to perform many of the tasks in this chapter. It is assumed that you have already installed Moab Workload Manager and installed, bootstrapped, and started Moab Accounting Manager before performing the steps discussed in this chapter.



For testing or demo purposes, an initialization script is available that provides a similar affect to running the example commands in this chapter to minimally set up MAM for the strict-allocation accounting mode with a small amount of dummy sample data. It will not perform the Moab configuration steps described in this chapter. It can be cleaned up by running the `hpc-cleanup.sh` script.

```
$ ./hpc-strict-allocation.sh
```

Set the Strict Allocation Accounting Mode

Set the `AMCFG[mam]` MODE parameter to strict-allocation in `moab.cfg` and set the `accounting.mode` parameter to strict-allocation in both the `goldd.conf`

and `gold.conf` files. Since strict allocation is the default accounting mode in both Moab Workload Manager and Moab Accounting Manager, it may not be necessary to do anything here unless you were previously using a different accounting mode.

Example 3-1: Setting the Accounting Mode to strict-allocation

`AMCFG[] MODE` parameter must be set in the Moab server configuration file (`moab.cfg`). After editing the `moab.cfg` file, you will need to restart `moab`.

```
# vi /opt/moab/etc/moab.cfg
AMCFG[mam] MODE=strict-allocation

# mschedctl -R
```

The `accounting.mode` parameter must be set in the server and client configuration files (`goldd.conf` and `gold.conf`). After editing the `goldd.conf` file, you will need to restart `goldd`.

```
$ vi /opt/mam/etc/goldd.conf
accounting.mode = strict-allocation

$ vi /opt/mam/etc/gold.conf
accounting.mode = strict-allocation

$ goldd -r
```

Decide on a Currency and Set the Currency Precision

Since we will be calculating charges, we will need to decide which currency unit a MAM credit represents and sets the currency precision accordingly. For this example we will define a currency in which one credit represents the value of using one processor core for one hour. We will assume for simplicity that a processor-hour on one machine will have the same value as a processor-hour on another machine. Charge rates will be specified relative to this currency unit. Monetary transactions such as deposits and charges will be specified in terms of this currency. Since we want to be able to track and account for short jobs, we will specify a currency precision of two so that our currency credits will be represented as a floating point number with two decimal places. If instead we were to have chosen to use processor-seconds as the currency base, we would want to set the `currency.precision` value to zero since processor seconds can easily be represented as an integer with no decimal places. If we were to have chosen to use dollars as the currency base, we would have set the `currency.precision` value to two.

Example 3-2: Setting the Currency Precision to Two

The currency precision value must be set in the server and client configuration files (`goldd.conf` and `gold.conf`). It must also be set in the GUI configuration file (`goldg.conf`) if you will be using the web GUI. If you make changes in `goldd.conf`, you must restart `goldd`.

```
$ vi /opt/mam/etc/goldd.conf  
currency.precision = 2  
  
$ vi /opt/mam/etc/gold.conf  
currency.precision = 2  
  
$ goldd -r
```

Define Charge Rates

Since we are charging, we must establish the charge rates for the usage. In our example, we will define a charge rate that charges 1 credit for each processor-hour utilized by the job. See [Managing Charge Rates on page 97](#) for more detailed information on setting up charge rates.

Example 3-3: Define a Charge Rate for Processors

```
$ gmkrate -n Processors -z 1/h -d "1 credit per processor-hour"  
Successfully created 1 charge rate  
  
$ glsrate  
  
Name      Value   Amount Description  
-----  
Processors    1/h     1 credit per processor-hour
```

Define Accounts

Next we will define some accounts and assign users to the accounts. We will also associate each account with an organization so that usage reports can be generated for the organization level as well as the account and user level. We will create accounts for biology, chemistry, and film and assign them some users. The biology and chemistry account will be associated with the sciences organization while the film account will be associated with the arts organization. See [Managing Accounts on page 47](#) for more information on setting up accounts.

Example 3-4: Define the Biology, Chemistry, and Film Accounts

```
$ gmkaccount -a biology -o sciences -u amy,bob -d "Biology Department"
Successfully created 1 account

$ gmkaccount -a chemistry -o sciences -u amy,dave -d "Chemistry Department"
Successfully created 1 account

$ gmkaccount -a film -o arts -u bob,dave -d "Film Department"
Successfully created 1 account

$ glsaccount

Name      Active Users      Organization Description
-----
biology   True    amy, bob    sciences      Biology Department
chemistry True    amy, dave   sciences      Chemistry Department
film      True    bob, dave   arts          Film Department
```

Create Funds

The next task will be to create the funds which will hold the allocated credits. A fund is much like a numbered bank account, where credits can be deposited and are defined by constraints that distinguish who or what can use the contained credits and for what purposes. In this example, we will create a fund for each of the three accounts. The reason that funds are defined separately from accounts is that it is possible to create multiple funds for the same account. For example, you might have a fund that can be used for the chemistry account only when running the red cluster, and another fund that is used for the chemistry account when using a certain quality of service. See [Managing funds on page 57](#) for more detailed information on setting up funds.

In this example, we will assume that we want to establish a periodic allocation cycle with predesignated allocation amounts being deposited on a quarterly schedule. In order to facilitate this, we will associate a default deposit amount with the science funds. For the biology fund, we will configure it to make a resetting deposit for 5000 credits for each period. The chemistry fund is going to be disabled at the end of the allocation period. The film account will remain unaffected by allocation renewals. See [Managing Allocations on page 69](#) for more information on periodic allocations.

Example 3-5: Create a Fund for Each of the Three Accounts

```
$ gmkfund -a biology -n "biology" --default-deposit 5000
Successfully created 1 fund with id 1 and 1 constraint

$ gmkfund -a chemistry -n "chemistry" --default-deposit 0
Successfully created 1 fund with id 2 and 1 constraint

$ gmkfund -a film -n "film"
Successfully created 1 fund with id 3 and 1 constraint

$ glsfund

ID  Name      Constraints          Allocated Balance DefaultDeposit Description
---+-----+-----+-----+-----+-----+-----+-----+
1   biology   Account=biology    0.00   0.00   5000.00
2   chemistry Account=chemistry  0.00   0.00   0.00
3   film     Account=film       0.00   0.00
```

Make Deposits

Now we need to allocate credits to these funds by making deposits to them. An allocation has a start and end time associated with it declaring the time frame in which it can be used (defaulting to a start time of the present and an end time of infinity). It can also have a credit limit which defines the extent to which the allocation is allowed to go negative. Allocations can be reset on a periodic basis or future allocations with different time frames can be pre-created within a fund to establish an allocation cycle and set expectations for credit expenditure. See [Managing Allocations on page 69](#) and [Making Deposits on page 60](#) for additional information.

In this example, we will allocate 5000 and 3000 credits to the biology and chemistry accounts respectively. The film account will be given a credit limit of 2000 credits which allows them to charge up to 2000 credits before rectifying their fund. When making a deposit we must specify the fund we are depositing into unless the fund can be unambiguously determined by its constraint references (i.e. there is only a single fund associated with the account biology). In the next example, we will utilize the fund's deposit amount in the first deposit, specify the amount explicitly in the second deposit, and establish a credit allocation in the third deposit.

Example 3-6: Making Deposits

```
$ gdeposit -a biology
Successfully deposited 5000.00 credits into fund 1
Successfully created 1 allocation

$ gdeposit -z 3000 -a chemistry
Successfully deposited 3000.00 credits into fund 2
Successfully created 1 allocation

$ gdeposit -L 2000 -a film
No credits were deposited into fund 3
Successfully created 1 allocation
```

Let's examine the allocations we just created and its effect on the funds.

```
$ glsalloc
Id Fund StartTime EndTime InitialDeposit Allocated CreditLimit Remaining
PercentUsed
-- -----
1 1 2015-08-09 18:18:56 Infinity 5000.00 5000.00 0.00 5000.00
0.00
2 2 2015-08-09 18:18:56 Infinity 3000.00 3000.00 0.00 3000.00
0.00
3 3 2015-08-09 18:18:57 Infinity 0.00 0.00 2000.00 0.00
0.00

$ glsfund
Id Name Constraints Allocated Balance DefaultDeposit Description
-- -----
1 biology Account=biology 5000.00 5000.00 5000.00
2 chemistry Account=chemistry 3000.00 3000.00 0.00
3 film Account=film 0.00 0.00
```

Check the Balance

We can verify the resulting balance (see [Querying the Balance on page 62](#)).

Example 3-7: Let's Look at Amy's Balance

```
$ gbalance -u amy
Id Name Balance CreditLimit Available
-- -----
1 biology 5000.00 0.00 5000.00
1 chemistry 3000.00 0.00 3000.00
```

Automate Allocation Renewal

To facilitate the automatic renewal of our allocations, we will create a repeating event that resets all funds (see [Creating Events on page 106](#)) at the beginning of each new quarter.

Example 3-8: Create an Automatic Allocation Renewal Event

```
$ gmkevent --fire-command "Fund Reset" -s "2016-01-01" --rearm-period "3 months^"
Successfully created 1 event

$ glsevent

Id FireCommand FireTime      ArmTime          RalarmPeriod EndTime Notify
RalarmOnFailure FailureCommand CatchUp CreationTime      Description

-----
1 Fund Reset   2016-01-01 2015-08-09 18:21:28 3 months^           False
True      2015-08-09 18:21:28
```

Run a Job

Let's submit a job and examine the effects on the accounting system.

Example 3-9: Submit a Job

```
$ echo sleep 300 | mschedule -A chemistry -l procs=12,walltime=600
```

The Usage Lien

When a job starts, Moab Workload Manager typically creates a lien (or hold) against the appropriate allocations based on the estimated duration of the job. We will examine the effect of a running job on the accounting system (see [Managing Liens on page 75](#)).

Example 3-10: Examine the Effect of a Running Job on the Accounting System

```
$ glslien

Id Instance Amount StartTime          EndTime          Duration UsageRecord Funds
Description

-----
1 74        2.00 2015-08-09 18:22:42 2015-08-09 18:22:42 600      1           2
```

This lien will decrease our available balance by the amount reserved.

```
$ gbalance -u amy -a chemistry

Id Name      Balance Reserved Effective CreditLimit Available
-----
2 chemistry 3000.00    2.00    2998.00    0.00    2998.00
```

The actual allocation has not changed.

```
$ glsalloc -a chemistry

Id Fund Active StartTime EndTime Amount CreditLimit Deposited
Description
-- --
-- 
2 2 True 2015-08-09 18:18:56 Infinity 3000.00 0.00 3000.00
```

Note that the lien resulted in the initial creation of a usage record for the job with Stage Reserve.

```
$ glsusage

Id Type Instance Charge Stage User Group Account Organization Class
QualityOfService Machine Nodes Processors CPUTime Memory Disk Duration StartTime
EndTime Description
-- --
-- 
1 Job 74 0.00 Reserve amy faculty chemistry sciences batch normal
colony 1 12 0
```

The Usage Charge

After a job completes, any associated liens are removed and a charge is issued against the appropriate allocations based on the resources and actual wallclock time used by the job. An allocation is debited and the usage record is modified with the charge and usage information.

Example 3-11: Examine the Effect of a Completed Job on the Accounting System

Your allocation will now have gone down by the amount of the charge.

```
$ glsalloc -u amy -a chemistry

Id Fund StartTime EndTime InitialDeposit Allocated CreditLimit Remaining
PercentUsed
-- --
-- 
2 2 2015-08-09 18:18:56 Infinity 3000.00 3000.00 0.00 2999.00
0.03
```

However, your balance actually goes up (because the lien that was removed was larger than the actual charge).

```
$ gbalance -u amy -a chemistry

Id Name Balance Reserved Effective CreditLimit Available
-- --
2 chemistry 2999.00 0.00 2999.00 0.00 2999.00
```

A usage record for the job was updated as a side-effect of the charge (see [Querying Usage Records on page 84](#)).

```
$ glsusage
Id Type Instance Charge Stage User Group Account Organization Class
QualityOfService Machine Nodes Processors CPUTime Memory Disk Duration StartTime
EndTime Description
-----
-----
1 Job 74 1.00 Charge amy faculty chemistry sciences batch normal
colony 1 12 300 2015-08-09 18:22:42 2015-08-
09 18:27:42
```

Usage Refund

Now, we will illustrate the effect of issuing a refund for the user's job. For more information, see [Issuing Usage Refunds on page 88](#).

Example 3-12: Refund the Job

```
$ grefund -J 74
Successfully refunded 1.00 credits to usage record 1 for instance 74
```

Our balance is back as it was before the job ran.

```
$ gbalance -u amy -a chemistry
Id Name Balance Reserved Effective CreditLimit Available
-----
2 chemistry 3000.00 0.00 3000.00 0.00 3000.00
```

The allocation, of course, is likewise restored.

```
$ glsalloc -u amy -a chemistry
Id Fund StartTime EndTime InitialDeposit Allocated CreditLimit Remaining
PercentUsed
-----
2 2 2015-08-09 18:18:56 Infinity 3000.00 3000.00 0.00 3000.00
0.00
```

Notice that the usage charge is now zero because the job has been fully refunded.

```
$ glsusage
Id Type Instance Charge Stage User
Group Account Organization Class QualityOfService Machine Nodes Processors CPUTim
e Memory Disk Duration StartTime
EndTime Description
-----
1 Job 74 0.00 Charge amy faculty chemistry sciences batch normal
colony 1 12 300 2015-08-09 18:22:42 2015-
08-09 18:27:42
```

List Transactions

Let's list the transactions relating to this job (see [Querying Transactions on page 103](#)).

Example 3-13: Listing Transaction Details for This Job

```
$ glstrans -J 74 --full

  Id   Object      Action Actor   Key Child Instance Count Amount Delta Balance
Remaining User Account    Machine Fund Allocation UsageRecord Duration Description
Details

CreationTime          ModificationTime     Deleted RequestId TransactionId
-----  -----  -----
--  -----  -----  -----
-----  -----  -----
-----  -----  -----
-----  -----  -----
-----  -----  -----
6481 UsageRecord Create root    1        74      1           0
                         1
Charge=0,Deleted=False,Stage=Reserve,Type=Job
                                         2015-08-
09 18:22:42 2015-08-09 18:22:42 False    8615      6481
6484 UsageRecord Reserve root    1        74      1           2.00
      amy chemistry colony      1           600
Class=batch,Group=research,ItemizedCharges:=12 [Processors] * 0.0002777777777777778
[ChargeRate{Processors}] * 600 [Duration] =
2,Nodes=1,Organization=sciences,Processors=12,QualityOfService=normal
                                         2015-08-09 18:22:42 2015-08-09 18:22:42 False
8615      6484
6489 UsageRecord Charge root    1    74      74      1           1.00 -1.00 2999.00
2999.00 amy chemistry colony  2    2           1           300
CPUTime=1800,Class=batch,EndTime:=1413997758,Group=research,ItemizedCharges:=12
[Processors] * 0.00027777777777778 [ChargeRate{Processors}] * 300 [Duration] =
1,Nodes=1,Organization=sciences,Processors=12,QualityOfService=normal,StartTime:=14139
97458 2015-08-09 18:27:42 2015-08-09 18:27:42 False    8627      6489
6495 UsageRecord Refund root    1        74      1           1.00  1.00 3000.00
3000.00
                                         2015-08-09
18:28:58 2015-08-09 18:28:58 False    8636      6495
```

Examine Fund Statement

Finally, you can examine the fund statement for the activities (see [Obtaining a Fund Statement on page 64](#)).

Example 3-14: You Can Request an Itemized Fund Statement Over All Time for Use amy and the chemistry Account (fund 2)

```
$ gstatement -u amy -a chemistry
#####
# Includes fund 2 (chemistry)
# Generated on Tue Aug 9 18:29:53 2015.
#
# Reporting fund activity from -infinity to now.
#
#####
Beginning Balance: 0.00
-----
Total Credits: 3001.00
Total Debits: -1.00
-----
Ending Balance: 30000.00
#####
Credit Detail #####
Object Action Instance Amount Balance Time
-----
Fund Deposit 3000.00 3000.00 2015-08-09 18:18:56
UsageRecord Refund 74 1.00 3000.00 2015-08-09 18:28:58
#####
Debit Detail #####
Object Action Instance Account User Machine Amount Balance Time
-----
UsageRecord Charge 74 chemistry amy colony -1.00 2999.00 2015-08-09 18:27:42
#####
End of Report #####

```


Chapter 4 Fast Allocation Setup Guide

This chapter will walk you through the typical steps needed to set up Moab Workload Manager and Moab Accounting Manager to use the fast allocation accounting mode.

If you want to enforce allocations by debiting funds in Moab Accounting Manager, but need higher throughput by eliminating the lien and quote operations of the strict allocation accounting mode, you can use the fast allocation accounting mode. With the fast allocation accounting mode, Moab Workload Manager checks a cached account balance, and jobs or reservations may be prevented from starting or continuing after the balance has become zero or negative. As with the strict allocation accounting mode, you establish limits on the use of compute resources by your various parties. This is done by associating a cost for the usage by deciding on a currency unit, generically referred to as credits, whether based on a real currency such as dollars, or a reference currency such as billing units or processor-seconds, and then creating charge rates based on this currency. Funds are established to contain credit allocations attributed to specific accounts. Users are designated as members of the accounts. Deposits are made into funds associated with the accounts creating allocations. An allocation cycle may be established whereby allocations are considered for renewal on a regular periodic basis (such as yearly, quarterly or monthly).

Before a job is started, Moab Workload Manager will check its internal cache to verify that the user has a positive account balance. When a job completes, the user's funds will be debited via a charge, usage information will be recorded for the job and Moab's account balance cache is updated. Since Moab Accounting Manager is not contacted at job submission or start time in order to verify account membership, additional configuration is needed in Moab to synchronize account information with Moab Accounting Manager. Additionally, since the cache in Moab Workload Manager supports only account based funds, when using the fast allocation accounting mode, funds having no constraints or having non-account constraints should not be used.



You will need to be a Moab Accounting Manager System Administrator to perform many of the tasks in this chapter. It is assumed that you have already installed Moab Workload Manager and installed, bootstrapped, and started Moab Accounting Manager before performing the steps discussed in this chapter.

- i** For testing or demo purposes, an initialization script is available that provides a similar affect to running the example commands in this chapter to minimally set up MAM for the fast-allocation accounting mode with a small amount of dummy sample data. It will not perform the Moab configuration steps described in this chapter. It can be cleaned up by running the hpc-cleanup.sh script.

```
$ ./hpc-fast-allocation.sh
```

Set the Fast Allocation Accounting Mode

Set the AMCFG[mam] MODE parameter to fast-allocation in `moab.cfg` and set the accounting.mode parameter to fast-allocation in both the `goldd.conf` and `gold.conf` files.

Example 4-1: Setting the Accounting Mode to fast-allocation

AMCFG[] MODE parameter must be set in the Moab server configuration file (`moab.cfg`). After editing the `moab.cfg` file, you will need to restart moab.

```
# vi /opt/moab/etc/moab.cfg
AMCFG [mam] MODE=fast-allocation

# mschedctl -R
```

The accounting.mode parameter must be set in the server and client configuration files (`goldd.conf` and `gold.conf`). After editing the `goldd.conf` file, you will need to restart `goldd`.

```
$ vi /opt/mam/etc/goldd.conf
accounting.mode = fast-allocation

$ vi /opt/mam/etc/gold.conf
accounting.mode = fast-allocation

$ goldd -r
```

Additional Moab Configuration

Since Moab will be checking an internal account balance cache when starting jobs and reservations instead of contacting Moab Accounting Manager, we need to periodically update Moab Workload Manager with account information from Moab Accounting Manager so that Moab can correctly apply default accounts and enforce account memberships. Additionally, it is beneficial to poll the account balances periodically so that external actions such as new deposits, transfers, etc. will be reflected properly in Moab's account balance cache.

Example 4-2: Configuring Moab to Synchronize Account Information

We will set AMCFG[] CREATECRED=TRUE in order to enable Moab to query accounts, users, user membership in accounts, and users' default accounts from Moab Accounting Manager and define them in Moab. We will set the AMCFG[] REFRESHPERIOD parameter to the interval that we want Moab to update these credential updates as well as its account balance cache. We will also set the ENFORCEACCOUNTACCESS parameter to TRUE in order to tell Moab to restrict users to use only accounts to which they belong.

```
# vi /opt/moab/etc/moab.cfg
AMCFG [mam] CREATECRED=TRUE
AMCFG [mam] REFRESHPERIOD=2:00:00
ENFORCEACCOUNTACCESS TRUE
# mschedctl -R
```

Decide on a Currency and Set the Currency Precision

Since we will be calculating charges, we will need to decide what currency unit a MAM credit represents and set the currency precision accordingly. For this example we will define a currency in which one credit represents the value of using one processor core for one hour. We will assume for simplicity that a processor-hour on one machine will have the same value as a processor-hour on another machine. Charge rates will be specified relative to this currency unit. Monetary transactions such as deposits and charges will be specified in terms of this currency. Since we want to be able to track and account for short jobs, we will specify a currency precision of two so that our currency credits will be represented as a floating point number with two decimal places. If instead we were to have chosen to use processor-seconds as the currency base, we would want to set the currency.precision value to zero since processor seconds can easily be represented as an integer with no decimal places. If we were to have chosen to use dollars as the currency base, we would have set the currency.precision value to two.

Example 4-3: Setting the Currency Precision to Two

The currency precision value must be set in the server and client configuration files (`goldd.conf` and `gold.conf`). It must also be set in the GUI configuration file (`goldg.conf`) if you will be using the web GUI. If you make changes in `goldd.conf`, you must restart `goldd`.

```
$ vi /opt/mam/etc/goldd.conf
currency.precision = 2
$ vi /opt/mam/etc/gold.conf
currency.precision = 2
$ goldd -r
```

Define Charge Rates

Since we are charging, we must establish the charge rates for the usage. In our example, we will define a charge rate that charges 1 credit for each processor-hour utilized by the job. See [Managing Charge Rates on page 97](#) for more detailed information on setting up charge rates.

Example 4-4: Define a Charge Rate for Processors

```
$ gmkrate -n Processors -z 1/h -d "1 credit per processor-hour"
Successfully created 1 charge rate

$ glsrate

Name      Value Amount Description
-----
Processors    1/h     1 credit per processor-hour
```

Define Accounts

Next we will define some accounts and assign users to the accounts. We will also associate each account with an organization so that usage reports can be generated for the organization level as well as the account and user level. We will create accounts for biology, chemistry and film and assign them some users. The biology and chemistry account will be associated with the sciences organization while the film account will be associated with the arts organization. See [Managing Accounts on page 47](#) for more information on setting up accounts.

Example 4-5: Define the Biology, Chemistry, and Film Accounts

```
$ gmkaccount -a biology -o sciences -u amy,bob -d "Biology Department"
Successfully created 1 account

$ gmkaccount -a chemistry -o sciences -u amy,dave -d "Chemistry Department"
Successfully created 1 account

$ gmkaccount -a film -o arts -u bob,dave -d "Film Department"
Successfully created 1 account

$ glsaccount

Name      Active Users      Organization Description
-----
biology   True    amy,bob    sciences      Biology Department
chemistry True    amy,dave   sciences      Chemistry Department
film      True    bob,dave   arts          Film Department
```

Create Funds

The next task will be to create the funds which will hold the allocated credits. A fund is much like a numbered bank account, where credits can be deposited and are defined by constraints that distinguish who or what can use the contained credits and for what purposes. In this example, we will create a fund for each of the three accounts. The reason that funds are defined separately from accounts is that it is possible to create multiple funds for the same account. For example, you might have a fund that can be used for the chemistry account only when running the red cluster, and another fund that is used for the chemistry account when using a certain quality of service. See [Managing funds on page 57](#) for more detailed information on setting up funds.

In this example, we will assume that we want to establish a periodic allocation cycle with predesignated allocation amounts being deposited on a quarterly schedule. In order to facilitate this, we will associate a default deposit amount with the science funds. For the biology fund, we will configure it to make a resetting deposit of 5000 credits for each period. The chemistry fund is going to be disabled at the end of the allocation period. The film account will remain unaffected by allocation renewals. See [Managing Allocations on page 69](#) for more information on periodic allocations.

Example 4-6: Create a Fund for Each of the Three Accounts

```
$ gmkfund -a biology -n "biology" --default-deposit 5000
Successfully created 1 fund with id 1 and 1 constraint

$ gmkfund -a chemistry -n "chemistry" --default-deposit 0
Successfully created 1 fund with id 2 and 1 constraint

$ gmkfund -a film -n "film"
Successfully created 1 fund with id 3 and 1 constraint

$ glsfund
```

Id	Name	Constraints	Allocated	Balance	DefaultDeposit	Description
1	biology	Account=biology	0.00	0.00	5000.00	
2	chemistry	Account=chemistry	0.00	0.00	0.00	
3	film	Account=film	0.00	0.00		

Make Deposits

Now we need to allocate credits to these funds by making deposits to them. An allocation has a start and end time associated with it declaring the time frame in which it can be used (defaulting to a start time of the present and an end time of infinity). It can also have a credit limit which defines the extent to which the allocation is allowed to go negative. Allocations can be reset on a periodic basis or future allocations with different time frames can be precreated within a fund to establish an allocation cycle and set expectations for credit expenditure. See [Managing Allocations on page 69](#) and [Making Deposits on page 60](#) for additional information.

In this example, we will allocate 5000 and 3000 credits to the biology and chemistry accounts respectively. The film account will be given a credit limit of 2000 credits which allows them to charge up to 2000 credits before settling their fund. When making a deposit we must specify the fund we are depositing into unless the fund can be unambiguously determined by its constraint references (i.e. there is only a single fund associated with the account biology). In the next example, we will utilize the fund's default deposit amount in the first deposit, specify the amount explicitly in the second deposit and establish a credit allocation in the third deposit.

Example 4-7: Making Deposits

```
$ gdeposit -a biology
Successfully deposited 5000.00 credits into fund 1
Successfully created 1 allocation

$ gdeposit -z 3000 -a chemistry
Successfully deposited 3000.00 credits into fund 2
Successfully created 1 allocation

$ gdeposit -L 2000 -a film
No credits were deposited into fund 3
Successfully created 1 allocation
```

Let's examine the allocations we just created and its effect on the funds.

```
$ glsalloc
Id Fund StartTime EndTime InitialDeposit Allocated CreditLimit Remaining
PercentUsed
--- -----
1 1 2015-08-09 18:18:56 Infinity      5000.00 5000.00 0.00 5000.00
0.00
2 2 2015-08-09 18:18:56 Infinity      3000.00 3000.00 0.00 3000.00
0.00
3 3 2015-08-09 18:18:57 Infinity      0.00 0.00 2000.00 0.00
0.00

$ glsfund
Id Name Constraints Allocated Balance DefaultDeposit Description
--- -----
1 biology Account=biology 5000.00 5000.00 5000.00
2 chemistry Account=chemistry 3000.00 3000.00 0.00
3 film Account=film 0.00 0.00
```

Check the Balance

We can verify the resulting balance (see [Querying the Balance on page 62](#)).

Example 4-8: Let's Look at Amy's Balance

```
$ gbalance -u amy
Id Name      Balance CreditLimit Available
-- -----
1 biology   5000.00    0.00  5000.00
1 chemistry 3000.00    0.00  3000.00
```

Automate Allocation Renewal

To facilitate the automatic renewal of our allocations, we will create a repeating event that resets all funds (see [Creating Events on page 106](#)) at the beginning of each new quarter.

Example 4-9: Create an Automatic Allocation Renewal Event

```
$ gmkevent --fire-command "Fund Reset" -s "2016-01-01" --rearm-period "3 months^"
Successfully created 1 event

$ glsevent
Id FireCommand FireTime      ArmTime          RalarmPeriod EndTime Notify
RalarmOnFailure FailureCommand CatchUp CreationTime       Description
-- -----
1 Fund Reset   2016-01-01 2015-08-09 18:21:28 3 months^           False
                    True      2015-08-09 18:21:28
```

Run a Job

Now, let's submit a job and examine the effects on the accounting system.

Example 4-10: Submit a Job

```
$ echo sleep 300 | msub -A chemistry -l procs=12,walltime=600
```

The Usage Charge

After a job completes, a charge is issued against the appropriate allocations based on the resources and actual wallclock time used by the job. An allocation is debited and the usage record is modified with the charge and usage information.

Example 4-11: Examine the Effect of a Completed Job on the Accounting System

Your allocation and balance will have gone down by the amount of the charge.

```
$ glsalloc -u amy -a chemistry

Id Fund StartTime           EndTime   InitialDeposit Allocated CreditLimit Remaining
PercentUsed
-- -----
-- -----
2 2    2015-08-09 18:18:56 Infinity      3000.00  3000.00      0.00  2999.00
0.03

$ gbalance -u amy -a chemistry

Id Name      Balance CreditLimit Available
-- -----
2 chemistry 2999.00     0.00   2999.00
```

The usage record for the job was updated as a side-effect of the charge (see [Querying Usage Records on page 84](#)).

```
$ glsusage

Id Type Instance Charge Stage User Group Account Organization Class
QualityOfService Machine Nodes Processors CPUTime Memory Disk Duration StartTime
EndTime          Description
-- -----
-- -----
-- -----
1 Job 74        1.00 Charge amy faculty chemistry sciences batch normal
colony 1       12                         300      2015-08-09 18:22:42 2015-08-
09 18:27:42
```

Usage Refund

Now, we will illustrate the effect of issuing a refund for the user's job (see [Issuing Usage Refunds on page 88](#)).

Example 4-12: Refund the Job

```
$ grefund -J 74

Successfully refunded 1.00 credits to usage record 1 for instance 74
```

Our balance is back as it was before the job ran.

```
$ gbalance -u amy -a chemistry

Id Name      Balance CreditLimit Available
-- -----
2 chemistry 3000.00     0.00   3000.00
```

The allocation, of course, is likewise restored.

```
$ glsalloc -u amy -a chemistry
```

Id	Fund	StartTime	EndTime	InitialDeposit	Allocated	CreditLimit	Remaining	PercentUsed
2	2	2015-08-09	18:18:56	Infinity	3000.00	3000.00	0.00	3000.00
				0.00				

Notice that the usage charge is now zero because the job has been fully refunded.

```
$ glsusage
Id Type Instance Charge Stage User Group Account Organization Class
QualityOfService Machine Nodes Processors CPUTime Memory Disk Duration StartTime
EndTime Description
-----
-----
1 Job 74      0.00 Charge amy faculty chemistry sciences      batch normal
    colony 1     12                               300      2015-08-09 18:22:42 2015-08-
09 18:27:42
```

List Transactions

Let's list the transactions relating to this job (see [Querying Transactions on page 103](#)).

Example 4-13: Listing Transaction Details for This Job

Examine Fund Statement

Finally, you can examine the fund statement for our activities (see [Obtaining a Fund Statement on page 64](#)).

Example 4-14: We Can Request an Itemized Fund Statement Over All Time for Use amy and the chemistry Account (fund 2)

```
$ gstatement -u amy -a chemistry
#####
#
# Includes fund 2 (chemistry)
# Generated on Tue Aug 9 18:29:53 2015.
# Reporting fund activity from -Infinity to Now.
#
#####
Beginning Balance:          0.00
-----
Total Credits:            3001.00
Total Debits:              -1.00
-----
Ending Balance:            3000.00
#####
##### Credit Detail #####
Object      Action  Instance Amount  Balance Time
-----
Fund        Deposit    3000.00 3000.00 2015-08-09 18:18:56
UsageRecord Refund    74       1.00   3000.00 2015-08-09 18:28:58
#####
##### Debit Detail #####
Object      Action  Instance Account  User Machine Amount Balance Time
-----
UsageRecord Charge 74      chemistry amy   colony   -1.00 2999.00 2015-08-09 18:27:42
#####
##### End of Report #####
#
```

Chapter 5 Notional Charging Setup Guide

This chapter will walk you through the typical steps needed to set up Moab Workload Manager and Moab Accounting Manager to use the notional charging accounting mode.

Some sites may want to use Moab Accounting Manager to calculate and record charges, but not to impose allocation limits or prevent any workload from running. With notional charging, charge rates will be used to calculate a cost for using resources, but there is no need to make deposits, debit funds or keep track of allocation limits. Although it would be possible to set up accounts and assign users to specific accounts, this chapter will assume that account membership is not going to be enforced. If you would prefer to enforce account membership, you can continue to use the notional charging accounting setup as described in this chapter, but you will need to additionally define accounts and account memberships as well as configure Moab to synchronize account information from Moab Accounting Manager as described in the [Fast Allocation Setup Guide on page 25](#). Liens, balance queries and quotes are not needed. Our main task is to define charge rates.

At the end of a job, Moab Workload Manager will send usage information to the accounting manager. Moab Accounting Manager will calculate a charge and store this with the job usage record.



You will need to be a Moab Accounting Manager System Administrator to perform many of the tasks in this chapter. It is assumed that you have already installed Moab Workload Manager and installed, bootstrapped, and started Moab Accounting Manager before performing the steps discussed in this chapter.



For testing or demo purposes, an initialization script is available that provides a similar affect to running the example commands in this chapter to minimally set up MAM for the notional-charging accounting mode with a small amount of dummy sample data. It will not perform the Moab configuration steps described in this chapter. It can be cleaned up by running the `hpc-cleanup.sh` script.

```
$ ./hpc-notional-charging.sh
```

Set the Notional Charging Accounting Mode

Set the `AMCFG[mam]` MODE parameter to notional-charging in `moab.cfg` and set the `accounting.mode` parameter to notional-charging in both the `goldd.conf` and `gold.conf` files.

Example 5-1: Setting the Accounting Mode to notional-charging

AMCFG[] MODE parameter must be set in the Moab server configuration file (moab.cfg). After editing the moab.cfg file, you will need to restart moab.

```
# vi /opt/moab/etc/moab.cfg
AMCFG[mam] MODE=notional-charging

# mschedctl -R
```

The accounting.mode parameter must be set in the server and client configuration files (goldd.conf and gold.conf). After editing the goldd.conf file, you will need to restart goldd.

```
$ vi /opt/mam/etc/goldd.conf
accounting.mode = notional-charging

$ vi /opt/mam/etc/gold.conf
accounting.mode = notional-charging

$ goldd -r
```

Decide on a Currency and Set the Currency Precision

Since we will be calculating charges, we will need to decide what currency unit a MAM credit represents and set the currency precision accordingly. For this example we will define a currency in which one credit represents the value of using one processor core for one hour. We will assume for simplicity that a processor-hour on one machine will have the same value as a processor-hour on another machine. Charge rates will be specified relative to this currency unit. Monetary transactions such as deposits and charges will be specified in terms of this currency. Since we want to be able to track and account for short jobs, we will specify a currency precision of two so that our currency credits will be represented as a floating point number with two decimal places. If instead we were to have chosen to use processor-seconds as the currency base, we would want to set the currency.precision value to zero since processor seconds can easily be represented as an integer with no decimal places. If we were to have chosen to use dollars as the currency base, we would have set the currency.precision value to two.

Example 5-2: Setting the Currency Precision to Two

The currency precision value must be set in the server and client configuration files (goldd.conf and gold.conf). It must also be set in the GUI configuration file (goldg.conf) if you will be using the web GUI. If you make changes in goldd.conf, you must restart goldd.

```
$ vi /opt/mam/etc/goldd.conf  
currency.precision = 2  
  
$ vi /opt/mam/etc/gold.conf  
currency.precision = 2  
  
$ goldd -r
```

Define Charge Rates

Since we are charging, we must establish the charge rates for the usage. In our example, we will define a charge rate that charges 1 credit for each processor-hour utilized by the job. See [Managing Charge Rates on page 97](#) for more detailed information on setting up charge rates.

Example 5-3: Define a Charge Rate for Processors

```
$ gmkrate -n Processors -z 1/h -d "1 credit per processor-hour"  
Successfully created 1 charge rate  
  
$ glsrate  
  
Name      Value   Amount Description  
-----  
Processors    1/h     1 credit per processor-hour
```

Run a Job

Now, let's submit a job and examine the effects on the accounting system.

Example 5-4: Submit a Job

```
$ echo sleep 300 | msub -A chemistry -l procs=12,walltime=600
```

The Usage Charge

After a job completes, a usage record is generated with the charge and resource usage information.

Example 5-5: List the Usage and Charge for Our Job

```
$ glsusage
Id Type Instance Charge Stage User Group Account Organization Class
QualityOfService Machine Nodes Processors CPUTime Memory Disk Duration StartTime
EndTime Description
-- -----
-- -----
1 Job 74      1.00 Charge amy faculty chemistry sciences batch normal
  colony 1     12                               300      2015-08-09 18:22:42 2015-08-
09 18:27:42
```

Usage Refund

Now, we will illustrate the effect of issuing a refund for the user's job (see [Issuing Usage Refunds on page 88](#)).

Example 5-6: Refund the Job

```
$ grefund -J 74
Successfully refunded 1.00 credits to usage record 1 for instance 74
```

Notice that the usage charge is now zero because the job has been fully refunded.

```
$ glsusage
Id Type Instance Charge Stage User Group Account Organization Class
QualityOfService Machine Nodes Processors CPUTime Memory Disk Duration StartTime
EndTime Description
-- -----
-- -----
1 Job 74      0.00 Charge amy faculty chemistry sciences batch normal
  colony 1     12                               300      2015-08-09 18:22:42 2015-08-
09 18:27:42
```

List Transactions

Let's list the transactions relating to this job (see [Querying Transactions on page 103](#)).

Example 5-7: Listing Transaction Details for This Job

```
$ glstrans -J 74 --full

Id Object      Action Actor   Key Child Instance Count Amount User Account
Machine Fund Allocation UsageRecord Duration Description Details

Deleted RequestId TransactionId                                         CreationTime           ModificationTime
----- ----- -----
-
-----
-----
-----
-----
6489 UsageRecord Charge    root     1    74      74          1        1.00 amy   chemistry colony
1                         300
CPUTime=1800,Class=batch,EndTime:=1413997758,Group=research,ItemizedCharges:=12
[Processors] * 0.0002777777777777778 [ChargeRate{Processors}] * 300 [Duration] =
1,Nodes=1,Organization=sciences,Processors=12,QualityOfService=normal,StartTime:=14139
97458 2015-08-09 18:27:42 2015-08-09 18:27:42 False    8627      6489
6495 UsageRecord Refund   root     1         74          1        1.00
1

2015-08-09 18:28:58 2015-08-09 18:28:58 False
8636      6495
```


Chapter 6 Usage Tracking Setup Guide

This chapter will walk you through the typical steps needed to set up Moab Workload Manager and Moab Accounting Manager to use the usage tracking accounting mode.

When used solely for usage tracking, Moab Accounting Manager logs resource usage in usage records. This usage can be queried to report what resources were used when and by whom. In this case, there is no need for charge rates, funds, allocations, liens or quotes. There is no need to define account membership.

At the end of a job, Moab Workload Manager will send usage information to the accounting manager. Moab Accounting Manager will store this information in a job usage record.



You will need to be a Moab Accounting Manager System Administrator to perform many of the tasks in this chapter. It is assumed that you have already installed Moab Workload Manager and installed, bootstrapped, and started Moab Accounting Manager before performing the steps discussed in this chapter.



For testing or demo purposes, an initialization script is available that provides a similar affect to running the example commands in this chapter to minimally set up MAM for the usage-tracking accounting mode with a small amount of dummy sample data. It will not perform the Moab configuration steps described in this chapter. It can be cleaned up by running the hpc-cleanup.sh script.

```
$ ./hpc-usage-tracking.sh
```

Set the Usage Tracking Accounting Mode

Set the AMCFG[mam] MODE parameter to usage-tracking in `moab.cfg` and set the accounting.mode parameter to usage-tracking in both the `goldd.conf` and `gold.conf` files.

Example 6-1: Setting the Accounting Mode to usage-tracking.

AMCFG[] MODE parameter must be set in the Moab server configuration file (`moab.cfg`). After editing the `moab.cfg` file, you will need to restart moab.

```
# vi /opt/moab/etc/moab.cfg
AMCFG [mam] MODE=usage-tracking

# mschedctl -R
```

The accounting.mode parameter must be set in the server configuration file (goldd.conf). After editing the goldd.conf file, you will need to restart goldd.

```
$ vi /opt/mam/etc/goldd.conf  
accounting.mode = usage-tracking  
  
$ goldd -r
```

Run a Job

Now, let's submit a job and examine the effects on the accounting system.

Example 6-2: Submit a Job

```
$ echo sleep 300 | msub -A chemistry -l procs=12,walltime=600
```

Query Job Usage Information

After a job completes, usage information is recorded. Let's examine the usage record that was created (see [Querying Usage Records on page 84](#)).

Example 6-3: List Usage Records

```
$ glsusage  
  
Id Type Instance Stage User Group Account Organization Class QualityOfService  
Machine Nodes Processors CPUPTime Memory Disk Duration StartTime EndTime  
Description  
-- -- -- -- -- -- -- -- -- --  
---- ---- ---- ----  
----  
1 Job 74 Charge amy faculty chemistry sciences batch normal  
colony 1 12 300 2015-09-17 15:42:43 2015-09-17  
15:47:22
```

Chapter 7 Managing Users

A user is a person authorized to use a resource or service. Default user properties include the common name, phone number, email address, default account, and description for that person. A user can be created, queried, modified, and deleted. By default, a standard user may only query his or her own user record.

User queries allow the specification of filter options which narrow down the users that will be returned to those belonging to the specified account.

Creating Users

To create a new user, use the command `gmkuser`:

```
gmkuser [-A | -I] [-n common_name] [-F phone_number] [-E email_address] [-a default_account] [-d description] [-X, --extension property=value]... [--debug] [--site site_name] [--help] [--man] [--quiet] [--verbose] [--version] {[-u] user_name}
```

Example 7-1: Creating a user

```
$ gmkuser -n "Smith, Robert F." -E "bob@bank.com" -F "(509) 555-1234" bob
Successfully created 1 user
```

Related Topics

[gmkuser on page 342](#)

Querying Users

To display user information, use the command `glsuser`:

```
glsuser [-A | -I] [-X, --extension property=value]... [-a account_name] [--full] [-show attribute_name,...] [--long] [--wide] [--format csv|standard] [--debug] [--site site_name] [--help] [--man] [--quiet] [--version] {[-u] user_pattern}
```

Example 7-2: Listing standard info about active users

```
$ glsuser -A
Name Active CommonName PhoneNumber EmailAddress
DefaultAccount Description
-----
-- -
amy True Wilkes, Amy (509) 555-8765 amy@bank.com
bob True Smith, Robert F. (509) 555-1234 bob@bank.com
```

Example 7-3: Displaying bob's phone number

```
$ glsuser -show PhoneNumber bob -quiet
(509) 555-1234
```

Example 7-4: Listing amy's accounts

```
$ glsuser -show Accounts amy -l -q
-----
chemistry
biology
```

Example 7-5: Listing all users belonging to the chemistry account

```
$ glsuser -show Name -a chemistry -q
-----
amy
dave
```

Related Topics

[glsuser on page 302](#)

Modifying Users

To modify a user, use the command `gchuser`:

```
gchuser [-A | -I] [-n common_name] [-F phone_number] [-E email_address]
[-a default_account] [-d description] [-X, --extension_property=value]...
[--debug] [--site site_name] [--help] [--man] [--quiet] [--verbose] [--version] {[-u] user_name}
```

Example 7-6: Deactivating a user

```
$ gchuser -I bob
Successfully modified 1 user
```

i In order for user validity enforcement to occur, the **Values** property for the **UsageRecord User** attribute must be set to '@User'.

```
$ goldsh Attribute Modify Object==UsageRecord Name==User Values=@User
```

Example 7-7: Setting a user's default account

```
$ gchuser -a chemistry amy
Successfully modified 1 user
```

Example 7-8: Changing a user's email address

```
$ gchuser -E "rsmith@cs.univ.edu" bob
Successfully modified 1 user
```

Related Topics

[gchuser on page 220](#)

Deleting Users

To delete a user, use the command `grmuser`:

```
grmuser [--debug] [--site site_name] [--help] [--man] [--quiet]
[--verbose] [--version] {[--u] user_name}
```

Example 7-9: Deleting a user

```
$ grmuser bob
Successfully deleted 1 user
```

Related Topics

[grmuser on page 393](#)

User Auto-Generation

If user auto-generation is enabled (this is the default), users are automatically created when first added as a member to an account or role. It is also possible to automatically create users when first encountered in a usage function (charge, reserve or quote). In order for user auto-generation to occur, the **AutoGen** property for the **User** object must be set to **True** (this is the default). Additionally, for user auto-generation to occur when a user is added as a member of another object (such as Account) via an association table (e.g. **AccountUser**), the **Values** property for the **user** attribute of the **Association** (e.g. **Name**) must be set to **@User**, indicating that that value should be constrained

to be a valid instance of the User object. For user auto-generation to occur when initially encountered in a usage function, the **Values** property of the user attribute of the UsageRecord object must be similarly set to `@User`. The auto-creation of users can be completely disabled by setting the **AutoGen** property for the User object to `False`.

Example 7-10: Enable auto-generation of users when initially seen in a charge

```
$ goldsh Attribute Modify Object==UsageRecord Name==User Values=@User  
Successfully modified 1 attribute
```

Example 7-11: Disable all auto-generation of users

```
$ goldsh Object Modify Name==User AutoGen=False  
Successfully modified 1 object
```

See [Object Auto-Generation on page 133](#) for more information about the auto-generation of objects.

Default User

It is possible to set a global default user to which usage would be ascribed in quotes, liens, or charges where no user is specified. This can be accomplished by setting the **DefaultValue** property for the User object to the desired user.

It is also possible to set a custom user default or a specific object, which will result in usage being ascribed to the specified user when the object is attributed to the usage. This is done by creating a default usage override modifier. For example, to specify that `acmeuser` be the default user for usage associated with the acme organization, you might first create an attribute called **DefaultUser** for the **Organization** Object with the **Values** property of `@?=User`. Then you would populate the new **DefaultUser** property for the acme organization with the value of `acmeuser`. See [Customizing Objects on page 131](#) for more information on default and other usage override modifiers.

Example 7-12: Assign a global default user

```
$ goldsh Object Modify Name==User DefaultValue=anonymous  
Successfully modified 1 object
```

Chapter 8 Managing Accounts

An account represents a work entity requiring the use of resources or services for a common purpose. Users may be designated as members of an account and may be allowed to share its allocations. If the special 'ANY' user is added to an account, then any user may use funds allocated to the account. The user members may be designated as active or inactive, and as an administrator for the account. Default account properties include the description, the organization it is part of, and whether or not it is active. An account can be created, queried, modified and deleted. An account's user membership can also be adjusted. By default, a standard user may only query accounts they belong to.

Account queries allow the specification of filter options which narrow down the accounts that will be returned to those having the specified users in them.

Creating Accounts

To create a new account, use the command gmkaccount:

```
gmkaccount [-A | -I] [-o organization_name] [-d description] [-X, --extension property=value]... [-u [^|!][+|-]user_name,...]... [-createFund=True|False] [--debug] [--site site_name] [--help] [--man] [--quiet] [--verbose] [--version] {[-a] account_name}
```

When defining users, the optional caret or exclamation symbol indicates whether the user should be created as an admin (^) or not (!) for the account. The optional plus or minus sign can precede each member to indicate whether the member should be created in the active (+) or inactive (-) state. By default, a user will be created in the active state but not an admin. Multiple users may be passed to the -u option in a comma-delimited list. Alternatively, multiple -u options may be specified.

- i** If the Fund object's AutoGen property is set to true (see [Fund Auto-generation on page 66](#)), a fund will be automatically created for the account (unless overridden with the --createFund option). The auto-generated fund will be associated with the new account.

Example 8-1: Creating an account

```
$ gmkaccount -d "Chemistry Department" chemistry
Successfully created 1 account
```

Example 8-2: Creating an account that can be used by any user

```
$ gmkaccount -d "Common Account" -u ANY common
Successfully created 1 account.
```

Example 8-3: Creating an account and specifying user members at the same time

In this example, we make amy the account admin and associate the account with the sciences organization.

```
$ gmkaccount -d "Chemistry Department" -u ^amy,bob,dave chemistry -o sciences
Successfully created 1 account
```

Related Topics

[gmkaccount on page 306](#)

Querying Accounts

To display account information, use the command `glsaccount`:

```
glsaccount [-A | -I] [-o organization_name] [-X, --extension_property=value]... [-u user_name] [--full] [--show_attribute_name,...] [--long] [--wide] [--format csv|standard] [--debug] [--site site_name] [--help] [--man] [--quiet] [--version] [[-a] account_pattern]
```

Example 8-4: Listing all info about all accounts

```
$ glsaccount
Name      Active    Users          Organization   Description
-----
biology   True      amy,^bob       sciences      Biology Department
chemistry True      ^amy,^dave     sciences      Chemistry Department
film      True      amy,^dave     arts          Film Department
```

Example 8-5: Displaying the name and user members of an account in long format

```
$ glsaccount --show Name,Users -long chemistry
Name      Users
-----
chemistry ^amy
dave
```

Example 8-6: Listing all account names

```
$ glsaccount --show Name --quiet
biology
chemistry
film
```

Example 8-7: Listing all accounts that have dave as a member

```
$ glsaccount --show Name -u dave --quiet
chemistry
film
```

Related Topics

[glsaccount on page 230](#)

Modifying Accounts

To modify an account, use the command `gchaccount`:

```
gchaccount [-A | -I] [-o organization] [-d description] [-X, --extension property=value]... [--add-user(s) [^|!][+|-]user_name,...]... [--del-user(s) user_name,...]... [--mod-user(s) [^|!][+|-]user_name,...]... [--debug] [--site site_man] [--help] [--man] [--quiet] [--verbose] [--version] {[[-a] account_name]}
```

User members may be added, removed or modified in an account. When adding user members to an account, the optional caret or exclamation symbol indicates whether the user should be created as an admin (^) or not (!) for the account. The optional plus or minus signs can precede each member to indicate whether the member should be created in the active (+) or inactive (-) state. When modifying user members of an account, the caret symbol or exclamation symbol indicates the user should be changed to become an admin (^) or not (!) for the account. The plus or minus signs indicate whether the user should be changed to become active (+) or inactive (-). If an active or admin modifier is not specified, that aspect of the user member will remain unchanged. If the **user.firstaccountdefault** server parameter is set to true, the first account that a user is added to will additionally become the default account for that user.

Example 8-8: Deactivating an account

```
$ gchaccount -I chemistry
Successfully modified 1 account
```

i In order for account validity enforcement to occur, the `Values` property for the `UsageRecord Account` attribute must be set to '@Account'.

```
$ goldsh Attribute Modify Object==UsageRecord Name==Account Values=@Account
```

Example 8-9: Adding users as members of an account

```
$ gchaccount --add-users jsmith,barney chemistry
Successfully added 2 users
```

Example 8-10: Deactivating a user in an account

```
$ gchaccount --mod-user -dave chemistry
Successfully modified 1 user
```

Deleting Accounts

To delete an account, use the command `grmaccount`:

```
grmaccount [--debug] [--site site_name] [--help] [--man] [--quiet] [--verbose] [--version] {[-a] account_name}
```

Example 8-11: Deleting an account

```
$ grmaccount chemistry
Successfully deleted 1 account
```

Related Topics

[grmaccount on page 368](#)

Account Auto-Generation

It is possible to have accounts be created automatically when first encountered in a usage function (charge, reserve or quote). It is also possible for accounts to be automatically created when initially added as a member of another object. In order for account auto-generation to occur, the **AutoGen** property for the Account object must be set to `True`. This is the default. For account auto-generation to occur when initially encountered in a usage function, the **Values** property of the account attribute of the UsageRecord object must be set to `@Account`. Additionally, for account auto-generation to occur when an account is added as a member of another object (such as the Organization object) via an association table (e.g. OrganizationAccount), the **Values** property for the account attribute of the Association (e.g. Name) must be set to `@Account`, indicating that that value should be constrained to be a valid instance of the Account object. The auto-creation of accounts can be completely disabled by setting the **AutoGen** property for the Account object to `False`.

Example 8-12: Enable auto-generation of accounts when initially seen in a charge

```
$ goldsh Attribute Modify Object==UsageRecord Name==Account Values=@Account
Successfully modified 1 attribute
```

Example 8-13: Disable all auto-generation of accounts

```
$ goldsh Object Modify Name==Account AutoGen=False  
Successfully modified 1 object
```

See [Object Auto-Generation on page 133](#) for more information about the auto-generation of objects.

Default Account

It is possible to set a global default account to which usage would be ascribed in quotes, liens, or charges where no account is specified. This can be accomplished by setting the **DefaultValue** property for the **Account** object to the desired account name.

A per-user default account can be established by setting the **DefaultAccount** property for the user. If the **user.firstaccountdefault** server parameter is set to true (the default), the first account that a user is added to will automatically become the default account for that user. Otherwise, you can use the `gchuser` command along with the `-a` option to set or change the default account for the user.

It is also possible to set a custom account default for a specific object, which will result in usage being ascribed to the specified account when the object is attributed to the usage. This is done by creating a default usage override modifier. For example, to specify a default account of testing for the beta organization, you might first create an attribute called **DefaultAccount** for the **Organization** Object with the **Values** property of `@?=Account`. Then you would populate the new **DefaultAccount** property for the beta organization with the value of testing. See [Customizing Objects on page 131](#) for more information on default and other usage override modifiers.

Example 8-14: Assign a global default account

```
$ goldsh Object Modify Name==Account DefaultValue=common  
Successfully modified 1 object
```


Chapter 9 Managing Organization

An organization is a virtual organization in which accounts are grouped. An account may only belong to a single organization while an organization may have multiple accounts. For example, an account may represent a project or cost-center while an organization may represent an institutional department or business division. The purpose of defining organizations is to support the ability to produce reporting for higher-order organizational entities beyond the individual account. Default organization properties include a name and a description. An organization can be created, queried, modified, and deleted.

Creating Organizations

To create a new organization, use the command `gmkorg`:

```
gmkorg [-d description] [-X, --extension property=value]... [--  
debug] [--site site man] [--help] [--man] [--quiet] [--  
verbose] [--version] {[[-o] organization_name]
```

Example 9-1: Creating an organization

```
$ gmkorg -d "Sciences Department" sciences  
Successfully created 1 organization
```

Related Topics

[gmkorg on page 323](#)

Querying Organizations

To display organization information, use the command `glsorg`:

```
glsorg [-X, --extension property=value]... [--full] [--show  
attribute_name,...] [--format csv|standard] [--debug] [--site  
site man] [--help] [--man] [--quiet] [--version] {[[-o]  
organization_pattern]
```

Example 9-2: Listing all organization names

```
$ glsorg --show Name -q  
+ arts  
+ sciences
```

Related Topics

[glsorg on page 272](#)

Modifying Organizations

To modify an organization, use the command `gchorg`:

```
gchorg [-d description] [-x, --extension property=value]... [--  
debug] [--site site man] [--help] [--man] [--quiet] [--  
verbose] [--version] {[-o] organization name}
```

Example 9-3: Changing an organization's description

```
$ gchorg -d "Art College" art  
Successfully modified 1 organization
```

Related Topics

[gchorg on page 199](#)

Deleting Organizations

To delete an organization, use the command `grmorg`:

```
grmorg [--debug] [--site site name] [--help] [--man] [--quiet]  
[--verbose] [--version] {[-o] organization name}
```

Example 9-4: Deleting an organization

```
$ grmorg arts  
Successfully deleted 1 organization
```

Related Topics

[grmorg on page 381](#)

Organization Auto-Generation

It is possible to have organizations be created automatically when initially added as a member of another object. In order for organization auto-generation to occur, the **AutoGen** property for the Organization object must be set to `True`. This is the default. For organization auto-generation to occur when initially encountered in a usage function, the `Values` property of the organization attribute of the UsageRecord object must be set to `@Organization`. Additionally, for organization auto-generation to occur when an organization is added as a member of another object (such as a hypothetical Site object) via an association table (e.g. SiteOrganization), the **Values** property for the organization attribute of the Association (e.g. Name) must be set to `@Organization`, indicating that that value should be

constrained to be a valid instance of the Organization object. The auto-creation of organizations can be completely disabled by setting the **AutoGen** property for the Organization object to False.

Example 9-5: Enable auto-generation of organizations when initially seen in a charge

```
$ goldsh Attribute Modify Object==UsageRecord Name==Organization Values=@Organization  
Successfully modified 1 attribute
```

Example 9-6: Disable all auto-generation of organizations

```
$ goldsh Object Modify Name==Organization AutoGen=False  
Successfully modified 1 object
```

See [Object Auto-Generation on page 133](#) for more information about the auto-generation of objects.

Default Organization

It is possible to set a global default organization to which usage would be ascribed in quotes, liens, or charges where no organization is specified. This can be accomplished by setting the **DefaultValue** property for the **Organization** object to the desired organization name.

It is also possible to set an organization default for a specific object, which will result in usage being ascribed to the specified organization when the object is attributed to the usage. This is done by creating a default usage override modifier. For example, to specify that retail be the default organization for usage associated with the user amy, you might first create an attribute called **DefaultOrganization** for the User Object with the **Values** property of @?=Organization. Then you would populate the new **DefaultOrganization** property for the amy user with the value of retail. See the chapter on [Customizing Objects on page 131](#) for more information on default and other usage override modifiers.

Example 9-7: Assign a global default organization

```
$ goldsh Object Modify Name==Organization DefaultValue=whitecloud  
Successfully modified 1 object
```


Chapter 10 Managing funds

A fund is a container for a time-bounded reference currency called credits for which the usage is restricted by constraints that define how the credits must be used. Much like with a bank, a fund is a repository for these resource or service credits which are added through deposits and debited through withdrawals and charges. Each fund has a set of constraints designating which entities (such as Users, Accounts, Machines, Classes, Organizations, etc.) may access the encapsulated credits or for which aspects of usage the funds are intended (QualityOfService, GeographicalArea, etc.). Fund constraints may also be negated with an exclamation point leading the constraint value.

Funds may have a name which is not necessarily unique for the fund. Funds may also have a priority which will influence the order of fund selection when charging. A default deposit amount can be set for a fund which is used when the amount is not specified for a deposit. Derived properties such as Allocated, Balance, Effective, Available, Capacity, PercentRemaining, PercentUsed and Used can be displayed via the `glsfund` or `gbalance` commands (see the commands reference for [glsfund on page 253](#) or [gbalance on page 162](#) for more details). Operations include creating, querying, modifying, deleting and resetting funds as well as making deposits, withdrawals, transfers and balance queries. By default, a standard user may only query and view the balance for funds which pertain to them.

Credits are added to a fund via a deposit. If no amount is specified for the deposit, the fund's default deposit value is used for the deposit amount. When credits are deposited into a fund, they are associated with a time period within which they are valid. These time-bounded pools of credits are known as allocations. The initial deposit into a fund will create a new allocation having the specified or default time boundaries. A fund may be reset, causing the currently active allocation to end and creating a new allocation. Alternatively, allocations with predesignated start and end times may be created ahead of time. By using one of these methods to implement periodic allocations, it is possible to establish an allocation cycle. See the [Managing Allocations on page 69](#) chapter for more information on periodic allocations as well as credit limits and infinite allocations.

Funds may be nested. Hierarchically nested funds may be useful for the delegation of management roles and responsibilities. Deposit shares may be established that assist to automate a trickle-down effect for credits deposited at higher level funds. Additionally, an optional overflow feature allows charges against lower level funds to trickle up the hierarchy.

Some fund operations (Query, Deposit, Withdraw, and Refund) allow the specification of filter options which narrow down the funds that will be acted on for that operation. There are two fund filter types that can be employed: Exclusive and NonExclusive. If an exclusive filter type is used, the query will return only the funds for which the specified filters meet all constraints for usage. Another way to think of an exclusive filter is to ask if usage were to be

posted given only the specified filter options as ACLs, which funds would be eligible for charging? For example, Fund Query FilterType:=Exclusive Filter:=User=scottmo would not return a fund with the sole constraint Machine=blue because Machine=blue was not included in the filters. Not only must the filters be a non-conflicting superset of the fund constraints, but all constraint dependencies must also be satisfied (for example, an appropriate user may need to be specified with the account). If a non-exclusive filter type is used, the query will return all funds for which the filters do not specifically exclude the constraints. The query assumes that if constraints are not specified within the filters, they can be assumed as a wildcard and will return all funds that are not specifically excluded by the filter. For example, Fund Query FilterType:=NonExclusive Filter:=User=scottmo would return a fund whose only constraint was Machine=blue (because it does not conflict) but would not return a fund with the constraint User=bob (because it does conflict).

Creating Funds

gmkfund is used to create a new fund. You can specify a fund name, a description, and any number of fund constraints. If a name is not specified and constraints are specified, a name will be automatically generated based on the constraints. A new unique id is automatically generated for the fund.

```
gmkfund [-n fund_name] [--priority fund_priority] [--default-deposit deposit_amount] [-d description] [-X, --extension-property=value]... [-u user_name,...]... [-g group_name,...]... [-a account_name,...]... [-o organization_name,...]... [-c class_name,...]... [-m machine_name,...]... [--constraint constraint_name=[!]constraint_value,...]... [--parent parent_fund_id] [--debug] [--site site_name] [--help] [--man] [--quiet] [--verbose] [--version]
```

i It is possible to have funds be created automatically when accounts are created by setting the Fund object's AutoGen property to true (see [Fund Auto-generation on page 66](#)). The auto-generated fund will be associated with the new account.

Example 10-1: Creating a fund valid for the chemistry account

```
$ gmkfund -a chemistry -n "Chemistry"
Successfully created 1 fund with id 7 and 1 constraint
```

Example 10-2: Creating a wide-open fund that can be used by anyone for anything

```
$ gmkfund -n "Windfall"
Successfully created 1 fund with id 8
```

Example 10-3: Creating a fund valid toward all biology account members except for dave and just the machine colony

```
$ gmkgfund --constraint Account=biology,User!=dave,Machine=colony -n "Biology on Colony not for Dave"
Successfully created 1 fund with id 9 and 3 constraints
```

Related Topics

[gmkgfund on page 314](#)

Querying Funds

To display fund information, use the command `glsfund`:

```
glsfund [-A | -I] [-n fund_name] [-X, --extension property=value]... [-u user_name] [-g group_name] [-a account_name] [-o organization_name] [-c class_name] [-m machine_name] [--filter filter_name=filter_value]... [--filter-type Exclusive|NonExclusive] [--full] [--show attribute_name,...] [--long] [--wide] [--format csv|standard] [-h, --hours] [--debug] [--site site man] [--help] [--man] [--quiet] [--version] [-f] fund_id]
```

Example 10-4: Listing all info about all funds with multi-valued fields displayed in a multi-line format

```
$ glsfund --long
```

			Allocated	Balance	DefaultDeposit	Description
1	biology	Account=biology	25000000	25000000	25000000	
2	chemistry for amy	User=amy	35000000	34802392	35000000	
		Account=chemistry				
3	chemistry not amy	User!=amy	50000000	50000000	50000000	
		Account=chemistry				
4	film on colony	Account=film		0	0	
		Machine=colony				

Example 10-5: Wide listing all info about all funds useable by amy

```
$ glsfund -u amy
```

			Allocated	Balance	DefaultDeposit
1	biology	Account=biology	25000000	25000000	25000000
2	chemistry for amy	Account=chemistry,User=amy	35000000	34802392	35000000
4	film on colony	Machine=colony,Account=film	0	0	0

Related Topics

[glsfund on page 253](#)

Modifying Funds

To modify a fund, use the command `gchfund`:

```
gchfund { [-n fund_name] [--priority fund_priority] [--default-deposit deposit_amount] [-d description] [-X, --extension property=value]... [-add-constraint constraint_name=[!] constraint_value,...] [--del-constraint(s) constraint_name [=constraint_value],...]... [--parent parent_fund_id] } | { --reset [--all]} [-u user_name] [-g group_name] [-a account_name] [-o organization_name] [-c class_name] [-m machine_name] [--filter filter_name=filter_value]... [--filter-type Exclusive|NonExclusive] [--debug] [--site site_name] [--help] [--man] [--quiet] [--verbose] [--version] ([-f] fund_id)
```

Example 10-6: Adding a constraint to a fund so that it can only be used by the acme organization

```
$ gchfund --add-constraint Organization=acme 7
Successfully created 1 constraint
```

Example 10-7: Setting the default deposit amount for a fund

```
$ gchfund --default-deposit 5000000 -f 1
Successfully modified 1 fund
```

Example 10-8: Resetting a fund

```
$ gchfund --reset 1
Successfully deposited 5000000 credits into fund 1
Successfully stopped 1 allocation
Successfully created 1 allocation
```

Related Topics

[gchfund on page 189](#)

Making Deposits

`gdeposit` is used to deposit time-bounded resource credits into a fund resulting in the creation or increase of an allocation. (See [Managing Allocations on page 69](#) for information about managing allocations). The start time will default to `-infinity` and the end time will default to `infinity` if not specified. Filter options can be specified to help select a unique fund for the deposit. If multiple funds are matched by the filters, the matching funds will be listed and you will be prompted to respecify the deposit with one of the fund ids. If an allocation for the deposit fund is found having the start and end times for the deposit, the amount of the allocation will be increased by the deposit amount. Otherwise, a new allocation will be created for the fund with the amount of the deposit. If no

funds match your criteria, if fund auto-generation is enabled, a fund will be created and the deposit made into it. Otherwise, the deposit will fail (the fund will need to be first created using `gmkfund`).

Deposits may be used to extend the debit ceiling by specifying an amount for the deposit (with the `-z` option) or extend the credit floor by specifying a credit limit for the deposit (with the `-L` option) or a combination of both options may be used. Additionally, `Infinity` may be used for either of these option values when Moab Accounting Manager is coupled with a database that supports IEEE Standard 754 for Floating-Point Arithmetic (e.g. PostgreSQL).

```
gdeposit [-L credit_limit] [-s start_time] [-e end_time] [-d
description] [-f fund_id] [-i allocation_id] [-u user_name] [-
g group_name] [-a account_name] [-o organization_name] [-c
class_name] [-m machine_name] [--filter filter_name=filter_
value]... [--filterType Exclusive|NonExclusive] [--create-fund
True|False] [--reset] [-h, --hours] [--debug] [--site site_
name] [--help] [--man] [--quiet] [--verbose] [--version] [[-z]
amount]
```

Example 10-9: Making a deposit into fund 1

```
$ gdeposit -z 360000000 -f 1
Successfully created 1 allocation
```

Example 10-10: Making a deposit "into" an account

If an account has a single fund then a deposit can be made against the account.

```
$ gdeposit -z 360000000 -a chemistry
Successfully deposited 360000000 credits into fund 2
```

Example 10-11: Creating a credit allocation

```
$ gdeposit -L 100000000000 -f 3
Successfully created 1 allocation
```

Example 10-12: Making a reset deposit

Stop the active allocation within a fund and create a new allocation.

```
$ gdeposit -f 4 -z 36000000 --reset
Successfully deposited 36000000 credits into fund 4
Successfully stopped 1 allocation
Successfully created 1 allocation
```

Example 10-13: Creating an infinite allocation

```
$ gdeposit -z Infinity -f 5
Successfully deposited inf credits into fund 5
Successfully created 1 allocation
```

i The use of infinite allocations requires the use of a database that supports the IEEE Standard 754 for Floating-Point Arithmetic (e.g. PostgreSQL).

Example 10-14: Creating a future quarterly allocation

```
$ gdeposit -s 2015-10-01 -e 2016-01-01 -z 25000000 -a biology
Successfully created 1 allocation
```

Related Topics

[gdeposit on page 223](#)

Querying the Balance

To display balance information, use the command `gbalance`:

```
gbalance [-u user_name] [-g group_name] [-a account_name] [-o organization_name] [-c class_name] [-m machine_name] [--filter filter_name=filter_value]... [--filterType Exclusive|NonExclusive] [--ignore-ancestors] [--full] [--show attribute_name,...] [--long] [--wide] [--format csv|standard] [-h, --hours] [--debug] [--site site_name] [--help] [--man] [-quiet] [--version]
```

Example 10-15: Querying amy's balance

```
$ gbalance -u amy
ID  Name      Balance Reserved Effective CreditLimit Available
--  --
1   biology    2785.87  103.22   2682.65      0.00   2682.65
2   chemistry  1785.87  0.00     1785.87      0.00   1785.87
```

Example 10-16: List the available balances that amy can charge against along with the constraints on those balances

```
$ gbalance -u amy --show Available,Constraints
Available Constraints
-----
25000000 Account=biology
34802392 Account=chemistry,User=amy
0 Machine=colony,Account=film
```

Related Topics

[gbalance on page 162](#)

Personal Balance

The `mybalance` has been provided as a wrapper script to show users their personal balance. It provides a list of balances for the funds that they can charge to:

```
mybalance [-h, --hours] [--help] [--man]
```

Example 10-17: List my fund balances

```
$ mybalance
Name          Available
-----
biology      25000000
chemistry for amy 34802392
```

Example 10-18: List my balance in (Processor) hours

```
$ mybalance -h
Name          Available
-----
biology      6944.44
chemistry for amy 9667.33
```

Related Topics

[mybalance on page 415](#)

Making Withdrawals

A withdrawal can be used to debit a fund without being associated with the usage charge from some item. To issue a withdrawal, use the command `gwithdraw`:

```
gwithdraw [-f fund_id] [-i allocation_id] [-u user-name] [-g group_name] [-a account_name] [-o organization_name] [-c class_name] [-m machine_name] [--filter filter_name=filter_value]... [--filter-type Exclusive|NonExclusive] [-d description] [-h, --hours] [--debug] [--site site_name] [--help] [--man] [--quiet] [--verbose] [--version] {[-z] amount}
```

Example 10-19: Making a withdrawal

```
$ gwithdraw -z 12800 -f 1 -d "Grid Tax"
Successfully withdrew 12800 credits from fund 1
```

Example 10-20: Making a withdrawal "from" an account

If an account has a single fund then a withdrawal can be made against the account.

```
$ gwithdraw -z 12800 -a biology
Successfully withdrew 12800 credits from fund 1
```

If more than one fund exists for the account or filter, you will be asked to be more specific:

```
$ gwithdraw -z 12800 -a chemistry
Multiple funds were matched for the withdrawal.
Please respecify using one of the following fund ids:
2 [chemistry for amy]
3 [chemistry not amy]
```

Related Topics

[gwithdraw on page 405](#)

Making Transfers

To issue a transfer between funds, use the command `gtransfer`. If the allocation id is specified, then only credits associated with the specified allocation will be transferred, otherwise, only active credits will be transferred. Fund transfers preserve the allocation time periods associated with the resource or service credits from the source to the destination funds. Source and destination filters may be used if they result in a single source fund and single destination fund.

```
gtransfer {--from-fund source_fund_id &| --from-allocation
source_allocation_id &| --from-filter filter_name=filter
value...} {--to-fund destination_fund &| --to-allocation
destination_allocation_id &| --to-filter filter_name=filter
value...} [--filter-type Exclusive|NonExclusive] [-d
description] [-h, --hours] [--debug] [--site site_name] [--
help] [--man] [--quiet] [--verbose] [--version] {[-z] amount}
```

Example 10-21: Transferring credits between two funds

```
$ gtransfer --from-fund 1 --to-fund 2 10000
Successfully transferred 10000 credits from fund 1 to fund 2
```

Example 10-22: Transferring credits between two single-fund accounts

```
$ gtransfer --from-filter Account=biology --to-filter Account=chemistry 10000
Successfully transferred 10000 credits from fund 1 to fund 2
```

Related Topics

[gtransfer on page 400](#)

Obtaining a Fund Statement

To generate a fund statement, use the command `gstatement`. For a specified time frame it displays the beginning and ending balances as well as the total credits and debits to the fund over that period. This is followed by an itemized report of the debits and credits. Summaries of the debits and credits will be displayed instead of the itemized report if the `--summarize` option is specified. If filter options are specified instead of a fund, then the statement will consist of information merged from all funds valid toward the specified entities.

```
gstatement [ -f fund_id] [-n fund_name] [-u user_name] [-g group_name] [-a account_name] [-o organization_name] [-c class_name] [-m machine_name] [--filter filter_name=filter_value]... [--filter-type Exclusive|NonExclusive] [-s start_time] [-e end_time] [--summarize] [-h, --hours] [--debug] [--site site_man] [--help] [--man] [--version]
```

Example 10-23: Generating a fund statement for all chemistry funds for the fourth quarter of 2014

```
$ gstatement -a chemistry -s 2014-10-01 -e 2015-01-01 --summarize
#####
#
# Includes fund 3 (chemistry not amy)
# Includes fund 2 (chemistry for amy)
# Generated on Mon Feb 7 18:44:23 2015.
# Reporting fund activity from 2014-10-01 to 2015-01-01.
#
#####
Beginning Balance: 0
-----
Total Credits: 90122212
Total Debits: -5308668
-----
Ending Balance: 84813544
#####
Credit Summary #####
Object Action Amount
-----
Fund Deposit 90100000
UsageRecord Refund 22212
#####
Debit Summary #####
Object Action Account User Machine Amount Count
-----
UsageRecord Charge chemistry amy colony -19744 239
#####
End of Report #####
```

i The fields which are used as default discriminators in the detail section of the `gstatement` command (which are by default Account, User, and Machine) can be customized by setting the **statement.show** configuration parameter in `gold.conf`.

Related Topics

[gstatement on page 395](#)

Deleting Funds

To delete a fund, use the command `grmfund`:

```
grmfund [--debug] [--site site_name] [--help] [--man] [--quiet]
[--verbose] [--version] {[-f] fund_id}
```

Example 10-24: Deleting a fund

```
$ grmfund 2
Successfully deleted 1 fund
```

Related Topics

[grmfund on page 375](#)

Fund Auto-generation

It is possible to enable the auto-generation of funds by setting the **AutoGen** property of the Fund object to `True`. When creating a new account, if fund auto-generation is enabled, a fund will automatically be created for the account (unless overridden with the `--create-fund` option). The fund will be usable only by usage attributed to the new account. Additionally, if fund auto-generation is set, a deposit that does not match an existing fund will automatically generate a fund using the filters as constraint options. Objects associated with the constraint that have **AutoGen** set to `True` will be auto-generated as well (unless overridden with the `--create-fund` option).

Example 10-25: Enable auto-generation of funds

```
$ goldsh Object Modify Name==Fund AutoGen=True
Successfully modified 1 object
```

Hierarchical Funds

A hierarchy can be established between funds. When creating a fund or by modifying it later, one can specify a parent fund id via the `--parent` option to establish the object fund as a child of the specified parent fund. A fund may have multiple children funds but only a single parent fund.

Example 10-26: Establishing a child relationship with another fund

```
$ gchfund --parent 3 -f 6
Successfully added 1 parent
```

Deposit shares may be established between the parent fund and its children that assist to automate a trickle-down effect for funds deposited at higher level funds (**DepositShare** is an attribute of the FundFund association object). Deposit shares are integers and are treated as a percentage of each deposit

and the sum of the deposit shares for a fund's children may not exceed 100. If the deposit shares for the children of a fund totals less than 100, the difference is taken to be the share of the deposit that will be allocated to the parent. When a deposit is made into a parent fund, for each child fund that has a non-zero deposit share a recursive deposit amounting to the designated percentage of the parent deposit is issued to that child. After the share amounts have been deposited to each of the child funds, the remaining percentage of the deposit is allocated to the parent fund. This effect is recursive with each child. If a start time and/or end time are specified in the original deposit, these time frames will be recursively applied to all descendant deposits. You have to use the goldsh interactive control program to manage deposit shares. For the FundFund association object, the Fund is the parent and the Id is the child.

Example 10-27: Establishing a 10% deposit share between a parent and a child fund

```
$ goldsh FundFund Modify Fund==3 Id==6 DepositShare=10
Fund      Id      DepositShare  Overflow
-----  -----
3          6        10            False
Successfully modified 1 fundFund
```

An overflow policy may be established between the parent fund and its children to enable a trickle-up effect for charges, liens, and quotes from the lower level funds (Overflow is an attribute of the FundFund association object). The Overflow attribute is a boolean value (True or False). If the overflow value between a child and its parent is set to True, any charges, liens, or quotes issued against the child fund that cannot be satisfied by the balance in the child fund, will recursively issue the unsatisfied portion of the charge, lien, or quote against the parent fund. If the charge, lien, or quote cannot be satisfied by the ancestors, no charges, liens, or quotes will result against any of funds. The balance in the descendant funds will be depleted before ancestor funds. This effect is recursive with each parent. If a parent fund is linked with overflow to a child fund and a charge, lien, or quote overflows to the parent fund, the constraints of the parent fund will not be checked against the properties of the item. One must use the goldsh control program to manage the overflow policy. For the FundFund association object, the Fund is the parent and the Id is the child.

Example 10-28: Enabling overflow between a parent and a child fund

```
$ goldsh FundFund Modify Fund==3 Id==6 Overflow=True
Fund      Id      DepositShare  Overflow
-----  -----
3          6        10            True
Successfully modified 1 fundFund
```

Fund Priority

By default, when an item can charge to multiple funds, funds with more constraints are chosen over funds with fewer constraints. For example, if the user `amy` is charging against the chemistry account for usage of an item and there are two viable funds, one with a single constraint (e.g. `Account=chemistry`) and another with two constraints (e.g. `Account=chemistry and User=amy`), credits will be taken from the more specific fund (with 2 constraints) before they are taken from the more general fund (with 1 constraint). To override this behavior, it is possible to give a priority to a fund. The priority factor of a fund has higher precedence than the specificity (constraint count) of the fund. Thus, all else being equal, if a fund with a lower number of constraints is given a higher priority than a fund with a higher number of constraints, the higher priority fund will be depleted first. Other factors, such as the end time of the allocation or whether there is an existing lien for the item against a fund, have a higher precedence than the specificity of the fund. If you want the allocations in a particular fund to be chosen before allocations that expire sooner or that have a lien, you may need to specify fund priorities that are in the millions (see [Allocation Precedence on page 73](#) for a discussion of the manner of sorting allocations for charging).

Example 10-29: Setting a fund priority

```
$ gchfund -f 3 --priority 1  
Successfully modified 1 fund
```

Chapter 11 Managing Allocations

An allocation is a time-bound pool of credits associated with a fund. A fund may have multiple allocations, each for use during a different time period. Normally, only a single allocation will be active within a fund at any given time.

Allocations are normally created via a fund deposit. An allocation has an amount, an initial deposit, and an allocated value. The **Amount** attribute tracks the current amount of credits in the allocation. The **InitialDeposit** attribute stores the amount originally deposited into an allocation when it is initially created. The **Allocated** attribute stores the current adjusted allocated amount. It begins set to the initial deposit amount and is incremented with each crediting deposit or incoming transfer and decremented with each withdrawal or outgoing transfer. When a deposit is made to a fund, if a matching allocation already exists with the appropriate time period, the existing allocation is modified. Otherwise, a new allocation is created. A resetting deposit will end the currently active allocation and create a new allocation.

An allocation has a start time and an end time that defines the time period during which the allocation may be used. If a start time or end time is specified when making a deposit, an existing allocation having the specified boundary times will be credited. If no start time or end time is specified, the active allocation will be credited. If no matching or active allocations can be found, a new allocation will be created with the specified or default start and end time (the start time defaults to the present and the end time defaults to infinity). An active flag is automatically updated to `True` if the allocation is within its valid time frame or `False` if it is not. An allocation that becomes active because the current time is greater than its start time undergoes an activation which normally registers as a credit to the fund. An allocation that becomes inactive because the current time is greater than its end time undergoes a deactivation which normally registers as a debit to the fund.

By using multiple allocations that expire in regular intervals it is possible to implement a use-it-or-lose-it policy and establish an allocation cycle. There are two primary methods to implement periodic allocations. In the first method, called Resetting Allocations, funds are reset (ending the current allocation and creating a new one) at the beginning of each allocation period. By setting and maintaining an appropriate default deposit amount for each fund, the process of resetting funds can be simplified. The periodic reset can be performed either by making a resetting deposit for each fund (e.g. `gdeposit -f 1 --reset`) which allows you to override default deposit amounts, by calling the `reset` action for each fund (e.g. `gchfund -f 1 --reset`) which allows you to select which funds to reset, or by invoking a `reset` across all funds (e.g. `gchfund --reset --all`). The effect of any of these commands is to end the currently active allocation in the fund and then make a fresh deposit. The fund's default deposit amount is used any time the amount is not specified in a deposit (as in the case of a fund reset command). If the default deposit amount is positive,

the currently active allocation is ended and a new allocation is created with the default amount. If the default deposit amount is set to a value of zero, the active allocation is ended and no new allocation is created. If the default deposit amount is not set, the fund's allocations are not affected. The reset can be performed via a scheduled event or via a cron script. If default deposit amounts are kept up-to-date (including being zeroed out for funds that are slated to end and being unset for funds that you do not want affected by the reset), automation of this method can be as simple as creating a single periodic event with a FireCommand of "Fund Reset (see [Creating Events on page 106](#)). In the second method, called Expiring Allocations, funds with predesignated start and end times are created head of time. When the beginning of an allocation period is reached, the currently active allocation automatically expires and the next one automatically becomes active. A future allocation is created by making a deposit while specifying a start time and an end time in the future (e.g. `gdeposit -f 1 -s 2015-10-01 -e 2016-01-01`). This method can also take advantage of default deposit amounts. The overall effect of either of these methods is very similar.

By default, Moab Accounting Manager attempts to enforce Discrete allocations, or ensure that allocations within a fund are non-overlapping (in time) and non-reusable (each allocation period should use a distinct allocation). This behavior is designated by the **allocation.enforediscrete** server configuration parameter. If set to `true`, this policy prevents new allocations within a fund from overlapping existing ones. Enabling this policy helps to improve clarity when reporting on allocation usage during a particular period. If set to `false`, overlapping allocations within a fund can be created. This might be useful if you want to allow the remaining balance from a prior allocation period to carry over into the new allocation period. With overlapping allocations, it is harder to describe what percentage of a group's allocation has been used. This policy is applied when making deposits that create new allocations, when making transfers that create new allocations, or when modifying the start and end times of an existing allocation. It is possible to override the configured policy for an individual command by specifying the **EnforcedDiscrete** option (e.g. `gdeposit --option name=EnforceDiscrete value=False`).

An allocation may have a credit limit representing the amount by which it can go negative. Thus, by having a positive balance in the **Amount** field, the fund is like a debit account, implementing a pay-first use-later model. By establishing a credit limit instead of depositing an initial balance, the fund will be like a credit account, implementing a use-first pay-later model. These strategies can be combined by depositing some amount of funds coupled with a credit limit, implementing a form of overdraft protection where the funds will be used down to the negative of the credit limit.

It is possible for the allocation **Amount** or **CredLimit** to be set to Infinity (via a deposit). If the amount is infinite, debits will not decrease the balance. An infinite deposit will result in an infinite Allocated amount. If the credit limit is infinite, there will be no negative limit for debits. It is not possible to have infinite charges, liens, quotes, withdrawals, refunds, or transfers. However, it is possible to have infinite allocation activations, deactivations, and deletions.

This capability is only available when using a database that supports IEEE Standard 754 for Floating-Point Arithmetic (e.g. PostgreSQL).

Operations include querying, modifying, and deleting allocations. Allocations can be created by a fund deposit, creating a fund with allocation auto-generation enabled, refunding a usage record, or a transfer between funds. Allocations may also be indirectly modified via charges, withdrawals, transfers, or refunds. By default, a standard user may only query allocations which pertain to them.

Allocation queries allow the specification of filter options which filter the allocations to those with funds meeting the specified fund constraints. There are two allocation filter types that can be employed: `Exclusive` and `NonExclusive`. If an exclusive filter type is used, the query will return only allocations relating to funds for which the specified filters meet all constraints. For example, `Allocation Query FilterType:=Exclusive Filter:=User=scottmo` would not return an allocation for a fund with the sole constraint `Machine=blue`. If a non-exclusive filter type is used, the query will return all allocations relating to funds for which the filters do not specifically exclude the constraints. The query assumes that if constraints are not specified within the filters, they can be assumed as a wildcard and will return all allocations involving funds that are not specifically excluded by the filter. For example, `Allocation Query FilterType:=NonExclusive Filter:=User=scottmo` would return an allocation with a fund whose only constraint was `Machine=blue` but would not return an allocation with a fund with the constraint `User=bob`.

Creating Allocations

Allocations are normally created by making fund deposits via the `gdeposit` command (See [Making Deposits on page 60](#)).

Querying Allocations

To display allocation information, use the command `glsalloc`:

```
glsalloc [-A | -I | {[ -s start_time] [ -e end_time]}] [-f fund_id]
[-X, --extension property=value]... [-u user_name] [-g group_name]
[-a account_name] [-o organization_name] [-c class_name]
[-m machine_name] [--filter filter_name=filter_value]...
[--filter-type Exclusive|NonExclusive] [--include-ancestors]
[--full] [--show attribute_name,...] [--format csv|standard] [-h,
--hours] [--debug] [--site site_name] [--help] [--man] [--quiet]
[--version] [ -i] allocation_id]
```

Example 11-1: Listing allocations for fund 1

```
$ glsalloc -f 1

  Id Fund Active StartTime EndTime    InitialDeposit Allocated CreditLimit Remaining
PercentUsed
-- -----
1 1   True  2015-01-01 2015-04-01      25000000  25000000      0 24974400
  0.10
2 1   False 2015-04-01 2015-07-01      25000000  25000000      0 25000000
  0.00
3 1   False 2015-07-01 2015-10-01      25000000  25000000      0 25000000
  0.00
4 1   False 2015-10-01 2016-01-01      25000000  25000000      0 25000000
  0.00
```

Related Topics[glsalloc on page 235](#)

Modifying Allocations

To modify an allocation, use the command `gchalloc`:

```
gchalloc [-s start_time] [-e end_time] [-L credit_limit] [-d
description] [-X, --extension property=value]... [-h, --hours]
[--debug] [--site site_name] [--help] [--man] [--quiet] [--
verbose] [--version] {[-i] allocation_id}
```

Example 11-2: Changing the end time for an allocation

```
$ gchalloc -e "2016-01-01" 4
Successfully modified 1 allocation
```

Example 11-3: Changing the credit limit for an allocation

```
$ gchalloc -L 500000000000 -i 2
Successfully modified 1 allocation
```

Related Topics[gchalloc on page 172](#)

Deleting Allocations

To delete an allocation, use the command `grmalloc`:

```
grmalloc (-I|{[-i] allocation_id}) [--debug] [--site site_name]
[--help] [--man] [--quiet] [--verbose] [--version]
```

Example 11-4: Deleting an allocation

```
$ grmalloc 4
Successfully deleted 1 allocation
```

Example 11-5: Purging inactive allocations

```
$ grmalloc -I
Successfully deleted 2 allocations
```

Related Topics

[grmalloc on page 370](#)

Allocation Auto-generation

It is possible to enable the auto-generation of allocations by setting the **AutoGen** property of the Allocation object to `True`. When creating a new fund, if allocation auto-generation is enabled, an allocation will automatically be created for the fund via a deposit. The deposit will use the default amount and default credit limit (defined in the **DefaultValue** property of the Allocation Amount and Allocation CreditLimit attributes). The default action for allocation auto-generation is to create an allocation with an infinite credit limit.

Example 11-6: Enable auto-generation of allocations

```
$ goldsh Object Modify Name==Allocation AutoGen=True
Successfully modify 1 object
```

Allocation Precedence

When issuing a charge (or a lien or quote) for the usage of a resource or service, the feasible allocations are sorted according to a weight given to them for that transaction. The weight for each allocation is calculated as follows: If the instance has a current lien against one or more allocations, these allocations are given a value of `10000000 + int((2147483647 - <end_epoch_time>) / 86400)`. Thus, these reserved allocations will generally have the highest precedence (subject to large fund priorities), with those that expire sooner being used first. For the remaining non-nested funds, allocations will be given a value of `100 * int((2147483647 - <end_epoch_time>) / 86400) + 10 * <fund_priority> + <constraint_count>`. Thus, sooner expiring allocations will be used before later expiring allocations, fund priority will be the next highest factor (assuming small priority values of 1-10), followed by the number of constraints on the fund (more specific funds will be used before more general funds). Of course, since priority is configurable, a sufficiently large priority (in the millions) can be used to override the

precedence of earlier expiring allocations or even allocations with liens. Lastly, nested funds that become feasible because of overflow to ancestor funds have a negative weighting and are used last, with the earliest expiring allocations being used before later expiring allocations and closer level ancestors being depleted before ancestor funds that are at more distant levels. These allocations are given a weight of $<distance * 100000> - <end_epoch_time>$. After all feasible allocations are sorted according to the above rules, the charge (or lien or quote) will be applied against the allocations one by one in sorted order (highest value first) until the request is fulfilled, or until it fails due to insufficient funds. If a transaction is not able to be satisfied in whole, the entire transaction will fail and no partial debits will be applied.

Chapter 12 Managing Liens

A lien is a hold placed against an allocation. Before usage of a resource or service begins, a lien is made against one or more allocations within the requesting user's applicable fund(s). Subsequent usage requests will also post liens while the available balance (active allocations minus liens) allows. When the usage ends, the lien is removed and the actual charge is made to the allocation(s). This procedure ensures that usage will only be permitted so long as the requestors have sufficient funds.

Associated with a lien is the instance name (name of the item being used such as the job id), the usage record (which contains the item details), a start time and end time for the lien and a description. The lien will automatically expire and no longer count against the user's balance after the end time passes. Each lien will be associated with held amounts from one or more allocations. Operations include creating, querying, modifying, and deleting liens. By default, a standard user may only query liens attributed to them.

Lien queries allow the specification of filter options which narrow down the liens that will be returned. There are two lien filter types that can be employed:

AttributedTo and **ImpingesUpon**. If **ImpingesUpon** is used, the query will return all liens associated with funds satisfying the filters. For example, `Lien Query FilterType:=ImpingesUpon Filter:=User=scottmo` will return all liens impinging on Funds usable by scottmo. If **AttributedTo** is used, the query will return all liens associated with usage records satisfying the filters. For example, `Lien Query FilterType:=AttributedTo Filter:=User=scottmo` will return all liens for resources or services allocated to scottmo.

When a lien is created via the UsageRecord Reserve action (such as via `greserve`), if another lien exists with the same instance name, the default behavior is to leave the old lien in place (and create the new one alongside it). This behavior assumes that the other lien is probably a separate lien created by a resource or service manager that reuses instance ids. However, alternate behaviors may be specified via the mutually exclusive **Modify** or **Replace** options. If the **Replace** option is specified, any pre-existing liens with matching instance names will first be deleted, thereby ensuring only one lien per instance name at a time. If the **Modify** option is specified, a pre-existing lien with matching instance name will be modified to have the new properties (but keeping the same lien id), and can be used to extend a lien. This might be used with incremental charging to dynamically stretch liens along a little at a time as needed. (See [Making a Usage Lien on page 86](#) for a description of the action using these options).

Liens may be granted a grace period (in seconds), which is defined as the difference between the validity period of the lien (end time minus start time) and the expected duration of the usage. The purpose of a grace period is to account for the fact that we may not know precisely when the usage will begin and the lien needs to remain in force during the lifetime of the usage. One can

apply a desired grace period for a lien by setting the end time longer than the specified duration. Alternatively, a grace duration option can be specified with the duration when creating a lien via `greserve` as a helper to computing a relatively adjusted end time.

Creating Liens

Liens are normally created with the `greserve` command (See [Making a Usage Lien on page 86](#)).

However, it is also possible to create a manual lien against specified allocations using the `gmklien` command. A lien object and its allocation associations will be created. Unlike `greserve`, no calculated lien amount will be returned nor will a usage record be created or updated with the lien. Furthermore, `gmklien` will not perform any checking to ensure that the specified allocations have a sufficient active balance to support the lien.

```
gmklien [-J instance_name|job_id] [-s start_time] {-e end_time |  
-t lien_duration] [-d description] [-X, --extension  
property=value]... {-A allocation_id<-fund_id=sublien  
amount,...}... [--debug] [--site site_name] [--help] [--man]  
[--quiet] [--verbose] [--version]
```

Example 12-1: Creating a manual lien

```
$ gmklien -J weekend_run -t 84600 -A "5<-2=3600"  
Successfully created 1 lien
```



Use of the `gmklien` command bypasses the normal mechanisms that prevent more liens from being placed against an allocation than it can support. Use `greserve` instead if you wish to avoid the possibility of oversubscribing the allocations.

Related Topics

[gmklien on page 319](#)

Querying Liens

To display lien information, use the command `glslien`:

```
glslien [-A | -I] [-J instance_pattern | job_id_pattern] [-X, --  
extension property=value]... [-u user_name] [-g group_name] [-  
a account_name] [-o organization_name] [-c class_name] [-m  
machine_name] [--filter filter_name=filter_value]... [--  
filter-type AttributedTo|ImpingesUpon] [--full] [--show]
```

```
attribute_name,...] [--long] [--wide] [-format csv|standard]
[-h, --hours] [--debug] [--site site_name] [--help] [--man] [--quiet]
[--version] {[--] lien_id}
```

Example 12-2: Listing all info about all liens for amy

\$ glslien -u amy						
<u>Id</u>	<u>Instance</u>	<u>Amount</u>	<u>StartTime</u>	<u>EndTime</u>	<u>UsageRecord</u>	<u>Funds</u>
<u>Description</u>						
3	PBS.1234.4	57600	2015-04-06 21:21:48	2015-04-06 22:31:48	7	2

Example 12-3: Listing all info about all liens that impinge against dave's balance

\$ glsres -u dave --filter-type ImpingesUpon						
<u>Id</u>	<u>Instance</u>	<u>Amount</u>	<u>StartTime</u>	<u>EndTime</u>	<u>UsageRecord</u>	<u>Funds</u>
<u>Description</u>						
4	batch.12	7600	2015-04-06 15:30:34	2015-04-06 15:41:50	244	3

Related Topics

[glslien on page 260](#)

Modifying Liens

To modify a lien, use the command `gchlien`:

```
gchlien [-s start_time] [-e end_time] [-t lien_duration] [-d
description] [-X, --extension property=value]... [--debug] [--site
site_name] [--help] [--man] [--quiet] [--verbose] [--version] {[--] lien_id}
```

Example 12-4: Changing the expiration time of a lien

```
$ gchlien -e "2015-06-06 14:43:02" 1
Successfully modified 1 lien
```

Related Topics

[gchlien on page 195](#)

Deleting Liens

To delete a lien, use the command `grmlien`:

```
grmlien {-I | {-J instance_name} | {[-l] lien_id}} [--debug] [--site site_name] [--help] [--man] [--quiet] [--verbose] [--version]
```

Example 12-5: Deleting a lien by instance (or job id)

```
$ grmlien -J PBS.1234.0  
Successfully deleted 1 lien
```

Example 12-6: Deleting a lien by Lien Id

```
$ grmlien 1  
Successfully deleted 1 lien
```

Example 12-7: Purging stale liens

```
$ grmlien -I  
Successfully deleted 2 liens
```

Related Topics

[grmlien on page 377](#)

Chapter 13 Managing Quotes

A quotation provides a way to determine beforehand how much would be charged for a job. When a guaranteed quote is requested, the charge rates applicable to the usage request are saved and a quote id is returned. Charge rates may be specified with the quote or the standard rates may be used in the quote calculation. When the lien and the final charge are issued, the quote id can be referenced to ensure that the saved quote charge rates are used instead of current standard values. A quotation has an expiration time after which it cannot be used. A quotation may also be used to verify that the given job has sufficient funds and meets the policies necessary for the charge to succeed.

Associated with a quote is the id, the instance name (name of the item being used such as the job id), the amount quoted (assuming full use of the quoted resources or services), the usage record (which contains the usage details), a start and end time for the quote, a duration (how long the item is expected to be used), a boolean indicating whether the quote is pinned or unpinned, and a description. Each guaranteed quote will be associated with one or more saved charge rates. Operations include creating, querying, modifying and deleting quotes. By default, a standard user may only query quotes attributed to them.

Quote queries allow the specification of filter options which narrow down the quotes that will be returned. The query will return all quotes associated with usage records satisfying the filters. For example, `Quote Query Filter:=User=scottmo` will return all quotes for resources or services allocated to scottmo.

A quote may be pinned (restricted to a particular instance) or unpinned (allowed to be used by any number of different instances). If a quote is pinned and has not been tied to a particular instance when initially created, it will be tied to the first instance that claims it. Once pinned to an instance, it can then be used repeatedly by that same instance until the quote expires, but not by any other instance. If a quote is not pinned, any instances may use the quoted rates while the quote is active.

A quote may be granted a grace period, which is defined as the difference between the validity period of the quote (end time minus start time) and the expected duration of the usage in seconds. The purpose of a grace period is to account for the fact that we may not know precisely when the usage will begin and the quote needs to be valid during the time of completion of the usage in order for the guaranteed charge rates to be applied. One can apply a desired grace period for a quote by setting the end time longer than the specified duration. Alternatively, a grace duration option can be specified with the duration when creating a quote via `gquote` as a helper to computing a relatively adjusted end time.

A distinction may be made between quotes and quote templates, both of which use the `Quote` object. A quote will always return a cost estimate and will be

associated with a specific usage record. A quote template provides a way to bundle together a package of special charge rates that can be applied to quotes, liens, and charges. Quote templates use the same Quote object as regular quotes but they are not associated with a usage record and do not generate a quote amount.

In calculating a price, a quote will use (in order of lower to higher precedence) the standard charge rates, the charge rates from a specified quote template, the specified override charge rates, or an externally specified charge amount. In saving guaranteed charge rates, the standard charge rates pertaining to the specified usage record properties will be used unless overridden by a specified quote template or specified charge rates.

There are several key purposes for using quotes and quote templates. First, a quote may be requested to discover the cost of using a resource or service. If this is your sole purpose, then you may want to use the `gquote` command with the `--costOnly` option. Second, a quote can be used to check whether the requestor has sufficient access and funds to use the requested resource. This may be accomplished by invoking the `gquote` command without the `--costOnly` option. Third, a quote or a quote template can be used to lock-in current or specified charge rates for use in future liens and charges. If the details of the usage are known and you would like to get a quote amount with a quote id that can be referenced to guarantee the quoted charge rates, you may use the `gquote` command with the `--guarantee` option. Override charge rates may be factored in to the cost estimate of the quote by using the `gquote` command with the `--rate` option. If specific override charge rates need to be saved or guaranteed for future use within a quote, lien, or charge without generating a cost estimate, create a pinned quote template by using the `gmkquote` command with the `--pin` and `-R` options. If it is necessary to create a quote template that can be used to override the standard charge rates for multiple instances, use the `gmkquote` command with the `--nopin` and `-R` options.

Creating Quotes

Quotes are normally generated by the resource management system with the `gquote` command before an instance uses requested resources or services (see [Obtaining Usage Quotes on page 85](#)).

Creating Quote Templates

Quote templates may be created by using the `gmkquote` command. Quote templates provide a way to bundle together a package of special charge rates that can be applied to quotes, liens, and charges.

```
gmkquote [[--pin] [-J instance_name|job_id] | --nopin] [-s start_time] {-e end_time | -t quote_duration} [-d description] [-X, --extension property=value]... { -R charge_rate_name [{charge_rate_value}] = charge_rate_amount,... }... [--debug] [--site site_name] [--help] [--man] [--quiet] [--verbose] [--version]
```

Example 13-1: Creating a pinned quote template

```
$ gmkquote --pin -J vpc.1 -t 86400 -R  
Processors=1.5/s,QualityOfService{Premium}=*1.7  
Successfully created 1 quote template with id 17
```

Example 13-2: Creating an unpinned quote template

```
$ gmkquote --nopin -t 86400 -R Disk=2.5/s,License{Matlab}=4/s  
Successfully created 1 quote template with id 18
```

i Use of the gmkquote command will not result in a cost estimate or the creation of a usage record. Use gquote instead if you wish to obtain a quote for usage.

Related Topics

[gmkquote on page 326](#)

Querying Quotes

To display quote information, use the command glsquote:

```
glsquote [-J instance_name] [-A | -I] [-X, --extension property=value]... [-u user_name] [-g group_name] [-a account_name] [-o organization_name] [-c class_name] [-m machine_name] [--filter filter_name=filter_value]... [--full] [--show_attribute_name,...] [--long] [--wide] [--format csv|standard] [-h, --hours] ] [--debug] [--site site_name] [--help] [--man] [--quiet] [--version] [[-q] quote_id]
```

Example 13-3: Listing all quotes for user amy on machine colony

```
$ glsquote -u amy -m colony  
Id Amount Pinned Instance UsageRecord StartTime EndTime D  
ururation ChargeRates Description  
--- --- --- --- --- --- --- ---  
1 57600 True 242 2015-04-06 12:49:53 2015-04-  
13 13:09:58 3600 Processors:1/s
```

Related Topics

[glsquote on page 275](#)

Modifying Quotes

To modify a quote, use the command gchquote:

```
gchquote [-s start_time] [-e end_time] [-d description] [-X, --  
extension property=value]... [--debug] [--site site_name] [--  
help] [--man] [--quiet] [--verbose] [--version] {[-q] quote  
id}
```

Example 13-4: Changing the expiration time of a quote

```
$ gchquote -e "2015-05-01" 1  
Successfully modified 1 quote
```

Related Topics

[gchquote on page 204](#)

Deleting Quotes

To delete a quote, use the command grmquote:

```
grmquote {-I | {[-q] quote_id}} [--debug] [--site site_name] [-  
--help] [--man] [--quiet] [--verbose] [--version]
```

Example 13-5: Deleting a quote

```
$ grmquote 1  
Successfully deleted 1 quote
```

Example 13-6: Purging stale quotes

```
$ grmquote -I  
Successfully deleted 2 quotes
```

Related Topics

[grmquote on page 383](#)

Chapter 14 Managing Usage Records

Moab Accounting Manager can track the usage of resources and services on your system, recording the charge and the details of the usage in a usage record. A usage record is created when a resource or service manager requests a guaranteed quote for usage, places a lien for usage, or charges for the usage of an item. Usage records can also be created directly via `UsageRecord Create (gmkusage)`. A refund can be invoked to credit a charge amount back to the originating fund. Usage records can also be queried, modified, or deleted. By default, a standard user may only query usage records attributed to them.

In a typical use case, a quote might be used to discover how much it would cost to use an item (resource or service) and to verify the user had sufficient access to the item and funds to cover the requested usage. Just before the item is about to be used, a lien (or hold) might be placed against the user's allocated credits for the requested usage. After the usage is complete, a charge for the actual usage can be debited from their fund and the lien removed.

As is the case for other Moab Accounting Manager objects, usage records are highly customizable. One may remove most usage record properties and add new usage record properties. Refer to the section [Customizing the Usage Record Object on page 89](#) for examples of customizing usage records.

Creating a Usage Record

In most cases, usage records will be created by the resource management system via the `API` or with the `gquote`, the `greserve` or the `gcharge` command.

However, it is also possible to create usage records directly using the `gmkusage` command:

```
gmkusage [-T usage_record_type] [-S service_id] [-u user_name]
[-g group_name] [-a account_name] [-o organization_name] [-c
class_name] [-Q quality_of_service] [-m machine_name] [-N
nodes] [-P processors] [-C cpu_time] [-M memory] [-D disk] [-t
usage_duration] [-s start_time] [-S service_id] [-e end_time]
[-d description] [-X | --extension property=value]... [--
debug] [--site site_name] [--help] [--man] [--quiet] [--
verbose] [--version] {[--J] instance_name|job_id}
```

Example 14-1: Creating a usage record

```
$ gmkusage -u jsmith -a chem -m cluster -X Charge=2468 -P 2 -t 1234 -J PBS.1234.0
Successfully created 1 usage record with id 246
```

- i** The fields which are displayed by default by this command can be customized by setting the **usagerecord.show** configuration parameter in gold.conf.
- i** Use of the gmkusage command to record usage will not result in the debiting of a user's allocation. Use gcharge instead if you wish to charge for the usage.

Related Topics

[gmkusage on page 336](#)

Querying Usage Records

To display usage record information, use the command glsusage:

```
glsusage [-T usage_record_type] [[-J] instance_name_pattern|job_id_pattern] [-S service_id] [-u user_name] [-g group_name] [-a account_name] [-o organization_name] [-c class_name] [-Q quality_of_service] [-m machine_name] [--stage stage] [-s start_time] [-e end_time] [-X, --extension property=value]... [--full] [--show attribute_name,...] [--format csv|standard] [-h, --hours] [--debug] [--site site_name] [--help] [--man] [-quiet] [--verbose] [[-j] usage_record_id]
```

Example 14-2: Show specific info about usage tallied by amy

```
$ glsusage --show=Type,Instance,Account,Machine,Charge -u amy
Type   Instance    Account    Machine    Charge
-----  -----
Job    PBS.1234.0  chemistry   colony     22212
```

Related Topics

[glsusage on page 295](#)

Modifying a Usage Record

It is possible to modify a usage record by using the command gchusage:

```
gchusage [-T usage_record_type] [-S service_id] [-u user_name] [-g group_name] [-a account_name] [-o organization_name] [-c class_name] [-Q quality_of_service] [-m machine_name] [-N nodes] [-P processors] [-C cpu_time] [-M memory] [-D disk] [-t usage_duration] [-s start_time] [-e end_time] [-d description] [-X, --extension property_name=value]... [--debug] [--site
```

```
site_name] [--help] [--man] [--quiet] [--verbose] [--version]  
{[-j] usage_record_id | -J instance_name|job_id}
```

Example 14-3: Changing a usage record

```
$ gchusage -Q HalfPrice -X Charge=1234 -d "Benchmark" -J PBS.1234.0  
Successfully modified 1 usage record
```



Changing a recorded charge in this manner will not change the allocated balance (see [Issuing Usage Refunds on page 88](#) to refund a charge).

Related Topics

[gchusage on page 213](#)

Deleting a Usage Record

To delete a usage record, use the command grmusage:

```
grmusage [--debug] [--site site_name] [--help] [--man] [--quiet]  
[--verbose] [--version] {[-j] usage_record_id | -J  
instance_name|job_id}
```

Example 14-4: Deleting a usage record

```
$ grmusage -J PBS.1234.0  
Successfully deleted 1 usage record
```

Related Topics

[gmkusage on page 336](#)

Obtaining Usage Quotes

Usage quotes can be used to determine how much it will cost to use a resource or service. Provided the cost-only option is not specified, this step will additionally verify that the submitter has sufficient funds and meets all the allocation policy requirements for the usage, and can be used at the submission of the usage request as an early filter to prevent the usage from getting blocked when it tries to obtain a lien to start later. If a guaranteed quote is requested, a quote id is returned and can be used in the subsequent charge to guarantee the rates that were used to form the original quote. A guaranteed quote has the side effect of creating a quote record and a permanent usage record. A quote id will be returned which can be used with the lien and charge to claim the quoted charge rates. A cost-only quote can be used to determine how much would be charged for usage without verifying sufficient funds or

checking to see if the charge could succeed. A breakdown of the charges in the quote can be returned by specifying the `--itemize` option with the `--verbose` option.

To request a usage quote, use the command `gquote`:

```
gquote [-T usage_record_type] [-S service_id] [-u user_name] [-g group_name] [-a account_name] [-o organization] [-c class_name] [-Q quality_of_service] [-m machine_name] [-N nodes] [-P processors] [-C cpu_time] [-M memory] [-D disk] [-X, --extension_property=value]... [-t quote_duration [-G grace_duration]] [-s quote_start_time] [-e quote_end_time] [-d quote_description] [-z quote_amount] [--cost-only | --guarantee] [-R charge_rate_name[{charge_rate_value}]=charge_rate_amount,...]... [-h, --hours] [--itemize] [--debug] [--site site_name] [--help] [--man] [--quiet] [--verbose] [--version] [[-j] usage_record_id] [-q quote_template_id] [-J instance_name|job_id]
```

Example 14-5: Requesting a quote

```
$ gquote -a chemistry -u amy -m colony -P 2 -t 3600
Successfully quoted 7200 credits
```

Example 14-6: Requesting a guaranteed quote

```
$ gquote -a chemistry -u amy -m colony -P 16 -t 3600 --guarantee
Successfully quoted 57600 credits with quote id 1 and usage record id 86
$ glsquote
      Id  Amount  UsageRecord  StartTime          EndTime          Duration  Used   Cha
      rgeRates      Description
      ---  -----  -----  -----  -----  -----
      1    57600     86    2015-04-06 10:09:58  2015-04-
      06 11:09:58  3600      0  Processors:1/s
```

i It is possible to establish a system default machine, project or user to be used in job functions (charge, reserve or quote) when left unspecified (see [Server Configuration on page 150](#)).

Related Topics

[gquote on page 346](#)

Making a Usage Lien

A usage lien can be used to place a hold on the user's fund before usage starts to ensure that the credits will be there when it completes. The `replace` option

may be specified if you want the new lien to replace existing liens of the same instance name (associated with the same usage record). The modify option may be specified to dynamically extend any existing lien with the same instance name with the specified characteristics instead of creating a new one. See [Managing Liens on page 75](#) for more information about these options.

To create a usage lien use the command `greserve`:

```
greserve [-T usage_record_type] [-S service_id] [-u user_name]
[-g group_name] [-a account_name] [-o organization] [-c class
name] [-Q quality_of_service] [-m machine_name] [-N nodes] [-P
processors] [-C cpu_time] [-M memory] [-D disk] [-X, --
extension_property=value] ... [-t lien_duration [-G grace
duration]] [-s lien_start_time] [-e lien_end_time] [-d lien
description] [-z lien_amount] [--modify|--replace] [-R charge
rate_name [{charge_rate_value}]=charge_rate_amount,...]... [-
h, --hours] [--itemize] [--debug] [--site site_name] [--help]
[--man] [--quiet] [--verbose] [--version] [[-j] usage_record
id] [-q quote_id] {-J instance_name|job_id}
```

Example 14-7: Creating a lien

```
$ greserve -J PBS.1234.0 -a chemistry -u amy -m colony -P 2 -t 3600
Successfully reserved 7200 credits with lien id 37 for instance PBS.1234.0 and created
usage record id 87
```

Related Topics

[greserve on page 359](#)

Charging for Usage

A usage charge debits the appropriate allocations based on the attributes of the usage. The charge is calculated based on factors including the resources and services used, the usage time, and other quality-based factors (see [Managing Charge Rates on page 97](#)). By default, any liens associated with the charge will be removed. The incremental option may be specified if you want associated liens to be reduced instead of removed. If a usage record already exists for the instance being charged it will be updated with the data properties passed in with the charge request, otherwise a new usage record will be created.

A quote id can be specified to use a previously quoted set of charge rates. This will also ensure the charge will update the usage record instantiated with the quote. A lien id can be specified to help match up a charge with its lien (this may assist in deleting the correct lien if instance ids are not unique). This will also ensure the charge will update the usage record that may have been instantiated by the lien.

Although, by default, Moab Accounting Manager will calculate the charge for the usage using its default charge rates or using the charge rates saved by a referenced quote or quote template, it is possible to specify override charge rates via the rate option. Alternatively, it is possible to designate an externally calculated charge by specifying the charge amount with the Charge option (-z option to gcharge).

To charge for a usage use the command gcharge:

```
gcharge [-T usage record type] [-S service id] [-u user name] [-g group name] [-a account name] [-o organization name] [-c class name] [-Q quality of service] [-m machine name] [-N nodes] [-P processors] [-C cpu time] [-M memory] [-D disk] [-x usage state] [-X, --extension property=value]... [-t charge duration] [-s charge start time] [-e charge end time] [-d charge description] [-z charge amount] [-f fund id] [--incremental] [-R charge rate name[{charge rate value}]=charge rate amount,...]... [-h, --hours] [--itemize] [--debug] [--site site name] [--help] [--man] [--quiet] [--verbose] [--version] {[-j] usage record id} [-q quote id] [-l lien id] {-J instance name|job id}
```

Example 14-8: Issuing a usage charge

```
$ gcharge -J PBS.1234.0 -a chemistry -u amy -m colony -P 2 -t 1234
Successfully charged 2468 credits for instance PBS.1234.0
1 lien was removed
```

Related Topics

[gcharge on page 176](#)

Issuing Usage Refunds

A charged amount can be credited back in part or in whole by issuing a usage refund. This action attempts to lookup the referenced usage record to ensure that the refund does not exceed the original charge and so that the charge entry can be updated. If multiple matches are found (such as the case when instance names (such as job ids) are non-unique), this command will return the list of matched usage records with unique ids so that the correct usage record can be specified for the refund.

To issue a refund for a usage charge, use the command grefund:

```
grefund [-z amount] [-f fund id] [-d description] [-A] [--filter filter name=filter value]... [--filter-type Exclusive|NonExclusive] [-h, --hours] [--debug] [--site site name] [--help] [--man] [--quiet] [--verbose] [--version] {-J instance name|job id} | {[-j] usage record id}
```

Example 14-9: Issuing a usage refund

```
$ grefund -J PBS.1234.0
Successfully refunded 19744 credits for instance PBS.1234.0
```

Related Topics

[grefund on page 355](#)

Customizing the Usage Record Object

The usage record object as natively defined can be customized with the attributes you want to track in your use cases. The chapter on [Customizing Objects on page 131](#) goes into some detail on the customization syntax. However, since this may be a common requirement, this section will provide a few examples on modifying, adding and deleting usage record attributes and getting them to be tracked and show up in queries.

Usage record discriminators are those properties which are considered primary differentiators between usage, lien, and quote records. Usage record discriminators are used in the dynamic web portal as filters for the listing, modification, and deletion of usage records, liens, and quotes. The default usage record discriminators are Type, User, Group, Account, Organization, Class, QualityOfService, and Machine. Any new attributes added to the usage record object will become usage record discriminators. Removing a discriminator attribute from the usage record object will necessarily remove it as a usage record discriminator as well. It will be necessary to log out and back in after adding or removing a discriminator in order for it to be reflected in the web GUI.

Example 14-10: Setting VM as the default Usage Record Type

As installed, the usage record type defaults to Job. The default value can be set to NULL if there should be no default value, or to any other default value. This example will demonstrate how to set the default usage record type to VM.

```
$ goldsh Attribute Modify Object=UsageRecord Name=Type DefaultValue=VM
Successfully modified 1 attribute
```

Example 14-11: Adding a UsageRecord Application Field (and discriminator)

Let's say you would like to track the application run by your usage scenarios. First, you would add **Application** as an Attribute of the UsageRecord Object.

```
$ goldsh Attribute Create Object=UsageRecord Name=Application DataType=String
Successfully created 1 attribute
```

If you want the new attribute to show up in glsusage, you must add it to the **usagerecord.show** string in gold.conf.

```
$ vi /opt/mam/etc/gold.conf

usagerecord.show =
Id,Instance,User,Group,Organization,Account,QualityOfService,Machine,Stage,Charge,Processors,Nodes,Application,Duration,StartTime,EndTime,Description
```

If you want to filter the usage records by **Application**, (such as listing all usage records associated with the specified application), use the **-X** (or **--extension**) option in `glsusage`.

```
$ glsusage -X Application=foo --show=Type,Instance,Charge,User,Application
Type Instance Charge User Application
-----
Job PBS.1234.0 19744 amy foo
```

You could also use **Application** as the basis of a ChargeRate. See the [Managing Charge Rates on page 97](#) chapter for details on how to do this.

Although the initial step above allows the application value to be tracked in the usage record, it is also possible to add it as an attribute of the Transaction table so that it will be automatically populated from actions having assignments, conditions, options and data values referring to the **Application**.

```
$ goldsh Attribute Create Object=Transaction Name=Application DataType=String
Successfully created 1 attribute
```

Additionally, the `gstatement` client command can show Application as one of its discriminators (which are Account, User, and Machine by default) in its debit detail. These statement discriminators are specified by the **--show** argument to `gstatement` and can be configured with the **statement.show** configuration parameter in `gold.conf`.

Example 14-12: Removing the UsageRecord Class Field

Let's say you were not interested in tracking the class. First, you would delete **Class** as an Attribute of the UsageRecord Object.

```
$ goldsh Attribute Delete Object==UsageRecord Name==Class
Successfully deleted 1 attribute
```

Next, we need to make sure `glsusage` doesn't try to list the class.

```
$ vi /opt/mam/etc/gold.conf

usagerecord.show =
Id,Instance,User,Account,Machine,QualityOfService,Stage,Charge,Processors,Nodes,Application,Duration,StartTime,EndTime,Description
```

If the attribute you want to delete is also an attribute in the Transaction table, you could delete it from there as well.

Usage Record Property Verification

If a usage record property has an object associated with it, you may want to verify that when that usage record property is specified in a scheduling action (Charge, Reserve, Quote), it verifies that that property is a valid instance of its object type. You can apply a simple verification to a usage record property by setting the property's **Values** attribute to an @ sign followed by the name of the object.

Example 14-13: Ensure that an organization specified in a charge actually exists

```
$ goldsh Attribute Modify Object==UsageRecord Name==Organization Values=@Organization
Successfully modified 1 attribute
```

See [Managing Attributes on page 134](#) for more information about setting the **Values** attribute.

Usage Record Property Defaults

It is possible to set defaults for usage record properties when they are not specified in the usage data for a charge, lien, or quote. There are two cases which must be considered — when the property has an object associated with it and when the property does not.

If a property does not have an object associated with it, simply set the **DefaultValue** attribute for the property's **UsageRecord** Attribute object to the desired value.

Example 14-14: Setting a system-wide simple default class of batch for usage functions

```
$ goldsh Attribute Modify Object==UsageRecord Name==Class DefaultValue=batch
Successfully modified 1 attribute
```

If a property does have an object associated with it, you will need to both set the **DefaultValue** attribute for the property's **UsageRecord** Attribute object to the desired value AND set the **DefaultValue** attribute for the corresponding object to the desired value.

Example 14-15: Setting a system-wide simple default user of anonymous for usage functions

```
$ goldsh Attribute Modify Object==UsageRecord Name==User DefaultValue=anonymous
Successfully modified 1 attribute
```

See [Global Object-Based Defaults on page 134](#) for more information about setting default values for objects. See [Local Attribute-Based Defaults on page 138](#) for more information about setting default values for attributes.

Usage Record Property Auto-Generation

It is possible for usage record properties which have object definitions to automatically create the referenced objects the first time they are encountered in a usage function (charge, reserve or quote). To do this, the referenced object must be set to `AutoGen=True` and the **Values** attribute for the `UsageRecord` attribute corresponding to the object must be set to a string consisting of the @ sign followed by the object name.

Example 14-16: Setting the Usage Record Type to auto-generate Items for usage functions

For example, let's assume there were many usage record types that could be charged for (Food, Book, Haircut) and that you had already created an Item object. It would be possible to automatically generate a new Item instance each time a new usage record type was referenced in a charge operation.

```
$ goldsh Object Modify Name==Item AutoGen=True
Successfully modified 1 object

$ goldsh Attribute Modify Object==UsageRecord Name==Type Values=@Item
Successfully modified 1 attribute
```

See [Object Auto-Generation on page 133](#) for more information about the auto-generation of objects.

Usage Record Property Instantiators

It is possible to establish a dynamic correlation between usage record properties in which one usage record property can instantiate another. For example, if a user is specified in a charge but no account is specified then the user's default account should be applied to the fund constraints and logged; or if an account is specified in a charge but not its organization then the organization corresponding to that account should be looked up and applied to the fund constraints and logged. Three usage record property instantiator types are currently supported and are configured by prefixing the property instance's `Values` foreign object reference with the appropriate characters: Assign if not defined (@?=), Assign if not different (@!=), Assign always (@:=). We shall look at each of these individually and in different terms.

Applying a correlated default (@?=) — If property X is specified with the value `x` in the usage record and property Y is not specified in the usage record and if the object instance referred to by `x` has a correlated default value of `y'` for property Y', then `y'` will be applied as the default value for property Y in the usage record. For example, we could establish the notion of a default account for a user.

Example 14-17: Establishing a default account for a user

First we add a `DefaultAccount` attribute (the name is arbitrary) to the `User` object and give it a **Values** property of `@?=Account`.

```
$ goldsh Attribute Create Object=User Name=DefaultAccount DataType=String
Values="\\"@?=Account\\\" Description=\"\\\"Default Account\\\""
Successfully created 1 attribute
```

Then we can establish the default account for user scottmo to be chemistry.

```
User Modify Name==scottmo DefaultAccount=chemistry
Successfully modified 1 user
```

Subsequently, when a Charge, Lien, or Quote is issued that specifies the User scottmo but does not specify the Account, the chemistry Account will be applied to the charge as if originally specified in the usage record charge data.

Applying a correlated verification (@!=) — If property X is specified with the value x in the usage record and property Y is specified with the value y in the usage record and if the object instance referred to by x has a correlated verification value of y' for the property Y' and if y' does not equal y , then fail with an error message. Additionally, if property X is specified with the value x in the usage record and property Y is not specified in the usage record and if the object instance referred to by x has a correlated verification value of y' for property Y', then y' will be applied as the default value for property Y in the usage record. For example, we could establish a parent-child relationship between organizations and accounts in which explicitly specified incongruities result in a failure.

Example 14-18: Establishing an override hierarchy with accounts and organizations

First we add a VerifyOrganization attribute (the name is arbitrary) to the Account object and give it a **Values** property @!=Organization.

```
$ goldsh Attribute Create Object=Account Name=VerifyOrganization DataType=String
Values="\\"@!=Organization\\\" Description=\"\\\"Verify Organization\\\""
Successfully created 1 attribute
```

Then we can establish the verify organization for account chemistry to be sciences.

```
$ goldsh Account Modify Name==chemistry VerifyOrganization=sciences
Successfully modified 1 account
```

Subsequently, when a Charge, Lien, or Quote is issued that specifies the Account chemistry and specifies the wrong Organization (e.g. arts), the transaction will fail with an error message. Additionally, when a Charge, Lien, or Quote is issued that specifies the Account chemistry but does not specify the Organization, the Organization sciences will be applied to the charge as if originally specified in the usage record charge data.

Applying a correlated override (@:=) — If property X is specified with the value x in the usage record and if the object instance referred to by x has a correlated override value of y' for property Y', then y' will be applied as the override value for property Y in the usage record. For example, we could

establish a parent-child relationship between organizations and accounts in which explicitly specified incongruities are silently overridden with the value from the child.

Example 14-19: Establishing an override hierarchy with accounts and organizations

First we add an `OverrideOrganization` attribute (the name is arbitrary) to the `Account` object and give it a **Values** property of `@:=Organization`.

```
$ goldsh Attribute Create Object=Account Name=OverrideOrganization DataType=String  
Values="\"@:=Organization\\"" Description="\"Override Organization\\""  
Successfully created 1 attribute
```

Then we can establish the override organization for account `chemistry` to be `sciences`.

```
$ goldsh Account Modify Name==chemistry OverrideOrganization=sciences  
Successfully modified 1 account
```

Subsequently, when a Charge, Reserve or Quote is issued that specifies the Account `chemistry` and specifies either the wrong Organization (e.g. `arts`) or no Organization, the Organization `sciences` will be silently applied to the charge as if originally specified in the usage record charge data.

Chapter 15 Managing Itemized Charges

The itemized charge table provides an ability to display the components of a composite charge in a line item format. Each charge transaction will write the components of its charge into the charge record so that you can get a line-item breakdown of each charge for usage including the names, values, rates, scaling factors, charge amounts and details listed for each component of the charge. This capability is enabled by setting `charge.itemization = true` in the `goldd.conf` (it is false by default).

Itemized charges may only be queried. They are created automatically in charge transactions and there are no command line clients to change or remove them.

Additionally, an itemize option can be specified for quotes, liens, and charges to include an itemized charge breakdown in the response data instead of a single line with the amount.

Querying Itemized Charges

To display itemized charge information, use the command `glscharge`:

```
glscharge [-j usage_record_id] [-J instance_name] [-n usage_property_name] [-s start_time] [-e end_time] [--full] [--show_attribute_name,...] [--format csv|standard] [-h, --hours] [--debug] [--site site_name] [--help] [--man] [--quiet] [--version]
```

Example 15-1: Listing all itemized charge information

\$ glscharge								
UsageRecord	Instance	Name	Value	Duration	Rate	ScalingFactor	Amount	Creatio
nTime		Description						
24	job.1	Storage	100	86400	1.157e-07	1		1 2015-
04-05 17:49:41								
25	job.2	Processors	4	86400	5.787e-07	1		20 2015-
04-05 17:49:42								
25	job.2	Memory	4096	86400	1.13e-08	1		4 2015-
04-05 17:49:42								
26	job.3	Processors	1	86400	5.787e-05	1		5 2015-
04-05 17:49:43								
26	job.3	Memory	1004	86400	1.13e-08	1		1 2015-
04-05 17:49:43								

Related Topics

[glscharge on page 242](#)

Displaying Itemized Charges for a Transaction

In addition to the itemized charge table, Moab Accounting Manager captures the itemized charges for usage record charges, liens, and guaranteed quotes in the details of the transaction. The itemized charges show the details for the formula used to calculate the charge for the transaction. To display the itemized charges for a scheduling transaction, parse the details from the command `glstxn --full -A Charge|Reserve|Quote`:

Example 15-2: Extract the itemized charges for a job charge

```
$ glstxn -A Charge -J PBS.1234.1 -q --show Details | perl -pe 's/.*(ItemizedCharges[^,]*).*/\1/'  
ItemizedCharges:=4 [Processors] * 5.787e-05 [ChargeRate{Processors}] * 86400  
[Duration] + 4096 [Memory] * 1.13e-08 [ChargeRate{Memory}] * 86400 [Duration] = 24
```

Chapter 16 Managing Charge Rates

Charge rates establish how much to charge for usage. A charge rate consists of its name, an optional value, and the amount. Charge rates are applied when usage properties matching the charge rate names are found in the usage data. In order for a charge rate of a given name to be applied, a usage record attribute of the same name must exist. For example, a charge rate having the name Processors will be applied if Processors is defined as a Usage Record attribute and the incoming usage data for the charge request contains a property called Processors that matches the value specified in the charge rate.

There are two basic types of charge rates: Name-valued charge rates and Numeric-valued charge rates.

- Name-valued charge rates charge rates are used for usage properties that take strings for values (e.g. `QualityOfService=premium` or `Account=chemistry`). The charge rate that is applied will be determined by a lookup of the usage property value to see if there is a matching charge rate value. A default rate may be specified by creating a name-valued charge rate with an empty charge rate value. Multiple values may be assigned to the same rate via separate charge rate definitions or by combining the values in a single charge rate value separated by commas.
- Numeric-valued charge rates are used for usage properties that take numbers for values (e.g. `Processors=2` or `CpuTime=12.67`). The charge rate amount that is applied will be multiplied by the usage property value. The charge rate value is commonly left blank to be taken as the default rate for the full range of usage property values. A particular value may also be specified as the charge rate value which means that that rate will only be used if the usage property value exactly matches the charge rate value. A half-bounded expression may be used by specifying a less than or greater than sign with an optional equal sign, followed by the number. For example, the charge rate value `<=4` would match a usage property value of `x` if `x <= 4`. A charge rate value may also be specified as a range (of the form `<number>[-<number>]`). For example, the range `1-4` would be match a usage property value of `x` if `1 <= x <= 4`. If you need to be more specific about the boundedness of the ranges, you may replace the dash with a less than sign with an optional equal sign on either side of it to indicate whether the endpoints are included. For example, the range `1<4` would match if `1 < x < 4`, `1=<4` would match if `1 <= x < 4`, `1<=4` would match if `1 < x <= 4` and `1=<4` would match if `1 <= x <= 4`. So you might use ranges like `1=<2`, `2=<4`, `4=<8`, and `>=8`. Multiple values or value ranges having the same charge rate may be specified in a single expression separated by commas.

A charge rate amount may have an operation modifier which dictates the way the rate is factored into the charge calculation. For example, consumption-based charge rates or usage fees will often be additive in nature while quality-based charge rates may be multiplicative. The additive charge rates may be

further distinguished by whether they should be added before or after the multiplicative charge rates are applied. The charge formula can be represented in the following form: $(S(\text{Pre-Additive Rates}) * P(\text{Multiplicative Rates})) + S(\text{Post-Additive Rates})$. Thus, there are three operation modifiers: Pre-Additive, Multiplicative and Post-Additive.

Composite type	Description
Pre-Additive	Pre-additive modifiers are applied to charge rates that should be added together before any charge multipliers are applied. A pre-additive modifier is specified by prepending a plus sign '+' to the charge rate amount. Since pre-additive is the most commonly specified operation modifier, a charge rate amount without an operation modifier will be assumed to be pre-additive by default.
Multiplicative	Multiplicative modifiers are applied to charge rates that should be multiplied together with other multiplicative charge rates and with the sum of the pre-additive charge rates. A multiplicative modifier is specified by prepending an asterisk '*' to the charge rate amount.
Post-Additive	Post-additive modifiers are applied to charge rates that should be added together after any charge multipliers are applied. A post-additive modifier is specified by appending a plus sign '+' to the charge rate amount.

A pre-additive charge rate may have a time-based modifier which dictates that the charge should be multiplied by the amount of time the feature was used. For example, it is common for the processor charge to be multiplied by the amount of time the processors were used. A time-based modifier is specified by appending a forward slash '/' to the charge rate amount, followed by one of the following time designators: s (per-second), m (per-minute), h (per-hour), d (per-day), W (per-week), M (per-month), Y (per-year). As an example, a per-hour time-based modifier is specified by appending '/h' to the charge rate amount and will cause a charge to be multiplied by the number of hours the feature was used. Technically, a rate with a time-based modifier will be multiplied by the number of seconds the feature was used, then divided by the number of seconds corresponding to the time designator (e.g. 3600), and will ultimately be rounded to the number of decimal places in the currency precision.

A pre-additive charge rate may have a divisor modifier which dictates that the charge should be divided by the specified integer. A divisor modifier is specified by appending a forward slash '/' to the charge rate amount, followed by an integer number. A divisor modifier can be used in lieu of expressing a small decimal fraction charge rate such as when converting a value from MegaBytes to GigaBytes. If a divisor modifier is used in conjunction with a time-based modifier, the divisor modifier must precede the time-based modifier.

A charge rate may have one or more conditions which dictates additional qualifications that must be met in order for the charge rate to be applied. A condition is specified by prepending `<propertyName>=<propertyValue>`

followed by a question mark '?' to the value field of the charge rate. If you want Processors to apply a special charge rate (e.g. .5/s) for user amy, the charge rate value should consist of the string "User=amy?". Additionally, you may combine charge rate conditions with either a pipe symbol '|' for or, or an ampersand symbol '&' for and. For example, User=amy|User=dave? or User=amy&Project=chemistry?. You may not combine ands and ors in the same charge rate value.

Creating Charge Rates

To create a new charge rate, use the command `gmkrate`:

```
gmkrate -n charge_rate_name [-x charge_rate_value] [-d description] [--debug] [--site site_name] [--help] [--man] [--quiet] [--verbose] [--version] {[-z] charge_rate_amount}
```

i If a usage record attribute does not exist for the name of the charge rate you are creating, you must first create the corresponding usage record property. See [Customizing the Usage Record Object on page 89](#).

Example 16-1: Creating a numeric-valued pre-additive charge rate with a divisor and a time-based modifier

```
$ gmkrate -n Memory -z 1/1024/s
Successfully created 1 charge rate
```

Example 16-2: Creating a name-valued pre-additive charge rate

```
$ gmkrate -n License -x Matlab -z 5
Successfully created 1 charge rate
```

Example 16-3: Creating a numeric-valued pre-additive charge rate without a time-based modifier

```
$ gmkrate -n CpuTime -z 1
Successfully created 1 charge rate
```

Example 16-4: Creating a numeric-valued multiplicative charge rate

```
$ gmkrate -n Discount -z *1
Successfully created 1 charge rate
```

Example 16-5: Creating a couple of name-valued multiplicative charge rates and a default rate

```
$ gmkrate -n QualityOfService -x Premium -z *2
Successfully created 1 charge rate
$ gmkrate -n QualityOfService -J BottomFeeder -z *0.5
Successfully created 1 charge rate
```

```
$ gmkrate -n QualityOfService -z *1  
Successfully created 1 charge rate
```

Example 16-6: Creating a numeric-valued post-additive charge rate

```
$ gmkrate -n Shipping -z 25+  
Successfully created 1 charge rate
```

Example 16-7: Creating a name-valued post-additive charge rate

```
$ gmkrate -n Zone -x Asia -z 200+  
Successfully created 1 charge rate
```

Example 16-8: Creating a couple of conditional numeric-valued pre-additive charge rates

```
$ gmkrate -n Disk -x User=dave? -z 0.2/s  
Successfully created 1 charge rate  
$ gmkrate -n Disk -x User=mike? -z 0.5/s  
Successfully created 1 charge rate
```

Example 16-9: Creating some numeric-valued pre-additive charge rate ranges and a default

```
$ $ gmkrate -n Processors -x 1-4 -z 2/s  
Successfully created 1 charge rate  
$ gmkrate -n Processors -x 5-8 -z 1.5/s  
Successfully created 1 charge rate  
$ gmkrate -n Processors -z 1/s  
Successfully created 1 charge rate
```

Example 16-10: Creating some numeric-valued pre-additive rate ranges for floating point values (without time-based modifiers)

```
$ $ gmkrate -n Power -x '<2' -z 0.005  
Successfully created 1 charge rate  
$ $ gmkrate -n Power -x '2=<4' -z 0.004  
Successfully created 1 charge rate  
$ $ gmkrate -n Power -x '>=4' -z 0.003  
Successfully created 1 charge rate
```

Example 16-11: Assigning multiple classes to run for free

```
$ $ gmkrate -n Class -x dev,test -z *0  
Successfully created 1 charge rate
```

Related Topics

[gmkrate on page 330](#)

Querying Charge Rates

To display charge rate information, use the command `glsrate`:

```
glsrate [-n charge_rate_name] [-x charge_rate_value] [--full] [-  
-show attribute_name,...] [--format csv|standard] [--debug] [-  
-site site_name] [--help] [--man] [--quiet] [--version]
```

Example 16-12: Listing all charge rates

Name	Value	Amount	Description
Class	dev,test	*0	
CpuTime		1	
Discount		*1	
Disk	User=dave?	0.2/s	
Disk	User=mike?	0.5/s	
License	Matlab	5/s	
Memory		1/1024/s	
Power	<2	0.005	
Power	2=<4	0.004	
Power	>=4	0.003	
Processors		1/s	
Processors	1-4	2/s	
Processors	5-8	1.5/s	
QualityOfService		*1	
QualityOfService	BottomFeeder	*0.5	
QualityOfService	Premium	*2	
Shipping		25+	
Zone	Asia	200+	

Related Topics

[glsrate on page 281](#)

Modifying Charge Rates

To modify a charge rate, use the command `gchrate`:

```
gchrate [-n charge_rate_name] [-x charge_rate_value] [-z charge_  
rate_amount] [-d description] [--debug] [--site site_name] [-  
-help] [--man] [--quiet] [--verbose] [--version]
```

Example 16-13: Changing a charge rate

```
$ gchrate -n License -x Matlab -z 4/s  
Successfully modified 1 charge rate
```

Related Topics

[gchrate on page 207](#)

Deleting Charge Rates

To delete a charge rate, use the command `grmrate`:

```
grmrate -n charge_rate_name [-x charge_rate_value] [--debug] [--site site_name] [--help] [--man] [--quiet] [--verbose] [--version]
```

Example 16-14: Deleting a charge rate

```
$ grmrate -n Memory  
Successfully deleted 1 charge rate
```

Related Topics

[grmrate on page 386](#)

Chapter 17 Managing Transactions

Moab Accounting Manager logs all modifying transactions in a detailed transaction journal (queries are not recorded). Previous transactions can be queried but not modified or deleted. By default, a standard user may only query transactions performed by them.

Querying Transactions

To display transaction information, use the command `glstrans`:

```
glstrans [-O object] [-A action] [-k primary key value] [-U
actor] [-f fund id] [-i allocation id] [-u user name] [-a
account name] [-m machine name] [-j usage record id] [-J
instance name|job id] [-s start time] [-e end time] [-T
transaction id] [-R request id] [-X, --extension
property=value]... [--full] [--show attribute name,...] [--
format csv|standard] [-h, --hours] [--debug] [--site site
name] [--help] [--man] [--quiet] [--version]
```

Example 17-1: List all deposits made in 2015

```
$ glstrans -A Deposit -s 2015-01-01 -e 2016-01-01
```

Example 17-2: List everything done by amy since the beginning of 2015

```
$ glstrans -U amy -s 2015-01-01
```

Example 17-3: List all transactions related to job moab.1

```
$ glstrans -J moab.1
```

Example 17-4: List all transactions affecting charge rates

```
$ glstrans -O ChargeRate
```

Related Topics

[glstrans on page 288](#)

Customizing the Transaction Object

The transaction record as natively defined can be customized with the attributes you want to track in your use cases. It is possible to add additional attributes to the Transaction table so that it will be automatically populated

from actions having assignments, conditions, options and data values referring to the attribute.

Transaction discriminators are those properties which are considered primary differentiators between transaction records (besides the metadata differentiators of object, action, and instance). Transaction discriminators are used in the dynamic web portal as filters for the listing of transaction records. Any new attributes added to the Transaction object will become transaction discriminators. Removing a discriminator attribute from the transaction object will necessarily remove it as a transaction discriminator as well. It will be necessary to log out and back in after adding or removing a discriminator in order for it to be reflected in the web GUI.

Example 17-5: Adding an Organization field to the Transaction record (which also makes it a discriminator)

```
$ goldsh Attribute Create Object=Transaction Name=Organization DataType=String  
Successfully created 1 attribute
```

Chapter 18 Managing Events

Moab Accounting Manager has an internal event scheduler that can be configured to execute Moab Accounting Manager actions at a designated time in the future or on a periodic basis. Valid actions on an event include Create, Query, Fire, Modify, Refresh and Delete. Event attributes include Id, FireCommand, ArmTime, FireTime, RearmPeriod, EndTime, Notify, RearmOnFailure, FailureCommand, CatchUp and Description.

There are two server configuration parameters which affect event scheduling: **event.scheduler** which specifies whether the event scheduler is enabled or not (it is disabled by default) and **event.pollinterval** which is the period in minutes that the event scheduler uses to fire events. The poll interval must divide evenly into the number of minutes in a day (1440).

The command(s) to be fired by an event are expressed in a serialized form of the request identical to the syntax used in the interactive control program (goldsh). There are two commands that can be configured in an event: the **FireCommand** which is the command to be executed when the event is fired, and the **FailureCommand** which is the command to be executed if the fired command results in an unsuccessful response status. The **FireTime** is the target time for the event to be triggered by the event scheduler. The actual fire time may be dependent on the state of the server and will be recorded in the **CreationTime** property of the corresponding "Event Fire" Transaction. An event may also be fired manually with the Event Fire action.

The **RearmPeriod** is a time period expression specifying when the event will be rearmed. This period expression is of the form: "<period>[[@<instant>] [~|^]!!]" where period may be something like 1 day, 2 hours, or 5 minutes. Instant locks the period to a specific instant within the time period such as 1 day @ hour 12 or 1 month @ day 3. The modifiers indicate whether the time period should be relative to now (!), or relative to the start of this (~) designator (month or minute, etc.), or relative to the start of the first (^) designator (month or minute, etc.). For example, assuming the FireTime was 7:15, if you specified 4 hours ! as the rearm period it would be rearmed at 11:15, if you specified 4 hours ~ as the rearm period it would be rearmed at 11:00, and if you specified 4 hours ^ as the rearm period it would be rearmed at 8:00.

The **ArmTime** is the time the event was last armed or fired. This field is used as a reference time to be able to derive how long the event has been waiting to happen. This field will be initially set to mark the moment the first FireTime is set and updated thereafter to indicate the last time the event was fired. In the case where an event does not have a FireTime set, this field may be set manually and used in a similar manner. If we consider the time between event firings as "laps", this could be thought of as the Lap Start Time. If the **RearmOnFailure** boolean is set to `False`, the event will not be rearmed if the command was unsuccessful. If set to `True`, the event will be evaluated for rearming even if the command response has a status of `Failure`. The

standard default is `False`. If the **CatchUp** boolean is set to `True` and the server was down during the time this event should have fired, the event scheduler will attempt to make up for the past due events by progressively firing them (rearming based on previous arm time) until catching up to the present. The actions will still show as having occurred in the present rather than in the past. If set to `False`, and the server is brought back up after an outage, the event scheduler will still fire immediately for a past due event, but it will only fire once and then rearm relative to the current time.

A Notification method can be specified via the **Notify** parameter and is of the form: `[+--] [<delivery_method>:] [<recipient>] [, [+--] [<delivery_method>:] [<recipient>]]*`. If the term is a `-`, the notification is sent only on failure. If the term is a `+`, the notification is sent only on success. Otherwise the notification is always sent. There can be multiple notify expressions separated by a comma. All applicable notifications will be sent. See the chapter on [Managing Notifications on page 109](#) for more information about delivery method and recipient.

Creating Events

To create a new event, use the command `gmkevent`:

```
gmkevent --fire-command goldsh_command [-s fire_time] [-e end_time] [--rearm-period rearm_period] [--rearm-on-failure True|False] [--failure-command failure_command] [--notify notification_url] [--catch-up (True)|False] [-d description] [--debug] [--site site_name] [--help] [--man] [--quiet] [--verbose] [--version]
```

Example 18-1: Creating an automatic allocation renewal event

```
$ gmkevent --fire-command "Fund Reset" -s "2016-01-01" --rearm-period "3 months^"
Successfully created 1 event
```

Related Topics

[gmkevent on page 310](#)

Querying Events

To display event information, use the command `glsevent`:

```
glsevent [-s start_time] [-e end_time] [--full] [--show attribute_name,...] [--format csv|standard] [--debug] [--site site_name] [--help] [--man] [--quiet] [--version] [[-E] event_id]
```

Example 18-2: Listing all events

```
$ glsevent
  Id FireCommand FireTime      ArmTime          RalarmPeriod EndTime Notify
  RalarmOnFailure FailureCommand CatchUp CreationTime      Description
  -----
  1 Fund Reset    2016-01-01 2015-11-09 10:31:28 3 months^
                True      2015-11-09 10:31:28
                                False
```

Related Topics[glsevent on page 248](#)

Modifying Events

To modify an event, use the command gchevent:

```
gchevent --fire-command fire_command [-s fire_time] [-e end_time]
                                         [--rearm-period rearm_period] [--rearm-on-failure True|False]
                                         [--failure-command failure_command] [--notify notification_url]
                                         [--catch-up (True)|False] [-d description]
                                         [--debug] [--site site_name] [--help] [-man] [--quiet] [--verbose]
                                         [--version] {[-E] event_id}
```

Example 18-3: Changing an event's rearm period to be monthly

```
$ gchevent --rearm-period "1 month" 1
Successfully modified 1 event
```

Related Topics[gchevent on page 185](#)

Deleting Events

To delete an event, use the command grmevent:

```
grmevent [--debug] [--site site_name] [--help] [-man] [--quiet]
          [--verbose] [--version] {[-E] event_id}
```

Example 18-4: Deleting an event

```
$ grmevent 1
Successfully deleted 1 event
```

Related Topics[grmevent on page 373](#)

Chapter 19 Managing Notifications

When event commands are executed (asynchronously), the success or failure of the operation is communicated back to the initiator via a notification. When an event is created, you may specify the **Notify** option which will associate a notification method with the event. Currently there is only one **DeliveryMethod** implemented which is `Store`. With the `Store` delivery method, command response information is stored as instances of the `Notification` object. These messages can later be retrieved by the initiator via a `Notification Query`. Payments can also route a notification method down to their associated events via a `Notify` option.

The notification attributes include **`Id`** (auto-generated), **`Type`**, **`Event`**, **`Status`**, **`Code`**, **`Message`**, **`Key`**, **`Recipient`**, **`EndTime`** and **`CreationTime`**. Stored notifications can be queried on any of these conditions. The notification type distinguishes what type of command resulted in the notification (Fire or Failure). The notification key is the value of the primary key of the object instance that the command acted on (e.g. the Payment Id). The recipient could be a user name or any tag that identifies the intended reader for the notification. The `Notification Query` supports a **`Delete`** option, which if set to `True`, will delete the notifications after they have been queried. Additionally, stored notifications have an `EndTime` after which they are automatically deleted by Gold. The `Notification` actions include `Send`, `Refresh`, `Create`, `Query`, `Delete` and `Modify`.

There are two server configuration parameters which affect notifications: **`notification.deliverymethod`** which dictates which `deliverymethod` is used by default if unspecified and **`notification.duration`** which defines how long notifications stick around if the `Store` delivery method is used.

Querying Notifications

To display notification information, use the command `glsnot`:

```
glsnot [-E event_id] [-T notification_type] [-k primary_key_value] [-u recipient] [-x status] [-s start_time] [-e end_time] [--delete] [--full] [--show attribute_name,...] [--format csv|standard] [--debug] [--site site_name] [--help] [--man] [--quiet] [--version] [[-N] notification_id]
```

Example 19-1: Listing all failure notifications

```
$ glsnot -x Failure
Id Event Type Status Code Message
Key Recipient EndTim
e CreationTime
-----
-----
-----
4 20 Fire Failure 782 Payment Begin failed starting payment: Failed creating payment
      starting lien: Insufficient balance to reserve usage (Instance Moab.1)
      \nClearing the event fire time.\nThe controlling event has been deleted. 9
amy 2015-04-23 13:35:01 2015-04-09 13:35:01
```

Related Topics

[glsnot on page 267](#)

Deleting Notifications

To delete a notification, use the command `grmnot`:

```
grmnot [--debug] [--site site_name] [--help] [--man] [--quiet]
[--verbose] [--version] {[-N] notification_id}
```

Example 19-2: Deleting a notification

```
$ grmnot 4
Successfully deleted 1 notification
```

Example 19-3: Deleting all successful notifications

To delete many notifications, query them with the `--delete` option:

```
$ glsnot -x Success --delete

Id Event Type Status Code Message

Key Re
cipient EndTime CreationTime
-----
```

4 20 Fire Failure 782 Payment Begin failed starting payment: Failed creating payment starting lien: Insufficient balance to reserve usage (Instance Moab.1)
\ nClearing the event fire time.\ nThe controlling event has been deleted.

9 amy 2015-04-23 13:35:01 2015-04-09 13:35:01

1 11 Fire Success 000 Payment Begin: Successfully charged 10 credits for instance Moab.1\ nSuccessfully charged 20 credits for instance Moab.2\ nSuccessfully charged 20 credits for instance Moab.3\ nSuccessfully started payment (6)
and created 3 liens\ nClearing the event fire time.\ nThe controlling event has been deleted.

6 scottmo 2015-04-23 13:28:02 2015-04-09 13:28:02

2 14 Fire Success 000 Payment Begin: Successfully charged 10 credits for instance Moab.1\ nSuccessfully charged 20 credits for instance Moab.2\ nSuccessfully charged 20 credits for instance Moab.3\ nSuccessfully started payment (7)
and created 3 liens\ nClearing the event fire time.\ nThe controlling event has been deleted.

7 amy 2015-04-23 13:31:02 2015-04-09 13:31:02

3 17 Fire Success 000 Payment Begin: Successfully charged 10 credits for instance Moab.1\ nSuccessfully charged 20 credits for instance Moab.2\ nSuccessfully charged 20 credits for instance Moab.3\ nSuccessfully started payment (8)
and created 3 liens\ nClearing the event fire time.\ nThe controlling event has been deleted.

8 amy 2015-04-23 13:32:02 2015-04-09 13:32:02

Successfully deleted 3 notifications

Related Topics

[grmnot on page 379](#)

Chapter 20 Managing Roles

Moab Accounting Manager uses instance-level role-based access controls to determine what users can perform what functions. Named roles are created, actions are associated with the roles, and users are assigned to these roles.

The actions for a role consist of a set of tuples of object, action and instance permitted by the role. In other words, each role action defines an object (whether specific or ANY), the action that can be taken on that object (whether specific or ANY) and the instance of the object that action can be taken on (whether specific or ANY).

In the base configuration, there are three default roles: `SystemAdmin`, `Anonymous` and `OVERRIDE`. Other configurations, such as the bank configuration, add additional roles. Roles can be added as desired. The three base roles are required for proper function of Moab Accounting Manager and should not be deleted. By default, the `SystemAdmin` role can perform any action on any object. This role is usually assigned to the super user. The `Anonymous` role is intended to define the actions available to your standard unprivileged user. This may include the ability to set your password, query certain public objects and modify objects that belong to you (implemented via the `OVERRIDE` role). The `OVERRIDE` role is a special role type that defines those actions that should use special business logic intrinsic to the routine that handles that object and action. For example, in the bank configuration, the `OVERRIDE` logic for the Account Query routine will only allow the standard user to see information about accounts for which he or she is a member. A given user's privileges will be the superset of the actions of all roles that apply to that user.

The instance indicates which specific instances of the object the action can be performed on. There are several special instance types that can be used in certain situations. The `ANY` instance is supported by all objects and permits the specified action on all instances of the specified object. The `SELF` instance applies to the user's own instance if the object is User, or to objects that have a User attribute associated with the user. The `MEMBERS` instance applies to objects for which the user is a direct member. The `ADMIN` instance applies to objects for which the user is designated as an administrator. Unless otherwise specified, the instance will default to a value of `ANY`.

Creating Roles

To create a new role, use the command `gmkrole`. Users and actions may be associated with the role at creation time. When assigning actions to a role, the object, action and instance must be specified in the form shown. Multiple actions or users may be specified for the role.

```
gmkrole [-d description] [-u user_name,...]... [-A object_name->action_name[instance_name],...]... [--debug] [--site site_name] [--help] [--man] [--quiet] [--verbose] [--version] {[-r] role_name}
```

Example 20-1: Creating a Manager role

```
$ gmkrole -r Manager -d "Manages Roles and Responsibilities"
```

```
Successfully created 1 role
```

Related Topics

[gmkrole on page 333](#)

Querying Roles

To display the role information, use the command `glsrole`:

```
glsrole [--full] [--show attribute_name,...] [--long] [--wide]
[--format csv|standard] [--debug] [--site site_name] [--help]
[--man] [--quiet] [--version] {[-r] role_name}
```

Example 20-2: Listing all roles along with users and descriptions

```
$ glsrole --show=Name,Users,Description
```

Name	Users	Description
AccountAdmin		Can update or view an account they are admin for
Anonymous	ANY	Things that can be done by anybody
OVERRIDE	ANY	A custom authorization method will be invoked
Schedule	root	Scheduler relevant Transactions
SystemAdmin	scottmo	Can update or view any object
UserServices		User Services

Example 20-3: Listing information about the scheduler role

```
$ glsrole --long Scheduler
```

Name	Users	Actions	Description
Scheduler	root	UsageRecord->Create (ANY) UsageRecord->Quote (ANY) UsageRecord->Reserve (ANY) UsageRecord->Charge (ANY) Lien->Delete (ANY)	Scheduler relevant Transactions

Related Topics

[glsrole on page 284](#)

Modifying Roles

To modify a role, use the command `gchrole`:

```
gchrole [-d description] [--add-user(s) user_name,...]... [--  
add-action(s) object_name->action_name[{:instance  
name}],...]... [--del-user(s) user_name,...]... [--del-action  
(s) object_name->action_name[{:instance_name}],...]... [--  
debug] [--site site_name] [--help] [--man] [--quiet] [--  
verbose] [--version] {[[-r] role_name]
```

Users may be added to a role or removed from a role. Actions also may be added to a role or removed from a role. When specifying actions, the instance will default to a value of ANY.

Example 20-4: Adding a user to a role

Let's add dave to our new Manager role:

```
$ gchrole --add-user dave -r Manager  
Successfully added 1 user
```

Example 20-5: Associating an action with a role

Allow the Manager to change role responsibilities:

```
$ gchrole --add-action "RoleAction->ANY" Manager -v  
Successfully added 1 action
```

Related Topics

[gchrole on page 209](#)

Deleting Roles

To delete a role, use the command `grmrole`:

```
grmrole [--debug] [--site site_name] [--help] [--man] [--quiet]  
[--verbose] [--version] {[[-r] role_name]}
```

Users may be added to a role or removed from a role. Actions also may be added to a role or removed from a role. When specifying actions, the instance will default to a value of ANY.

Example 20-6: Deleting the Manager role

Let's add dave to our new Manager role:

```
$ grmrole Manager  
Successfully deleted 1 role and 2 associations
```

Related Topics

[grmrole on page 388](#)

Chapter 21 Managing Passwords

Passwords must be established for each user who wishes to use the web-based GUI. Passwords must be at least eight characters and are stored in encrypted form. A `gchpasswd` command line client exists to aid a user or administrator in setting or changing a password. Other operations (deleting or listing password entries) must be performed using the interactive control program (`goldsh`). By default, a standard user may only set or change his or her own password. A system administrator may set or change any user's password.



Because Moab Accounting Manager caches password information for faster responsiveness, it will be necessary to restart the server after running `gchpasswd` for the GUI to accept that password change.

Setting Passwords

To set a new password, use the command `gchpasswd`. If the user name is not specified via an option or as the unique argument, then the invoking user will be taken as the user whose password will be set. The invoker will be prompted for the new password.

```
gchpasswd [--debug] [--site site_name] [--help] [--man] [--quiet] [--verbose] [--version] {[-u] user_name}
```

Example 21-1: Setting a password

```
$ gchpasswd amy  
Enter your new password:  
Successfully created 1 password
```

Related Topics

[gchpasswd on page 201](#)

Querying Passwords

To display password information, use the command `goldsh Password Query`:

```
goldsh Password Query [Show:=<"Field1,Field2,...">] [User==<UserName>] [ShowUsage:=True]
```

Example 21-2: List the users who have set passwords

```
$ goldsh Password Query Show:=User  
User  
-----  
amy  
gold
```

Deleting Passwords

To delete a password, use the command `goldsh Password Delete`:

```
goldsh Password Delete User==<User Name>]
```

i The `goldsh` control program allows you to make powerful and sweeping modifications to Moab Accounting Manager objects. Misuse of this command could result in the inadvertent deletion of all passwords.

Example 21-3: Deleting a password

```
$ goldsh Password Delete User==amy  
User    Password  
-----  
amy    HZYzwD20o1XIE/gxRYyFKP2sumkCluHm  
Successfully deleted 1 password
```

Chapter 22 Using the Gold Shell (goldsh)

goldsh is an interactive control program that can access all of the advanced functionality in Moab Accounting Manager.

- i** The goldsh control program allows you to make powerful and sweeping modifications to many objects with a single command. Inadvertent mistakes could result in modifications that are very difficult to reverse.

Usage

Goldsh commands can be invoked directly from the command line as arguments, or read from stdin (interactively or redirected from a file).

```
goldsh [--debug] [--site site name] [--help] [--man] [--format  
csv|standard] [--quiet] [--verbose] [--version] [command]
```

Example 22-1: Specifying the command as direct arguments

```
$ goldsh System Query  
-----  
Name          Version Description  
-----  
Moab Accounting Manager 8.1.2.0 Commercial Release
```

Example 22-2: Using the interactive prompt

```
$ goldsh  
gold> System Query  
-----  
Name          Version Description  
-----  
Moab Accounting Manager 8.1.2.0 Commercial Release  
gold> quit
```

Example 22-3: Reading commands from a file

```
$ cat >commands.gold <<EOF  
System Query  
quit  
EOF  
$ goldsh <commands.gold  
-----  
Name          Version Description  
-----  
Moab Accounting Manager 8.1.2.0 Commercial Release
```

Related Topics

[goldsh on page 410](#)

Command Syntax

goldsh commands are of the form:

```
<Object> [= <Alias>] [, <Object> [= <Alias>] ...] <Action> [
[<Conjunction>] [<Open_Parenthesis>... ] [<Object>.] 
<Name> <Operator> [<Subject>.] <Value> [<Close_Par-
enthesis>... ] ...]
```

The basic form of a command is `<Object> <Action> [<Name><Operator><Value>] *`. When an action is performed on more than one object, such as in a multi-object query, the objects are specified in a comma-separated list. Commands may accept zero or more predicates which may function as fields to return, conditions, update values, processing options, etc. Predicates, in their simplest form, are expressed as Name, Operator, Value tuples. Predicates may be combined via conjunctions with grouping specified with parentheses. When performing multi-object queries, names and values may need to be associated with their respective objects.

Valid conjunctions include:

Conjunction	Meaning
<code>&&</code>	and
<code> </code>	or
<code>&!</code>	and not
<code> </code>	or not

Open parentheses may be any number of literal open parentheses '('.

Name is the name of the condition, assignment, or option. When performing a multi-object query, a name may need to be prepended by its associated object separated by a period.

Valid operators include:

Operator	Meaning
<code>==</code>	equals
<code><</code>	less than

Operator	Meaning
>	greater than
<=	less than or equal to
>=	greater than or equal to
!=	not equal to
~	matches
=	is assigned
+=	is incremented by
-=	is decremented by
:=	option
::!	not option

Value is the value of the selection list, condition, assignment, or option. When performing a multi-object query, a value may need to be prepended by its associated object (called the subject) separated by a period.

Close parentheses may be any number of literal closing parentheses ')'.

Valid Objects

To list the objects available for use with commands in goldsh commands, use the goldsh command: Object Query

Example 22-4: Listing all objects

```
gold> Object Query Show:="Sort (Name)"  
  
Name  
-----  
Account  
AccountUser  
Action  
Allocation  
Attribute  
ChargeRate  
Constraint  
Fund  
FundFund  
Lien  
LienAllocation  
Object  
Organization  
Password  
Quote  
QuoteChargeRate  
Role  
RoleAction  
RoleUser  
System  
Transaction  
UsageRecord  
User
```

Valid Actions for an Object

To list the actions that can be performed on an object, use the goldsh command: Action Query

Example 22-5: Listing all actions associated with the Fund object

```
gold> Action Query Object==Fund Show:="Sort (Name)"  
  
Name  
-----  
Create  
Delete  
Deposit  
Modify  
Query  
Transfer  
Undelete  
Withdraw
```

Valid Predicates for an Object and Action

By appending the option ShowUsage:=True to a command, the syntax of the command is returned, expressed in SSSRMAP XML Message Format.

Example 22-6: Show the usage for Allocation Query

```

gold> Allocation Query ShowUsage:=True

<Request action="Query">
  <Object>Allocation<Object>
    [<Get name="Id" [op="Sort|Tros|Count|GroupBy|Max|Min"]></Get>]
    [<Get name="Fund" [op="Sort|Tros|Count|GroupBy|Max|Min"]></Get>]
    [<Get name="StartTime" [op="Sort|Tros|Count|GroupBy|Max|Min"]></Get>]
    [<Get name="EndTime" [op="Sort|Tros|Count|GroupBy|Max|Min"]></Get>]
    [<Get name="Amount" [op="Sort|Tros|Count|GroupBy|Max|Min|Sum|Average"]></Get>]
    [<Get name="CreditLimit" [op="Sort|Tros|Count|GroupBy|Max|Min|Sum|Average"]></Get>]
  ></Get>
  [<Get name="InitialDeposit" [op="Sort|Tros|Count|GroupBy|Max|Min|Sum|Average"]></Get>]
></Get>
  [<Get name="Allocated" [op="Sort|Tros|Count|GroupBy|Max|Min|Sum|Average"]></Get>]
  [<Get name="Active" [op="Sort|Tros|Count|GroupBy"]></Get>]
  [<Get name="Description" [op="Sort|Tros|Count|GroupBy|Max|Min"]></Get>]
  [<Where name="Id" [op="EQ|NE|GT|GE|LT|LE (EQ)"] [conj="And|Or (And)"]
  [group="<Integer Number>Integer Number"]</Where>]
  [<Where name="Fund" [op="EQ|NE|GT|GE|LT|LE|Match|NotMatch (EQ)"] [conj="And|Or
  (And)"] [group="<Integer Number>Fund Name"]</Where>]
  [<Where name="StartTime" [op="EQ|NE|GT|GE|LT|LE (EQ)"] [conj="And|Or (And)"]
  [group="<Integer Number>YYYY-MM-DD [ hh:mm:ss ]|infinity|infinity|now"]</Where>]
  [<Where name="EndTime" [op="EQ|NE|GT|GE|LT|LE (EQ)"] [conj="And|Or (And)"]
  [group="<Integer Number>YYYY-MM-DD [ hh:mm:ss ]|infinity|infinity|now"]</Where>]
  [<Where name="Amount" [op="EQ|NE|GT|GE|LT|LE (EQ)"] [conj="And|Or (And)"]
  [group="<Integer Number>Decimal Number"]</Where>]
  [<Where name="CreditLimit" [op="EQ|NE|GT|GE|LT|LE (EQ)"] [conj="And|Or (And)"]
  [group="<Integer Number>Decimal Number"]</Where>]
  [<Where name="InitialDeposit" [op="EQ|NE|GT|GE|LT|LE (EQ)"] [conj="And|Or (And)"]
  [group="<Integer Number>Decimal Number"]</Where>]
  [<Where name="Allocated" [op="EQ|NE|GT|GE|LT|LE (EQ)"] [conj="And|Or (And)"]
  [group="<Integer Number>Decimal Number"]</Where>]
  [<Where name="Active" [op="EQ|NE (EQ)"] [conj="And|Or (And)"]
  [group="<Integer Number>True|False"]</Where>]
  [<Where name="Description" [op="EQ|NE|GT|GE|LT|LE|Match|NotMatch (EQ)"]
  [conj="And|Or (And)"] [group="<Integer Number>Description"]</Where>]
  [<Option name="Filter">True|False (False)</Option>]
  [<Option name="FilterType">Exclusive|NonExclusive (NonExclusive)</Option>]
  [<Option name="IncludeAncestors">True|False (False)</Option>]
  [<Option name="Time">YYYY-MM-DD [ hh:mm:ss ]</Option>]
  [<Option name="Unique">True|False (False)</Option>]
  [<Option name="Limit">{Integer Number}</Option>]
  [<Option name="Offset">Integer Number</Option>]
  [<Option name="ShowHidden">True|False (False)</Option>]
  [<Option name="ShowUsage">True|False (False)</Option>]
<Request>
```

Common Options

There are a number of options that may be specified for all commands. These options include: ShowUsage

ShowUsage — This option may be included with any command to cause the command to return a usage message in SSSRMAP XML Message Format.

Common Actions Available for Most Objects

There are a number of actions that are available for most objects. These actions include Query, Create, Modify, Delete, and Undelete. Commands involving these actions inherit some common structure unique to the action type.

Query Action

The Query action is used to query objects. It accepts selections that describe the attributes (fields) to return (including aggregation operations on those attributes), conditions that select which objects to return the attributes for, and other options unique to queries.

Selections

Selections use the Show option to specify a list of the attributes to return for the selected object. If selections are not specified, a default set of attributes (defaulting to those not marked as hidden) will be returned.

Name = Show

Op = :=

Value = "attribute1,attribute2,attribute3,..."

Aggregation operators may be applied to attributes by enclosing the target attribute in parenthesis and prepending the name of the desired operator. The aggregation operators that can be applied depend on the datatype of the attribute.

Valid selection operators include:

Sort — Ascending sort

Tros — Descending sort

Count — Count

Max — Maximum value

Min — Minimum value

Average — Average value

Sum — Sum

GroupBy — Group other aggregations by this attribute

Additionally, aliases can be applied to selections so that columns can be renamed as desired. Aliases are expressed by adding =*<Alias>* to the target attribute name (and after the trailing parenthesis of the aggregation, if specified).

For example: Allocation Query Show:="GroupBy) Fund) , Sum(Amount) =Total"

Conditions	<p>Conditions are used to select which objects the action is to be performed on.</p> <p>Name = Name of the attribute to be tested</p> <p>Op = conditional operator</p> <p>Value = The object or value against which the attribute is tested</p> <p>Valid condition operators include:</p> <ul style="list-style-type: none"> == Equal to != Not equal to < Less than > Greater than <= Less than or equal to >= Greater than or equal to ~ Matches !~ Does not match <p>Matching uses the wildcards * and ? (equivalent to SQL % and _ respectively) in a manner similar to file globbing. * matches zero or more unspecified characters and ? matches exactly one unspecified character. For example <code>mscf*</code> matches objects having the specified attributes whose values start with the letters <code>mscf</code>, while <code>mscf?</code> matches objects having the specified attributes whose values start with <code>mscf</code> and have a total of exactly five characters.</p>
Options	<p>Options indicate processing options that affect the result.</p> <p>Name = Name of the option</p> <p>Op = :=</p> <p>Value = Value of the option</p> <p>Valid options for query actions include:</p> <ul style="list-style-type: none"> ShowHidden:=True False (False) Includes hidden attributes in the result Time:=YYYY-MM-DD [hh:mm:ss] Run the command as if it were the specified time Unique:=True False (False) Display only unique results (like DISTINCT in SQL) Limit:={Integer Number} Limit the results to the number of objects specified

Example 22-7: Return the number of inactive liens

```
gold> Lien Query EndTime<now Show:="Count(Id)"
Id
-----
8
```

Create Action

The Create action is used to create a new object. It accepts assignments that describe the values of the attributes to be set.

Assignments	Assignments specify values to be assigned to attributes in the new object. Name = Name of the attribute being assigned a value Op = = (is assigned) Value = The new value being assigned to the attribute
-------------	--

Example 22-8: Add a new account member

```
gold> AccountUser Create Account=chemistry Name=scottmo
Account      Name      Active     Admin
-----
chemistry    scottmo   True      False
Successfully created 1 accountUser
```

Modify Action

The Modify action is used to modify existing objects. It accepts conditions that select which objects will be modified and predicates that describe the values of the attributes to be set.

Assignments	Assignments specify values to be assigned to attributes in the selected objects. Name = Name of the attribute being assigned a value Op = assignment operators {=, +=, -=} Value = The value being assigned to the attribute Valid assignment operators include: = is assigned += is incremented by -= is decremented by
-------------	---

Conditions	<p>Conditions are used to select which objects the action is to be performed on.</p> <p>Name = Name of the attribute to be tested</p> <p>Op = conditional operator</p> <p>Value = The object or value against which the attribute is tested</p> <p>Valid condition operators include:</p> <ul style="list-style-type: none"> <code>==</code> Equal to <code>!=</code> Not equal to <code><</code> Less than <code>></code> Greater than <code><=</code> Less than or equal to <code>>=</code> Greater than or equal to <code>~</code> Matches <code>!~</code> Does not match <p>Matching uses the wildcards <code>*</code> and <code>?</code> (equivalent to SQL <code>%</code> and <code>_</code> respectively) in a manner similar to file globbing. <code>*</code> matches zero or more unspecified characters and <code>?</code> matches exactly one unspecified character. For example <code>mscf*</code> matches objects having the specified attributes whose values start with the letters <code>mscf</code>, while <code>mscf?*</code> matches objects having the specified attributes whose values start with <code>mscf</code> and have a total of exactly five characters.</p>
------------	--

Example 22-9: Change/set scottmo's phone number and email address

```
gold> User Modify Name==scottmo PhoneNumber="(509) 376-2204"
EmailAddress="scottmo@adaptivecomputing.com"
```

Name	Active	CommonName	PhoneNumber	EmailAddress
		DefaultAccount	Description	
scottmo	True	Jackson, Scott M.	(509) 376-	
2204		scottmo@adaptivecomputing.com		

Successfully modified 1 user

Example 22-10: Extend all liens against account chemistry by 10 days

```
gold> Lien Modify EndTime+=864000 Instance=="job.1"
```

Id	Fund	Amount	Instance	UsageRecord	User	Project	Machine	EndTime
			Description					
1	2	57600	PBS.1234.0	1	amy	chemistry	colony	2015-
04-16 10:47:30								

Successfully modified 1 lien

Delete Action

The Delete action is used to delete objects. It accepts conditions that select which objects are to be deleted.

Conditions	<p>Conditions are used to select which objects the action is to be performed on.</p> <p>Name = Name of the attribute to be tested</p> <p>Op = conditional operator</p> <p>Value = The object or value against which the attribute is tested</p> <p>Valid condition operators include:</p> <ul style="list-style-type: none"> <code>==</code> Equal to <code>!=</code> Not equal to <code><</code> Less than <code>></code> Greater than <code><=</code> Less than or equal to <code>>=</code> Greater than or equal to <code>~</code> Matches <code>!~</code> Does not match <p>Matching uses the wildcards <code>*</code> and <code>?</code> (equivalent to SQL <code>%</code> and <code>_</code> respectively) in a manner similar to file globbing. <code>*</code> matches zero or more unspecified characters and <code>?</code> matches exactly one unspecified character. For example <code>mscf*</code> matches objects having the specified attributes whose values start with the letters <code>mscf</code>, while <code>mscf?</code> matches objects having the specified attributes whose values start with <code>mscf</code> and have a total of exactly five characters.</p>
------------	---

Example 22-11: Get rid of the pesky Jacksons

```
gold> User Delete CommonName~"Jackson*"
Name      Active    CommonName          PhoneNumber        EmailAddress
          DefaultAccount           Description
-----  -----
scottmo   True     Jackson, Scott M.   (509) 376-
2204      scottmo@adaptivecomputing.gov
Successfully deleted 1 user and 1 association
```

Undelete Action

The Undelete action is used to restore deleted objects. It accepts conditions that select which objects are to be undeleted.

Conditions	<p>Conditions are used to select which objects the action is to be performed on.</p> <p>Name = Name of the attribute to be tested</p> <p>Op = conditional operator</p> <p>Value = The object or value against which the attribute is tested</p> <p>Valid condition operators include:</p> <ul style="list-style-type: none"> <code>==</code> Equal to <code>!=</code> Not equal to <code><</code> Less than <code>></code> Greater than <code><=</code> Less than or equal to <code>>=</code> Greater than or equal to <code>~</code> Matches <code>!~</code> Does not match <p>Matching uses the wildcards <code>*</code> and <code>?</code> (equivalent to SQL <code>%</code> and <code>_</code> respectively) in a manner similar to file globbing. <code>*</code> matches zero or more unspecified characters and <code>?</code> matches exactly one unspecified character. For example <code>mscf*</code> matches objects having the specified attributes whose values start with the letters <code>mscf</code>, while <code>mscf?</code> matches objects having the specified attributes whose values start with <code>mscf</code> and have a total of exactly five characters.</p>
------------	---

Example 22-12: Resurrect the deleted users that were active

```
gold> User Undelete Active==True
Name      Active    CommonName          PhoneNumber        EmailAddress
          DefaultAccount           Description
-----
scottmo   True     Jackson, Scott M.   (509) 376-
2204      scottmo@adaptivecomputing.com
Successfully undeleted 1 user and 1 association
```

Multi-Object Queries

Goldsh supports multi-object queries (table joins). Multiple objects are specified via a comma-separated list and attributes need to be prefixed by the associated object.

Example 22-13: Print the sums for active balance and allocated amounts grouped by account

```
gold> Allocation,Constraint Query
Show:="GroupBy(Constraint.Value)=Account,Sum(Allocation.Amount)=Balance,Sum
(Allocation.Deposited)=Allocation"
Constraint.Fund==Allocation.Fund Constraint.Name==Account
Allocation.Active==True
Account      Balance      Allocation
-----
biology      193651124    360000000
chemistry    296167659    360000000
```

Example 22-14: Show all active accounts for amy's privileges

```
gold> RoleUser,RoleAction Query
Show:="RoleAction.Object,RoleAction.Name=Action"
RoleUser.Role==RoleAction.Role && ( RoleUser.Name==amy || 
RoleUser.Name==ANY ) Unique:=True

Object          Action
-----
Account        Query
AccountUser    Query
Action         Query
Allocation     Query
Attribute      Query
ChargeRate    Query
Constraint    Query
Fund           Query
FundFund      Query
Lien           Query
LienAllocation Query
Object         Query
Organization   Query
Password      ANY
Quote          Query
QuoteChargeRate Query
Role           Query
RoleAction     Query
RoleUser       Query
System         Query
Transaction   Query
UsageRecord   Query
User          Query
```

i Although the forgoing was a good example of a join request, it should be understood that it is not a straightforward way to determine the full extent of a user's privileges. Some of the actions may be tied to specific object instances and many of them are associated with an override method which may not actually permit the user access to any instances of the object.

Using

Show:="RoleUser.Role,RoleUser.Name=User,RoleAction.Object,Rol
eAction.Name=Action,RoleAction.Instance" may be revealing in this
regard. See the chapter on [Managing Roles on page 113](#) for more
information about managing roles.

Chapter 23 Customizing Objects

Moab Accounting Manager provides the ability to dynamically create new objects or customize or delete existing objects through the interactive control program (goldsh).

- i The object customizations described in this chapter will be noticeable in subsequent goldsh queries (and in the web GUI after a fresh login). Client commands may need to be modified to properly interact with changed objects or attributes.
- i The goldsh control program allows you to make powerful and sweeping modifications to many objects with a single command. Inadvertent mistakes could result in modifications that are very difficult to reverse.

Managing Objects

In Moab Accounting Manager, Objects correspond to tables in the repository which have Attributes (such as Name and Color) and Actions (such as Query and Modify). A specific instance of an object is described as an Instance and has Properties (the specific values of the attributes for that object). The instance is uniquely referred to via its primary key(s) (such as its Name or Id).

An object must have a name and may have a description. An object may be set to auto-generate its instances when first seen (see [Object Auto-Generation on page 133](#)) and/or a default value may be designated for the object (see [Global Object-Based Defaults on page 134](#)).

Objects may reference other objects. If a single instance of an object references only a single instance of another object (for example, a usage record may only have one user), then it is sufficient for the first object to have an attribute field for the second object (the UsageRecord object has an attribute called User). However, if there may be a many-to-many relationship between objects (for example, an account may have multiple users and a user may belong to multiple accounts), then it is necessary to maintain a separate object as an association table (e.g. AccountUser). When creating an association object, the object should be given an appropriate name (e.g. AccountUser), it should be marked as an association (`Association=True`), and an object needs to be designated for the parent (e.g. Account) and the child (e.g. User). The association object itself may have additional attributes that provide qualitative information about the association (e.g. a particular AccountUser association may be active or be an administrator).

Creating a Custom Object

To create a new object, use the command `goldsh Object Create`. When an object is created, the 5 default actions are automatically created for the object: Create, Delete, Modify, Query and Undelete. A number of default metadata attributes are created as well: CreationTime, ModificationTime, Deleted, RequestId and TransactionId. These attributes are normally hidden in regular queries.

```
goldsh Object Create Name=<Object Name> [AutoGen=True|False]
[DefaultValue=<Default Value>] [Description=<Description>]
[Association=True|False] [Child=<Child Object>]
[Parent=<Parent Object>] [ShowUsage:=True]
```

Example 23-1: Creating a Node Object

```
$ goldsh Object Create Name=Node Description=\"Node Information\"
Successfully created 1 object and 5 actions
```

Example 23-2: Add a node name attribute

```
$ goldsh Attribute Create Object=Node Name=Name DataType=String PrimaryKey=True
Successfully created 1 attribute
```

Example 23-3: Add a processor count attribute

```
$ goldsh Attribute Create Object=Node Name=Processors DataType=Integer
Successfully created 1 attribute
```

Querying Objects

To display object information, use the command `goldsh Object Query`.

```
goldsh Object Query [Name=<Object Name>]
>Show:=Name,AutoGen,DefaultValue,Description,Association,Paren
t,Child] [ShowUsage:=True]
```

Example 23-4: List Information for the Node Object

```
$ goldsh Object Query Name==Node
Name Association Parent Child DefaultValue AutoGen Description
---- ----- ---- -----
Node False
```

Modifying an Object

It is possible to modify an object by using the command `goldsh Object Modify`.

```
goldsh Object Query [Name=<Object Name>] [AutoGen=True|False]
[DefaultValue=Default Value] [Description=Description]
[Association=True|False] [Child=Child Object]
[Parent=Parent Object] [ShowUsage:=True]
```

Example 23-5: Changing the Node object's description

```
$ goldsh Object Modify Name==Node Description="\"Host Information\""
Successfully modified 1 object
```

Deleting an Object

To delete an object, use the command `goldsh Object Delete`. When an object is deleted, all associated attributes, actions and other associations are automatically deleted as well.

```
goldsh Object Delete [Name=<Object Name>] [ShowUsage:=True]
```

Example 23-6: Deleting the Node Object

```
$ goldsh Object Delete Name==Node
Successfully deleted 1 object
```

i This is a very dangerous operation and could result in the deletion of all object definitions requiring database repair. The `goldsh` control program allows you to make powerful and sweeping modifications to many objects with a single command. Be sure to specify conditions for the object you want to delete.

Object Auto-Generation

It is possible to have object instances be automatically generated the first time they are referenced in designated contexts. For example, you might want a user be auto-generated when newly added to an account. You could have an organization auto-generated when specified as the default for a user. You could have a cost-center be auto-generated when referenced in a usage record. To do this, the referenced object must be set to `AutoGen=True` and the **Values** property for the attribute that you want to trigger the auto-generation must be set to a string consisting of the @ sign followed by the object name.

Example 23-7: Auto-generate an account's organization

For example, let's assume that your accounts belong to specific organizations that you may want to run a report against but you don't want to define all of the organizations up front. It would be possible to automatically generate a new organization instance each time an undefined organization is specified for an account.

```
$ goldsh Object Modify Name==Organization AutoGen=True
Successfully modified 1 object
$ goldsh Attribute Modify Object==Account Name==Organization Values=@Organization
Successfully modified 1 attribute
```

See [Usage Record Property Auto-Generation on page 91](#) for a discussion of auto-generating objects referenced in usage records.

Global Object-Based Defaults

It is possible to set a global default for an object that will be applied to all attributes referencing this object. When a new instance of an object is being created which has an attribute referring to another object via its **Values** property, if that attribute has not been specified and you want it to default to the global default, you will need to set the **DefaultValue** attribute for the referenced object to the desired value.

Example 23-8: Setting a system-wide simple default organization called general

```
$ goldsh Object Modify Name==Organization DefaultValue=general
Successfully modified 1 object
```

Thereafter each (non-association) object which has an attribute with a **Values** property set to @Organization will default to general if that attribute is not specified. Perhaps we would want the default value to be taken for the organization when a new account is created.

```
$ goldsh Attribute Modify Object==Account Name==Organization Values=@Organization
Successfully modified 1 attribute
```

See [Local Attribute-Based Defaults on page 138](#) for more information about setting default values for attributes. See [Usage Record Property Defaults on page 91](#) for more information about setting default values for usage record properties.

Managing Attributes

Objects can have any number of fields called Attributes. When an object is first created, a number of attributes are created for the object by default. These are: **CreationTime** (time the object was first created), **ModificationTime** (time the object was last updated), **Deleted** (whether the object is deleted or not), **RequestId** (request id that resulted in the last modification of the object), **TransactionId** (transaction id that resulted in the last modification of the object).

An attribute must have a name and be associated with an object.

An attribute will have a data type which can be one of (**AutoGen**, **TimeStamp**, **Boolean**, **Float**, **Integer**, **Currency**, **String**) and defaults to String. A data type of AutoGen means the field will be a primary key of type integer which will assume the next auto-incremented value from the g_key_generator table. TimeStamps are epoch times stored in integer format. Booleans are strings constrained to the values of True or False (or unset). Float is used to store decimal or floating point values. Currency is like Float but may have special business logic for handling currency values.

An object may have zero or more attributes which are primary keys (PrimaryKey==True), the combination of which are used to uniquely identify an object instance. Moab Accounting Manager will try to ensure that there can only be one object instance with the exact same set of values of its primary keys.

A required attribute (Required==True), must be either specified or be derived via a default value or other dynamic mechanism when the object is created. It can also not be unset.

A fixed attribute (Fixed==True), may not be changed from its initial value.

An attribute may be constrained to certain values via the **Values** attribute. The values may be constrained to members of a list expressed as a parenthesized comma-delimited list of strings (i.e. Values=" (Brazil,China,France,Russia,USA) "). Alternatively, the values may be constrained to be an instance of a particular object type (like a foreign key constraint) by assigning to the Values attribute the name of an object prefixed by the @ sign (e.g. Values="@Account" -- which would constrain the value of this attribute to be a valid account name). Stronger versions of the @-prefixed object-constrained values may be used in Quote, Reserve and Charge actions to enforce dynamic interactions between usage record properties such as to assign default values if not defined (e.g. Values="@?=Account"), verification values which evoke an error if they differ (e.g. Values="@!=Account"), or designated values which always overwrite the value (e.g. Values="@:=Account"). See [Usage Record Property Instantiators on page 92](#) for more information.

A default value may be assigned to an attribute via the **DefaultValue** attribute. When a new instance of an object is created, if a property is not specified for the attribute, the default value will be used.

The **Sequence** attribute determines which order an object's attributes will be listed in for queries if no selection list is specified in the query. Attributes with smaller sequence numbers will appear before attributes with larger sequence numbers. The Sequence attribute is also used to enforce a proper attribute display ordering in the web GUI.

The **Hidden** attribute specifies whether an attribute should be shown in a query by default or not. Hidden attributes can be seen in queries by specifying the **ShowHidden** option with a value of True.

The **Description** field is a location to describe the meaning of the attribute and is used in the GUI for field descriptions.

Adding an Attribute to an Object

To create a new attribute for an object, use the command `goldsh Attribute Create`:

```
goldsh Attribute Create Object=<Object Name> Name=<Attribute Name>
[DataType=AutoGen|TimeStamp|Boolean|Float|Integer|Currency|
(String)] [PrimaryKey=True| (False)] [Required=True| (False)]
[Fixed=True| (False)] [Values=<Foreign Key or List of Values>]
[DefaultValue=<Default Value>] [Sequence=<Integer Number>]
[Hidden=<True| (False)>] [Description=<Description>]
[ShowUsage:=True]
```

Example 23-9: Adding a Country Attribute to User

```
$ goldsh Attribute Create Object=User Name=Country Values="\"
(Brazil,China,France,Russia,USA)\"" DefaultValue=USA
Successfully created 1 attribute
```

Example 23-10: Tracking Submission Time in Usage records

```
$ goldsh Attribute Create Object=UsageRecord Name=SubmissionTime DataType=TimeStamp
Successfully created 1 attribute
```

Querying Attributes

To display attribute information, use the command `goldsh Attribute Query`:

```
goldsh Attribute Query Object=<Object Name> Name=<Attribute Name>
[Show:=Object,Name,DataType,PrimaryKey,Required,Fixed,Values,Default
Value,Sequence,Hidden,Description] [ShowHidden:=True]
[ShowUsage:=True]
```

Example 23-11: List the attributes of the Node object

```
$ goldsh Attribute Query Object==Node
Object Name          DataType PrimaryKey Required Fixed Values DefaultValue Sequence
ce Hidden Description
```

Object Name	DataType	PrimaryKey	Required	Fixed	Values	DefaultValue	Sequence
Node Processors	Integer	False	False	False		20	
Node Name	String	True	True	True		10	
Node TransactionId	Integer	False	False	True		990	
Node RequestId	Integer	False	False	True		980	
Node Deleted	Boolean	False	False	True		970	
Node ModificationTime	TimeStamp	False	False	True		960	
Node CreationTime	TimeStamp	False	False	True		950	
Node First Created							

Modifying an Attribute

To modify an attribute, use the command `goldsh Attribute Modify`:

```
goldsh Attribute Modify Object==<Object Name> Name==<Attribute Name>
[Required=True| (False)] [Fixed=True| (False)] [Values=<Foreign
Key or List of Values>] [DefaultValue=<Default Value>]
[Sequence=<Integer Number>] [Hidden=<True| (False)>]
[Description=<Description>] [ShowUsage:=True]
```

- i** The goldsh control program allows you to make powerful and sweeping modifications to many objects with a single command. A mistake made using this command could result in the inadvertent modification of all attributes.

Example 23-12: Change Account Organization values to not be restricted to the set of organization instances

```
$ goldsh Attribute Modify Object==Account Name==Organization Values=NULL
Successfully modified 1 attribute
```

Removing an Attribute From an Object

To delete an attribute, use the command `goldsh Attribute Delete`:

```
goldsh Attribute Delete Object==<Object Name> Name==<Attribute Name>
[ShowUsage:=True]
```

- i** The goldsh control program allows you to make powerful and sweeping modifications to many objects with a single command. A mistake made using this command could result in the inadvertent deletion of all attributes.

- i** When using Moab Accounting Manager as an accounting manager, certain objects and attributes are assumed to exist. For example, a call to UsageRecord Charge would fail if you had deleted the **Allocation Amount** attribute. The Attribute Undelete command might come in useful in such a case.

Example 23-13: Removing the Organization attribute from Account

```
$ goldsh Attribute Delete Object==Account Name==Organization
Successfully deleted 1 attribute
```

Example 23-14: Perhaps we don't care to track the QualityOfService attribute in a Usage record

```
$ goldsh Attribute Delete Object==UsageRecord Name==QualityOfService
Successfully deleted 1 attribute
```

Local Attribute-Based Defaults

It is possible to set a specific default for an object attribute that will be applied when an instance of that object is created but the attribute is not specified. This type of default is intended for attributes which do not refer to another object or which should vary from the object global default. This default value is assigned to an attribute via the **DefaultValue** attribute. When a new instance of the associated object is created, if a property is not specified for the attribute, the specified default value will be used. A local attribute default will have precedence over a global object default.

```
goldsh Attribute Delete Object=<Object Name> Name=<Attribute Name> [ShowUsage:=True]
```

Example 23-15: Setting a default organization just for the account object

```
$ goldsh Attribute Modify Object==Account Name==Organization DefaultValue=university
Successfully modified 1 attribute
```

Example 23-16: Setting a default phone for the user object

```
$ goldsh Attribute Modify Object==User Name==PhoneNumber DefaultValue="\"NoPhone\""
Successfully modified 1 attribute
```

See [Global Object-Based Defaults on page 134](#) for more information about setting default values for objects.

See [Usage Record Property Defaults on page 91](#) for more information about setting default values for usage record properties.

Managing Actions

Moab Accounting Manager defines which actions can be performed by which objects. When an object is first created, five basic actions are created for the object by default. These are: Create, Modify, Query, Delete and Undelete. Specific code must exist in Moab Accounting Manager modules in order for objects to support additional actions.

An action is uniquely specified by its name and the object with which it is associated. An action also has a description and a boolean display attribute which governs whether this action should be displayed in the web GUI or not.

Adding an Action to an Object

To specify that an action is allowed for an object, use the command `goldsh Action Create`:

```
goldsh Action Create Object=<Object Name> Name=<Action Name>
[Display=True| (False)] [Description=<Description>]
>ShowUsage:=True]
```

Example 23-17: Adding a Modify Action to Transaction

```
$ goldsh Action Create Object=Transaction Name=Modify Description=Modify
Successfully created 1 action
```

Querying Actions

To display action information, use the command `goldsh Action Query`:

```
goldsh Action Query [Object==<Object Name>] [Name==<Attribute
Name>] [Show:=Object,Name,Display,Description]
>ShowUsage:=True]
```

Example 23-18: List the actions of the Node object

```
$ goldsh Action Query Object==Node
Object Name      Display Description
-----
Node  Create     False   Create
Node  Delete     False   Delete
Node  Modify     False   Modify
Node  Query      False   Query
Node  Undelete   False   Undelete
```

Modifying an Action

To modify an action, use the command `goldsh Action Modify`:

```
goldsh Action Modify [Object==<Object Name>] [Name==<Attribute Name>] [Display=True|False] [Description=<Description>] [ShowUsage:=True]
```

- i** The goldsh control program allows you to make powerful and sweeping modifications to many objects with a single command. A mistake made using this command could result in the inadvertent modification of all actions.

Example 23-19: Display all Node actions but Undelete in the web GUI

```
$ goldsh Action Modify Object==Node Name!=Undelete Display=True  
Successfully modified 4 actions
```

Removing an Action From an Object

To delete an action from an object, use the command `goldsh Action Delete`:

```
goldsh Action Delete [Object==<Object Name>] [Name==<Attribute Name>] [ShowUsage:=True]
```

- i** The goldsh control program allows you to make powerful and sweeping modifications to many objects with a single command. A mistake made using this command could result in the inadvertent modification of all actions.



When using Moab Accounting Manager as an accounting manager, certain actions are assumed to exist. Be careful what you delete!

Example 23-20: Do not allow accounts to be deleted

```
$ goldsh Action Delete Object==Account Name==Delete  
Successfully deleted 1 action
```

Examples Creating Custom Objects

Creating a custom object normally involves defining a new object and adding attributes to the object.

Example 23-21: Creating a License object to track license usage and charges.

Invoke the Moab Accounting Manager control program in interactive mode.

```
$ goldsh
```

Create the License Object.

```
gold> Object Create Name=License Description=License
Successfully created 1 object and 5 actions
```

Next, define its attributes. Give each record a unique id (so the record can be more easily modified), a license type that can be one of (Matlab,Mathematica,Compiler,AutoCAD,Oracle), the user who is using it, the start and end time, how many instances of the license were used, and how much was charged.

```
gold> Attribute Create Object=License Name=Id DataType=AutoGen PrimaryKey=True
Description="Record Id"
Successfully created 1 attribute

gold> Attribute Create Object=License Name=Type DataType=String Required=True Values="
(Matlab,Mathematica,Compiler,AutoCAD,Oracle)" Fixed=True Description="License Type"
Successfully created 1 attribute

gold> Attribute Create Object=License Name=User Required=True Values="@User"
Description="User Name"
Successfully created 1 attribute

gold> Attribute Create Object=License Name=StartTime DataType=TimeStamp
Description="Start Time"
Successfully created 1 attribute

gold> Attribute Create Object=License Name=EndTime DataType=TimeStamp Description="End
Time"
Successfully created 1 attribute

gold> Attribute Create Object=License Name=Count DataType=Integer Description="Number
of Licenses Used"
Successfully created 1 attribute

gold> Attribute Create Object=License Name=Charge DataType=Currency
Description="Amount Charged"
Successfully created 1 attribute
```

Finally, since we would like to manage licenses from the web GUI, set `Display=True`.

```
gold> Action Modify Object==License Name!=Undelete Display=True
Successfully modified 4 actions
```

When done, exit the goldsh prompt.

```
gold> quit
```

That's about it. Licenses should now be able to be managed via the GUI and goldsh. The data source will need to use one of the methods of interacting with Moab Accounting Manager (see [Methods of Interacting with Moab Accounting Manager on page 146](#)) in order to push license record usage info to Moab Accounting Manager.

Apart from being used as an accounting manager, Moab Accounting Manager can be used as a generalized information service. It can be used to manage just about any object-oriented information over the web. For example, Moab Accounting Manager could be used to provide meta-schedulers with machine/user mappings, or node/resource information.

Example 23-22: Using Moab Accounting Manager as a Grid Map File.

Invoke the goldsh control program in interactive mode.

```
$ goldsh
```

Create the GridMap Object.

```
gold> Object Create Name=GridMap Description="Online Grid Map File"  
Successfully created 1 object and 5 actions
```

Next, define its attributes. Each entry will consist of a userid (which will serve as the primary key) and a required public X.509 certificate.

```
gold> Attribute Create Object=GridMap Name=User PrimaryKey=True Values=@User  
Description="User Name"  
Successfully created 1 attribute  
gold> Attribute Create Object=GridMap Name=Certificate DataType=String Required=True  
Description="X.509 Public Key"  
Successfully created 1 attribute
```

Exit the goldsh prompt.

```
gold> quit
```

From this point, a peer service will need to use one of the methods of interacting with Moab Accounting Manager (see [Interaction Methods](#)) in order to query the GridMap information.

Chapter 24 Integration

This chapter provides information on integrating with the resource management system (Moab Workload Manager) and Moab Web Services. It also contains methods for interacting with Moab Accounting Manager.

- [Integrating with Moab Workload Manager on page 143](#)
- [Integrating with Moab Web Services on page 145](#)
- [Methods of Interacting with Moab Accounting Manager on page 146](#)

Integrating with Moab Workload Manager

Moab Workload Manager can be configured to interact with Moab Accounting Manager to track and charge for resources utilized by jobs and reservations. You will need to use Moab HPC Suite — Enterprise Edition in order to have support for Moab Accounting Manager.

Select an Appropriate Accounting Management Interface Type

There are two accounting manager interface types that Moab can use to interact with Moab Accounting Manager: MAM, which makes direct calls to MAM over the SSS wire protocol, and Native, in which customizable scripts are invoked to communicate with Moab Accounting Manager. The MAM accounting manager interface is the default as it is usually faster. The Native accounting manager interface can be used if higher customizability is needed, or if you need to interface with a third party accounting or allocation system. See *Accounting Manager Interface Types in the Moab Workload Manager Administrator Guide* for more information. Choose the accounting manager interface type that is right for your needs and remember it. This information will be used in a later step.

Run Configure --with-am

It may be necessary or advantageous when installing Moab Workload Manager to run configure with certain accounting related options.

Configure Moab to use the Moab Accounting Manager by running ./configure with the applicable options when installing Moab:

- `--with-am[=TYPE]` - Enable accounting management with the specified accounting manager interface type (mam or native) [mam].
- `--with-am-dir=DIR` - Uses the specified prefix directory for the accounting manager if installed in a non-default location.

The --with-am option specifies the accounting manager interface type that you want to use as either mam, which is the default, or native. Specifying this option will add essential entries into Moab configuration files. Although these entries may be added manually later, this step facilitates configuration by adding parameters appropriate for your selected accounting manager interface type.

Use --with-am-dir to specify the prefix directory for Moab Accounting Manager if it has been installed in a non-default location. This value is used to help the native accounting manager scripts find the Moab Accounting Manager libraries and server connection information.

Example 24-1: Configuring Moab to Use the Direct Accounting Manager Interface

```
$ ./configure --with-am
```

Edit the Moab Server Configuration File

Add or uncomment the essential AMCFG lines in the moab.cfg file.

Example 24-2: Configuring Moab to Use the MAM Accounting Manager Interface

If you are using the direct (MAM) accounting manager interface, at a minimum, you must tell Moab to use AMCFG [] TYPE=MAM. Additionally, if your Moab Accounting Manager server is running on a different host than the Moab Workload Manager server, you must specify the hostname via the AMCFG[] HOST parameter.

```
$ vi /opt/moab/etc/moab.cfg
AMCFG[mam] TYPE=MAM HOST=localhost
```

Example 24-3: Configuring Moab to Use the Native Accounting Manager Interface

If you are using the script (Native) accounting manager interface, at a minimum, you must tell Moab to use AMCFG [] TYPE=NATIVE. Moab Workload Manager will default to using a set of stock scripts to interact with Moab Accounting Manager.

```
$ vi /opt/moab/etc/moab.cfg
AMCFG[mam] TYPE=NATIVE
```

Edit the Moab Private Configuration File

If you have chosen to use the direct MAM accounting manager interface type, you will need to configure Moab to have Moab Accounting Manager's symmetric key for secure authentication. This step is not necessary when using the Native accounting manager interface type since the secret key can be securely derived from Moab Accounting Manager and used via the connection libraries.

Example 24-4: Configuring Moab to Communicate Securely with Moab Accounting Manager

Add or uncomment a CLIENTCFG [AM:mam] KEY parameter line in moab-private.cfg. Copy the **token.value** parameter in /opt/mam/etc/site.conf into the KEY value in /opt/moab/etc/moab-private.cfg.

```
# vi /opt/moab/etc/moab-private.cfg
CLIENTCFG [AM:mam] KEY=UiW7EihzKyUyVQg6dKirDhV3
```

Restart Moab Workload Manager

Restart Moab in order for the configuration changes to take effect.

Example 24-5: Restarting Moab

```
# service moab restart
```

Integrating with Moab Web Services

Moab Web Services can be configured to interact with Moab Accounting Manager order to be able to perform RESTful web service queries against accounting objects in Moab Accounting Manager.

Edit the MWS HPC Configuration File

Uncomment and set the following parameters in /opt/mws/etc/mws.d/mws-config-hpc.groovy:

- **mam.secretKey** - Set to the value of the token.value parameter in /opt/mam/etc/site.conf
- **mam.server** - Set to the hostname of the MAM server
- **mam.port** - Set to the port of the MAM server (defaults to 7112)

Example 24-6: Configuring Moab Web Services to Communicate with MAM Accounting Manager

```
$ vi /opt/mws/etc/mws.d/mws-config-hpc.groovy

mam.secretKey = "UiW7EihzKyUyVQg6dKirDhV3"
mam.server = "localhost"
mam.port = 7112
```

Restart Moab Web Services

Restart tomact in order for the MWS configuration changes to take effect.

Example 24-7: Restarting Moab Web Services

```
# service tomcat6 restart
```

Methods of Interacting with Moab Accounting Manager

There are essentially five ways of interacting with Moab Accounting Manager. Let's consider a simple usage charge in each of the different ways.

Using the Appropriate Command-line Client

From inside a script, or by invoking a system command, you can use a command-line client (one of the "g" commands in the bin directory).

Example 24-8: To issue a charge at the completion of job usage, you could use gcharge:

```
gcharge -J Moab.1234 -a chemistry -u amy -m colony -P 2 -t 3600 -X Duration=3600
```

Using the Interactive Control Program

The interactive control program, goldsh, will issue a charge for a job expressed in xml.

Example 24-9: To issue a charge you must invoke the Charge action on the Job object:

```
goldsh UsageRecord Charge
Data:="<UsageRecord><Instance>Moab.1234</Instance><Account>chemistry</Account><User>amy</User><Machine>colony</Machine><Processors>2</Processors><Duration>3600</Duration></UsageRecord>" Duration:=3600
```

Use the Perl API

The Perl API exposes the full functionality of Moab Accounting Manager. The client commands can be examined as sample code. Use perldoc on the modules in lib/Gold for function documentation.

Example 24-10: To make a charge via this interface you might do something like:

```
use Gold;

my $request = new Gold::Request(object => "UsageRecord", action => "Charge");
my $usageRecord = new Gold::Datum("UsageRecord");
$usageRecord->addProperty("Instance", "Moab.1234");
$usageRecord->addProperty("Account", "chemistry");
$usageRecord->addProperty("User", "amy");
$usageRecord->addProperty("Machine", "colony");
$usageRecord->addProperty("Processors", "2");
$usageRecord->addProperty("Duration", "3600");
$request->setDatum($usageRecord);
$request->setOption("Duration", "3600");
my $response = $request->getResponse();
print $response->getStatus(), ": ", $response->getMessage(), "\n";
```

Use the Java API

Although deprecated, the Java API may still be usable to interact with Moab Accounting Manager. The javadoc command can be run on the contrib/java/gold directory to generate documentation for the gold java classes.

Example 24-11: To make a charge via this interface you might do something like:

```
import java.util.*;
import gold.*;

public class Test
{
    public static void main(String [] args) throws Exception
    {
        Gold.initialize();
        Request request = new Request("UsageRecord", "Charge");
        Datum usageRecord = new Datum("UsageRecord");
        usageRecord.addProperty("Instance", "Moab.1234");
        usageRecord.addProperty("Account", "chemistry");
        usageRecord.addProperty("User", "amy");
        usageRecord.addProperty("Machine", "colony");
        usageRecord.addProperty("Processors", "2");
        usageRecord.addProperty("Duration", "3600");
        request.setDatum(usageRecord);
        request.setOption("Duration", "3600");
        Response response = request.getResponse();
        System.out.println(response.getStatus() + ":" + response.getMessage() + "\n");
    }
}
```

Communicating via the SSSRMAP Protocol

Finally, it is possible to interact with Moab Accounting Manager by directly using the SSSRMAP Wire Protocol and Message Format over the network (see [SSS Resource Management and Accounting documentation](#)). This will entail building the request body in XML, appending an XML digital signature, combining these in an XML envelope framed in an HTTP POST, sending it to the server, and parsing the similarly formed response. The Moab Workload Manager communicates with Moab Accounting Manager via this method.

Example 24-12: The message might look something like:

```
POST /SSSRMAP HTTP/1.1
Content-Type: text/xml; charset="utf-8"
Transfer-Encoding: chunked

190
<?xml version="1.0" encoding="UTF-8"?>
<Envelope>
  <Body>
    <Request action="Charge" actor="scottmo">
      <Object>UsageRecord</Object>
      <Data>
        <UsageRecord>
          <Instance>Moab.1234</Instance>
          <Account>chemistry</Account>
          <User>amyh</User>
          <Machine>colony</Machine>
          <Processors>2</Processors>
          <Duration>3600</Duration>
        </UsageRecord>
      </Data>
      <Option name="Duration">3600</Option>
    </Request>
  </Body>
  <Signature>
    <DigestValue>azu4obZswzBt89OgATukBeLyt6Y=</DigestValue>
    <SignatureValue>YXE/C08XX3RX4PMU1bWju+5/E5M=</SignatureValue>
    <SecurityToken type="Symmetric"></SecurityToken>
  </Signature>
</Envelope>
0
```

Chapter 25 Configuration Files

Moab Accounting Manager uses four configuration files: one for the connection information (`site.conf`), one for the server (`goldd.conf`), one for the clients (`gold.conf`) and one for the graphical user interface (`goldg.conf`). For configuration parameters that have hard-coded defaults, the default value is specified within brackets.

Site Configuration

The site configuration file specifies the connection information for the current site such as the server host name, port, backup server, default security method and the symmetric key. Optionally, it may also have blocks that specify connection information for other sites. This file should be readable only by the accounting admin user.

Example 25-1: The following is an example `site.conf` file

```
server.host = red-head1
backup.host = red-head2
server.port = 7071
token.type = Symmetric
token.value = pBaIapJqbfLd8NiyzTJeffFXW

[white]
server.host = white-head1
server.port = 7071
token.value = F17wOkioUpyjdqJ8ckvWK_ta

[blue]
server.host = blue-head1
server.port = 7071
token.value = gVSeQ8Diz5O3pzj01y4inGWq
```

The following configuration parameters may be set in the site configuration file (`site.conf`).

backup.host — The hostname of the backup server. Each site can have both a primary server and a hot-standby backup server. They should either point to the same database or separate instances of replicated database. If `backup.host` is specified, clients will try communicating with the primary server first, and if the connection fails, they will try communicating with the backup server.

server.host — The hostname of the server

server.port [7112] — The port that the server listens on

token-type [Symmetric] — Indicates the default security token type to be used in both authentication and encryption. Valid token types include `Password` and `Symmetric`. The default is `Symmetric`.

token.value — When using the `Symmetric` token type, `token.value` is the secret key. It is a base64-encoded symmetric key used between clients and the server for authentication and encryption.

Server Configuration

The following configuration parameters may be set in the server configuration file (`goldd.conf`).

Parameter	Description
accounting.mode [strict-allocation]	The accounting mode can be one of <code>usage-tracking</code> , <code>notional-charging</code> , <code>fast-allocation</code> , or <code>strict-allocation</code> . If <code>usage-tracking</code> is specified, charges will simply result in the creation of usage records with no charge value. No charge will be calculated and allocations will not be debited. If <code>notional-charging</code> is specified, a charge will be calculated and recorded with the usage record, but allocations are not debited. If <code>fast-allocation</code> is specified, usage records will be updated with charge amounts and allocations will be debited, but liens will not be used to protect the allocation from simultaneous use. If <code>strict-allocation</code> is specified, usage records will be updated with charge amounts and allocations will be debited, and liens will be used to protect the allocation from simultaneous use.
currency.itemization [false]	Enables (<code>true</code>) or disables (<code>false</code>) the storing of itemized charges to the Charge table for charge transactions.
currency.precision [0]	Indicates the number of decimal places in the resource credit currency. For example, if you will be dealing with an integer billable unit like processor-seconds, use 0 (which is the default). If you will be charging dollars and cents, then use 2. This parameter should be the same in the <code>goldd.conf</code> and <code>gold.conf</code> files.
database.datasource [DBI:Pg:dbname=mam;host=localhost]	The Perl DBI data source name for the database you wish to connect to
database.password	The password to be used for the database connection (if any)
database.user	The username to be used for the database connection (if any)
event.scheduler [false]	Specifies whether the event scheduler is enabled (<code>true</code>) or not (<code>false</code>)

Parameter	Description
event.pollinterval [5]	The period in minutes that the event scheduler uses to check and fire events. The poll interval must divide evenly into the number of minutes in a day (1440).
log4perl.appenders.Log.filename	Used by log4perl to set the base name of the log file
log4perl.appenders.Log.max	Used by log4perl to set the number of rolling backup logs
log4perl.appenders.Log.size	Used by log4perl to set the size the log will grow to before it is rotated
log4perl.appenders.Log.Threshold	Used by log4perl to set the debug level written to the log. The logging threshold can be one of TRACE, DEBUG, INFO, WARN, ERROR, and FATAL
log4perl.appenders.Screen.Threshold	Used by log4perl to set the debug level written to the screen. The logging threshold can be one of TRACE, DEBUG, INFO, WARN, ERROR, and FATAL
notification.deliverymethod [store]	Specifies which delivery method is used by default if unspecified
notification.duration [1209600]	Defines how long in seconds that stored notifications persist before being automatically deleted. The default is two weeks.
response.chunksize [0]	Indicates the line length in the data response that will trigger message segmentation (or truncation). A value of 0 (zero) means unlimited, i.e. that the server will not truncate or segment large responses unless overridden by a chunksize specification in a client request. The response chunksize will be taken to be the smaller of the client and server chunksize settings.
security.authentication [true]	Indicates whether incoming message authentication is required
security.encryption [false]	Indicates whether incoming message encryption is required
super.user [root]	The primary Moab Accounting Manager system admin which by default can perform all actions on all objects. The super user is sometimes used as the actor in cases where an action is invoked from within another action.
user.firstaccountdefault [true]	If set to true, the first account that a user is added to will become the default account for that user. This default value is true.

Client Configuration

The following configuration parameters may be set in the client configuration file (`gold.conf`):

Parameter	Description
account.show [Name,Active,Users,Organization,Description]	The default fields shown by <code>glsaccount</code>
accounting.context [hpc]	By specifying the accounting context (either <code>hpc</code> or <code>cloud</code>), the behavior of some client commands are adjusted to show the proper fields for that context. The default is <code>hpc</code> .
accounting.mode [strict-allocation]	The accounting mode can be one of <code>usage-tracking</code> , <code>notional-charging</code> , <code>fast-allocation</code> , or <code>strict-allocation</code> . The value of this parameter may modify the default fields displayed by certain commands such as <code>glsusage</code> .
allocation.enforcediscrete [true]	If enabled (the default), new allocations will be prevented from overlapping existing ones. This policy helps to improve clarity when reporting on allocation usage during a particular period.
allocation.show [Id,Fund,StartTime,EndTime,InitialDeposit,Allocated,CreditLimit,Remaining,PercentUsed]	The default fields shown by <code>glsalloc</code>
balance.show [Id,Name,Balance,Reserved,Effective,CreditLimit,Available]	The default fields shown by <code>gbalance</code>
currency.precision [0]	Indicates the number of decimal places in the credit currency. For example, if you will be dealing with integer billable units like processor-seconds, use 0 (which is the default). If you will be charging dollars and cents, then use 2. This parameter should be the same in the <code>goldd.conf</code> and <code>gold.conf</code> files.

Parameter	Description
event.show	[Id,FireCommand,FireTime,ArmTime,RearmPeriod,EndTime,Notify,RearmOnFailure,FailureCommand,CatchUp,CreationTime,Description] -- The default fields shown by <code>glsevent</code>
fund.show [Id,Name,Constraints,Allocated,Balance,DefaultDeposit,Description]	The default fields shown by <code>glsfund</code>
lien.show [Id,Instance,Mount,StartTime,EndTime,UsageRecord,Funds,Description]	The default fields shown by <code>glslien</code>
log4perl.appendер.Log.filename	Used by log4perl to set the base name of the log file
log4perl.appendер.Log.max	Used by log4perl to set the number of rolling backup logs
log4perl.appendер.Log.size	Used by log4perl to set the size the log will grow to before it is rotated
log4perl.appendер.Log.Threshold	Used by log4perl to set the debug level written to the log. The logging threshold can be one of TRACE, DEBUG, INFO, WARN, ERROR, and FATAL.
log4perl.appendер.Screen.Threshold	Used by log4perl to set the debug level written to the screen. The logging threshold can be one of TRACE, DEBUG, INFO, WARN, ERROR, and FATAL.
notification.show	[Id,Event,Type,Status,Code,Message,Key,Recipient,EndTime,CreationTime] --The default fields shown by <code>glsnot</code>
organization.show [Name,Description]	The default fields shown by <code>glsorg</code>
quote.show [Id,Mount,Pinned,Instance,UsageRecord,StartTime,EndTime,Duration,Char	The default fields shown by <code>glsquote</code>

Parameter	Description
response.chunking [false]	Indicates whether large responses should be chunked (segmented). If set to false, large responses will be truncated
response.chunksize [1000]	Indicates the line length in the data response that will trigger message segmentation (or truncation). A value of 0 (zero) means unlimited, i.e., that the client will accept the chunksize set by the server. The response chunksize will be taken to be the smaller of the client and server chunksize settings.
security.authentication [true]	Indicates whether outgoing message are signed
security.encryption [false]	Indicates whether outgoing messages are encrypted

Parameter	Description
security.promotion [suidperl]	When using the symmetric key for security authentication or encryption, since the <code>site.conf</code> file is readable only by the accounting admin user, a method must be employed to temporarily elevate privileges in order to encrypt the communication with the symmetric key. One of two security promotion methods may be selected: <code>suidperl</code> or <code>gauth</code> . Suidperl allows a Perl script to temporarily elevate privileges to the owner of the script if the setuid bit is set on the file. This method is recommended when <code>suidperl</code> can be installed on the system. If you prefer not to use <code>suidperl</code> or if it is not available for your system (such as with Perl 5.12 and higher), you will need to use the <code>gauth</code> security promotion method. Gauth is a setuid binary that allows the request body to be passed in as standard input and returns the authenticated digest and signature. Currently, only <code>suidperl</code> can be used for encryption of client communication. The security promotion method should be configured at install time by specifying the <code>--with-promotion</code> configuration parameter and defaults to <code>suidperl</code> when it is available.
statement.show [Account,User,Machine]	The default discriminator fields in <code>gstatement</code>
transaction.show [Id, Object, Action, Actor, Name, Child, Instance, Count, Amount, Delta, Balance, User]	The default fields shown by <code>glstrans</code>
usagerecord.show [Id, Type, Instance, Charge, Stage, Quote, User, Group, Account, Organization, Class, QualityOfService, Machine, Node]	The default fields shown by <code>glsusage</code>
user.show [Name, Active, CommonName, PhoneNumber, EmailAddress, DefaultAccount, Description]	The default fields shown by <code>gluser</code>

GUI Configuration

The following configuration parameters may be set in the GUI configuration file (`goldg.conf`).

Parameter	Description
currency.enablehours [false]	If set to true, the graphical user interface will include a ShowHours radio button (defaulting to <code>True</code>) for certain panels (e.g. Fund Deposit, Query, Statement, Transfer, Withdraw) that will allow the currency inputs or outputs to be divided by 3600.
currency.precision [0]	Indicates the number of decimal places in the credit currency. For example, if you will be dealing with integer billable units like processor-seconds, use 0 (which is the default). If you will be charging dollars and cents, then use 2. This parameter should be the same in the <code>goldd.conf</code> and <code>gold.-conf</code> files.
gui.style [viewpoint]	Modifies the appearance and behavior of the web GUI to be consistent with use within viewpoint or for standalone use. Valid values are <code>legacy</code> or <code>viewpoint</code> . The default is <code>viewpoint</code> .
log4perl.appendер.Log.filename	Used by log4perl to set the base name of the log file
log4perl.appendер.Log.max	Used by log4perl to set the number of rolling backup logs
log4perl.appendер.Log.size	Used by log4perl to set the size the log will grow to before it is rotated
log4perl.appendер.Log.Threshold	Used by log4perl to set the debug level written to the log. The logging threshold can be one of TRACE, DEBUG, INFO, WARN, ERROR, and FATAL.
response.chunking [false]	Indicates whether large responses should be chunked (segmented). If set to false, large responses will be truncated.
response.chunksize [1000]	Indicates the line length in the data response that will trigger message segmentation (or truncation). A value of 0 (zero) means unlimited, i.e. that the client will accept the chunksize set by the server. The response chunksize will be taken to be the smaller of the client and server chunksize settings.
security.authentication [true]	Indicates whether outgoing message are signed
security.encryption [false]	Indicates whether outgoing messages are encrypted

Parameter	Description
security.promotion [suidperl]	When using the symmetric key for security authentication or encryption, since the <code>site.conf</code> file is readable only by the accounting admin user, a method must be employed to temporarily elevate privileges in order to encrypt the communication with the symmetric key. One of two security promotion methods may be selected: <code>suidperl</code> or <code>gauth</code> . Suidperl allows a Perl script to temporarily elevate privileges to the owner of the script if the setuid bit is set on the file. This method is recommended when <code>suidperl</code> can be installed on the system. If you prefer not to use <code>suidperl</code> , or if it is not available for your system (such as with Perl 5.12 and higher), you will need to use the <code>gauth</code> security promotion method. Gauth is a setuid binary that allows the request body to be passed in as standard input and returns the authenticated digest and signature. Currently, only <code>suidperl</code> can be used for encryption of client communication. The security promotion method should be configured at install time by specifying the <code>--with-promotion</code> configuration parameter and defaults to <code>suidperl</code> when it is available.
statement.discriminators	The Fund Statement page will group summary entries in the debit detail by these transaction properties.

Appendix A Commands Reference

Moab Accounting Manager provides a server daemon and client commands for use by administrators and end users.

Common Command Options

Most Moab Accounting Manager commands support the following common options.

Option	Description
--help	Brief command option summary
--man	Full command documentation
--site	Obtain response from specified site
--version	Display product version

List of Commands

Command	Description
gbalance	Display balance information
gchaccount	Modify an account
gchalloc	Modify an allocation
gcharge	Create a usage charge
gchevent	Modify an event
gchfund	Modify a fund
gchlien	Modify a lien
gchorg	Modify an organization

Command	Description
gchpasswd	Set a user password
gchquote	Modify a quote
gchrate	Modify a charge rate
gchrole	Modify a role
gchusage	Modify a usage record
gchuser	Modify a user
gdeposit	Issue a deposit
glsaccount	Query accounts
glsalloc	Query allocations
glscharge	Query charges
glsconfig	Query configuration
glsevent	Query events
glsfund	Query funds
glslien	Query liens
glsnot	Query stored notifications
glsorg	Query organizations
glsquote	Query quotes
glsrate	Query charge rates
glsrole	Query roles
glstrans	Query transactions

Command	Description
<u>glsusage</u>	Query usage records
<u>glsuser</u>	Query users
<u>gmkaccount</u>	Create a new account
<u>gmkevent</u>	Create a new event
<u>gmkfund</u>	Create a new fund
<u>gmklien</u>	Create a lien
<u>gmkorg</u>	Create a new organization
<u>gmkquote</u>	Create a quote template
<u>gmkrate</u>	Create a new charge rate
<u>gmkrole</u>	Create a new role
<u>gmkusage</u>	Create a new usage record
<u>gmkuser</u>	Create a new user
<u>gquote</u>	Quote for usage
<u>grefund</u>	Issue a usage refund
<u>greserve</u>	Reserve for usage
<u>grmaccount</u>	Delete an account
<u>grmalloc</u>	Delete an allocation or purge stale allocations
<u>grmevent</u>	Delete an event
<u>grmfund</u>	Delete a fund
<u>grmlien</u>	Delete a lien

Command	Description
grmnot	Delete a stored notification
grmorg	Delete an organization
grmquote	Delete a quote
grmrate	Delete a charge rate
grmrole	Delete a role
grmusage	Delete a usage record
grmuser	Delete a user
gstatement	Display fund statement
gtransfer	Issue a transfer
gwithdraw	Issue a withdrawal
goldsh	Interactive shell for gold
goldd	Moab Accounting Manager server
mybalance	Display personal balance information

gbalance

Synopsis

```
gbalance [-u <user_name>] [-q <group_name>] [-a <account_name>]
[-o <organization_name>] [-c <class_name>] [-m <machine_name>]
[--filter <filter_name>=<filter_value>] ... [--filter-type
<filter_type>] [--ignore-ancestors] [--full] [--show
<attribute_name>, ...] [--long] [--wide] [--format <output_
type>] [-h, --hours] [--debug] [--site <site_name>] [--help]
[--man] [--quiet] [--version]
```

Overview

`gbalance` displays balance information for funds having active allocations.

Options

-a	
Format	<code>-a <account_name></code>
Default	---
Description	Displays the balance available to the specified account.
-c	
Format	<code>-c <class_name></code>
Default	---
Description	Displays the balance available to the specified class.
-g	
Format	<code>-g <group_name></code>
Default	---
Description	Displays the balance available to the specified group.
-h, --hours	
Format	<code>-h</code> or <code>--hours</code>
Default	---
Description	Displays time-based credits in hours. In cases where the currency is measured in resource-seconds (like processor-seconds), the currency is divided by 3600 to display resource-hours.

-m

Format	<code>-m <machine_name></code>
Default	---
Description	Display the s balance available to the specified machine.

-o

Format	<code>-o <organization_name></code>
Default	---
Description	Displays the balance available to the specified organization.

-u

Format	<code>-u <user_name></code>
Default	---
Description	Displays the balance available to the specified user.

--debug

Format	<code>--debug</code>
Default	---
Description	Logs debugging information to the screen.

--filter

Format	<code>--filter <filter_name>=<filter_value></code>
Default	---

--filter**Description**

Displays the balance for funds where constraints do not conflict with the specified filters. You can use multiple filter options by logically ANDing them together.

--filter-type**Format**

`--filter-type <filter_type>`

Default

NonExclusive

Description

Selects the filtering type. If you use the exclusive filter type, a fund will only be matched if the specified filters meet all constraints. If you use the non-exclusive filter type, a fund will be matched as long as the specified filters do not conflict with the constraints.

--format**Format**

`--format <output_type>`

Default

standard

Description

Data output format. Valid values include standard and csv.

--full**Format**

`--full`

Default

Description

Displays all attributes.

--help**Format**

`--help`

Default

Description

Displays a brief help message.

--ignore-ancestors

Format	--ignore-ancestors
Default	---
Description	Does not include hierarchical ancestor funds in the result.

--long

Format	--long
Default	---
Description	Displays multi-valued fields in a multi-line format.

--man

Format	--man
Default	---
Description	Displays the full documentation.

--quiet

Format	--quiet
Default	---
Description	Suppresses headers and success messages.

--show

Format	--show <attribute_name>[,<attribute_name>...]
Default	---

--show**Description**

Valid attributes include the following:

- **Allocated** – Adjusted allocation. This value stores the effective allocated amount based on the initial deposit and subsequent allocation adjustments via deposits, withdrawals or transfers.
- **Available** – Total amount currently available for charging (Balance - Reserved + CreditLimit).
- **Balance** – Sum of active allocation amounts remaining within this fund. It does not take into account current liens.
- **Capacity** – Total expendable amount (Allocated + CreditLimit).
- **Constraints** – Constraints on fund usage.
- **CreationTime** – Time this fund was created.
- **CreditLimit** – Sum of active credit limits within this fund.
- **Deleted** – Boolean indicating whether this fund is deleted.
- **Description** – Fund description
- **Effective** – Effective allocation total not blocked by liens (Balance - Reserved).
- **Id** – Fund id.
- **ModificationTime** – Time this fund was last modified.
- **Name** – Fund name.
- **PercentRemaining** – Percentage of allocation remaining (Balance * 100 / Capacity).
- **PercentUsed** – Percentage of allocation used (Used * 100 / Capacity).
- **RequestId** – Id of the last modifying request.
- **Reserved** – Sum of active lien amounts against this fund.
- **TransactionId** – Id of the last modifying transaction.
- **Used** – Total amount used from this allocation (Allocated - Balance).

--site**Format**

--site <site_name>

Default

Description

Obtains a response from specified site.

--wide**Format**

--wide

Default

Description

Displays multi-valued fields in a single-line, comma-separated format.

--version	
Format	--version
Default	---
Description	Displays the product version.

Related Topics

[Querying the Balance on page 62](#)

[Commands Reference on page 159](#)

gchaccount

Synopsis

```
gchaccount [-A | -I] [-o <organization_name>] [-d <description>]
[-X, --extension <property>=<value>]... [--add-user(s) [^|!]
[+|-]<user_name>, ...]... [--del-user(s) <user_name>, ...]... [-
-mod-user(s) [^|!][+|-]<user_name>, ...]... [--debug] [--site
<site_name>] [--help] [--man] [--quiet] [--verbose] [--
version] { [-a] <account_name>}
```

Overview

gchaccount modifies an account.

Options

-a	
Format	[-a] <account_name>
Default	---
Description	Specifies the name of the account to modify.

-A	
Format	<code>-A</code>
Default	---
Description	Activates the account.

-d	
Format	<code>-d <description></code>
Default	---
Description	Modifies the account description.

-I	
Format	<code>-I</code>
Default	---
Description	Deactivates the account.

-O	
Format	<code>-o <organization_name></code>
Default	---
Description	Modifies the name of the organization to which the account belongs.

-X, --extension	
Format	<code>-X or --extension <property>=<value></code>
Default	---
Description	Modifies an extension property. You can specify any number of extra field assignments.

--add-user**Format**

```
--add-user [^|!] [+|-] <user_name>[, [^|!] [+|-] <user_name>...]
```

Default

Description

Adds user members of the account. The optional caret or exclamation symbol indicates whether the user should be created as an administrator (^) or not (!) for the account. The optional plus or minus signs can precede each member to indicate whether the member should be created in the active (+) or inactive (-) state. By default, a user will be created in the active state but not an administrator. You can pass multiple users to the --add-user option in a comma-delimited list or you can specify multiple --add-user options.

--debug**Format**

```
--debug
```

Default

Description

Logs debug information to the screen.

--del-user**Format**

```
--del-user <user_name>[, <user_name>...]
```

Default

Description

Removes user members from the account. You can pass multiple users to the --del-user option in a comma-delimited list or specify multiple --del-user options.

--help**Format**

```
--help
```

Default

Description

Displays a brief help message.

--man	
Format	--man
Default	---
Description	Displays full documentation.

--mod-user	
Format	--mod-user [^ !] [+ -] <user_name>[, [^ !] [+ -] <user_name>...]
Default	---
Description	Modifies user members of the account. The caret symbol or exclamation symbol indicates the user should be changed to become an administrator (^) or not (!) for the account. The plus or minus signs indicate whether the user should be changed to become active (+) or inactive (-). If you do not specify an active or admin modifier, that aspect of the user member will remain unchanged. You can pass multiple users to the --mod-user option in a comma-delimited list or you can specify multiple --mod-user options.

--quiet	
Format	--quiet
Default	---
Description	Suppresses headers and success messages.

--site	
Format	--site <site_name>
Default	---
Description	Obtains a response from specified site.

--verbose**Format** --verbose**Default** ---**Description** Displays modified object details.**--version****Format** --version**Default** ---**Description** Displays the product version.

Related Topics

[Modifying Accounts on page 49](#)[Commands Reference on page 159](#)

gchalloc

Synopsis

```
gchalloc [-s <start_time>] [-e <end_time>] [-L <credit_limit>] [-d <description>] [-X, --extension <property>=<value>]... [-h, --hours] [--debug] [--site <site_name>] [--help] [--man] [--quiet] [--verbose] [--version] {[ -i] <allocation_id>}
```

Overview

`gchalloc` modifies an allocation. This includes changing the credit limit or description or adjusting the start time or end time.

Options

-d**Format** -d <description>

-d	
Default	---
Description	Modifies the allocation description.

-e	
Format	<code>-e <end_time></code>
Default	---
Description	Specifies a new end time in the format YYYY-MM-DD [hh:mm:ss] -Infinity Infinity Now

-h, --hours	
Format	<code>-h</code> or <code>--hours</code>
Default	---
Description	Treats currency as specified in hours. In cases where the currency is measured in resource-seconds (like processor-seconds), this option allows you to specify the credit limit in resource hours.

-i	
Format	<code>[-i]<allocation_id></code>
Default	---
Description	The ID of the allocation to modify.

-L	
Format	<code>-L <credit_limit></code>
Default	---
Description	Specifies a new credit limit.

-s**Format**`-s <start_time>`**Default**`---`**Description**

Specifies a new start time in the format YYYY-MM-DD[hh:mm:ss] | -Infinity|Infinity|Now

-X, --extension <property>**Format**`-X or --extension <property>=<value>`**Default**`---`**Description**

Modifies an extension property. You can specify any number of extra field assignments.

--debug**Format**`--debug`**Default**`---`**Description**

Logs debug information to the screen.

--help**Format**`--help`**Default**`---`**Description**

Displays a brief help message.

--man**Format**`--man`**Default**`---`

--man	
Description	Displays full documentation.

--quiet	
Format	--quiet
Default	---
Description	Suppresses headers and success messages.

--site	
Format	--site <site_name>
Default	---
Description	Obtains a response from specified site.

--verbose	
Format	--verbose
Default	---
Description	Displays modified object details.

--version	
Format	--version
Default	---
Description	Displays the product version.

Related Topics

- [Modifying Allocations on page 72](#)
- [Commands Reference on page 159](#)

gcharge

Synopsis

```
gcharge [-T <usage_record_type>] [-S <service_id>] [-u <user_name>] [-g <group_name>] [-a <account_name>] [-o <organization_name>] [-c <class_name>] [-Q <quality_of_service>] [-m <machine_name>] [-N <nodes>] [-P <processors>] [-C <cpu_time>] [-M <memory>] [-D <disk>] [-x <usage_state>] [-X, --extension <property>=<value>] ... [-t <charge_duration>] [-s <charge_start_time>] [-e <charge_end_time>] [-d <charge_description>] [-z <charge_amount>] [-f <fund_id>] [--incremental] [-R <charge_rate_name>[<charge_rate_value>]] =<charge_rate_amount>, ...] ... [-h, --hours] [--itemize] [--debug] [--site <site_name>] [--help] [--man] [--quiet] [--verbose] [--version] [[-j] <usage_record_id>] [-q <quote_id>] [-l <lien_id>] {-J <instance_name>} <job_id>}
```

Overview

gcharge charges for resource or service usage.

Options

-a

Format -a <account_name>

Default ---

Description Specifies the account to charge.

-c

Format -c <class_name>

Default ---

Description Specifies the class of queue used.

-C	
Format	<code>-C <cpu_time></code>
Default	---
Description	Specifies the CPU time used.

-d	
Format	<code>-d <charge_description></code>
Default	---
Description	Specifies an explanatory message for the charge. You can pass the overall usage description via the extension property option (<code>-X Description=<description></code>).

-D	
Format	<code>-D <disk></code>
Default	---
Description	Specifies the amount of disk space used.

-e	
Format	<code>-e <charge_end_time></code>
Default	Now
Description	Specifies an end time for the charge in the format <code>YYYY-MM-DD [hh:mm:ss] -Infinity Infinity Now</code> You can pass the overall usage end time via the extension property option (<code>-X EndTime=<end_time></code>).

-f	
Format	<code>-f <fund_id></code>

-f	
Default	---
Description	Specifies the fund ID to charge.
-g	
Format	<code>-g <group_name></code>
Default	---
Description	Specifies the name of the group to charge.
-h, --hours	
Format	<code>-h</code> or <code>--hours</code>
Default	---
Description	Displays time-based credits in hours. In cases where the currency is measured in resource-seconds (like processor-seconds), the currency is divided by 3600 to display resource-hours.
-j	
Format	<code>[-j] <usage_record_id></code>
Default	---
Description	Specifies the usage record id for the charge (if already created with <code>gmkusage</code> , <code>gquote</code> , <code>greserve</code> or a previous <code>gcharge</code>). Use <code>-j</code> to charge an existing usage record if the instance name (such as a job ID) is ambiguous or if a usage has already been debited and you want to charge an additional amount to the same usage record.
-J	
Format	<code>-J <instance_name> <job_id></code>
Default	---

-J**Description**

Specifies the instance name or job ID for the charge if known. This can sometimes be non-unique, such as when a resource manager recycles job IDs, and does not always unambiguously identify a usage record to charge. In such cases, look up and specify the usage record ID for the charge.

-l**Format**

`-l <lien_id>`

Default

Description

Specifies the lien ID, which MAM will use to match up the right usage record ID and remove the correct lien, if ambiguous.

-m**Format**

`-m <machine_name>`

Default

Description

Specifies the name of the cluster, cloud, or system used.

-M**Format**

`-M <memory>`

Default

Description

Specifies the amount of memory used.

-N**Format**

`-N <nodes>`

Default

Description

Specifies the number of nodes used.

-o

Format	<code>-o <organization_name></code>
Default	---
Description	Specifies the organization name.

-P

Format	<code>-P <processors></code>
Default	---
Description	Specifies the number of processors used.

-q

Format	<code>-q <quote_id></code>
Default	---
Description	Specifies the quote MAM should use to determine charge rates.

-Q

Format	<code>-Q <quality_of_service_name></code>
Default	---
Description	Specifies the quality of service used.

-R

Format	<code>-R <charge_rate_name>[<charge_rate_value>]=<charge_rate_amount> [,<charge_rate_name>[<charge_rate_value>]]=<charge_rate_amount>...]</code>
Default	---

-R**Description**

Specifies the charge rates in the charge. The specified rates override the general rates or rates guaranteed through a quote. You can pass multiple charge rates -R option in a comma-delimited list or specify multiple -R options.

-S**Format**

`-s <charge_start_time>`

Default

`Now - <duration>`

Description

Specifies a start time for the charge in the format `YYYY-MM-DD [hh:mm:ss] | -Infinity | Infinity | Now`

You can pass the overall usage end time via the extension property option (`-X StartTime=<start_time>`).

-S**Format**

`-S <service_id>`

Default

`---`

Description

Specifies the root service ID (cloud).

-t**Format**

`-t <charge_duration>`

Default

`---`

Description

Specifies the wallclock duration for the charge in seconds. You can pass the total wallclock time for the lifetime usage via the extension property option (`-X Duration=<duration_in_seconds>`).

-T**Format**

`-T <usage_record_type>`

-T	
Default	---
Description	Specifies the usage record type (Job, Reservation, etc.).
-u	
Format	<code>-u <user_name></code>
Default	---
Description	Specifies the user name.
-x	
Format	<code>-x <usage_state></code>
Default	---
Description	Specifies the final status of the usage.
-X, --extension	
Format	<code>-X or --extension <property>=<value></code>
Default	---
Description	Specifies an extension property. You can specify any number of extra field assignments.
-z	
Format	<code>-z <charge_amount></code>
Default	---
Description	Specifies the charge amount if calculated externally.

--debug	
Format	--debug
Default	---
Description	Logs debugging information to the screen.

--help	
Format	--help
Default	---
Description	Displays a brief help message.

--incremental	
Format	--incremental
Default	---
Description	Debits any associated liens instead of removing them.

--itemize	
Format	--itemize
Default	---
Description	Returns the composite charge information in the response data. This must be used in conjunction with the -verbose flag to display the data.

--man	
Format	--man
Default	---

--man

Description	Displays the full documentation.
-------------	----------------------------------

--quiet

Format	--quiet
Default	---
Description	Suppresses headers and success messages.

--site

Format	--site <site_name>
Default	---
Description	Obtains a response from specified site.

--verbose

Format	--verbose
Default	---
Description	Displays modified object details.

--version

Format	--version
Default	---
Description	Displays the product version.

Related Topics

[Charging for Usage on page 87](#)[Commands Reference on page 159](#)

gchevent

Synopsis

```
gchevent [--fire-command <fire_command>] [-s <fire_time>] [-e
<end_time>] [--rearm-period <rearm_period>] [--rearm-on-
failure <boolean>] [--failure-command <failure_command>] [--notify
<notification_url>] [--catch-up <boolean>] [-d
<description>] [--debug] [--site <site_name>] [--help] [--man]
[--quiet] [--verbose] [--version] {[-E] <event_id>}
```

Overview

gchevent modifies an event.

Options

-d	
Format	<code>-d <description></code>
Default	---
Description	Specifies a new description.

-e	
Format	<code>-e <end_time></code>
Default	---
Description	Specifies the time this event becomes inactive.

-E	
Format	<code>-E <event_id></code>
Default	---
Description	Specifies the ID of the event to modify.

-s	
Format	<code>-s <fire_time></code>
Default	---
Description	Specifies a new target time for the event to be triggered by the event scheduler. The actual fire time may be dependent on the state of the server and will be recorded in the CreationTime property of the corresponding "Event Fire" Transaction. An event may also be fired manually with the goldsh Event Fire action.
--catch-up	
Format	<code>--catch-up <boolean></code>
Default	True
Description	If you set the --catch-up boolean to True and the server was down during the time this event should have fired, the event scheduler will attempt to make up for the past due events by progressively firing them (rearming based on previous arm time) until it catches up to the present. The actions will still show as having occurred in the present rather than in the past. If set to False and the server is brought back up after an outage, the event scheduler will still fire immediately for a past due event, but it will only fire once and then rearm relative to the current time.
-debug	
Format	<code>--debug</code>
Default	---
Description	Logs debugging information to the screen.
--failure-command	
Format	<code>--failure-command <goldsh_command></code>
Default	---
Description	Specifies a new command MAM should execute if the fired command results in an unsuccessful response status. This command is expressed in a serialized form of the request identical to the syntax used in the interactive control program (goldsh). The option argument will need to be appropriately quoted and/or escaped in order to avoid misinterpretation or alteration by the shell.

--help	
Format	--help
Default	---
Description	Displays a brief help message.

--man	
Format	--man
Default	---
Description	Displays the full documentation.

--notify	
Format	--notify [+--] [<delivery_method>:] [recipient]
Default	Logs all event statuses to the Notification table.
Description	A Notification method logs the result of the fired command. If the term is a -, the notification is sent only on failure. If the term is a +, the notification is sent only on success. Otherwise the notification is always sent. See Managing Notifications on page 109 for more information about delivery method and recipient.

--quiet	
Format	--quiet
Default	---
Description	Suppresses headers and success messages.

--rearm-on-failure	
Format	--rearm-on-failure <boolean>

--rearm-on-failure

Default	False
Description	If you set the --rearm-on-failure boolean to False, MAM will not rearm the event if the command was unsuccessful. If you set it to True, MAM will evaluate the event for rearming even if the command response has a status of Failure.

--rearm-period

Format	--rearm-period <period>[@instant] [~ ^] !]
Default	---
Description	The --rearm-period is a time period expression specifying when MAM will rearm the event. This period expression is of the form: <period>[@instant] [~ ^] !] where <period> may be something like 1 day, 2 hours, or 5 minutes. Instant locks the period to a specific instant within the time period such as 1 day @ hour 12 or 1 month @ day 3. The modifiers indicate whether the time period should be relative to now (!), or relative to the start of this (~) designator (month, minute, or other unit), or relative to the start of the first (^) designator (month, minute, or other unit). For example, assuming the fire time was 7:15, if you specified 4 hours ! as the rearm period, MAM would rearm it at 11:15; if you specified 4 hours ~ as the rearm period, MAM would rearm it at 11:00; and if you specified 4 hours ^ as the rearm period, MAM would rearm it at 8:00.

--site

Format	--site <site_name>
Default	---
Description	Obtains a response from specified site.

--verbose

Format	--verbose
Default	---
Description	Displays modified object details.

--version	
Format	--version
Default	---
Description	Displays the product version.

Related Topics

[Modifying Events on page 107](#)

[Commands Reference on page 159](#)

gchfund

Synopsis

```
gchfund { { [-n <fund_name>] [--priority <fund_priority>] [--default-deposit <deposit_amount>] [-d <description>] [-x, --extension <property>=<value>]... [--add-constraint <constraint_name>=[!]<constraint_value>,...]... [--del-constraint(s) <constraint_name>[=<constraint_value>],...]... [--parent <parent_fund_id>] } | { --reset [--all] } } [-u <user_name>] [-q <group_name>] [-a <account_name>] [-o <organization_name>] [-c <class_name>] [-m <machine_name>] [--filter <filter_name>=<filter_value>]... [--filter-type <filter_type>] [--debug] [--site <site_name>] [--help] [--man] [--quiet] [--verbose] [--version] [[-f] <fund_id>]
```

Overview

gchfund modifies a fund. This includes adding to or deleting from constraints for the account. After applying all filter options, if there is exactly one applicable fund, that fund will be modified. Otherwise, a list of funds will be displayed for the specified filters and you will be prompted to rerun gchfund against one of the enumerated funds.

Options

-a	
Format	-a <account_name>

-a

Default	---
Description	Specifies that the fund to modify should be restricted to one usable by the given account.

-c

Format	<code>-c <class_name></code>
Default	---
Description	Specifies that the fund to modify should be restricted to one usable by the given class.

-d

Format	<code>-d <description></code>
Default	---
Description	Specifies a new description

-f

Format	<code>[-f] <fund_id></code>
Default	---
Description	Specifies the ID of the fund to modify.

-g

Format	<code>-g <group_name></code>
Default	---
Description	Specifies that the fund to modify should be restricted to one usable by the given group.

-m	
Format	<code>-m <machine_name></code>
Default	---
Description	Specifies that the fund to modify should be restricted to one usable by the given machine.

-n	
Format	<code>-n <fund_name></code>
Default	---
Description	Specifies a new fund name.

-o	
Format	<code>-o <organization_name></code>
Default	---
Description	Specifies that the fund to modify should be restricted to one usable by the given organization.

-u	
Format	<code>-u <user_name></code>
Default	---
Description	Specifies that the fund to modify should be restricted to one usable by the given user.

-X, --extension	
Format	<code>-X or --extension <property>=<value></code>
Default	---
Description	Modifies an extension property. You can specify any number of extra field assignments.

--add-constraint**Format**

```
--add-constraint <constraint_name>=[!]<constraint_value>[,<constraint_name>=[!]<constraint_value>...]
```

Default

Description

Adds a constraint to the fund. The constraint value may be a perl5 regular expression. You can prepend an exclamation point to the constraint value to express a negation of the constraint. You can pass multiple constraints to the --add-constraint option in a comma-delimited list or specify multiple --add-constraint options.

--all**Format**

```
--all
```

Default

Description

Specifies that you want to reset all active allocations for all funds when you use it with the --reset option.

--debug**Format**

```
--debug
```

Default

Description

Logs debug information to the screen.

--default-deposit**Format**

```
--default-deposit <deposit_amount>
```

Default

Description

Specifies the default amount for any deposit that is made to this fund that does not specify a deposit amount.

--del-constraint

Format	--del-constraint <constraint_name>=<constraint_value>[, <constraint_name> [= <constraint_value>] ...]
Default	---
Description	Removes a constraint from the fund. You can pass multiple constraints to the --del-constraint option in a comma-delimited list or by specifying multiple --del-constraint options.

--filter

Format	--filter <filter_name>=<filter_value>
Default	---
Description	Restricts the fund to one without constraints that conflict with the specified filters. For example, gchfund --filter User=amy will restrict the fund to one usable by the user amy. You can specify multiple filter options by logically ANDing them together.

--filter-type

Format	--filter-type <filter_type>
Default	NonExclusive
Description	Selects the filtering type. If you use the exclusive filter type, a fund will only be matched if the specified filters meet all constraints. If you use the non-exclusive filter type, a fund will be matched as long as the specified filters do not conflict with the constraints.

--help

Format	--help
Default	---
Description	Displays a brief help message.

--man**Format** --man**Default** ---**Description** Displays full documentation.**--parent****Format** --parent <parent_fund_id>**Default** ---**Description** Sets a new parent fund, replacing the current parent fund if one exists.**--priority****Format** --priority <fund_priority>**Default** ---**Description** Sets a new fund priority.**--quiet****Format** --quiet**Default** ---**Description** Suppresses headers and success messages.**--reset****Format** --reset**Default** ---

--reset**Description**

Ends all active allocations and initiates a new default deposit. If the default deposit amount is positive, MAM creates a new allocation with this amount, otherwise no deposit is made and the fund becomes inactive. You may reset the allocations for a specified fund using the [\[-f\] fund_id](#) option, all funds using the [--all](#) option, or use filtering options to filter the funds to be reset. Do not use this option with any other modifying option.

--site**Format**

```
--site <site_name>
```

Default

Description

Obtains a response from specified site.

--verbose**Format**

```
--verbose
```

Default

Description

Displays modified object details.

--version**Format**

```
--version
```

Default

Description

Displays the product version.

Related Topics

[Modifying Funds on page 60](#)

[Commands Reference on page 159](#)

gchlien

Synopsis

```
gchlien [-s <start_time>] [-e <end_time>] [-t <lien_duration>]
[-d <description>] [-X, --extension <property>=<value>]... [--debug]
[--site <site_name>] [--help] [--man] [--quiet] [--verbose]
[--version] {[ -l] <lien_id>}
```

Overview

`gchlien` modifies a lien.

Options

-d	
Format	<code>-d <description></code>
Default	---
Description	Specifies a new description.

-e	
Format	<code>-e <end_time></code>
Default	---
Description	Specifies a new expiration time in the format YYYY-MM-DD[hh:mm:ss] -Infinity Infinity Now

-l	
Format	<code>[-l] <lien_id></code>
Default	---
Description	Specifies the ID of the lien to modify.

-s	
Format	<code>-s <start_time></code>

-s	
Default	---
Description	Specifies a new start time in the format YYYY-MM-DD[hh:mm:ss] -Infinity Infinity Now

-t	
Format	<code>-t <lien_duration></code>
Default	---
Description	Specifies the duration of the lien in seconds. Although the lien start time and end time are enforced, the duration is not authoritative. If the time frame between the end time and the start time is greater than the duration, the difference is the allotted grace period (which defaults to 10 minutes).

-X, --extension <property>	
Format	<code>-X or --extension <property>=<value></code>
Default	---
Description	Modifies an extension property. You can specify any number of extra field assignments.

--debug	
Format	<code>--debug</code>
Default	---
Description	Logs debug information to the screen.

--help	
Format	<code>--help</code>
Default	---

--help**Description**

Displays a brief help message.

--man**Format**

--man

Default

Description

Displays full documentation.

--quiet**Format**

--quiet

Default

Description

Suppresses headers and success messages.

--site**Format**

--site <site_name>

Default

Description

Obtains a response from specified site.

--verbose**Format**

--verbose

Default

Description

Displays modified object details.

--version	
Format	--version
Default	---
Description	Displays the product version.

Related Topics

[Modifying Liens on page 77](#)

[Commands Reference on page 159](#)

gchorg

Synopsis

```
gchorg [-d <description>] [-X, --extension
<property>=<value>]... [--debug] [--site <site_name>] [--help]
[--man] [--quiet] [--verbose] [--version] {[[-o] <organization_
name>]}
```

Overview

gchorg modifies an organization.

Options

-d	
Format	-d <description>
Default	---
Description	Specifies a new description.

-o	
Format	-o <organization_name>

-o**Default**

Description

Specifies the name of the organization to modify.

-X, --extension <property>**Format**

-X or --extension <property>=<value>

Default

Description

Modifies an extension property. You can specify any number of extra field assignments.

--debug**Format**

--debug

Default

Description

Logs debug information to the screen.

--help**Format**

--help

Default

Description

Displays a brief help message.

--man**Format**

--man

Default

Description

Displays full documentation.

--quiet	
Format	--quiet
Default	---
Description	Suppresses headers and success messages.

--site	
Format	--site <site_name>
Default	---
Description	Obtains a response from specified site.

--verbose	
Format	--verbose
Default	---
Description	Displays modified object details.

--version	
Format	--version
Default	---
Description	Displays the product version.

Related Topics

[Modifying Organizations on page 54](#)

[Commands Reference on page 159](#)

gchpasswd

Synopsis

```
gchpasswd [--debug] [--site <site_name>] [--help] [--man] [--quiet] [--verbose] [--version] [-[u] <user_name>]
```

Overview

`gchpasswd` sets a user password. If the user name is not specified via an option or as the unique argument, then the invoking user will be taken as the user whose password will be set. The invoker will be prompted for the new password.

Options

-u	
Format	<code>[-u] <user_name></code>
Default	---
Description	Specifies the name of user whose password is to be set. If no user is specified, the invoking user will be taken as the user whose password will be set.
--debug	
Format	<code>--debug</code>
Default	---
Description	Logs debug information to the screen.
--help	
Format	<code>--help</code>
Default	---
Description	Displays a brief help message.

--man	
Format	--man
Default	---
Description	Displays full documentation.

--quiet	
Format	--quiet
Default	---
Description	Suppresses headers and success messages.

--site	
Format	--site <site_name>
Default	---
Description	Obtains a response from specified site.

--verbose	
Format	--verbose
Default	---
Description	Displays modified object details.

--version	
Format	--version
Default	---
Description	Displays the product version.

Related Topics

[Setting Passwords on page 117](#)
[Commands Reference on page 159](#)

gchquote

Synopsis

```
gchquote [-s <start_time>] [-e <end_time>] [-d <description>] [-x, --extension <property>=<value>]... [--debug] [--site <site_name>] [--help] [--man] [--quiet] [--verbose] [--version] {[-q] <quote_id>}
```

Overview

gchquote modifies a quote.

Options

-d

Format -d <description>

Default ---

Description Specifies a new description.

-e

Format -e <end_time>

Default ---

Description Specifies a new expiration time in the format YYYY-MM-DD [hh:mm:ss] | -Infinity|Infinity|Now

-q

Format [-q] <quote_id>

-q	
Default	---
Description	Specifies the ID of the quote to modify.

-s	
Format	<code>-s <start_time></code>
Default	---
Description	Specifies a new start time in the format YYYY-MM-DD [hh:mm:ss] -Infinity Infinity Now

-X, --extension <property>	
Format	<code>-X or --extension <property>=<value></code>
Default	---
Description	Modifies an extension property. You can specify any number of extra field assignments.

--debug	
Format	<code>--debug</code>
Default	---
Description	Logs debug information to the screen.

--help	
Format	<code>--help</code>
Default	---
Description	Displays a brief help message.

--man**Format** --man**Default** ---**Description** Displays full documentation.**--quiet****Format** --quiet**Default** ---**Description** Suppresses headers and success messages.**--site****Format** --site <site_name>**Default** ---**Description** Obtains a response from specified site.**--verbose****Format** --verbose**Default** ---**Description** Displays modified object details.**--version****Format** --version**Default** ---**Description** Displays the product version.

Related Topics

[Modifying Quotes on page 82](#)

[Commands Reference on page 159](#)

gchrate

Synopsis

```
gchrate -n <charge_rate_name> [-x <charge_rate_value>] [-z <charge_rate_amount>] [-d <description>] [--debug] [--site <site_name>] [--help] [--man] [--quiet] [--verbose] [--version]
```

Overview

gchrate modifies a charge rate. Only the amount or the description of a charge rate may be modified.

Options

-d	
Format	<code>-d <description></code>
Default	---
Description	Specifies a new description.

-n	
Format	<code>-n <charge_rate_name></code>
Default	---
Description	Specifies the name of the charge rate to change.

-x	
Format	<code>-x <charge_rate_value></code>

-x	
Default	---
Description	Specifies the charge rate value expression to change. If you do not specify a value, an empty value is assumed.
-z	
Format	<code>-z <charge_rate_amount></code>
Default	---
Description	Specifies a new amount for the charge rate. The amount is an integer or decimal and may include operators indicating how to apply the charge rate as well as divisors and time-based units. See Managing Charge Rates on page 97 for more information.
--debug	
Format	<code>--debug</code>
Default	---
Description	Logs debug information to the screen.
--help	
Format	<code>--help</code>
Default	---
Description	Displays a brief help message.
--man	
Format	<code>--man</code>
Default	---
Description	Displays full documentation.

--quiet	
Format	--quiet
Default	---
Description	Suppresses headers and success messages.

--site	
Format	--site <site_name>
Default	---
Description	Obtains a response from specified site.

--verbose	
Format	--verbose
Default	---
Description	Displays modified object details.

--version	
Format	--version
Default	---
Description	Displays the product version.

Related Topics

[Modifying Charge Rates on page 101](#)

[Commands Reference on page 159](#)

gchrole

Synopsis

```
gchrole [-d <description>] [--add-user(s) <user_name>, ...] ... [-  
-add-action(s) <object_name>-><action_name>[<instance_name>], ...] ... [--  
-del-user(s) <user_name>, ...] ... [-del-action(s) <object_name>-><action_name>[<instance_name>], ...] ... [-  
-debug] [--site <site_name>] [--help] [--man] [--quiet] [--verbose] [--version] {[-r] <role_name>}
```

Overview

`gchaccount` modifies a role. This can include adding or removing users from a role and adding removing actions from a role.

Options

-d	
Format	<code>-d <description></code>
Default	---
Description	Specifies a new description.

-r	
Format	<code>[-r] <role_name></code>
Default	---
Description	Specifies the name of the role to modify.

--add-action	
Format	<code>--add-action <object_name>-><action_name>[<instance_name>] [, <object_name>-><action_name>[<instance_name>]] ...</code>
Default	---
Description	Adds actions to the role. You must specify the object, action and instance in the form shown. Unless specified, the instance will default to a value of ANY. You may pass multiple actions to the <code>--add-action</code> option in a comma-delimited list or by specifying multiple <code>--add-action</code> options.

--add-user	
Format	--add-user <user_name>[,<user_name>...]
Default	---
Description	Adds users to the role. You may pass multiple users to the --add-user option in a comma-delimited list or by specifying multiple --add-user options.
--debug	
Format	--debug
Default	---
Description	Logs debug information to the screen.
--del-action	
Format	--del-action <object_name>-><action_name>[{<instance_name>}] [,<object_name>-><action_name>[{<instance_name>}] ...]
Default	---
Description	Removes actions from a role. You must specify the object and action; however, the instance is optional. You may pass multiple actions to the --del-action option in a comma-delimited list or by specifying multiple --del-action options.
--del-user	
Format	--del-user <user_name>[,<user_name>...]
Default	---
Description	Removes users from the role. You may pass multiple users to the --del-user option in a comma-delimited list or by specifying multiple --del-user options.

--help**Format**`--help`**Default**`---`**Description**

Displays a brief help message.

--man**Format**`--man`**Default**`---`**Description**

Displays full documentation.

--quiet**Format**`--quiet`**Default**`---`**Description**

Suppresses headers and success messages.

--site**Format**`--site <site name>`**Default**`---`**Description**

Obtains a response from specified site.

--verbose**Format**`--verbose`**Default**`---`**Description**

Displays modified object details.

--version	
Format	--version
Default	---
Description	Displays the product version.

Related Topics

[Modifying Roles on page 114](#)

[Commands Reference on page 159](#)

gchusage

Synopsis

```
gchusage [-T <usage_record_type>] [-S <service_id>] [-u <user_name>] [-q <group_name>] [-a <account_name>] [-o <organization_name>] [-c <class_name>] [-Q <quality_of_service>] [-m <machine_name>] [-N <nodes>] [-P <processors>] [-C <cpu_time>] [-M <memory>] [-D <disk>] [-t <usage_duration>] [-s <start_time>] [-e <end_time>] [-d <description>] [-x, --extension <property>=<value>]... [--debug] [--site <site_name>] [--help] [--man] [--quiet] [--verbose] [--version] {[ -j ] <usage_record_id> | -J <instance_name>|<job_id>}
```

Overview

gchusage modifies a usage record.

Options

-a	
Format	-a <account_name>
Default	---
Description	Specifies a new account name.

-c

Format	<code>-c <class_name></code>
Default	---
Description	Specifies a new class or queue.

-C

Format	<code>-C <cpu_time></code>
Default	---
Description	Specifies a new CPU time used.

-d

Format	<code>-d <description></code>
Default	---
Description	Specifies a new description.

-D

Format	<code>-D <disk></code>
Default	---
Description	Specifies a new amount of disk used.

-e

Format	<code>-e <end_time></code>
Default	---

-e	
Description	Specifies a new date and time the usage ended in the format YYYY-MM-DD [hh:mm:ss] -Infinity Infinity Now.
-g	
Format	<code>-g <group_name></code>
Default	---
Description	Specifies a new group name.
-j	
Format	<code>[-j] <usage_record_id></code>
Default	---
Description	Specifies the ID of the usage record to modify. Instance names can be non-unique (resource managers often recycle job IDs). This option allows you to specify a usage record uniquely using the unique gold identifier.
-J	
Format	<code>[-J] <instance_name> <job_id></code>
Default	---
Description	Specifies the instance name of the usage record(s) to modify. If there is exactly one matching usage record, MAM will modify that usage record. Otherwise, MAM will display a list of usage records for the specified instance and you will be prompted to rerun <code>gchusage</code> against one of the enumerated usage records.
-m	
Format	<code>-m <machine_name></code>
Default	---

-m**Description**

Specifies the new name of the cluster, cloud, or system used.

-M**Format**

`-M <memory>`

Default

Description

Specifies the new amount of memory used.

-N**Format**

`-N <nodes>`

Default

Description

Specifies a new number of nodes used.

-o**Format**

`-o <organization_name>`

Default

Description

Specifies a new organization name.

-P**Format**

`-P <processors>`

Default

Description

Specifies a new number of processors used.

-Q	
Format	<code>-Q <quality_of_service></code>
Default	---
Description	Specifies a new quality of service used.

-s	
Format	<code>-s <start_time></code>
Default	---
Description	Specifies a new date and time the usage started in the format YYYY-MM-DD [hh:mm:ss] -Infinity Infinity Now.

-S	
Format	<code>-S <service_id></code>
Default	---
Description	Specifies a new root service ID (cloud).

-t	
Format	<code>-t <usage_duration></code>
Default	---
Description	Specifies a new amount of time used in seconds.

-T	
Format	<code>-T <usage_record_type></code>
Default	---

-T

Description	Specifies the usage record type (Job or Reservation, for example).
--------------------	--

-u

Format	<code>-u <user_name></code>
---------------	-----------------------------------

Default	---
----------------	-----

Description	Specifies a new user name.
--------------------	----------------------------

-X, --extension

Format	<code>-X or --extension <property>=<value></code>
---------------	---

Default	---
----------------	-----

Description	Modifies an extension property. You can specify any number of extra field assignments.
--------------------	--

--debug

Format	<code>--debug</code>
---------------	----------------------

Default	---
----------------	-----

Description	Logs debug information to the screen.
--------------------	---------------------------------------

--help

Format	<code>--help</code>
---------------	---------------------

Default	---
----------------	-----

Description	Displays a brief help message.
--------------------	--------------------------------

--man	
Format	--man
Default	---
Description	Displays full documentation.

--quiet	
Format	--quiet
Default	---
Description	Suppresses headers and success messages.

--site	
Format	--site <site_name>
Default	---
Description	Obtains a response from specified site.

--verbose	
Format	--verbose
Default	---
Description	Displays modified object details.

--version	
Format	--version
Default	---
Description	Displays the product version.

Related Topics

[Modifying a Usage Record on page 84](#)
[Commands Reference on page 159](#)

gchuser

Synopsis

```
gchuser [-A | -I] [-n <common_name>] [-F <phone_number>] [-E
<email_address>] [-a <default_account>] [-d <description>] [-
X, --extension <property>=<value>] ... [--debug] [--site <site_
name>] [--help] [--man] [--quiet] [--verbose] [--version] { [-
u] <user_name>}
```

Overview

gchuser modifies a user.

Options

-a

Format	-a <default_account>
Default	---
Description	Specifies the account MAM will charge when no account is specified.

-A

Format	-A
Default	---
Description	Activates the user.

-d

Format	-d <description>
---------------	------------------

-d	
Default	---
Description	Specifies a new description.

-E	
Format	<code>-E <email_address></code>
Default	---
Description	Specifies a new email address.

-F	
Format	<code>-F <phone_number></code>
Default	---
Description	Specifies a new phone number.

-I	
Format	<code>-I</code>
Default	---
Description	Deactivates the user.

n	
Format	<code>-n <common_name></code>
Default	---
Description	Specifies the common name for the user.

-u**Format**

[-u] <user_name>

Default

Description

Specifies the name of the user to modify.

-X, --extension <property>**Format**

-X or --extension <property>=<value>

Default

Description

Modifies an extension property. You can specify any number of extra field assignments.

--debug**Format**

--debug

Default

Description

Logs debug information to the screen.

--help**Format**

--help

Default

Description

Displays a brief help message.

--man**Format**

--man

Default

Description

Displays full documentation.

--quiet	
Format	--quiet
Default	---
Description	Suppresses headers and success messages.

--site	
Format	--site <site_name>
Default	---
Description	Obtains a response from specified site.

--verbose	
Format	--verbose
Default	---
Description	Displays modified object details.

--version	
Format	--version
Default	---
Description	Displays the product version.

Related Topics

[Modifying Users on page 44](#)

[Commands Reference on page 159](#)

gdeposit

Synopsis

```
gdeposit [-L <credit_limit>] [-s <start_time>] [-e <end_time>]
[-d <description>] [-f <fund_id>] [-i <allocation_id>] [-u
<user_name>] [-g <group_name>] [-a <account_name>] [-o
<organization_name>] [-c <class_name>] [-m <machine_name>] [--filter
<filter_name>=<filter_value>] ... [--filter-type
<filter_type>] [--create-fund=<boolean>] [--reset] [-h, --
hours] [--debug] [--site <site_name>] [--help] [--man] [--quiet]
[--verbose] [--version] [ [-z] <amount>]
```

Overview

`gdeposit` makes time-bound deposits into funds. After applying all filter options, if there is exactly one debitible fund for the specified criteria, MAM makes a deposit into that fund. If multiple funds match the specified criteria, MAM displays a list of matching funds and prompt you to respecify the deposit against one of the enumerated funds. If no funds match your criteria, if auto-generation is turned on for the fund object, or the --create-fund flag is asserted, MAM creates a fund and make a deposit into it; otherwise, the deposit fails (the fund will need to be created with [gmkfund](#)).

Options

-a	
Format	<code>-a <account_name></code>
Default	---
Description	Restricts the fund for the deposit to one usable by the specified account.

-c	
Format	<code>-c <class_name></code>
Default	---
Description	Restricts the fund for the depositto one usable by the specified class.

-d	
Format	<code>-d <description></code>
Default	---
Description	Specifies the reason for the deposit. The annotation applies to the transaction description (seen via gltrans), not the allocation description.
-e	
Format	<code>-e <end_time></code>
Default	Infinity
Description	Specifies the end time for the allocation to be credited in the format YYYY-MM-DD [hh:mm:ss] -Infinity Infinity Now.
-f	
Format	<code>-f <fund_id></code>
Default	Infinity
Description	Specifies the ID of the fund into which the deposit will be made.
-g	
Format	<code>-g <group_name></code>
Default	---
Description	Specifies that the fund for the deposit should be restricted to one usable by the specified group.
-h	
Format	<code>-h or --hours</code>

-h	
Default	---
Description	Treats currency as specified in hours. In systems where the currency is measured in resource-seconds (like processor-seconds), this option allows you to specify the amount and credit limit in resource-hours.
-i	
Format	<code>-i <allocation_id></code>
Default	---
Description	Specifies the allocation in which to make the deposit. This option is incompatible with the -L option.
-L	
Format	<code>-L <credit_limit></code>
Default	---
Description	Creates a new allocation with the specified credit limit. This option is incompatible with the -i option.
-m	
Format	<code>-m <machine_name></code>
Default	---
Description	Restricts the fund for the deposit to one usable by the specified machine.
-o	
Format	<code>-o <organization_name></code>
Default	---
Description	Restricts the fund for the deposit to one usable by the specified organization.

-s	
Format	<code>-s <start_time></code>
Default	Infinity
Description	Specifies the start time for the allocation to be credited in the format YYYY-MM-DD [hh:mm:ss] -Infinity Infinity Now.
-u	
Format	<code>-u <user_name></code>
Default	---
Description	Restricts the fund for the deposit to one usable by the specified user.
-z	
Format	<code>[-z] <amount></code>
Default	---
Description	Specifies the amount to deposit.
--create-fund	
Format	<code>--create-fund <boolean></code>
Default	---
Description	Overrides the fund auto-generation setting. Setting this option to <code>True</code> creates a default fund for this deposit. Setting this option to <code>False</code> inhibits the creation of a default fund for this deposit.
--debug	
Format	<code>--debug</code>

--debug**Default**

Description

Logs debug information to the screen.

--filter**Format**`--filter <filter_name>=<filter_value>`**Default**

Description

Restricts the fund to one where constraints do not conflict with the specified filters. For example, `gchfund --filter User=amy` restricts the fund to one usable by the user `amy`. You may specify multiple filter options by logically ANDing them together.

--filter-type**Format**`--filter-type <filter_type>`**Default**

NonExclusive

Description

Specifies the filtering type. If you use the exclusive filter type, MAM will only match a fund if the specified filters meet all constraints. If you use the non-exclusive filter type, MAM will match a fund as long as the specified filters do not conflict with the constraints.

--help**Format**`--help`**Default**

Description

Displays a brief help message.

--man**Format**`--man`**Default**

--man**Description**

Displays full documentation.

--quiet**Format**

--quiet

Default

Description

Suppresses headers and success messages.

--reset**Format**

--reset

Default

Description

Ends the current allocation and creates a new allocation with the deposit.

--site**Format**

--site <site_name>

Default

Description

Obtains a response from specified site.

--verbose**Format**

--verbose

Default

Description

Displays modified object details.

--version	
Format	--version
Default	---
Description	Displays the product version.

Related Topics

- [Making Deposits on page 60](#)
- [Commands Reference on page 159](#)

glsaccount

Synopsis

```
glsaccount [-A | -I] [-o <organization_name>] [-X, --extension
<property>=<value>] ... [-u <user_name>] [--full] [--show
<attribute_name>, ...] [--long] [--wide] [--format <output_
type>] [--debug] [--site <site_name>] [--help] [--man] [--
quiet] [--version] [[-a] <account_pattern>]
```

Overview

glsaccount displays account information.

You can customize the fields this command displays by default by setting the **account.show** configuration parameter in `gold.conf`.

Options

-a	
Format	<code>[-a] <account_pattern></code>
Default	---
Description	Displays only accounts matching the pattern. If no pattern is specified then all accounts are displayed.

-A	
Format	<code>-A</code>
Default	---
Description	Displays only active accounts.

-I	
Format	<code>-I</code>
Default	---
Description	Displays only inactive accounts.

-O	
Format	<code>-o <organization_name></code>
Default	---
Description	Displays only accounts belonging to the specified organization. The following wildcards are supported: <ul style="list-style-type: none"> • * – matches any number of characters • ? – matches a single character

-U	
Format	<code>-u <user_name></code>
Default	---
Description	Displays only accounts that have the specified user as a member.

-X	
Format	<code>-X</code> or <code>--extension <property>=<value></code>

-X**Default**

Description

Specifies an extension property. You can specify any number of extra custom conditions.

--debug**Format**`--debug`**Default**

Description

Logs debug information to the screen.

--format**Format**`--format <output_format>`**Default**

standard

Description

Specifies the data output format. Valid values include standard and csv.

--full**Format**`--full`**Default**

Description

Displays all attributes.

--help**Format**`--help`**Default**

Description

Displays a brief help message.

--long	
Format	--long
Default	---
Description	Long format. Displays multi-valued fields in a multi-line format.

--man	
Format	--man
Default	---
Description	Displays full documentation.

--quiet	
Format	--quiet
Default	---
Description	Suppresses headers and success messages.

--show	
Format	--show <attribute_name>[,<attribute_name>...]
Default	---

-show**Description**

Displays only the specified attributes in the order specified. Valid attributes include the following:

- **Active** – Boolean indicating whether this account is active or not.
- **CreationTime** – Time this account was created.
- **Deleted** – Boolean indicating whether this account is deleted or not.
- **Description** – Account description.
- **ModificationTime** – Time this account was last modified.
- **Name** – Account name.
- **Organization** – Organization to which the account belongs.
- **RequestId** – ID of the last modifying request.
- **TransactionId** – ID of the last modifying transaction.
- **Users** – List of users belonging to the account. A caret prefixing a user name indicates that the user is an account admin. A minus sign prefixing a user name indicates that the user is an inactive member of the account.

-site**Format**

--site <site_name>

Default

Description

Obtains a response from specified site.

-version**Format**

--version

Default

Description

Displays the product version.

-wide**Format**

--wide

Default

Description

Wide format. Displays multi-valued fields in a single-line, comma-separated format.

Related Topics

[Querying Accounts on page 48](#)

[Commands Reference on page 159](#)

glsalloc

Synopsis

```
glsalloc [-A | -I | {[-s <start_time>] [-e <end_time>]}] [-f
<fund_id>] [-X, --extension <property>=<value>]... [-u <user_
name>] [-g <group_name>] [-a <account_name>] [-o
<organization_name>] [-c <class_name>] [-m <machine_name>] [--_
filter <filter_name>=<filter_value>]... [--filter-type
<filter_type>] [--include-ancestors] [--full] [--show
<attribute_name>, ...] [--format <output_type>] [-h, --hours]
[--debug] [--site <site_name>] [--help] [--man] [--quiet] [--_
version] [[-i] <allocation_id>]
```

Overview

`glsalloc` displays allocation information.

You can customize the fields this command displays by default by setting the **allocation.show** configuration parameter in `gold.conf`.

Options

-a	
Format	<code>-a <account_name></code>
Default	---
Description	Displays only allocations usable by the specified account.

-A	
Format	<code>-A</code>
Default	---
Description	Displays only active allocations.

-c

Format	<code>-c <class_name></code>
Default	---
Description	Displays only allocations usable by the specified class.

-e

Format	<code>-e <end_time></code>
Default	---
Description	Displays only allocations that start before the specified end time in the format YYYY-MM-DD [hh:mm:ss] -Infinity Infinity Now.

-f

Format	<code>-f <fund_id></code>
Default	---
Description	Displays only the allocations associated with the specified fund.

-g

Format	<code>-g <group_name></code>
Default	---
Description	Displays only allocations usable by the specified group.

-h

Format	<code>-h or --hours</code>
Default	---

-h

Description	Displays time-based credits in hours. In cases where the currency is measured in resource-seconds (like processor-seconds), the currency is divided by 3600 to display resource-hours.
--------------------	--

-i

Format	<code>[-i] <allocation_id></code>
Default	---
Description	Displays only the allocation with the specified ID.

-I

Format	<code>-I</code>
Default	---
Description	Displays only inactive allocations.

-m

Format	<code>-m <machine_name></code>
Default	---
Description	Displays only allocations usable by the specified machine.

-o

Format	<code>-o <organization_name></code>
Default	---
Description	Displays only accounts usable to the specified organization.

-s**Format**`-s <start_time>`**Default**`---`**Description**

Displays only allocations that end after the specified start time in the format YYYY-MM-DD [hh:mm:ss] |-Infinity|Infinity|Now.

-u**Format**`-u <user_name>`**Default**`---`**Description**

Displays only allocations usable by the specified user.

-X**Format**`-X or --extension <property>=<value>`**Default**`---`**Description**

Specifies an extension property. You can specify any number of extra custom conditions.

--debug**Format**`--debug`**Default**`---`**Description**

Logs debug information to the screen.

--filter**Format**`--filter <filter_name>=<filter_value>`**Default**`---`

--filter**Description**

Displays allocations where fund constraints comply with the specified filters. For example, `gls-fund --filter User=amy` displays funds usable by the user amy. You can specify multiple filter options by logically ANDing them together.

--filter-type**Format**

`--filter-type <filter_type>`

Default

NonExclusive

Description

Specifies the filtering type. If you use exclusive filter type, MAM only matches a fund if the specified filters meet all constraints. If you specify the non-exclusive filter type, MAM matches a fund as long as the specified filters do not conflict with the constraints.

--format**Format**

`--format <output_format>`

Default

standard

Description

Specifies a data output format. Valid values include standard and csv.

--full**Format**

`--full`

Default

Description

Displays all attributes.

--help**Format**

`--help`

Default

Description

Displays a brief help message.

--include-ancestors**Format** --include-ancestors**Default** ---**Description** Includes ancestors of the selected allocations.**--man****Format** --man**Default** ---**Description** Displays full documentation.**--quiet****Format** --quiet**Default** ---**Description** Suppresses headers and success messages.**--show****Format** --show <attribute_name>[,<attribute_name>...]**Default** ---

--show

Description	Valid attributes include the following:
	<ul style="list-style-type: none"> • Active – Boolean indicating whether this allocation is active or not. • Adjustments – Total of subsequent adjustments to the initial deposit via deposits, withdrawals or transfers (Allocated - InitialDeposit). • Allocated – Adjusted allocation. This value stores the effective allocated amount based on the initial deposit and subsequent allocation adjustments via deposits, withdrawals or transfers. • Available – Amount currently available for charging. If the allocation is active, this is Remaining - Reserved + CreditLimit. If the allocation is inactive, this is zero. • Balance – Active allocation balance. If the allocation is active, this is the remaining allocation amount (Remaining). If the allocation is inactive, this is zero. • Capacity – Total expendable amount (Allocated + CreditLimit). • CreationTime – Time this allocation was created. • CreditLimit – Determines how far in the negative this allocation is permitted to be used (enforced in quotes and liens). • Deleted – Boolean indicating whether this allocation is deleted or not. • Description – Allocation description. • Effective – Effective balance not blocked by liens. If the.allocation is active, this is Remaining - Reserved. If the allocation is inactive, this is zero. • EndTime – Time this allocation becomes inactive. • Fund – Fund ID. • FundName – Fund name. • Id – Allocation ID. • InitialDeposit – Amount of the first deposit into this allocation. • ModificationTime – Time this allocation was last modified. • PercentRemaining – Percentage of allocation remaining (remaining * 100 / Capacity). • PercentUsed – Percentage of allocation used (Used * 100 / Capacity). • Remaining – Remaining allocation amount. • Reserved – Sum of active lien amounts against this allocation. • RequestId – ID of the last modifying request. • StartTime – Time this allocation becomes active. • TransactionId – ID of the last modifying transaction. • Used – Amount used from this allocation (Allocated - Remaining).

--site

Format	<code>--site <site_name></code>
Default	---
Description	Obtains a response from specified site.

--version	
Format	--version
Default	---
Description	Displays the product version.

Related Topics

- [Querying Allocations on page 71](#)
- [Commands Reference on page 159](#)

glscharge

Synopsis

```
glscharge [-j <usage_record_id>] [-J <instance_name>] [-n <usage_property_name>] [-s <start_time>] [-e <end_time>] [--full] [--show <attribute_name>, ...] [--format <output_type>] [-h, --hours] [--debug] [--site <site_name>] [--help] [--man] [--quiet] [--version]
```

Overview

glscharge displays allocation information.

Options

-e	
Format	-e <end_time>
Default	---
Description	Displays charges occurring before the specified time in the format YYYY-MM-DD [hh:mm:ss] -Infinity Infinity Now

-h

Format	<code>-h</code> or <code>--hours</code>
Default	---
Description	Displays time-based credits in hours. In cases where the currency is measured in resource-seconds (like processor-seconds), the currency is divided by 3600 to display resource-hours.

-j

Format	<code>-j <usage_record_id></code>
Default	---
Description	Displays only charges associated with the specified usage record.

-J

Format	<code>-J <instance_name></code>
Default	---
Description	Displays only charges against the specified instance (such as a job ID).

-n

Format	<code>-n <usage_record_property_name></code>
Default	---
Description	Displays only charges against the specified usage property.

-s

Format	<code>-s <start_time></code>
Default	---

-s**Description**

Displays charges occurring after the specified time in the format YYYY-MM-DD [hh:mm:ss] | -Infinity|Infinity|Now

--debug**Format**

--debug

Default

Description

Logs debug information to the screen.

--format**Format**

--format <output_format>

Default

standard

Description

Specifies a data output format. Valid values include standard and csv.

--full**Format**

--full

Default

Description

Displays all attributes.

--help**Format**

--help

Default

Description

Displays a brief help message.

--man	
Format	--man
Default	---
Description	Displays full documentation.

--quiet	
Format	--quiet
Default	---
Description	Suppresses headers and success messages.

--show	
Format	--show <attribute_name>[,<attribute_name>...]
Default	---
Description	<p>Displays only the specified attributes in the order specified. Valid attributes include the following:</p> <ul style="list-style-type: none"> • Amount – Amount charged. • CreationTime – Time this charge was created. • Deleted – Boolean indicating whether this allocation is deleted or not. • Description – Charge description. • Details – Details of the formula used in calculating the charge. • Duration – Amount of time the item was used in seconds. • Instance – Instance name (such as job ID) for the charge. • ModificationTime – Time this charge was last modified. • Name – Usage record property name (also charge rate name). • Rate – Base charge rate. • RequestId – ID of the last modifying request. • ScalingFactor – Product of all applicable multipliers (discounts and premiums) applied to the base rate. • TransactionId – ID of the last modifying transaction. • UsageRecord – Usage record ID. • Value – Usage record property value.

--site

Format	--site <site_name>
Default	---
Description	Obtains a response from specified site.

--version

Format	--version
Default	---
Description	Displays the product version.

Related Topics

[Querying Itemized Charges on page 95](#)[Commands Reference on page 159](#)

glsconfig

Synopsis

```
glsconfig [-c on page 246|-s on page 247|-g on page 247] [-p
parameter_pattern] [--help] [--man] [--quiet] [--version]
```

Overview

glsconfig is used to display configuration information. It will only display enabled (uncommented) parameter values. If neither -c, -s nor -g are specified, configuration parameters from all configuration files will be displayed.

Options

-c

Format	-c
---------------	----

-c	
Default	---
Description	Display only client configuration parameters.

-g	
Format	<code>-g</code>
Default	---
Description	Display only GUI configuration parameters.

-s	
Format	<code>-s</code>
Default	---
Description	Display only server configuration parameters.

-p	
Format	<code>-p <parameter_pattern></code>
Default	---
Description	Display only configuration parameters matching the specified pattern. The following wildcards are supported: * – Matches any number of characters ? – Matches a single character

--help	
Format	<code>--help</code>
Default	---

--help**Description**

Brief help message.

--man**Format**

--man

Default

Description

Full documentation.

--quiet**Format**

--quiet

Default

Description

Suppress headers and parameter names.

--version**Format**

--version

Default

Description

Display product version.

Related Topics

[Commands Reference on page 159](#)

glsevent

Synopsis

```
glsevent [-s <start_time>] [-e <end_time>] [--full] [--show <attribute_name>, ...] [--format <output_type>] [--debug] [--]
```

`site <site_name>] [--help] [--man] [--quiet] [--version] [[-E] <event_id>]`

Overview

glsevent displays event information.

You can customize the fields this command displays by default by setting the **event.show** configuration parameter in `gold.conf`.

Options

-e	
Format	<code>-e <end_time></code>
Default	---
Description	Displays events with a prospective fire time occurring before the specified time in the format YYYY-MM-DD[hh:mm:ss] -Infinity Infinity Now
-E	
Format	<code>[-E] <event_id></code>
Default	---
Description	Displays only the event with the specified ID.
-s	
Format	<code>-s <start_time></code>
Default	---
Description	Displays events with a prospective fire time occurring after the specified time in the format YYYY-MM-DD[hh:mm:ss] -Infinity Infinity Now
--debug	
Format	<code>--debug</code>

--debug**Default**

Description

Logs debug information to the screen.

--format**Format**`--format <output_format>`**Default**

standard

Description

Specifies a data output format. Valid values include standard and csv.

--full**Format**`--full`**Default**

Description

Displays all attributes.

--help**Format**`--help`**Default**

Description

Displays a brief help message.

--man**Format**`--man`**Default**

Description

Displays full documentation.

--quiet	
Format	--quiet
Default	---
Description	Suppresses headers and success messages.

--show	
Format	--show <attribute_name>[,<attribute_name>...]
Default	---

-show	
Description	<p>Displays only the specified attributes in the order specified. Valid attributes include the following:</p> <ul style="list-style-type: none"> • ArmTime – Time the event was last armed or fired. This field is used as a reference time to be able to derive how long the event has been waiting to happen. This field will be initially set to mark the moment the first FireTime is set and updated thereafter to indicate the last time the event was fired. In the case where an event does not have a FireTime set, this field may be set manually and used in a similar manner. If we consider the time between event firings as "laps," this could be thought of as the Lap Start Time. • CatchUp – If set to <code>True</code> and Gold was down during the time this event should have fired, Gold will attempt to make up for the past due events by progressively firing them (rearming based on previous arm time) until catching up to the present. The actions will still show as having occurred in the present rather than in the past. If set to <code>False</code>, and Gold is brought back up after an outage, Gold will still fire immediately for a past due event, but it will only fire once and then rearm relative to the current time. • CreationTime – Time this event was created. • Deleted – Boolean indicating whether this event is deleted or not. • Description – Event description. • EndTime – Time after which an event having a rearm period will be deleted. • FailureCommand – Serialized Gold request string to be executed if the fired command results in an unsuccessful response status. They syntax is the same as used to invoke commands within the goldsh prompt. • FireCommand – Serialized Gold request string to be executed when the event is fired. They syntax is the same as used to invoke commands within the goldsh prompt. • FireTime – Target time for the event to be triggered. The actual fire time may be dependent on the state of the server and will be recorded in the CreationTime property of the corresponding "Event Fire" Transaction. • Id – Event ID. • ModificationTime – Time this event was last modified. • Notify – Expression specifying where to send a notification of the response for the fire command and the failure command. The notification expression is of the form: <code>[+--] [delivery_method:] [recipient] [, [+--] [delivery_method:] [recipient]] *</code> (For example, <code>-store:amy</code>). If the term is a <code>-</code>, the notification is sent only on failure. If the term is a <code>+</code>, the notification is sent only on success. Otherwise the notification is always sent. There can be multiple notify expressions separated by a comma. • RearmPeriod – Time period expression specifying when the event will be rearmed. This period expression is of the form: <code>period[[@instant] [~ ^] !]</code> where period may be something like 1 day, 2 hours, or 5 minutes. Instant locks the period to a specific instant within the time period such as 1 day @ hour 12 or 1 month @ day 3. The modifiers indicate whether the time period should be relative to now (<code>!</code>), or relative to the start of this (<code>~</code>) designator (month or minute, etc.), or relative to the start of the first (<code>^</code>) designator (month or minute, etc.). • RearmOnFailure – If set to <code>False</code>, the event will not be rearmed if the command was unsuccessful. If set to <code>True</code>, the event will be evaluated for rearming even if the command response has a status of Failure. The standard default value for this boolean is <code>False</code>. • RequestId – ID of the last modifying request.

--show

- **TransactionId** – ID of the last modifying transaction.

--site

Format `--site <site_name>`

Default `---`

Description Obtains a response from specified site.

--version

Format `--version`

Default `---`

Description Displays the product version.

Related Topics

[Querying Events on page 106](#)

[Commands Reference on page 159](#)

glsfund

Synopsis

```
glsfund [-A | -I] [-n <fund_name>] [-X, --extension
<property>=<value>]... [-u <user_name>] [-g <group_name>] [-a
<account_name>] [-o <organization_name>] [-c <class_name>] [-m
<machine_name>] [--filter <filter_name>=<filter_value>]... [--filter-type <filter_type>] [--full] [--show <attribute_name>, ...] [--long] [--wide] [--format <output_type>] [-h, --hours]
[--debug] [--site <site_name>] [--help] [--man] [--quiet] [--version] [ [-f] <fund_id>]
```

Overview

glsfund displays fund information.

You can customize the fields this command displays by default by setting the **fund.show** configuration parameter in `gold.conf`.

Options

-a	
Format	<code>-a <account_name></code>
Default	---
Description	Displays only funds valid toward the specified account.

-A	
Format	<code>-A</code>
Default	---
Description	Displays funds with active allocations.

-c	
Format	<code>-c <class_name></code>
Default	---
Description	Displays only funds usable by the specified class.

-f	
Format	<code>[-f] <fund_id></code>
Default	---
Description	Displays only the funds with the specified ID.

-g	
Format	<code>-g <group_name></code>
Default	---
Description	Displays only funds usable by the specified group.

-h	
Format	<code>-h</code> or <code>--hours</code>
Default	---
Description	Displays time-based credits in hours. In cases where the currency is measured in resource-seconds (like processor-seconds), the currency is divided by 3600 to display resource-hours.

-I	
Format	<code>-I</code>
Default	---
Description	Displays only funds with inactive allocations.

-m	
Format	<code>-m <machine_name></code>
Default	---
Description	Displays only funds valid toward the specified machine.

-n	
Format	<code>-n <fund_name></code>
Default	---

-n**Description**

Displays only funds with the specified name.

-o**Format**`-o <organization_name>`**Default**

Description

Displays only funds valid toward the specified organization.

-u**Format**`-u <user_name>`**Default**

Description

Displays only funds valid toward the specified user.

-x**Format**`-X or --extension <property>=<value>`**Default**

Description

Specifies an extension property. You can specify any number of extra custom conditions.

--debug**Format**`--debug`**Default**

Description

Logs debug information to the screen.

--filter

Format	--filter <filter_name>=<filter_value>
Default	---
Description	Displays funds where constraints do not conflict with the specified filters. For example, glsfund -f User=amy displays funds usable by the user amy. You may specify multiple filter options by logically ANDing them together.

--filter-type

Format	--filter-type <filter_type>
Default	NonExclusive
Description	Selects the filtering type. If you use the exclusive filter, MAM only matches a fund if the specified filters meet all constraints. If you use the non-exclusive filter type, MAM will match a fund as long as the specified filters do not conflict with the constraints.

--format

Format	--format <output_format>
Default	standard
Description	Specifies a data output format. Valid values include standard and csv.

--full

Format	--full
Default	---
Description	Displays all attributes.

--help

Format	--help
---------------	--------

--help**Default**

Description

Displays a brief help message.

--long**Format**

--long

Default

Description

Long format. Displays multi-valued fields in a multi-line format.

--man**Format**

--man

Default

Description

Displays full documentation.

--quiet**Format**

--quiet

Default

Description

Suppresses headers and success messages.

--show**Format**

--show <attribute_name>[,<attribute_name>...]

Default

--show**Description**

Displays only the specified attributes in the order specified. Valid attributes include the following:

- **Allocated** – Adjusted allocation. This value stores the effective allocated amount based on the initial deposit and subsequent allocation adjustments via deposits, withdrawals or transfers.
- **Allocations** – Lists the active allocations in this fund in the format `id:amount:start_time:end_time`.
- **Balance** – Sum of active allocation amounts within this fund.
- **Children** – Lists the children funds in the format `id[(deposit_share)] [^]` where the carat symbol (^) is displayed if Overflow is True.
- **Constraints** – Constraints on fund usage.
- **CreationTime** – Time this fund was created.
- **CreditLimit** – Sum of active credit limits within this fund.
- **DefaultDeposit** – Used for the deposit amount if a deposit is made without specifying an amount.
- **Deleted** – Boolean indicating whether this fund is deleted or not.
- **Description** – Fund description.
- **Id** – Fund ID.
- **InitialDeposit** – Initial deposit for current allocation.
- **ModificationTime** – Time this fund was last modified.
- **Name** – Fund name.
- **Parent** – Displays the parent fund in the format `id[(deposit_share)] [^]` where the carat symbol (^) is displayed if Overflow is True.
- **Priority** – Fund priority.
- **RequestId** – ID of the last modifying request.
- **TransactionId** – ID of the last modifying transaction.

--site**Format**

`--site <site_name>`

Default

Description

Obtains a response from specified site.

--version**Format**

`--version`

Default

--version**Description**

Displays the product version.

--wide**Format**

--wide

Default

Description

Wide format. Displays multi-valued fields in a single-line comma-separated format.

Related Topics

[Querying Funds on page 59](#)[Commands Reference on page 159](#)

glslien

Synopsis

```
glslien [-A|-I] [-J <instance_pattern>] [-X, --extension <property>=<value>]... [-u <user_name>] [-g <group_name>] [-a <account_name>] [-o <organization_name>] [-c <class_name>] [-m <machine_name>] [--filter <filter_name>=<filter_value>]... [--filter-type <filter_type>] [--full] [--show <attribute_name>,...] [--long] [--wide] [--format <output_type>] [-h, --hours] [--debug] [--site <site_name>] [--help] [--man] [--quiet] [--version] [[-l] <lien_id>]
```

Overview

`glslien` displays lien information.

You can customize the fields this command displays by default by setting the **lien.show** configuration parameter in `gold.conf`.

Options

-a	
Format	<code>-a <account_name></code>
Default	---
Description	Displays only liens against the specified account.

-A	
Format	<code>-A</code>
Default	---
Description	Displays only unexpired liens.

-c	
Format	<code>-c <class_name></code>
Default	---
Description	Displays only liens against the specified class.

-g	
Format	<code>-g <group_name></code>
Default	---
Description	Displays only liens against the specified group.

-h	
Format	<code>-h or --hours</code>
Default	---

-h**Description**

Displays time-based credits in hours. In cases where the currency is measured in resource-seconds (like processor-seconds), the currency is divided by 3600 to display resource-hours.

-I**Format**

`-I`

Default

Description

Displays only expired liens.

-J**Format**

`-J <instance_pattern>`

Default

Description

Displays only liens with the instance names (or job IDs) matching the pattern.

I**Format**

`[-l] <lien_id>`

Default

Description

Displays only the specified lien.

-m**Format**

`-m <machine_name>`

Default

Description

Displays only liens against the specified machine.

-o	
Format	<code>-o <organization_name></code>
Default	---
Description	Displays only liens against the specified organization.

-u	
Format	<code>-u <user_name></code>
Default	---
Description	Displays only liens against the specified user.

-X	
Format	<code>-X or --extension <property>=<value></code>
Default	---
Description	Specifies an extension property. You can specify any number of extra custom conditions.

--debug	
Format	<code>--debug</code>
Default	---
Description	Logs debug information to the screen.

--filter	
Format	<code>--filter <filter_name>=<filter_value></code>
Default	---

--filter**Description**

Displays liens where constraints do not conflict with the specified filters. For example, `glslien -f User=amy` will display liens usable by the user amy. You may specify multiple filter options by logically ANDing them together.

--filter-type**Format**

`--filter-type <filter_type>`

Default

AttributedTo

Description

Selects the filtering type. If you use the AttributedTo filter type, the query returns all liens associated with usage records satisfying the filters. If you use the ImpingesUpon filter type, the query returns all liens affecting the balances of funds satisfying the filters.

--format**Format**

`--format <output_format>`

Default

standard

Description

Specifies a data output format. Valid values include standard and csv.

--full**Format**

`--full`

Default

Description

Displays all attributes.

--help**Format**

`--help`

Default

Description

Displays a brief help message.

--long	
Format	--long
Default	---
Description	Long format. Displays multi-valued fields in a multi-line format.

--man	
Format	--man
Default	---
Description	Displays full documentation.

--quiet	
Format	--quiet
Default	---
Description	Suppresses headers and success messages.

--show	
Format	--show <attribute_name>[,<attribute_name>...]
Default	---

-show**Description**

Displays only the specified attributes in the order specified. Valid attributes include the following:

- **Funds** – List of funds that the lien has holds against.
- **Allocations** – List of allocations that the lien has holds against in the format
 $<\text{allocation_id}><-\langle\text{fund_id}\rangle=\langle\text{reserved_amount}\rangle$.
- **Amount** – Reserved amount.
- **CreationTime** – Time this lien was created.
- **Deleted** – Boolean indicating whether this lien is deleted or not.
- **Description** – Lien description.
- **Duration** – Expected duration of the reserved usage in seconds.
- **EndTime** – Time the lien becomes inactive.
- **Id** – Lien ID.
- **Instance** – The lien is against the specified instance (for instance, job ID).
- **ModificationTime** – Time this lien was last modified.
- **RequestId** – ID of the last modifying request.
- **StartTime** – Time the lien becomes active.
- **TransactionId** – ID of the last modifying transaction.
- **UsageRecord** – ID of the usage record associated with the lien and containing the usage properties.

-site**Format**

--site <site_name>

Default

Description

Obtains a response from specified site.

--version**Format**

--version

Default

Description

Displays the product version.

--wide	
Format	--wide
Default	---
Description	Wide format. Displays multi-valued fields in a single-line comma-separated format.

Related Topics

[Querying Liens on page 76](#)

[Commands Reference on page 159](#)

glsnot

Synopsis

```
glsnot [-E <event_id>] [-T <notification_type>] [-k <primary_key_value>] [-u <recipient>] [-x <status>] [-s <start_time>] [-e <end_time>] [--delete] [--full] [--show <attribute_name>, ...] [--format <output_type>] [--debug] [--site <site_name>] [--help] [--man] [--quiet] [--version] [[-N] <notification_id>]
```

Overview

glsnot displays stored notification information.

You can customize the fields this command displays by default by setting the **notification.show** configuration parameter in `gold.conf`.

Options

-e	
Format	-e <end_time>
Default	---
Description	Displays the notifications sent before the specified time in the format YYYY-MM-DD [hh:mm:ss] -Infinity Infinity Now.

-E**Format**`-E <event_id>`**Default**

Description

Displays only the notifications associated with the specified event ID.

-k**Format**`-k <primary_key_value>`**Default**

Description

Displays only the notifications associated with the specified primary key value. This value of the primary key of the object instance that the command acted on.

-N**Format**`[-N] <notification_id>`**Default**

Description

Displays only the notifications with the specified ID.

-s**Format**`-s <start_time>`**Default**

Description

Displays notifications sent after the specified time in the format YYYY-MM-DD [hh:mm:ss] | -Infinity | Infinity | Now

-x**Format**`-x <status>`

-X

Default	---
Description	Displays notifications having the specified status (such as Success or Failure).

-T

Format	<code>-T <notification_type></code>
Default	---
Description	Displays notifications of the specified type (such as Fire or Failure).

-u

Format	<code>-u <recipient></code>
Default	---
Description	Displays notifications having the specified recipient. This could be a user name or any tag that identifies the intended reader of this notification.

--debug

Format	<code>--debug</code>
Default	---
Description	Logs debug information to the screen.

--delete

Format	<code>--delete</code>
Default	---
Description	Deletes a notification after it has been queried.

--format**Format** --format <output_format>**Default** standard**Description** Specifies a data output format. Valid values include standard and csv.**--full****Format** --full**Default** ---**Description** Displays all attributes.**--help****Format** --help**Default** ---**Description** Displays a brief help message.**--man****Format** --man**Default** ---**Description** Displays full documentation.**--quiet****Format** --quiet**Default** ---**Description** Suppresses headers and success messages.

--show	
Format	--show <attribute_name>[, <attribute_name>...]
Default	---
Description	<p>Displays only the specified attributes in the order specified. Valid attributes include the following:</p> <ul style="list-style-type: none"> • Code – Event command exit code. • CreationTime – Time this notification was created. • Deleted – Boolean indicating whether this notification is deleted or not. • EndTime – Time after which a notification will be detected. • Event – Event ID. • Key – Object primary key value. • Id – Notification ID. • Message – Event command message. • ModificationTime – Time this notification was last modified. • Recipient – Recipient to notify. • Status – Event command status. • RequestId – ID of the last modifying request. • TransactionId – ID of the last modifying transaction. • Type – Displays the type of notification. Notifications may be created by event "Fire" commands or by event "Failure" commands.

--site	
Format	--site <site_name>
Default	---
Description	Obtains a response from specified site.

--version	
Format	--version
Default	---
Description	Displays the product version.

Related Topics

[Querying Notifications on page 109](#)

[Commands Reference on page 159](#)

glsorg

Synopsis

```
glsorg [-X, --extension <property>=<value>]... [--full] [--show
<attribute_name>, ...] [--format <output_type>] [--debug] [--site
<site_name>] [--help] [--man] [--quiet] [--version] [[-o]
<organization_pattern>]
```

Overview

`glsorg` displays organization information.

You can customize the fields this command displays by default by setting the **organization.show** configuration parameter in `gold.conf`.

Options

-o	
Format	<code>-o <organization_pattern></code>
Default	---
Description	<p>Displays only organizations matching the pattern. If no pattern is specified then all organizations are displayed. The following wildcards are supported:</p> <ul style="list-style-type: none"> * – matches any number of characters ? – matches a single character

-X	
Format	<code>-X</code> or <code>--extension <property>=<value></code>
Default	---
Description	Extension property. You may specify any number of extra custom conditions may be specified.

--debug	
Format	--debug
Default	---
Description	Logs debug information to the screen.

--format	
Format	--format <output_format>
Default	standard
Description	Specifies a data output format. Valid values include standard and csv.

--full	
Format	--full
Default	---
Description	Displays all attributes.

--help	
Format	--help
Default	---
Description	Displays a brief help message.

--man	
Format	--man
Default	---
Description	Displays full documentation.

--quiet	
Format	--quiet
Default	---
Description	Suppresses headers and success messages.
--show	
Format	--show <attribute_name>[,<attribute_name>...]
Default	---
Description	<p>Displays only the specified attributes in the order specified. Valid attributes include the following:</p> <ul style="list-style-type: none"> • CreationTime – Time this organization was created. • Deleted – Boolean indicating whether this organization is deleted or not. • Description – Organization description. • ModificationTime – Time this organization was last modified. • Name – Organization name. • RequestId – ID of the last modifying request. • TransactionId – ID of the last modifying transaction.
--site	
Format	--site <site_name>
Default	---
Description	Obtains a response from specified site.
--version	
Format	--version
Default	---
Description	Displays the product version.

Related Topics

[Querying Organizations on page 53](#)

[Commands Reference on page 159](#)

glsquote

Synopsis

```
glsquote [-J <instance_name>] [-A|-I] [-X, --extension
<property>=<value>]... [-u <user_name>] [-q <group_name>] [-a
<account_name>] [-o <organization_name>] [-c <class_name>] [-m
<machine_name>] [--filter <filter_name>=<filter_value>] [--full]
[--show <attribute_name>,...] [--long] [--wide] [--format
<output_type>] [-h,--hours] [--debug] [--site <site_name>]
[--help] [--man] [--quiet] [--version] [[-q] <quote_id>]
```

Overview

glsquote displays quote information.

You can customize the fields this command displays by default by setting the **quote.show** configuration parameter in `gold.conf`.

Options

-a	
Format	<code>-a <account_name></code>
Default	---
Description	Displays only quotes for the specified account.

-A	
Format	<code>-A</code>
Default	---
Description	Displays only unexpired quotes.

-c

Format	<code>-c <class_name></code>
Default	---
Description	Displays only quotes for the specified class.

-g

Format	<code>-g <group_name></code>
Default	---
Description	Displays only quotes for the specified group.

-h

Format	<code>-h</code> or <code>--hours</code>
Default	---
Description	Displays time-based credits in hours. In cases where the currency is measured in resource-seconds (like processor-seconds), the currency is divided by 3600 to display resource-hours.

-I

Format	<code>-I</code>
Default	---
Description	Displays only expired quotes.

-J

Format	<code>-J <instance_name></code>
Default	---

-J**Description**

Displays only quotes with the specified instance name or job ID.

-m**Format**`-m <machine_name>`**Default**

Description

Displays only quotes for the specified machine.

-o**Format**`-o <organization_name>`**Default**

Description

Displays only quotes for the specified organization.

-q**Format**`[-q] <quote_id>`**Default**

Description

Displays only information for the specified quote.

-u**Format**`-u <user_name>`**Default**

Description

Displays only quotes for the specified user.

-X**Format**`-X or --extension <property>=<value>`**Default**`---`**Description**

Specifies an extension property. You can specify any number of extra custom conditions.

--debug**Format**`--debug`**Default**`---`**Description**

Logs debug information to the screen.

--filter**Format**`--filter <filter_name>=<filter_value>`**Default**`---`**Description**

Displays quotes where constraints do not conflict with the specified filters. For example, `glsquote --filter User=amy` will display funds usable by the user amy. You may specify multiple filter options by logically ANDing them together.

--format**Format**`--format <output_format>`**Default**`standard`**Description**

Specifies a data output format. Valid values include standard and csv.

--full**Format**`--full`

--full

Default	---
Description	Displays all attributes.

--help

Format	--help
Default	---
Description	Displays a brief help message.

--long

Format	--long
Default	---
Description	Long format. Displays multi-valued fields in a multi-line format.

--man

Format	--man
Default	---
Description	Displays full documentation.

--quiet

Format	--quiet
Default	---
Description	Suppresses headers and success messages.

-show	
Format	--show <attribute_name>[,<attribute_name>...]
Default	---
Description	<p>Displays only the specified attributes in the order specified. Valid attributes include the following:</p> <ul style="list-style-type: none"> • Amount – Quoted amount. • ChargeRate – Saved charge rates to be used when the quote is referenced. These are displayed in the format <charge_rate_name>[{<charge_rate_value>}] =<charge_rate_amount> • CreationTime – Time this quote was created. • Deleted – Boolean indicating whether this quote is deleted or not. • Description – Quote description. • Duration – Expected duration of the quoted usage in seconds. • EndTime – Time the quote becomes inactive. • Id – Quote ID. • Instance – The quote may only be used by the specified instance. • ModificationTime – Time this quote was last modified. • Pinned – Boolean indicating whether the quote is pinned or not. • RequestId – ID of the last modifying request. • StartTime – Time the quote becomes active. • TransactionId – ID of the last modifying transaction. • UsageRecord – ID of the usage record associated with the quote and containing the usage properties.

-site	
Format	--site <site_name>
Default	---
Description	Obtains a response from specified site.

--version	
Format	--version
Default	---
Description	Displays the product version.

--wide

Format	--wide
Default	---
Description	Wide format. Displays multi-valued fields in a single-line comma-separated format.

Related Topics[Querying Quotes on page 81](#)[Commands Reference on page 159](#)

glslrate

Synopsis

```
glslrate [-n <charge_rate_name>] [-x <charge_rate_value>] [--full] [--show <attribute_name>, ...] [--format <output_type>] [--debug] [--site <site_name>] [--help] [--man] [--quiet] [--version]
```

Overview

glslrate displays charge rate information.

Options

-n

Format	<code>-n <charge_rate_name></code>
Default	---
Description	Displays only charge rates of the specified name.

-x

Format	<code>-x <charge_rate_value></code>
---------------	---

-x**Default**

Description

Displays only charge rates having the specified value.

--debug**Format**

--debug

Default

Description

Logs debug information to the screen.

--format**Format**

--format <output_format>

Default

standard

Description

Specifies a data output format. Valid values include standard and csv.

--full**Format**

--full

Default

Description

Displays all attributes.

--help**Format**

--help

Default

Description

Displays a brief help message.

--man	
Format	--man
Default	---
Description	Displays full documentation.
--quiet	
Format	--quiet
Default	---
Description	Suppresses headers and success messages.
--show	
Format	--show <attribute_name>[,<attribute_name>...]
Default	---
Description	<p>Displays only the specified attributes in the order specified. Valid attributes include the following:</p> <ul style="list-style-type: none"> • Amount – Charge rate amount. The amount is an integer or decimal number and may include operators indicating how to apply the charge rate as well as divisors and time-based units. See Managing Charge Rates on page 97 for more information. • CreationTime – Time this charge rate was created. • Deleted – Boolean indicating whether this charge rate is deleted or not. • Description – Charge rate description. • ModificationTime – Time this charge rate was last modified. • Name – Charge rate name (such as Processors or License). • RequestId – ID of the last modifying request. • TransactionId – ID of the last modifying transaction. • Value – Charge rate value. For name-valued charge rates this is the usage property value corresponding to the rate. For numeric-valued charge rates this is the range of values corresponding to the rate. A blank value will function as a default charge rate. See Managing Charge Rates on page 97 for more information.

-site

Format	--site <site_name>
Default	---
Description	Obtains a response from specified site.

--version

Format	--version
Default	---
Description	Displays the product version.

Related Topics

[Querying Charge Rates on page 101](#)[Commands Reference on page 159](#)

glsrole

Synopsis

```
glsrole [--full] [--show <attribute_name>, ...] [--long] [--wide]
[--format <output_type>] [--debug] [--site <site_name>] [--help]
[–man] [–quiet] [–version] [[–r] <role_name>]
```

Overview

glsrole displays role information.

Options

-r

Format	[–r] <role_name>
Default	---

-r**Description**

Displays information for only the specified role.

--debug**Format**

--debug

Default

Description

Logs debug information to the screen.

--format**Format**

--format <output_format>

Default

standard

Description

Specifies a data output format. Valid values include standard and csv.

--full**Format**

--full

Default

Description

Displays all attributes.

--help**Format**

--help

Default

Description

Displays a brief help message.

--long**Format** --long**Default** ---**Description** Long format. Displays multi-valued fields in a multi-line format.**--man****Format** --man**Default** ---**Description** Displays full documentation.**--quiet****Format** --quiet**Default** ---**Description** Suppresses headers and success messages.**--show****Format** --show <attribute_name>[,<attribute_name>...]**Default** ---

--show**Description**

Displays only the specified attributes in the order specified. Valid attributes include the following:

- **Actions** – List of actions permitted by the role. Actions are displayed in the format object->action{instance}
- **CreationTime** – Time this role was created.
- **Deleted** – Boolean indicating whether this role is deleted or not.
- **Description** – Role description.
- **ModificationTime** – Time this role was last modified.
- **Name** – Role name.
- **RequestId** – ID of the last modifying request.
- **TransactionId** – ID of the last modifying transaction.
- **Users** – List of users granted access to the role.

--site**Format**

--site <site_name>

Default

Description

Obtains a response from specified site.

--version**Format**

--version

Default

Description

Displays the product version.

--wide**Format**

--wide

Default

Description

Wide format. Displays multi-valued fields in a single-line comma-separated format.

Related Topics

[Querying Roles on page 114](#)

[Commands Reference on page 159](#)

glstrans

Synopsis

```
glstrans [-O <object>] [-A <action>] [-k <primary_key_value>] [-U
<actor>] [-f <fund_id>] [-i <allocation_id>] [-u <user_name>]
[-a <account_name>] [-m <machine_name>] [-j <usage_record_id>]
[-J <instance_name>|<job_id>] [-s <start_time>] [-e <end_
time>] [-T <transaction_id>] [-R <request_id>] [--filter
<filter_name>=<filter_value>] [-X, --extension
<property>=<value>] ... [--full] [--show <attribute_name>, ...]
[--long] [--wide] [--format <output_type>] [-h, --hours] [--_
debug] [--site <site_name>] [--help] [--man] [--quiet] [--_
version]
```

Overview

glstrans displays transaction information.

You can customize the fields this command displays by default by setting the **transaction.show** configuration parameter in gold.conf.

Options

-a

Format -a <account_name>

Default ---

Description Displays only transactions involving the specified account.

-A

Format -A <action>

Default ---

Description Displays only transactions invoking the specified action.

-e	
Format	<code>-e <end_time></code>
Default	---
Description	Displays transactions occurring before the specified time in the format YYYY-MM-DD [hh:mm:ss] -Infinity Infinity Now
-f	
Format	<code>-f <fund_id></code>
Default	---
Description	Displays only transactions involving the specified fund.
-h	
Format	<code>-h or --hours</code>
Default	---
Description	Displays time-based credits in hours. In cases where the currency is measured in resource-seconds (like processor-seconds), the currency is divided by 3600 to display resource-hours.
-i	
Format	<code>-i <allocation_id></code>
Default	---
Description	Displays only transactions logged against the specific allocation.
-j	
Format	<code>-j <usage_record_id></code>

-j**Default**

Description

Displays only transactions affecting the given usage record.

-J**Format**`-J <instance_name|job_id>`**Default**

Description

Displays only transactions affiliated with the given instance name or job ID.

-k**Format**`-k <primary_key_value>`**Default**

Description

Displays only transactions involving the objects having the specified primary key value (i.e. having the specified Id or Name) or associations with the given parent name.

-m**Format**`-m <machine_name>`**Default**

Description

Displays only transactions involving the specified machine.

-O**Format**`-O <object>`**Default**

Description

Displays only transactions performing actions on the given object type.

-R

Format	<code>-R <request_id></code>
Default	---
Description	Displays only transactions with the specified request ID. A unique request ID is associated with each request, while each request may be associated with more than one transaction.

-S

Format	<code>-s <start_time></code>
Default	---
Description	Displays transactions occurring after the specified time in the format YYYY-MM-DD [hh:mm:ss] -Infinity Infinity Now

-T

Format	<code>-T <transaction_id></code>
Default	---
Description	Displays only transactions with the specified transaction ID. A transaction occurs when an action is invoked on an object. A complex request may involve multiple transactions.

-U

Format	<code>-u <user_name></code>
Default	---
Description	Displays only transaction involving the specified user.

-U

Format	<code>-U <actor></code>
---------------	-------------------------------

-U**Default**

Description

Displays only transactions invoked by the specified user.

-X**Format**`-X or --extension <property>=<value>`**Default**

Description

Specifies an extension property. You can specify any number of extra custom conditions.

--debug**Format**`--debug`**Default**

Description

Logs debug information to the screen.

--format**Format**`--format <output_format>`**Default**

standard

Description

Specifies a data output format. Valid values include standard and csv.

--full**Format**`--full`**Default**

Description

Displays all attributes.

--help	
Format	--help
Default	---
Description	Displays a brief help message.

--man	
Format	--man
Default	---
Description	Displays full documentation.

--quiet	
Format	--quiet
Default	---
Description	Suppresses headers and success messages.

--show	
Format	--show <attribute_name>[,<attribute_name>...]
Default	---

-show**Description**

Valid attributes include the following:

- **Account** – Account name associated with the transaction.
- **Action** – Action name.
- **Actor** – User that performed the action.
- **Allocation** – Allocation ID associated with the transaction.
- **Amount** – Amount.
- **Balance** – Effective active balance. If the allocation is active, this is the same as the remaining allocation amount (Remaining). If the allocation is inactive, this is zero.
- **Child** – If the transaction object is an association, this is the value of the child.
- **Count** – Number of objects affected by the transaction.
- **CreationTime** – Time this transaction was created.
- **Deleted** – Boolean indicating whether this transaction is deleted or not.
- **Delta** – Change (positive or negative) to the effective active balance of an allocation (Balance). This may differ in some cases from the change in the actual allocation amount (Remaining). For example, if an allocation expires, a negative Delta will be recorded for the event, while the remaining allocation amount has not changed. On the other hand, a modification of the amount in an expired allocation will be recorded as a Delta of zero.
- **Description** – Transaction description.
- **Details** – Additional assignments, conditions, options, and other details of the transaction are recorded here when there is no applicable transaction property to store them in.
- **Duration** – Expected duration of the transaction in seconds.
- **Fund** – Fund ID associated with the transaction.
- **Id** – Transaction ID.
- **Instance** – Instance name.
- **Key** – If the transaction object is an association, this is the value of the parent, otherwise this is the value of the primary key (ID or name) of the object.
- **Machine** – Machine name associated with the transaction.
- **ModificationTime** – Time this transaction was last modified.
- **Object** – Object name.
- **Remaining** – Remaining allocation amount. If an allocation amount has the potential for being affected by this transaction, this field stores the remaining allocation amount after the transaction completed. Note that for expired allocations, this will still record the allocation's actual remaining amount, even though the allocation's effective active balance (Balance) may be zero. Thus it is possible for the Remaining amount to change even though the Delta is zero or the Remaining amount to remain unchanged even though the Delta is non-zero.
- **RequestId** – ID of the last modifying request.
- **TransactionId** – ID of the last modifying transaction.
- **UsageRecord** – ID of the usage record associated with the transaction.
- **User** – User name associated with the transaction.

--site	
Format	--site <site_name>
Default	---
Description	Obtains a response from specified site.

--version	
Format	--version
Default	---
Description	Displays the product version.

Related Topics

[Querying Transactions on page 103](#)

[Commands Reference on page 159](#)

glsusage

Synopsis

```
glsusage [-T <usage_record_type>] [-J <instance_name_pattern>|<job_id_pattern>] [-S <service_id>] [-u <user_name>]
[-g <group_name>] [-a <account_name>] [-o <organization_name>]
[-c <class_name>] [-Q <quality_of_service>] [-m <machine_name>] [--stage <stage>] [-s <start_time>] [-e <end_time>] [-X,
--extension <property>=<value>]... [--full] [--show
<attribute_name>, ...] [--format <output_type>] [-h, --hours] [-d,
--debug] [-site <site_name>] [--help] [-man] [--quiet] [--version]
[[[-j] <usage_record_id>]]
```

Overview

glsusage displays usage record information.

You can customize the fields this command displays by default by setting the **usagerecord.show** configuration parameter in `gold.conf`.

Options

-a

Format	<code>-a <account_name></code>
Default	---
Description	Displays only usage records affiliated with the specified account.

-c

Format	<code>-c <class_name></code>
Default	---
C	Specifies the class or queue name.

-e

Format	<code>-e <end_time></code>
Default	---
Description	Ended before the specified time in the format YYYY-MM-DD [hh:mm:ss] -Infinity Infinity Now.

-g

Format	<code>-g <group_name></code>
Default	---
Description	Displays only usage records affiliated with the specified group

-h

Format	<code>-h</code> or <code>--hours</code>
Default	---

-h**Description**

Displays time-based credits in hours. In cases where the currency is measured in resource-seconds (like processor-seconds), the currency is divided by 3600 to display resource-hours.

-j**Format**

`[-j] <usage_record_id>`

Default

Description

Displays the usage record with the specified ID.

-J**Format**

`-J <instance_name>|<job_pattern>`

Default

Description

Displays only usage records matching the specified instance name or job id pattern. The following wildcards are supported:

- * – matches any number of characters
- ? – matches a single character

-m**Format**

`-m <machine_name>`

Default

Description

Displays only usage records affiliated with the specified machine.

-o**Format**

`-o <organization_name>`

Default

Description

Displays only usage records affiliated with the specified organization.

-Q

Format `-Q <quality_of_service>`

Default ---

Description Displays only usage records affiliated with the given quality of service .

-s

Format `-s <start_time>`

Default ---

Description Ended on or after the specified time in the format YYYY-MM-DD [hh:mm:ss] | -Infinity | Infinity | Now.

-stage

Format `--stage <stage>`

Default ---

Description Last stage completed by the usage (Quote, Create, Reserve, Charge).

-S

Format `-S <service_id>`

Default ---

Description Display on usage records associated with the specified root service ID (cloud).

-T

Format `-T <usage_record_type>`

Default ---

-T

Description	Displays only usage records associated with the specified type (such as Job or Reservation).
--------------------	--

-u

Format	<code>-u <user_name></code>
---------------	-----------------------------------

Default	---
----------------	-----

Description	Displays only usage records affiliated with the given user.
--------------------	---

-X

Format	<code>-X</code> or <code>--extension <property>=<value></code>
---------------	--

Default	---
----------------	-----

Description	Specifies an extension property. You can specify any number of extra custom conditions.
--------------------	---

--debug

Format	<code>--debug</code>
---------------	----------------------

Default	---
----------------	-----

Description	Logs debug information to the screen.
--------------------	---------------------------------------

--format

Format	<code>--format <output_format></code>
---------------	---

Default	standard
----------------	----------

Description	Specifies a data output format. Valid values include standard and csv.
--------------------	--

--full**Format**`--full`**Default**`---`**Description**

Displays all attributes.

--help**Format**`--help`**Default**`---`**Description**

Displays a brief help message.

--man**Format**`--man`**Default**`---`**Description**

Displays full documentation.

--quiet**Format**`--quiet`**Default**`---`**Description**

Suppresses headers and success messages.

--show**Format**`--show <attribute_name>[,<attribute_name>...]`**Default**`---`

--show

Description	Displays only the specified attributes in the order specified. Valid attributes include the following:
	<ul style="list-style-type: none"> • Account – Account name associated with the usage. • Charge – Cumulative amount charged. • Class – Class or queue name associated with the usage. • CPUTime – CPU time used. • CreationTime – Time this usage record was created. • Deleted – Boolean indicating whether this usage record is deleted or not. • Description – Usage description. • Disk – Amount of disk used. • Duration – Expected duration of the usage. • EndTime – Overall end time of the usage. • Id – Usage record ID. • Instance – Instance name (job ID). • JobTemplate – Job template name (cloud). • Machine – Cluster name (HPC) or virtual or physical machine name (cloud). • Memory – Amount of memory used. • ModificationTime – Time this usage record was last modified. • Nodes – Number of nodes used (HPC). • Organization – Organization name associated with the usage. • Processors – Number of cores or processors used. • QualityOfService – Quality of service associated with the usage. • Quote – Associated quote ID. • RequestId – ID of the last modifying request. • Service – Root service ID (cloud). • ServiceComponent – Leaf service ID (cloud). • Stage – Last affecting action (Create, Quote, Reserve, Query). • StartTime – Overall start time of the usage. • TransactionId – ID of the last modifying transaction. • Type – Usage record type. • User – User name associated with the usage.

--site

Format	<code>--site <site_name></code>
Default	---
Description	Obtains a response from specified site.

--version	
Format	--version
Default	---
Description	Displays the product version.

Related Topics

[Querying Usage Records on page 84](#)

[Commands Reference on page 159](#)

glsuser

Synopsis

```
glsuser [-A|-I] [-X, --extension <property>=<value>]... [-a
<account_name>] [--full] [--show <attribute_name>,...] [--
long] [--wide] [--format <output_type>] [--debug] [--site
<site_name>] [--help] [--man] [--quiet] [--version] [[-u]
<user_pattern>]
```

Overview

glsuser displays user information.

You can customize the fields this command displays by default by setting the **user.show** configuration parameter in `gold.conf`.

Options

-a	
Format	-a <account_name>
Default	---
Description	Displays only users affiliated with the specified account.

-A	
Format	<code>-A</code>
Default	---
Description	Displays only active users.

-I	
Format	<code>-I</code>
Default	---
Description	Displays only inactive users.

-U	
Format	<code>[-u] <user_pattern></code>
Default	---
Description	Displays only users matching the pattern. If you do not specify a pattern then all users are displayed. The following wildcards are supported: <ul style="list-style-type: none"> • * – matches any number of characters • ? – matches a single character

-X	
Format	<code>-X or --extension <property>=<value></code>
Default	---
Description	Specifies an extension property. You can specify any number of extra custom conditions.

--debug	
Format	<code>--debug</code>

--debug**Default**

Description

Logs debug information to the screen.

--format**Format**`--format <output_format>`**Default**

standard

Description

Specifies a data output format. Valid values include standard and csv.

--full**Format**`--full`**Default**

Description

Displays all attributes.

--help**Format**`--help`**Default**

Description

Displays a brief help message.

--long**Format**`--long`**Default**

Description

Long format. Displays multi-valued fields in a multi-line format.

--man	
Format	--man
Default	---
Description	Displays full documentation.

--quiet	
Format	--quiet
Default	---
Description	Suppresses headers and success messages.

--show	
Format	--show <attribute_name>[,<attribute_name>...]
Default	---
Description	<p>Displays only the specified attributes in the order specified. Valid attributes include the following:</p> <ul style="list-style-type: none"> • Active – Boolean indicating whether this user is active or not. • CommonName – Common name for the user. • CreationTime – Time this user was created • Deleted – Boolean indicating whether this user is deleted or not. • DefaultAccount – Default account. • Description – User description. • EmailAddress Email address. • ModificationTime – Time this user was last modified. • Name – User name. • PhoneNumber – Phone number. • Accounts – List of accounts to which the user belongs. • RequestId – ID of the last modifying request. • TransactionId – ID of the last modifying transaction.

--site	
Format	--site <site_name>

--site**Default**

Description

Obtains a response from specified site.

--version**Format**

--version

Default

Description

Displays the product version.

--wide**Format**

--wide

Default

Description

Wide format. Displays multi-valued fields in a single-line comma-separated format.

Related Topics

[Querying Users on page 43](#)[Commands Reference on page 159](#)

gmkaccount

Synopsis

```
gmkaccount [-A|-I] [-o <organization_name>] [-d <description>]
[-x, --extension <property>=<value>]... [-u [^|!][+|-]<user_
name>, ...]... [--create-fund=<boolean>] [--debug] [--site
<site_name>] [--help] [--man] [--quiet] [--verbose] [--
version] [[-a] <account_pattern>]
```

Overview

`gmkaccount` creates a new account. Users may be associated with the account. If you turn on auto-generation for the Fund object or assert the [--create-fund](#) flag, a fund will automatically be created for the account.

Options

-a	
Format	<code>-a <account_name></code>
Default	---
Description	Specifies the name of the new account

-A	
Format	<code>-A</code>
Default	---
Description	Activates the account

-d	
Format	<code>-d <description></code>
Default	---
Description	Specifies an account description.

-I	
Format	<code>-I</code>
Default	---
Description	Deactivates the account.

-o	
Format	<code>-o <organization_name></code>
Default	---
Description	Specifies the name of the organization to which the account belongs.
-u	
Format	<code>-u [^ !] [+ -] <user_name>[, [^ !] [+ -] <user_name>...]</code>
Default	---
Description	Defines user members of the account. The optional caret or exclamation symbol indicates whether the user should be created as an admin (^) or not (!) for the account. The optional plus or minus sign can precede each member to indicate whether the member should be created in the active (+) or inactive (-) state. By default, a user will be created in the active state but not an admin. You may pass multiple users may be passed to the -u option in a comma-delimited list or by specifying multiple -u options.
-X	
Format	<code>-X or --extension <property>=<value></code>
Default	---
Description	Specifies an extension property. You can specify any number of extra custom conditions.
--create-fund	
Format	<code>--create-fund <boolean></code>
Default	---
Description	Overrides the fund auto-generation setting. Setting this option to <code>True</code> creates a default fund for this account. Setting this option to <code>False</code> inhibits the creation of a default fund for this account.

--debug	
Format	--debug
Default	---
Description	Logs debug information to the screen.

--help	
Format	--help
Default	---
Description	Displays a brief help message.

--man	
Format	--man
Default	---
Description	Displays full documentation.

--quiet	
Format	--quiet
Default	---
Description	Suppresses headers and success messages.

--site	
Format	--site <site_name>
Default	---
Description	Obtains a response from specified site.

--verbose

Format	--verbose
Default	---
Description	Displays modified object details.

--version

Format	--version
Default	---
Description	Displays the product version.

Related Topics

- [Creating Accounts on page 47](#)
[Commands Reference on page 159](#)

gmkevent

Synopsis

```
gmkevent --fire-command <fire_command> [-s <fire_time>] [-e
<end_time>] [--rearm-period <rearm_period>] [--rearm-on-
failure <boolean>] [--failure-command <failure_command>] [--
notify <notification_url>] [--catch-up <boolean>] [-d
<description>] [--debug] [--site <site_name>] [--help] [--man]
[--quiet] [--verbose] [--version]
```

Overview

gmkevent creates a new event.

Options

-d	
Format	<code>-d <description></code>
Default	---
Description	Specifies an event description.

-e	
Format	<code>-e <end_time></code>
Default	---
Description	Specifies the time that this event becomes inactive.

-s	
Format	<code>-s <fire_time></code>
Default	---
Description	Specifies the target time for the event scheduler to trigger the event. The actual fire time may be dependent on the state of the server and will be recorded in the CreationTime property of the corresponding "Event Fire" Transaction. An event may also be fired manually with the goldsh Event Fire action.

--catch-up	
Format	<code>--catch-up <boolean></code>
Default	True
Description	If you set --catch-up to True and the server was down during the time this event should have fired, the event scheduler will attempt to make up for the past-due events by progressively firing them (rearming based on previous arm time) until it catches up to the present. The actions will still appear to have occurred in the present rather than in the past. If you set it to False and the server is brought back up after an outage, the event scheduler will still fire immediately for a past due event, but it will only fire once and then rearm relative to the current time.

--debug

Format	--debug
Default	---
Description	Logs debug information to the screen.

--failure-command

Format	--failure-command <goldsh_command>
Default	---
Description	Specifies the command MAM should execute if the fired command results in an unsuccessful response status. This command is expressed in a serialized form of the request identical to the syntax used in the interactive control program (goldsh). You must appropriately quote and/or escape the option argument to avoid misinterpretation or alteration by the shell.

--help

Format	--help
Default	---
Description	Displays a brief help message.

--man

Format	--man
Default	---
Description	Displays full documentation.

--notify

Format	--notify [+=] [<delivery_method>:] [<recipient>]
---------------	--

--notify	
Default	Log all event statuses to the Notification table.
Description	Causes MAM to log the result of the fired command. If the term is a -, the notification is sent only on failure. If the term is a +, the notification is sent only on success. Otherwise the notification is always sent. See the Managing Notifications on page 109 for more information about delivery method and recipient.
--quiet	
Format	--quiet
Default	---
Description	Suppresses headers and success messages.
--rearm-on-failure	
Format	--rearm-on-failure <Boolean>
Default	False
Description	If you set --rearm-on-failure to False, MAM will not rearm the event if the command was unsuccessful. If you set it to True, the event will be evaluated for rearming even if the command response has a status of Failure.
--rearm-period	
Format	--rearm-period <period>[@instant] [~ ^] !]
Default	---
Description	Specifies when the event will be rearmed. This period expression is of the form <period> [[@instant] [~ ^] !]], where <period> may be something like 1 day, 2 hours, or 5 minutes. Instant locks the period to a specific instant within the time period such as 1 day @ hour 12 or 1 month @ day 3. The modifiers indicate whether the time period should be relative to now (!), relative to the start of this (~) designator (month, minute, or other unit), or relative to the start of the first (^) designator (month, minute, or other unit). For example, assuming the fire time was 7:15, if you specified 4 hours ! as the rearm period it would be rearmed at 11:15; if you specified 4 hours ~ as the rearm period, it would be rearmed at 11:00; and if you specified 4 hours ^ as the rearm period it would be rearmed at 8:00.

-site

Format	--site <site_name>
Default	---
Description	Obtains a response from specified site.

-verbose

Format	--verbose
Default	---
Description	Displays modified object details.

-version

Format	--version
Default	---
Description	Displays the product version.

Related Topics

- [Creating Events on page 106](#)
[Commands Reference on page 159](#)

gmkfund**Synopsis**

```
gmkfund [-n <fund_name>] [--priority <fund_priority>] [--  

default-deposit <deposit_amount>] [-d <description>] [-X, --  

extension <property>=<value>]... [-u <user_name>, ...]... [-g  

<group_name>, ...]... [-a <account_name>, ...]... [-o  

<organization_name>, ...]... [-c <class_name>, ...]... [-m  

<machine_name>, ...]... [--constraint <constraint_name>=[!]  

<constraint_value>, ...]... [--parent <parent_fund_id>] [--
```

[debug](#)] [[--site <site_name>](#)] [[--help](#)] [[--man](#)] [[--quiet](#)] [[--verbose](#)] [[--version](#)]

Overview

`glsfund` creates new funds. MAM automatically generates a new ID for the fund. It essentially creates a new container into which time-bounded credits valid toward a specific set of constraints can be later credited and debited.

Options

-a

Format `-a <account_name>[,<account_name>...]`

Default ---

Description Specifies the account required by the fund. You may pass multiple accounts to the `-a` option in a comma-delimited list or by specifying multiple `-a` options may be specified.

-c

Format `-c <class_name>[,<class_name>...]`

Default ---

Description Specifies the class or queue required by the fund. You may pass multiple classes may be passed to the `-c` option in a comma-delimited list or by specifying multiple `-c` options.

-d

Format `-d <description>`

Default ---

Description Specifies a fund description.

-g

Format `-g <group_name>[,<group_name>...]`

-g	
Default	---
Description	Specifies the group required by the fund. You can pass multiple groups to the <code>-g</code> option in a comma-delimited list or by specifying multiple <code>-g</code> options.
-m	
Format	<code>-m <machine_name>[,<machine_name>...]</code>
Default	---
Description	Specifies the machine (cluster or cloud) the fund requires. You may pass multiple machines to the <code>-m</code> option in a comma-delimited list or by specifying multiple <code>-m</code> options .
-n	
Format	<code>-n <fund_name></code>
Default	---
Description	Specifies the fund name.
-o	
Format	<code>-o <organization_name>[,<organization_name>...]</code>
Default	---
Description	Specifies the organization the fund requiresd. You may pass multiple organizations may be passed to the <code>-o</code> option in a comma-delimited list or by specifying multiple <code>-o</code> options.
-u	
Format	<code>-u <user_name>[,<user_name>...]</code>
Default	---

-u**Description**

Specifies the user required by the fund. You may pass multiple users may be passed to the `-u` option in a comma-delimited list or by specifying multiple `-u` options.

-X, --extension**Format**

`-X` or `--extension <property>=<value>`

Default

Description

Specifies an extension property. You can specify any number of extra custom conditions.

--constraint**Format**

`--constraint <constraint_name>=<constraint_value>[, <constraint_name>=<constraint_value>...]`

Default

Description

Specifies a constraint for the fund. The constraint value may be a perl5 regular expression. You may prepend an exclamation point to the constraint value to express a negation of the constraint. You may specify multiple constraint options. For example, `--constraint User=amy --constraint Machine=colony` will make the credits in this fund valid only for the user amy on the machine colony. You may pass multiple constraints to the `--constraint` option in a comma-delimited list or specify multiple `--constraint`.

--debug**Format**

`--debug`

Default

Description

Logs debug information to the screen.

--default-deposit**Format**

`--default-deposit <default_amount>`

--default-deposit

Default	---
Description	Sets the default amount for any deposit that is made to this fund that does not specify a deposit amount.

--help

Format	--help
Default	---
Description	Displays a brief help message.

--man

Format	--man
Default	---
Description	Displays full documentation.

--quiet

Format	--quiet
Default	---
Description	Suppresses headers and success messages.

--parent

Format	--parent <parent_fund_id>
Default	---
Description	Associates the newly created fund as a child of the specified parent fund.

--priority	
Format	--priority <fund_priority>
Default	---
Description	Specifies the fund priority.

--site	
Format	--site <site_name>
Default	---
Description	Obtains a response from specified site.

--verbose	
Format	--verbose
Default	---
Description	Displays modified object details.

--version	
Format	--version
Default	---
Description	Displays the product version.

Related Topics

[Creating Funds on page 58](#)

[Commands Reference on page 159](#)

gmklien

Synopsis

```
gmklien [-J <instance_name>|<job_id>] [-s <start_time>] { [-e
<end_time>] | -t <lien_duration>} [-d <description>] [-X, --
extension <property>=<value>]... {-A <allocation_id><-<fund_
id>=<sublien_amount>, ...}... [--debug] [--site <site_name>] [-
--help] [--man] [--quiet] [--verbose] [--version]
```

Overview

`gmklien` creates a lien against specified allocations. MAM will create a lien object and its allocation associations. Unlike `greserve`, MAM will not return a calculated lien amount or create a usage record with the lien.

i This command bypasses the normal mechanisms that prevent more liens from being placed against an allocation than it can support.

Options

-A

Format

`-A <allocation_id><-<fund_id>=<sublien_amount>[,<allocation_id><-<fund_id>=<sublien_amount>...]`

Default

Description

Creates subliens against the specified allocations. You must specify at least one allocation expression with the lien. You may pass multiple allocation expressions to the `-A` option in a comma-delimited list or by specifying multiple `-A` options.

-d

Format

`-d <description>`

Default

Description

Specifies a description for the lien.

-e

Format

`-e <end_time>`

-e	
Default	Now
Description	Specifies the expiration time for the lien in the format YYYY-MM-DD [hh:mm:ss] -Infinity Infinity Now.
-J	
Format	<code>[-J] <instance_name> <job_id></code>
Default	---
Description	Specifies the instance name or job ID for the lien.
-s	
Format	<code>-s <start_time></code>
Default	Now
Description	Specifies a new start time for the lien in the format YYYY-MM-DD [hh:mm:ss] -Infinity Infinity Now.
-t	
Format	<code>-t <lien_duration></code>
Default	Lien end time minus start time
Description	Specifies the duration of the lien in seconds.
-X, --extension	
Format	<code>-X or --extension <property>=<value></code>
Default	---
Description	Modifies an extension property. You can specify any number of extra field assignments.

--debug**Format** --debug**Default** ---**Description** Logs debug information to the screen.**--help****Format** --help**Default** ---**Description** Displays a brief help message.**--man****Format** --man**Default** ---**Description** Displays full documentation.**--quiet****Format** --quiet**Default** ---**Description** Suppresses headers and success messages.**--site****Format** --site <site_name>**Default** ---**Description** Obtains a response from specified site.

--verbose	
Format	--verbose
Default	---
Description	Displays modified object details.

--version	
Format	--version
Default	---
Description	Displays the product version.

Related Topics

[Creating Liens on page 76](#)

[Commands Reference on page 159](#)

gmkorg

Synopsis

```
gmkorg [-d <description>] [-X, --extension
<property>=<value>]... [--debug] [--site <site_name>] [--help]
[--man] [--quiet] [--verbose] [--version] {[[-o] <organization_
name>]}
```

Overview

gmkorg creates a new organization.

Options

-d	
Format	-d <description>

-d**Default**

Description

Specifies a description for the organization.

-o**Format**`-o <organization_name>`**Default**

Description

Specifies the name of the organization.

-X, --extension <property>**Format**`-X or --extension <property>=<value>`**Default**

Description

Modifies an extension property. You can specify any number of extra field assignments.

--debug**Format**`--debug`**Default**

Description

Logs debug information to the screen.

--help**Format**`--help`**Default**

Description

Displays a brief help message.

--man	
Format	--man
Default	---
Description	Displays full documentation.

--quiet	
Format	--quiet
Default	---
Description	Suppresses headers and success messages.

--site	
Format	--site <site_name>
Default	---
Description	Obtains a response from specified site.

--verbose	
Format	--verbose
Default	---
Description	Displays modified object details.

--version	
Format	--version
Default	---
Description	Displays the product version.

Related Topics

- [Creating Organizations on page 53](#)
- [Commands Reference on page 159](#)

gmkquote

Synopsis

```
gmkquote [[--pin] [-J <instance_name>|<job_id>] | --nopin] [-s <start_time>] {-e <end_time> | -t <quote_duration>} [-d <description>] [-X, --extension <property>=<value>]... { -R <charge_rate_name>[ {<charge_rate_value>} ]=<charge_rate_amount>, ... }... [--debug] [--site <site_name>] [--help] [--man] [--quiet] [--verbose] [--version]
```

Overview

gmkquote creates a new chargeable quote template. MAM will create a quote object and its associated charge rates. Instances referencing the quote will use the override charge rates specified in the command. Unlike [gquote](#), gmkquote will not return a calculated quote amount or create a usage record with the quote.

Options

-d	
Format	<code>-d <description></code>
Default	---
Description	Specifies a description of the quote.

-e	
Format	<code>-e <end_time></code>
Default	---

-e

Description	Specifies the expiration time for the quote template in the format YYYY-MM-DD [hh:mm:ss] -Infinity Infinity Now. The rates associated with this quote may not be claimed after this time. If you do not specify an end time but did specify a duration, MAM will calculate the end time as start time + duration. If you specify both end time and duration but they are inconsistent, MAM will ignore the duration specification.
--------------------	--

-J

Format	<code>-J <instance_name> <job_id></code>
Default	---
Description	Specifies the instance name, such as a job ID, for the quote. You cannot specify an instance name if the quote is unpinned.

-R

Format	<code>-R <charge_rate_name>[{<charge_rate_value>}]=<charge_rate_amount> [, <charge_rate_name>[{<charge_rate_value>}]=<charge_rate_amount>...]</code>
Default	---
Description	Associates the specified charge rates with the quote. You must specify at least one charge rate expression with the quote. You may pass multiple charge rate expressions to the <code>-R</code> option in a comma-delimited list or specify multiple <code>-R</code> options.

-s

Format	<code>-s <start_time></code>
Default	Now
Description	Specifies a beginning time for the quote template in the format YYYY-MM-DD [hh:mm:ss] -Infinity Infinity Now. The rates associated with this quote may not be claimed before this time.

-t

Format	<code>-t <quote_duration></code>
Default	---
Description	Specifies the amount of time in seconds the rates in the quote template may be used. MAM uses the duration to calculate an end time (start time + duration) as an alternative to specifying the end time.

-X, --extension <property>

Format	<code>-X or --extension <property>=<value></code>
Default	---
Description	Modifies an extension property. You can specify any number of extra field assignments.

--debug

Format	<code>--debug</code>
Default	---
Description	Logs debug information to the screen.

--help

Format	<code>--help</code>
Default	---
Description	Displays a brief help message.

--man

Format	<code>--man</code>
---------------	--------------------

--man	
Default	---
Description	Displays full documentation.

--nopin	
Format	--nopin
Default	Not set
Description	Indicates that the quote is not to be pinned to a specific instance. An instance may use an unpinned quote while the quote is active.

--pin	
Format	--pin
Default	Set
Description	Indicates that the quote will be pinned to a specific instance. If you do not specify the instance when you create the quote, the first instance to claim it will become the pinned instance. Once a quote is pinned to a particular instance, no other instances may use the quote.

--quiet	
Format	--quiet
Default	---
Description	Suppresses headers and success messages.

--site	
Format	--site <site_name>
Default	---
Description	Obtains a response from specified site.

--verbose**Format** --verbose**Default** ---**Description** Displays modified object details.**--version****Format** --version**Default** ---**Description** Displays the product version.

Related Topics

[Creating Quote Templates on page 80](#)[Commands Reference on page 159](#)

gmkrat

Synopsis

```
gmkrat -n <charge_rate_name> [-x <charge_rate_value>] [-d <description>] [--debug] [--site <site_name>] [--help] [--man] [--quiet] [--verbose] [--version] {[-z] <charge_rate_amount>}
```

Overview

gmkrat creates a new charge rate.

Options

-d**Format** -d <description>**Default** ---

-d	
Description	Specifies a charge rate description.
-n	
Format	<code>-n <charge_rate_name></code>
Default	---
Description	Specifies the name of the usage record property for which the rate is charging, such as Processors or QualityOfService.
-x	
Format	<code>-x <charge_rate_value></code>
Default	---
Description	Specifies charge rate value. For name-valued charge rates, this is the usage property value corresponding to the rate. For numeric-valued charge rates, this is the range of values corresponding to the rate. A blank value will function as a default charge rate. See Managing Charge Rates on page 97 for more information.
-z	
Format	<code>-z <charge_rate_amount></code>
Default	---
Description	Specifies the rate for the charge. This is an integer or decimal number and may include operators that indicate how the charge is applies as well as divisors and time-based units. See Managing Charge Rates on page 97 for more information.
--debug	
Format	<code>--debug</code>
Default	---

--debug**Description**

Logs debug information to the screen.

--help**Format**

--help

Default

Description

Displays a brief help message.

--man**Format**

--man

Default

Description

Displays full documentation.

--quiet**Format**

--quiet

Default

Description

Suppresses headers and success messages.

--site**Format**

--site <site_name>

Default

Description

Obtains a response from specified site.

--verbose	
Format	--verbose
Default	---
Description	Displays modified object details.

--version	
Format	--version
Default	---
Description	Displays the product version.

Related Topics

- [Creating Charge Rates on page 99](#)
- [Commands Reference on page 159](#)

gmkrole

Synopsis

```
gmkrole [-d <description>] [-u <user_name>, ...]... [-A <object_name>--><action_name>[<instance_name>], ...]... [--debug] [--site <site_name>] [--help] [--man] [--quiet] [--verbose] [--version] {[ -r] <role_name>}
```

Overview

gmkrole creates a new role. You can associate users and actions with the role at creation time.

Options

-A**Format**

`-A <object_name>-><action_name>[{<instance_name>}] [, <object_name>-><action_name>[{<instance_name>}] ...]`

Default

ANY

Description

Adds actions to the role. You must specify the object, action and instance in the form shown. You may pass multiple actions to the `-A` option in a comma-delimited list or specify multiple `-A` options.

-d**Format**

`-d <description>`

Default

Description

Specifies a role description.

-r**Format**

`[-r] <role_name>`

Default

Description

Specifies the name of the new role.

-u**Format**

`-u <user_name>[, <user_name>...]`

Default

Description

Adds users to the role. You may pass multiple users to the `-u` option in a comma-delimited list or specify multiple `-u` options.

--debug**Format**

`--debug`

--debug

Default	---
Description	Logs debug information to the screen.

--help

Format	--help
Default	---
Description	Displays a brief help message.

--man

Format	--man
Default	---
Description	Displays full documentation.

--quiet

Format	--quiet
Default	---
Description	Suppresses headers and success messages.

--site

Format	--site <site_name>
Default	---
Description	Obtains a response from specified site.

--verbose

Format	--verbose
Default	---
Description	Displays modified object details.

--version

Format	--version
Default	---
Description	Displays the product version.

Related Topics

- [Creating Roles on page 113](#)
[Commands Reference on page 159](#)

gmkusage

Synopsis

```
gmkusage [-T <usage_record_type>] [-S <service_id>] [-u <user_name>]
[-g <group_name>] [-a <account_name>] [-o <organization_name>]
[-c <class_name>] [-Q <quality_of_service>] [-m <machine_name>]
[-N <nodes>] [-P <processors>] [-C <cpu_time>] [-M <memory>]
[-D <disk>] [-t <usage_duration>] [-s <start_time>] [-e <end_time>]
[-d <description>] [-X, --extension <property>=<value>]...
[--debug] [--site <site_name>] [--help] [--man] [--quiet]
[--verbose] [--version] {[ -J] <instance_name>} <job_id>
```

Overview

gmkusage creates a new usage record.

Options

-a	
Format	<code>-a <account_name></code>
Default	---
Description	Specifies an account name.

-c	
Format	<code>-c <class_name></code>
Default	---
Description	Specifies a class or queue.

-C	
Format	<code>-C <cpu_time></code>
Default	---
Description	Specifies CPU time used.

-d	
Format	<code>-d <description></code>
Default	---
Description	Specifies a description.

-D	
Format	<code>-D <disk></code>
Default	---
Description	Specifies the amount of disk used.

-e

Format	<code>-e <end_time></code>
Default	---
Description	Specifies the date and time the usage ended in the format YYYY-MM-DD [hh:mm:ss] -Infinity Infinity Now.

-g

Format	<code>-g <group_name></code>
Default	---
Description	Specifies a group name.

-J

Format	<code>[-J] <instance_name> <job_id></code>
Default	---
Description	Specifies the instance name or job ID of the new usage record.

-m

Format	<code>-m <machine_name></code>
Default	---
Description	Specifies the name of the cluster, cloud, or system used.

-M

Format	<code>-M <memory></code>
Default	---

-M

Description	Specifies the amount of memory used.
--------------------	--------------------------------------

-N

Format	<code>-N <nodes></code>
---------------	-------------------------------

Default	---
----------------	-----

Description	Specifies the number of nodes used.
--------------------	-------------------------------------

-O

Format	<code>-o <organization_name></code>
---------------	---

Default	---
----------------	-----

Description	Specifies an organization name.
--------------------	---------------------------------

-P

Format	<code>-P <processors></code>
---------------	------------------------------------

Default	---
----------------	-----

Description	Specifies the number of processors used.
--------------------	--

-Q

Format	<code>-Q <quality_of_service></code>
---------------	--

Default	---
----------------	-----

Description	Specifies the quality of service used.
--------------------	--

-s

Format	<code>-s <start_time></code>
Default	---
Description	Specifies the date and time the usage started in the format YYYY-MM-DD [hh:mm:ss] -Infinity Infinity Now.

-S

Format	<code>-S <service_id></code>
Default	---
Description	Specifies the root service ID (cloud).

-t

Format	<code>-t <usage_duration></code>
Default	---
Description	Specifies the amount of time used in seconds.

-T

Format	<code>-T <usage_record_type></code>
Default	---
Description	Specifies the usage record type (Job or reservation, for example).

-u

Format	<code>-u <user_name></code>
Default	---

-u**Description**

Specifies the user name.

-X, --extension**Format**`-X or --extension <property>=<value>`**Default**

Description

Modifies an extension property. You can specify any number of extra field assignments.

--debug**Format**`--debug`**Default**

Description

Logs debug information to the screen.

--help**Format**`--help`**Default**

Description

Displays a brief help message.

--man**Format**`--man`**Default**

Description

Displays full documentation.

--quiet**Format**`--quiet`**Default**`---`**Description**

Suppresses headers and success messages.

--site**Format**`--site <site_name>`**Default**`---`**Description**

Obtains a response from specified site.

--verbose**Format**`--verbose`**Default**`---`**Description**

Displays modified object details.

--version**Format**`--version`**Default**`---`**Description**

Displays the product version.

Related Topics

[Creating a Usage Record on page 83](#)[Commands Reference on page 159](#)

Synopsis

```
gmkuser [-A|-I] [-n <common_name>] [-F <phone_number>] [-E
<email_address>] [-a <default_account>] [-d <description>] [-
X, --extension <property>=<value>]... [--debug] [--site <site_
name>] [--help] [--man] [--quiet] [--verbose] [--version] {[-
u] <user_name>}
```

Overview

gmkuser creates a new user.

Options

-a

Format	<code>-a <default_account></code>
Default	---
Description	Specifies the account MAM will charge when no account is specified.

-A

Format	<code>-A</code>
Default	---
Description	Activates the user.

-d

Format	<code>-d <description></code>
Default	---
Description	Specifies a user description.

-E

Format	<code>-E <email_address></code>
---------------	---------------------------------------

Default	---
----------------	-----

Description	Specifies an email address.
--------------------	-----------------------------

-F

Format	<code>-F <phone_number></code>
---------------	--------------------------------------

Default	---
----------------	-----

Description	Specifies a phone number.
--------------------	---------------------------

-I

Format	<code>-I</code>
---------------	-----------------

Default	---
----------------	-----

Description	Deactivates the user.
--------------------	-----------------------

n

Format	<code>-n <common_name></code>
---------------	-------------------------------------

Default	---
----------------	-----

Description	Specifies the common name for the user.
--------------------	---

-u

Format	<code>[-u] <user_name></code>
---------------	-------------------------------------

Default	---
----------------	-----

Description	Specifies the new user's ID or name.
--------------------	--------------------------------------

-X, --extension <property>

Format	<code>-X or --extension <property>=<value></code>
Default	---
Description	Modifies an extension property. You can specify any number of extra field assignments.

--debug

Format	<code>--debug</code>
Default	---
Description	Logs debug information to the screen.

--help

Format	<code>--help</code>
Default	---
Description	Displays a brief help message.

--man

Format	<code>--man</code>
Default	---
Description	Displays full documentation.

--quiet

Format	<code>--quiet</code>
Default	---
Description	Suppresses headers and success messages.

-site

Format	--site <site_name>
Default	---
Description	Obtains a response from specified site.

-verbose

Format	--verbose
Default	---
Description	Displays modified object details.

-version

Format	--version
Default	---
Description	Displays the product version.

Related Topics

- [Creating Users on page 43](#)
[Commands Reference on page 159](#)

gquote**Synopsis**

```
gquote [-T <usage_record_type>] [-S <service_id>] [-u <user_name>]
[-g <group_name>] [-a <account_name>] [-o <organization_name>]
[-c <class_name>] [-Q <quality_of_service>] [-m <machine_name>]
[-N <nodes>] [-P <processors>] [-C <cpu_time>] [-M <memory>]
[-D <disk>] [-X, --extension <property>=<value>]... [-t <quote_duration>]
[-G <grace_duration>] [-s <quote_start_time>] [-e <quote_end_time>]
[-d <quote_description>] [-z <quote_amount>] [--cost-only | --
```

```
guarantee] [-R <charge_rate_name>[<charge_rate_value>}] [=<charge_rate_amount>, ...] ... [-h, --hours] [--itemize] [--debug] [--site <site_name>] [--help] [--man] [--quiet] [--verbose] [--version] [[-j] <usage_record_id>] [-q <quote_template_id>] [-J <instance_name>|<job_id>]
```

Overview

`gquote` obtains a quote for usage. This command and its options can estimate the cost of using resources or services, validate that a requester has sufficient access and funds to use the requested resources or services, and guarantee that the charge rates used to generate the quote do not change when applying subsequent liens and charges.

Options

-a	
Format	<code>-a <account_name></code>
Default	---
Description	Specifies the account name.

-c	
Format	<code>-c <class_name></code>
Default	---
Description	Specifies the class or queue used.

-C	
Format	<code>-C <cpu_time></code>
Default	---
Description	Specifies the estimated CPU time used.

-d

Format	<code>-d <quote_description></code>
Default	---
Description	Specifies an explanatory message for the quote. You can pass overall usage description via the extension property option (<code>-X Description=<description></code>).

-D

Format	<code>-D <disk></code>
Default	---
Description	Specifies the amount of disk used.

-e

Format	<code>-e <quote_end_time></code>
Default	---
Description	Specifies the end time for the quote in the format <code>YYYY-MM-DD [hh:mm:ss] -Infinity Infinity Now</code> .

-g

Format	<code>-g <group_name></code>
Default	---
Description	Specifies the group name.

-G

Format	<code>-G <grace_duration></code>
---------------	--

-G	
Default	---
Description	Specifies a grace period in seconds. If you specify the quote duration but not the quote end time, MAM will calculate the quote end time as the quote start time plus the quote duration plus the grace duration.

-h	
Format	<code>-h</code> or <code>--hours</code>
Default	---
Description	Displays time-based credits in hours. In cases where the currency is measured in resource-seconds (like processor-seconds), the currency is divided by 3600 to display resource-hours.

-j	
Format	<code>[-j] <usage_record_id></code>
Default	---
Description	Usage record ID for the quote if already created with <code>gmkusage</code> or a previous <code>gquote</code> .

-J	
Format	<code>-J <instance_name job_id></code>
Default	---
Description	Instance name or the job ID of the quote if known.

-m	
Format	<code>-m <machine_name></code>
Default	---
Description	Specifies the name of the cluster, cloud, or system used.

-M

Format	<code>-M <memory></code>
Default	---
Description	Specifies the amount of memory used.

-N

Format	<code>-N <nodes></code>
Default	---
Description	Specifies the number of nodes used.

-O

Format	<code>-o <organization_name></code>
Default	---
Description	Specifies the organization name.

-P

Format	<code>-P <processors></code>
Default	---
Description	Specifies the number of processors used.

-q

Format	<code>-q <quote_template_id></code>
Default	---
Description	Specifies quote template used to override standard charge rates.

-Q	
Format	<code>-Q <quality_of_service></code>
Default	---
Description	Specifies the quality of service used.
-R	
Format	<code>-R <charge_rate_name>[{<charge_rate_value>}]=<charge_rate_amount> [, <charge_rate_name>[{<charge_rate_value>}]=<charge_rate_amount>...]</code>
Default	---
Description	Uses the specified charge rates in the quote. The specified rates override the general rates. If you specify the --guarantee option, these charge rates will be saved and used when this quote is referenced in a charge action. You may pass multiple charge rates to the -R option in a comma-delimited list or you can specify multiple -R options.
-S	
Format	<code>-s <quote_start_time></code>
Default	---
Description	Specifies the start time for the quote in the format YYYY-MM-DD [hh:mm:ss] -Infinity Infinity Now. You can pass the overall usage start time via the extension property option (-X StartTime=<start_time>).
-S	
Format	<code>-S <service_id></code>
Default	---
Description	Specifies the root service ID (cloud).

-t

Format	<code>-t <quote_duration></code>
Default	---
Description	Specifies the wallclock duration for the quote in seconds. You can pass the estimated wallclock time for the total lifetime usage via the extension property option (<code>-X Duration=<usage_duration></code>).

-T

Format	<code>-T <usage_record_type></code>
Default	---
Description	Specifies the usage record type, such as job or reservation.

-u

Format	<code>-u <user_name></code>
Default	---
Description	Specifies the user name.

-X

Format	<code>-X or --extension <property>=<value></code>
Default	---
Description	Specifies an extension property. You can specify any number of extra custom conditions.

-z

Format	<code>-z <quote_amount></code>
---------------	--------------------------------------

-z	
Default	---
Description	Specifies the quote amount if calculated externally.

--cost-only	
Format	--cost-only
Default	---
Description	Returns the cost, ignoring all balance and validity checks. This option is mutually exclusive with --guarantee .

--debug	
Format	--debug
Default	---
Description	Logs debug information to the screen.

--guarantee	
Format	--guarantee
Default	---
Description	Guarantees the quote and returns a quote ID to secure the current charge rates. This results in the creation of a quote record and a permanent usage record. This option is mutually exclusive with --cost-only .

--help	
Format	--help
Default	---
Description	Displays a brief help message.

--itemize

Format	--itemize
Default	---
Description	Returns the composite charge information in the response data. You must use this in conjunction with the --verbose flag to display the data.

--man

Format	--man
Default	---
Description	Displays full documentation.

--quiet

Format	--quiet
Default	---
Description	Suppresses headers and success messages.

--site

Format	--site <site_name>
Default	---
Description	Obtains a response from specified site.

--verbose

Format	--verbose
Default	---

--verbose	
Description	Display modified object details.
--version	
Format	--version
Default	---
Description	Displays the product version.

Related Topics

[Obtaining Usage Quotes on page 85](#)
[Commands Reference on page 159](#)

grefund

Synopsis

```
grefund [-z <amount>] [-f <fund_id>] [-d <description>] [-A] [--filter <filter_name>=<filter_value>]... [--filter-type <filter_type>] [-h, --hours] [--debug] [--site <site_name>] [-h, --help] [--man] [--quiet] [--verbose] [--version] {-J <instance_name>}|<job_id> | [-j] <usage_record_id>}
```

Overview

grefund issues a refund for the specified usage. The command will return a list of usage records if the usage search does not yield a unique match. If you do not specify an amount, MAM will credit the appropriate allocations for the full amount the overall usage was charged. You may specify a lesser amount for a partial refund.

Options

-A	
Format	-A

-A	
Default	---
Description	When you use filters to select the fund to be refunded, specifies that only active allocations should be considered.
-d	
Format	<code>-d <description></code>
Default	---
Description	Specifies an explanatory message for the refund.
-f	
Format	<code>-f <fund_id></code>
Default	---
Description	Specifies the fund to be refunded to. If you omit this, MAM will look in the transaction table to determine to which fund it should deposit the refund.
-h, --hours	
Format	<code>-h</code> or <code>--hours</code>
Default	---
Description	Treats currency as specified in hours. In systems where the currency is measured in resource-seconds (like processor-seconds), this option allows the amount to be specified in resource-hours.
-j	
Format	<code>[-j] <usage_record_id></code>
Default	---

-j**Description**

Specifies the unique usage record identifier the accounting manager assigns to distinguish between usage with non-unique instance names.

-J**Format**

`-J <instance_name>|<job_id>`

Default

Description

Specifies the name of the instance (such as a job ID assigned by the local resource manager). This ID might not be unique among the historical list of usage records the accounting manager manages.

-z**Format**

`-z <amounts>`

Default

Description

Specifies the amount to refund. This amount must be non-negative and less than or equal to the amount charged for the overall usage.

--debug**Format**

`--debug`

Default

Description

Logs debug information to the screen.

--filter**Format**

`--filter <filter_name>=<filter_value>`

Default

--filter**Description**

By default, MAM refunds amounts to the same fund from which it took them. If you want to override this, by specifying filters you may restrict the fund to be refunded to one whose constraints are consistent with the specified filters. For example, `grefund --filter User=amy` will refund the amount to the fund usable by the user `amy`. You may specify multiple filter options by logically ANDing them together.

--filter-type**Format**

`--filter-type <filter_type>`

Default

NonExclusive

Description

Selects the filtering type. If you use the `Exclusive` filter type, a fund will only be matched if the specified filters meet all constraints. If you use the `NonExclusive` filter type, a fund will be matched as long as the specified filters do not conflict with the constraints.

--help**Format**

`--help`

Default

Description

Displays a brief help message.

--man**Format**

`--man`

Default

Description

Displays full documentation.

--quiet**Format**

`--quiet`

Default

--quiet**Description**

Suppresses headers and success messages.

--site**Format**`--site <site_name>`**Default**

Description

Obtains a response from specified site.

--verbose**Format**`--verbose`**Default**

Description

Displays modified object details.

--version**Format**`--version`**Default**

Description

Displays the product version.

Related Topics

[Issuing Usage Refunds on page 88](#)[Commands Reference on page 159](#)

greserve

Synopsis

```
greserve [-T <usage_record_type>] [-S <service_id>] [-u <user_name>]
          [-g <group_name>] [-a <account_name>] [-o
```

```

<organization_name> [-c <class_name>] [-Q <quality_of_
service>] [-m <machine_name>] [-N <nodes>] [-P <processors>]
[-C <cpu_time>] [-M <memory>] [-D <disk>] [-X, --extension
<property>=<value>]... [-t <lien_duration>] [-G <grace_
duration>] [-s <lien_start_time>] [-e <lien_end_time>] [-d
<lien_description>] [-z <lien_amount>] [--modify|--replace] [-_
R <charge_rate_name>[<charge_rate_value>]]=<charge_rate_
amount>, ...]... [-h, --hours] [--itemize] [--debug] [--site
<site_name>] [--help] [--man] [--quiet] [--verbose] [--_
version] [[-j] <usage_record_id>] [-q <quote_id>] [-J
<instance_name>|<job_id>]

```

Overview

greserve obtains a lien for usage.

Options

-a	
Format	<code>-a <account_name></code>
Default	---
Description	Specifies the account name.

-c	
Format	<code>-c <class_name></code>
Default	---
Description	Specifies the class or queue used.

-C	
Format	<code>-C <cpu_time></code>
Default	---
Description	Specifies the estimated CPU time used.

-d	
Format	<code>-d <lien_description></code>
Default	---
Description	Specifies an explanatory message for the lien. You can pass overall usage description via the extension property option (<code>-X Description=<description></code>).
-D	
Format	<code>-D <disk></code>
Default	---
Description	Specifies the amount of disk used.
-e	
Format	<code>-e <lien_end_time></code>
Default	<i>Now</i>
Description	Specifies the end time for the lien in the format <code>YYYY-MM-DD [hh:mm:ss] -Infinity Infinity Now</code> .
-g	
Format	<code>-g <group_name></code>
Default	---
Description	Specifies the group name.
-G	
Format	<code>-G <grace_duration></code>

-G	
Default	---
Description	Specifies a grace period in seconds. If you specify the lien duration but not the lien end time, MAM will calculate the lien end time as the lien start time plus the lien duration plus the grace duration.
-h	
Format	<code>-h</code> or <code>--hours</code>
Default	---
Description	Displays time-based credits in hours. In cases where the currency is measured in resource-seconds (like processor-seconds), the currency is divided by 3600 to display resource-hours.
-j	
Format	<code>[-j] <usage_record_id></code>
Default	---
Description	Usage record ID for the lien if already created with gmkusage , gquote , or a previous greserve . This places a hold against an existing usage record if the instance name (such as a job ID) is ambiguous or if usage has already been debited and you want to reserve an additional amount associated with the same usage record.
-J	
Format	<code>-J <instance_name job_id></code>
Default	---
Description	Instance name or the job ID of the lien if known. This can sometimes be non-unique, such as when a resource manager recycles job ids, and does not always unambiguously identify a usage record to reserve. In such cases, look up and specify the usage record ID for the lien.

-m	
Format	<code>-m <machine_name></code>
Default	---
Description	Specifies the name of the cluster, cloud, or system used.

-M	
Format	<code>-M <memory></code>
Default	---
Description	Specifies the amount of memory used..

-N	
Format	<code>-N <nodes></code>
Default	---
Description	Specifies the number of nodes used.

-O	
Format	<code>-o <organization_name></code>
Default	---
Description	Specifies the organization name.

-P	
Format	<code>-P <processors></code>
Default	---
Description	Specifies the number of processors used.

-q

Format	<code>[-q] <quote_id></code>
Default	---
Description	Specifies the quote used to determine charge rates.

-Q

Format	<code>-Q <quality_of_service></code>
Default	---
Description	Specifies the quality of service used.

-R

Format	<code>-R <charge_rate_name>[<charge_rate_value>]=<charge_rate_amount> [,<charge_rate_name>[<charge_rate_value>]]=<charge_rate_amount>...]</code>
Default	---
Description	Uses the specified charge rates in the lien. The specified rates override the general rates or rates guaranteed through a quote. You may pass multiple charge rates to the <code>-R</code> option in a comma-delimited list or specify multiple <code>-R</code> options.

-S

Format	<code>-s <lien_start_time></code>
Default	---
Description	Specifies the start time for the lien in the format <code>YYYY-MM-DD [hh:mm:ss] -Infinity Infinity Now</code> . You can pass the overall usage start time via the extension property option (<code>-X StartTime=<start_time></code>).

-S	
Format	<code>-S <service_id></code>
Default	---
Description	Specifies the root service ID (cloud).

-t	
Format	<code>-t <lien_duration></code>
Default	---
Description	Specifies the wallclock duration for the lien in seconds. You can pass the estimated wallclock time for the total lifetime usage via the extension property option (<code>-X Duration=<usage_duration></code>).

-T	
Format	<code>-T <usage_record_type></code>
Default	---
Description	Specifies the usage record type, such as job or reservation.

-u	
Format	<code>-u <user_name></code>
Default	---
Description	Specifies the user name.

-X	
Format	<code>-X or --extension <property>=<value></code>

-X	
Default	---
Description	Specifies an extension property. You can specify any number of extra custom conditions.
-z	
Format	<code>-z <lien_amount></code>
Default	---
Description	Specifies the lien amount if calculated externally.
--debug	
Format	<code>--debug</code>
Default	---
Description	Logs debug information to the screen.
--help	
Format	<code>--help</code>
Default	---
Description	Displays a brief help message.
--itemize	
Format	<code>--itemize</code>
Default	---
Description	Returns the composite charge information in the response data. You must use this in conjunction with the --verbose flag to display the data.

--man	
Format	--man
Default	---
Description	Displays full documentation.

--modify	
Format	--modify
Default	---
Description	Causes the reserve operation to augment existing liens instead of creating new ones. This new option is mutually exclusive with the --replace option which deletes existing matching liens and recreates a new one. The default action is to create a new lien even if a lien for an instance of the same name exists. The modify behavior supports extending liens out dynamically and is often used with incremental charging.

--quiet	
Format	--quiet
Default	---
Description	Suppresses headers and success messages.

--replace	
Format	--replace
Default	---
Description	If you specify this option, MAM will delete similarly named liens before creating this lien. The default action is to create a new lien while leaving any existing liens for instances of the same name. The replace option should be specified if you want this lien to replace existing liens for instances of the same name such as when a system reuses instance names. This new option is mutually exclusive with the --modify option which modifies any existing matching lien instead of creating a new one.

-site

Format	--site <site_name>
Default	---
Description	Obtains a response from specified site.

-verbose

Format	--verbose
Default	---
Description	Display modified object details.

-version

Format	--version
Default	---
Description	Displays the product version.

Related Topics

- [Making a Usage Lien on page 86](#)
[Commands Reference on page 159](#)

grmaccount

Synopsis

```
grmaccount [--debug] [--site <site_name>] [--help] [--man] [--quiet] [--verbose] [--version] {[-a] <account_name>}
```

Overview

grmaccount **deletes** an account.

Options

-a

Format	<code>[-a] <account_name></code>
Default	---
Description	Specifies the name of the account to be deleted.

--debug

Format	<code>--debug</code>
Default	---
Description	Logs debug information to the screen.

--help

Format	<code>--help</code>
Default	---
Description	Displays a brief help message.

--man

Format	<code>--man</code>
Default	---
Description	Displays full documentation.

--quiet

Format	<code>--quiet</code>
Default	---

--quiet**Description**

Suppresses headers and success messages.

--site**Format**`--site <site_name>`**Default**

Description

Obtains a response from specified site.

--verbose**Format**`--verbose`**Default**

Description

Display modified object details.

--version**Format**`--version`**Default**

Description

Displays the product version.

Related Topics

[Deleting Accounts on page 50](#)[Commands Reference on page 159](#)

grmalloc

Synopsis

```
grmalloc (-I | {-i allocation_id}) [--debug] [--site <site_name>] [--help] [--man] [--quiet] [--verbose] [--version]
```

Overview

grmalloc deletes an allocation or purges stale allocations.

Options

-i	
Format	<code>[-i] <allocation_id></code>
Default	---
Description	Specifies the allocation to be deleted.

-I	
Format	<code>-I</code>
Default	---
Description	Deletes inactive allocations.

--debug	
Format	<code>--debug</code>
Default	---
Description	Logs debug information to the screen.

--help	
Format	<code>--help</code>
Default	---
Description	Displays a brief help message.

--man**Format** --man**Default** ---**Description** Displays full documentation.**--quiet****Format** --quiet**Default** ---**Description** Suppresses headers and success messages.**--site****Format** --site <site_name>**Default** ---**Description** Obtains a response from specified site.**--verbose****Format** --verbose**Default** ---**Description** Display modified object details.**--version****Format** --version**Default** ---**Description** Displays the product version.

Related Topics

[Deleting Allocations on page 72](#)
[Commands Reference on page 159](#)

grmevent

Synopsis

```
grmevent [--debug] [--site <site_name>] [--help] [--man] [--quiet] [--verbose] [--version] {-E} <event_id>
```

Overview

grmevent deletes an event.

Options

-E	
Format	<code>[-E] <event_id></code>
Default	---
Description	Specifies the ID of the event to be deleted.

--debug	
Format	<code>--debug</code>
Default	---
Description	Logs debug information to the screen.

--help	
Format	<code>--help</code>
Default	---
Description	Displays a brief help message.

--man**Format** --man**Default** ---**Description** Displays full documentation.**--quiet****Format** --quiet**Default** ---**Description** Suppresses headers and success messages.**--site****Format** --site <site_name>**Default** ---**Description** Obtains a response from specified site.**--verbose****Format** --verbose**Default** ---**Description** Display modified object details.**--version****Format** --version**Default** ---**Description** Displays the product version.

Related Topics

[Deleting Events on page 107](#)

[Commands Reference on page 159](#)

grmfund

Synopsis

```
grmfund [--debug] [--site <site_name>] [--help] [--man] [--quiet] [--verbose] [--version] {-f} <fund_id>
```

Overview

grmfund deletes a fund.

Options

-f

Format	<code>[-f] <fund_id></code>
Default	---
Description	Specifies the fund to be deleted.

--debug

Format	<code>--debug</code>
Default	---
Description	Logs debug information to the screen.

--help

Format	<code>--help</code>
Default	---
Description	Displays a brief help message.

--man**Format** --man**Default** ---**Description** Displays full documentation.**--quiet****Format** --quiet**Default** ---**Description** Suppresses headers and success messages.**--site****Format** --site <site_name>**Default** ---**Description** Obtains a response from specified site.**--verbose****Format** --verbose**Default** ---**Description** Display modified object details.**--version****Format** --version**Default** ---**Description** Displays the product version.

Related Topics

[Deleting Funds on page 65](#)

[Commands Reference on page 159](#)

grmlien

Synopsis

```
grmlien { -I | {-J <instance_name>} | {-l <lien_id>} } [--debug]
[--site <site_name>] [--help] [--man] [--quiet] [--verbose] [--version]
```

Overview

grmlien deletes a lien or purges stale liens.

Options

-I	
Format	<code>-I</code>
Default	---
Description	Deletes expired liens.

-J	
Format	<code>-J <instance_name></code>
Default	---
Description	Specifies that the liens with the specified instance name, or job ID, will be deleted.

-l	
Format	<code>[-l] <lien_id></code>
Default	---

-l**Description**

Specifies the lien to be deleted.

--debug**Format**`--debug`**Default**`---`**Description**

Logs debug information to the screen.

--help**Format**`--help`**Default**`---`**Description**

Displays a brief help message.

--man**Format**`--man`**Default**`---`**Description**

Displays full documentation.

--quiet**Format**`--quiet`**Default**`---`**Description**

Suppresses headers and success messages.

--site	
Format	--site <site_name>
Default	---
Description	Obtains a response from specified site.

--verbose	
Format	--verbose
Default	---
Description	Display modified object details.

--version	
Format	--version
Default	---
Description	Displays the product version.

Related Topics

[Deleting Liens on page 77](#)

[Commands Reference on page 159](#)

grmnot

Synopsis

```
grmnot [--debug] [--site <site_name>] [--help] [--man] [--quiet] [--verbose] [--version] { [-N] <notification_id>}
```

Overview

grmnot deletes a stored notification.

Options

-N	
Format	<code>-N <notification_id></code>
Default	---
Description	Deletes expired liens.

--debug	
Format	<code>--debug</code>
Default	---
Description	Logs debug information to the screen.

--help	
Format	<code>--help</code>
Default	---
Description	Displays a brief help message.

--man	
Format	<code>--man</code>
Default	---
Description	Displays full documentation.

--quiet	
Format	<code>--quiet</code>
Default	---

--quiet**Description**

Suppresses headers and success messages.

--site**Format**`--site <site_name>`**Default**

Description

Obtains a response from specified site.

--verbose**Format**`--verbose`**Default**

Description

Display modified object details.

--version**Format**`--version`**Default**

Description

Displays the product version.

Related Topics

[Deleting Notifications on page 110](#)[Commands Reference on page 159](#)**grmorg****Synopsis**

```
grmorg [--debug] [--site <site_name>] [--help] [--man] [--quiet] [--verbose] [--version] { [-o] <organization_name>}
```

Overview

grmorg deletes an organization.

Options

-o

Format	<code>[-o] <organization_name></code>
Default	---
Description	Specifies the organization to delete.

--debug

Format	<code>--debug</code>
Default	---
Description	Logs debug information to the screen.

--help

Format	<code>--help</code>
Default	---
Description	Displays a brief help message.

--man

Format	<code>--man</code>
Default	---
Description	Displays full documentation.

--quiet

Format	--quiet
Default	---
Description	Suppresses headers and success messages.

--site

Format	--site <site_name>
Default	---
Description	Obtains a response from specified site.

--verbose

Format	--verbose
Default	---
Description	Display modified object details.

--version

Format	--version
Default	---
Description	Displays the product version.

Related Topics

- [Deleting Organizations on page 54](#)
[Commands Reference on page 159](#)

grmquote

Synopsis

```
grmquote { -I | { -q <quote_id> } } [--debug] [--site <site_name>] [--help] [--man] [--quiet] [--verbose] [--version]
```

Overview

grmquote deletes a quote or purges expired quotes.

Options

-I	
Format	<code>-I</code>
Default	---
Description	Deletes expired quotes.

-q	
Format	<code>[-q] <quote_id></code>
Default	---
Description	Specifies the quote to be deleted.

--debug	
Format	<code>--debug</code>
Default	---
Description	Logs debug information to the screen.

--help	
Format	<code>--help</code>
Default	---

--help	
Description	Displays a brief help message.

--man	
Format	--man
Default	---
Description	Displays full documentation.

--quiet	
Format	--quiet
Default	---
Description	Suppresses headers and success messages.

--site	
Format	--site <site_name>
Default	---
Description	Obtains a response from specified site.

--verbose	
Format	--verbose
Default	---
Description	Display modified object details.

--version

Format	--version
Default	---
Description	Displays the product version.

Related Topics

- [Deleting Quotes on page 82](#)
- [Commands Reference on page 159](#)

grmrate**Synopsis**

```
grmrate -n <charge_rate_name> [-x <charge_rate_value>] [--debug]
[--site <site_name>] [--help] [--man] [--quiet] [--verbose] [--version]
```

Overview

grmrate deletes a charge rate.

Options**-n**

Format	-n <charge_rate_name>
Default	---
Description	Specifies the charge rate to delete.

-x

Format	-x <charge_rate_value>
Default	---

-X

Description	Specifies the charge rate value to delete. If you do not specify a value, MAM will only delete a charge rate with an empty value.
--------------------	---

--debug

Format	--debug
---------------	---------

Default	---
----------------	-----

Description	Logs debug information to the screen.
--------------------	---------------------------------------

--help

Format	--help
---------------	--------

Default	---
----------------	-----

Description	Displays a brief help message.
--------------------	--------------------------------

--man

Format	--man
---------------	-------

Default	---
----------------	-----

Description	Displays full documentation.
--------------------	------------------------------

--quiet

Format	--quiet
---------------	---------

Default	---
----------------	-----

Description	Suppresses headers and success messages.
--------------------	--

-site

Format	--site <site_name>
Default	---
Description	Obtains a response from specified site.

-verbose

Format	--verbose
Default	---
Description	Display modified object details.

-version

Format	--version
Default	---
Description	Displays the product version.

Related Topics

- [Deleting Charge Rates on page 102](#)
[Commands Reference on page 159](#)

grmrole**Synopsis**

```
grmrole [--debug] [--site <site_name>] [--help] [--man] [--quiet] [--verbose] [--version] { [-r] <role_name>}
```

Overview

grmrole deletes a role.

Options

-r	
Format	<code>[-r] <role_name></code>
Default	---
Description	Specifies the role to delete.

--debug	
Format	<code>--debug</code>
Default	---
Description	Logs debug information to the screen.

--help	
Format	<code>--help</code>
Default	---
Description	Displays a brief help message.

--man	
Format	<code>--man</code>
Default	---
Description	Displays full documentation.

--quiet	
Format	<code>--quiet</code>
Default	---

--quiet**Description** Suppresses headers and success messages.**--site****Format** --site <site_name>**Default** ---**Description** Obtains a response from specified site.**--verbose****Format** --verbose**Default** ---**Description** Display modified object details.**--version****Format** --version**Default** ---**Description** Displays the product version.

Related Topics

[Deleting Roles on page 115](#)[Commands Reference on page 159](#)

grmusage

Synopsis

```
grmusage [--debug] [--site <site_name>] [--help] [--man] [--quiet]
[--verbose] [--version] {[-j] <usage_record_id>| -J
<instance_name>}
```

Overview

`grmusage` deletes a usage record.

Options

-j

Format	<code>[-j] <usage_record_id></code>
Default	---
Description	Specifies the ID of the usage record to delete. Instance names can be non-unique, because resource managers often recycle job IDs). This option enables specifying a unique usage record using the unique identifier.

-J

Format	<code>-J <instance_name></code>
Default	---
Description	Specifies the instance name to delete. Since instance names are assigned by externally and may be non-unique (such as job IDs assigned by a resource manager), all usage records with the specified instance name will be deleted.

--debug

Format	<code>--debug</code>
Default	---
Description	Logs debug information to the screen.

--help**Format**`--help`**Default**`---`**Description**

Displays a brief help message.

--man**Format**`--man`**Default**`---`**Description**

Displays full documentation.

--quiet**Format**`--quiet`**Default**`---`**Description**

Suppresses headers and success messages.

--site**Format**`--site <site name>`**Default**`---`**Description**

Obtains a response from specified site.

--verbose**Format**`--verbose`**Default**`---`**Description**

Display modified object details.

--version	
Format	--version
Default	---
Description	Displays the product version.

Related Topics

- [Deleting a Usage Record on page 85](#)
- [Commands Reference on page 159](#)

grmuser

Synopsis

```
grmuser [--debug] [--site <site_name>] [--help] [--man] [--quiet] [--verbose] [--version] [-u] <user_name>
```

Overview

grmuser deletes a user.

Options

-u	
Format	[-u] <user_name>
Default	---
Description	Specifies the name of the user to delete.

--debug	
Format	--debug
Default	---

--debug**Description**

Logs debug information to the screen.

--help**Format**

--help

Default

Description

Displays a brief help message.

--man**Format**

--man

Default

Description

Displays full documentation.

--quiet**Format**

--quiet

Default

Description

Suppresses headers and success messages.

--site**Format**

--site <site_name>

Default

Description

Obtains a response from specified site.

--verbose	
Format	--verbose
Default	---
Description	Display modified object details.

--version	
Format	--version
Default	---
Description	Displays the product version.

Related Topics

[Deleting Users on page 45](#)

[Commands Reference on page 159](#)

gstatement

Synopsis

```
gstatement [ [-f] <fund_id>] [ -n <fund_name>] [ -u <user_name>] [ -g <group_name>] [ -a <account_name>] [ -o <organization_name>] [ -c <class_name>] [ -m <machine_name>] [ --filter <filter_name>=<filter_value>] ... [ --filter-type <filter_type>] [ -s <start_time>] [ -e <end_time>] [ --summarize] [ -h, --hours] [ --debug] [ --site <site_name>] [ --help] [ --man] [ --version]
```

Overview

`gstatement` displays a fund statement. For a specified time frame it displays the beginning and ending balances as well as the total credits and debits to the fund over that period. This is followed by an itemized report of the debits and credits. You can use filters to select the funds you would like to review.

Options

-a	
Format	<code>-a <account_name></code>
Default	---
Description	Specifies that the statement will represent a combination of information for all the funds available for this account. Note that the statement may include information from other accounts if multiple accounts share the included funds.
-c	
Format	<code>-c <class_name></code>
Default	---
Description	Specifies that the statement will represent a combination of information for all the funds available for this class. Note that the statement may include information from other classes if multiple classes share the included funds.
-e	
Format	<code>-e <end_time></code>
Default	Now
Description	Specifies the end of the reporting period in the format <code>YYYY-MM-DD [hh:mm:ss] -Infinity Infinity Now</code> .
-f	
Format	<code>[-f] <fund_id></code>
Default	Infinity
Description	Specifies that MAM should make the fund statement for the specified fund.

-g	
Format	<code>-g <group_name></code>
Default	---
Description	Specifies that the statement will represent a combination of information for all the funds available for this group. Note that the statement may include information from other groups if multiple groups share the included funds.

-h	
Format	<code>-h</code> or <code>--hours</code>
Default	---
Description	Specifies that MAM should display time-based credits in hours. In cases where the currency is measured in resource-seconds (like processor-seconds), the currency is divided by 3600 to display resource-hours.

-m	
Format	<code>-m <machine_name></code>
Default	---
Description	Specifies that the statement will represent a combination of information for all the funds available for this machine. Note that the statement may include information from other machines if multiple machines share the included funds.

-n	
Format	<code>[-n] <fund_name></code>
Default	---
Description	Specifies that MAM will display the fund statement for funds with the given name.

-o	
Format	<code>-o <organization_name></code>
Default	---
Description	Specifies that the statement will represent a combination of information for all the funds available for this organization. Note that the statement may include information from other organizations if multiple organizations share the included funds.
-s	
Format	<code>-s <start_time></code>
Default	<code>-Infinity</code>
Description	Specifies the beginning of the reporting period in the format <code>YYYY-MM-DD [hh:mm:ss] -Infinity Infinity Now</code> .
-u	
Format	<code>-u <user_name></code>
Default	---
Description	Specifies that the statement will represent a combination of information for all the funds available for this user. Note that the statement may include information from other users if multiple machines share the included users.
--debug	
Format	<code>--debug</code>
Default	---
Description	Logs debug information to the screen.

--filter

Format	--filter <filter_name>=<filter_value>
Default	---
Description	Restricts the fund to one where constraints do not conflict with the specified filters. For example, gstatement --filter User=amy will restrict the fund to one usable by the user amy. You may specify multiple filter options that are logically ANDed together.

--filter-type

Format	--filter-type <filter_type>
Default	NonExclusive
Description	Specifies the filtering type. If you use the Exclusive filter type, MAM will only match a fund if the specified filters meet all constraints. If you use the NonExclusive filter type, MAM will match a fund as long as the specified filters do not conflict with the constraints.

--help

Format	--help
Default	---
Description	Displays a brief help message.

--man

Format	--man
Default	---
Description	Displays full documentation.

--quiet

Format	--quiet
---------------	---------

--quiet**Default**

Description

Suppresses headers and success messages.

--site**Format**`--site <site_name>`**Default**

Description

Obtains a response from specified site.

--summarize**Format**`--summarize`**Default**

Description

Displays transaction summaries only. Deposits, Refunds, Charges, and other properties will be shown as total as opposed to being itemized.

--version**Format**`--version`**Default**

Description

Displays the product version.

Related Topics

[Obtaining a Fund Statement on page 64](#)[Commands Reference on page 159](#)**gtransfer****Synopsis**

```
gtransfer {--from-fund <source_fund_id> &| --from-allocation
<source_allocation_id> &| --from-filter <filter_name>=<filter_
value>...} {--to-fund <destination_fund> &| --to-allocation
<destination_allocation_id> &| --to-filter <filter_
name>=<filter_value>...} [--filter-type <filter_type>] [-d
<description>] [-h, --hours] [--debug] [--site <site_name>] [-
-help] [--man] [--quiet] [--verbose] [--version] {[-z
<amount>]}
```

Overview

`gtransfer` issues a transfer between funds.

Options

`-d`

Format	<code>-d <description></code>
Default	---
Description	Specifies the reason for the transfer. The annotation applies to the transaction description (seen via glstrans), not the allocation description.

`-h`

Format	<code>-h</code> or <code>--hours</code>
Default	---
Description	Treats currency as specified in hours. In systems where the currency is measured in resource-seconds (like processor-seconds), this option allows the amount to be specified in resource-hours.

`-z`

Format	<code>[-z] <amount></code>
Default	---
Description	Specifies the amount to transfer.

--debug**Format**`--debug`**Default**`---`**Description**

Logs debug information to the screen.

--filter-type**Format**`--filter-type <filter_type>`**Default**`NonExclusive`**Description**

Specifies the filtering type. If you use the `Exclusive` filter type, MAM will only match a fund if the specified filters meet all constraints. If you use the `NonExclusive` filter type, MAM will match a fund as long as the specified filters do not conflict with the constraints.

--from-allocation**Format**`--from-allocation <source_allocation_id>`**Default**`---`**Description**

Transfers credits from the specified allocation ID only. If you omit the allocation, only credits from active allocations will transfer in the order of earliest expiring first.

--from-filter**Format**`--from-fund <filter_name=<filter_value>`**Default**`---`**Description**

If you specify one or more source filters and there is exactly one matching fund, MAM makes the transfer from that fund. Otherwise, it displays a list of funds for the specified filters and you will be prompted to respecify the transfer against one of the enumerated funds. You can specify multiple `--from-filter` options by logically ANDing them together.

--from-fund	
Format	--from-fund <source_fund_id>
Default	---
Description	Specifies the fund to be debited.

--help	
Format	--help
Default	---
Description	Displays a brief help message.

--man	
Format	--man
Default	---
Description	Displays full documentation.

--quiet	
Format	--quiet
Default	---
Description	Suppresses headers and success messages.

--site	
Format	--site <site_name>
Default	---
Description	Obtains a response from specified site.

--to-allocation

Format	--to-allocation <destination_allocation_id>
Default	---
Description	Transfers credits to the specified allocation ID only. If you omit the allocation, MAM transfers the credits to the allocation having the same start and end time as the source allocation the funds are taken from, or, if such an allocation is non-existent, MAM will create a new allocation in the target fund having the same start and end time.

--to-filter

Format	--to-filter <filter_name>-<filter_value>
Default	---
Description	If you specify one or more destination filters and there is exactly one matching fund, a transfer will be made to that fund. Otherwise, MAM displays a list of funds for the specified filters and you will be prompted to respecify the transfer against one of the enumerated funds. You may specify multiple --to-filter options by logically ANDing them together.

--to-fund

Format	--to-fund <destination_fund_id>
Default	---
Description	Specifies the fund to be credited.

--verbose

Format	--verbose
Default	---
Description	Displays modified object details.

--version	
Format	--version
Default	---
Description	Displays the product version.

Related Topics

[Making Transfers on page 64](#)

[Commands Reference on page 159](#)

gwithdraw

Synopsis

```
gwithdraw [-f <fund_id>] [-i <allocation_id>] [-u <user_name>]
[-q <group_name>] [-a <account_name>] [-o <organization_name>]
[-c <class_name>] [-m <machine_name>] [-filter <filter_name>=<filter_value>] ...
[--filter-type <filter_type>] [-d <description>] [-h, --hours] [--debug]
[--site <site_name>] [-help] [--man] [--quiet] [--verbose] [--version] {[ -z ]
<amount>}
```

Overview

`gwithdraw` makes a withdrawal from the specified fund.

Options

-a	
Format	-a <account_name>
Default	---
Description	Specifies that the fund for the withdrawal should be usable by the specified account.

-c**Format**`-c <class_name>`**Default**

Description

Specifies that the fund for the withdrawal should be usable by the specified class.

-d**Format**`-d <description>`**Default**

DescriptionSpecifies the reason for the withdrawal. The annotation applies to the transaction description (seen via [glstrans](#)), not the allocation description.**-f****Format**`-f <fund_id>`**Default**

Description

Specifies the ID of the fund from which MAM will make the withdrawal.

-g**Format**`-g <group_name>`**Default**

Description

Specifies that the fund for the withdrawal should be usable by the specified group.

-h**Format**`-h or --hours`

-h	
Default	---
Description	Specifies that MAM should display time-based credits in hours. In cases where the currency is measured in resource-seconds (like processor-seconds), this option allows the amount to be specified in resource-hours.
-i	
Format	<code>-i <allocation_id></code>
Default	---
Description	Withdraws credits from the specified allocation ID only. If you omit the allocation, MAM only withdraws credits from active allocations in the order of earliest expiring first.
-m	
Format	<code>-m <machine_name></code>
Default	---
Description	Specifies that the fund for the withdrawal should be usable by the specified machine
-o	
Format	<code>-o <organization_name></code>
Default	---
Description	Specifies that the fund for the withdrawal should be usable by the specified organization.
-u	
Format	<code>-u <user_name></code>
Default	---
Description	Specifies that the fund for the withdrawal should be usable by the specified user.

-z**Format**

`[-z] <amount>`

Default

Description

Specifies the amount to withdraw. You may also specify the amount as the sole argument.

--debug**Format**

`--debug`

Default

Description

Logs debug information to the screen.

--filter**Format**

`--filter <filter_name>=<filter_value>`

Default

Description

Restricts the fund to one where constraints do not conflict with the specified filters. For example, `gwithdraw --filter User=amy` will restrict the fund to one usable by the user `amy`. You may specify multiple filter options that are logically ANDed together.

--filter-type**Format**

`--filter-type <filter_type>`

Default

NonExclusive

Description

Specifies the filtering type. If you use the Exclusive filter type, MAM will only match a fund if the specified filters meet all constraints. If you use the NonExclusive filter type, MAM will match a fund as long as the specified filters do not conflict with the constraints.

--help**Format**

`--help`

--help	
Default	---
Description	Displays a brief help message.

--man	
Format	--man
Default	---
Description	Displays full documentation.

--quiet	
Format	--quiet
Default	---
Description	Suppresses headers and success messages.

--site	
Format	--site <site_name>
Default	---
Description	Obtains a response from specified site.

--verbose	
Format	--verbose
Default	---
Description	Displays modified object details.

--version	
Format	--version
Default	---
Description	Displays the product version.

Related Topics

- [Making Withdrawals on page 63](#)
- [Commands Reference on page 159](#)

goldsh

Synopsis

```
goldsh [ --debug] [ --site <site_name>] [ --help] [ --man] [ --format <output_type>] [ --quiet] [ --verbose] [ --version] [<command>]
```

Overview

goldsh is an interactive control program that can access all functionality available in Gold. You can invoke commands directly from the command line, or an interpreter can parse commands from stdin.

Commands follow the form:

<Object>[,<Object>...] <Action> [<Predicate>] ...

<Predicate> follows the form:

[<*Conjunction*>] [<*OpenParentheses*>] [<*Object*>.]<*Name Operator*> [<*Subject*>.]<*Value*> [<*CloseParentheses*>]

Where

- <*Conjunction*> defaults to "And" and includes:
 - && – and
 - || – or
 - & ! – and not
 - | ! – or not
- <*OpenParentheses*> can be any number of literal open parentheses '('.

- <Name> is the name of the condition, assignment or option.
- <Operator> is one of:
 - == – equals
 - < – less than
 - > – greater than
 - <= – less than or equal to
 - >= – greater than or equal to
 - != – not equal to
 - ~ – matches
 - !~ – does not match
 - = – assignment
 - += – increment
 - -= – decrement
 - := – option
 - :! – negated option
- <Value> is the value of the condition, assignment, or option and may be enclosed in double quotes to enclose spaces or special characters.
- <CloseParentheses> can be any number of literal close parentheses ')'.

You can specify the desired selections (columns to be displayed) in a query via a pseudo Show option with a value of comma-separated attribute names. It can optionally include an object, operator and alias. It will follow the form:

```
Show:=" [operator() [object.]name[=alias] [] [, [operator()
[object.]name[=alias] []]]] ...".
```

See [Using the Gold Shell \(goldsh\) on page 119](#) for more information on constructing requests.

Options

--debug	
Format	--debug
Default	---
Description	Logs debug information to the screen.

--format

Format	--format <standard csv>
Default	standard
Description	Specifies the data output format. Valid values include standard and csv.

--help

Format	--help
Default	---
Description	Displays a brief help message.

--man

Format	--man
Default	---
Description	Displays full documentation.

--quiet

Format	--quiet
Default	---
Description	Suppresses headers and success messages.

--site

Format	--site <site_name>
Default	---
Description	Obtains a response from specified site.

--verbose	
Format	--verbose
Default	---
Description	Displays modified object details.

--version	
Format	--version
Default	---
Description	Displays the product version.

Related Topics

- [Usage on page 119](#)
[Commands Reference on page 159](#)

goldd

Synopsis

```
goldd [-k, --stop] [-r, --restart] [-s, --start] [-l, --status]
[--help] [--man] [-d, --debug [<debug_level>]] [--version]
```

Overview

`goldd` is a forking server that listens for and services Moab Accounting Manager client requests. It handles the startup and daemonization, shutdown and restart of the application.

Options

-k, --stop	
Format	-k or --stop

-k, --stop**Default**

Description

Shuts down the server.

-l, --status**Format**

-l or --status

Default

Description

Displays the status of the server, indicating whether it is running or has stopped.

-r, --restart**Format**

-r or --restart

Default

Description

Restarts the server.



If MAM has been started under systemd, use `systemctl restart mam.service` instead of using this option.

-s, --start**Format**

-s or --start

Default

Description

Starts the server. This option is assumed in the absence of a stop or restart flag and may be omitted in a start request.

-d, --debug**Format**

-d or --debug [<debug_level>]

Default

DEBUG

-d, --debug**Description**

Logs debug information to the screen. You can supply an optional debug level parameter to indicate the logging threshold. It can be one of TRACE, DEBUG, INFO, WARN, ERROR and FATAL.

--help**Format**

--help

Default

Description

Displays a brief help message.

--man**Format**

--man

Default

Description

Displays full documentation.

--version**Format**

--version

Default

Description

Displays the product version.

Related Topics

[Commands Reference on page 159](#)

mybalance

Synopsis

mybalance [[-h](#), [--hours](#)] [[--help](#)] [[--man](#)]

Overview

`mybalance` displays balance information for the invoking user.

Options

-h, --hours

Format -h or --hours

Default ---

Description Displays balance in processor-hours (instead of processor-seconds).

--help

Format --help

Default ---

Description Displays a brief help message.

--man

Format --man

Default ---

Description Displays full documentation.

Related Topics

[Personal Balance on page 62](#)

[Commands Reference on page 159](#)