

Novell ZENworks® Orchestrator

1.2

January 31, 2008

JOB MANAGEMENT

www.novell.com



Novell®

Legal Notices

Novell, Inc. makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc. makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

Any products or technical information provided under this Agreement may be subject to U.S. export controls and the trade laws of other countries. You agree to comply with all export control regulations and to obtain any required licenses or classification to export, re-export or import deliverables. You agree not to export or re-export to entities on the current U.S. export exclusion lists or to any embargoed or terrorist countries as specified in the U.S. export laws. You agree to not use deliverables for prohibited nuclear, missile, or chemical biological weaponry end uses. See the [Novell International Trade Services Web page \(http://www.novell.com/info/exports/\)](http://www.novell.com/info/exports/) for more information on exporting Novell software. Novell assumes no responsibility for your failure to obtain any necessary export approvals.

Copyright © 2008 Novell, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher.

Novell, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed on the [Novell Legal Patents Web page \(http://www.novell.com/company/legal/patents/\)](http://www.novell.com/company/legal/patents/) and one or more additional patents or pending patent applications in the U.S. and in other countries.

Novell, Inc.
404 Wyman Street, Suite 500
Waltham, MA 02451
U.S.A.
www.novell.com

Online Documentation: To access the latest online documentation for this and other Novell products, see [the Novell Documentation Web page \(http://www.novell.com/documentation\)](http://www.novell.com/documentation).

Novell Trademarks

For Novell trademarks, see [the Novell Trademark and Service Mark list \(http://www.novell.com/company/legal/trademarks/tmlist.html\)](http://www.novell.com/company/legal/trademarks/tmlist.html).

Third-Party Materials

All third-party trademarks are the property of their respective owners.

Contents

About This Guide	7
1 Overview	9
1.1 The Role of the Job Manager	9
1.2 Before You Begin	9
1.2.1 Browser Requirements	9
1.2.2 Logging in to the Orchestrator Server	9
2 Using the ZENworks Orchestrator User Portal	11
2.1 Starting the Portal	11
2.1.1 Browser Requirements	11
2.1.2 Logging in to the ZENworks Orchestrator Server	11
2.2 Getting Started Using the Portal	11
2.2.1 Starting the Portal	12
2.2.2 Identifying Available Jobs	14
2.2.3 Starting a Local Job from the Server	14
2.2.4 Monitoring a Job	20
2.2.5 Viewing the Job Log	21
2.2.6 Pausing and Resuming a Running Job	21
2.2.7 Cancelling a Running Job	22
2.2.8 Logging Out of the Portal	22
A The ZOS Command Line Tool	23
A.1 List of ZOS Commands	23
A.2 Getting Started with the ZOS Command	24
A.2.1 Logging In to the Server for Orchestrator	24
A.2.2 Running a Job	25
A.2.3 Monitoring a Job	25
A.3 Details, Usage, and Syntax Examples of ZOS Commands	27
login	28
logout	29
password	30
list	31
jobinfo	32
run	33
status	35
jobs	36
priority	38
event	39
cancel	40
fail	41
pause	42
resume	43
listen	44
search	45
info	47
copy	48
delete	50

dir	51
mkdir	52
move	53
cat	54
log	55
head	56
tail	57

About This Guide

This *Job Management Guide* introduces the role and tools of the Novell® ZENworks® Orchestrator 1.2 job manager. The guide explains the role of the job manager, referred to also as the Orchestrator “end user.” It also provides an introductory overview of the Orchestrator User Portal and explains how to monitor and manage applications running on the Orchestrator by using the portal. Further, it provides a reference for the zos command line tool, a command line interface used for monitoring jobs. The guide is organized as follows:

- ♦ Chapter 1, “Overview,” on page 9
- ♦ Chapter 2, “Using the ZENworks Orchestrator User Portal,” on page 11
- ♦ Appendix A, “The ZOS Command Line Tool,” on page 23

Audience

This book is intended for the consumption of datacenter managers, IT or Operations administrators. It assumes that users of the product have the following background:

- ♦ General understanding of network operating environments and systems architecture.
- ♦ Knowledge of basic UNIX shell commands and text editors.

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation, or go to www.novell.com/documentation/feedback.html (<http://www.novell.com/documentation/feedback.html>) and enter your comments there.

Documentation Updates

For the most recent version of this *Installation and Getting Started Guide*, visit the [ZENworks Orchestrator 1.2 Web site](http://www.novell.com/documentation/zen_orchestrator12/) (http://www.novell.com/documentation/zen_orchestrator12/).

Documentation Conventions

In Novell documentation, a greater-than symbol (>) is used to separate actions within a step and items in a cross-reference path.

A trademark symbol (®, ™, etc.) denotes a Novell trademark. An asterisk (*) denotes a third-party trademark.

When a single pathname can be written with a backslash for some platforms or a forward slash for other platforms, the pathname is presented with a backslash. Users of platforms that require a forward slash, such as Linux or UNIX, should use forward slashes as required by your software.

Overview

1

The Novell ZOS Server Job Manager Reference Guide provides information on using both the thin-client user console and the other command line tools for submitting and monitor jobs running on the matrix. The guide provides in-depth information on the following management and deployment tools:

- ♦ [Section 1.1, “The Role of the Job Manager,” on page 9](#)
- ♦ [Section 1.2, “Before You Begin,” on page 9](#)

1.1 The Role of the Job Manager

The Job Manager is a designated operator in the system who has the permission to launch specific jobs to the ZENworks Orchestrator Server (ZOS). These jobs are assigned to you by your ZENworks administrator.

1.2 Before You Begin

Before you begin, you must have:

- ♦ A User account created by the ZOS Administrator.
- ♦ An internet browser.
- ♦ Jobs assigned to be run by you from the ZOS Administrator.

1.2.1 Browser Requirements

The following browsers support the User Portal:

- ♦ Internet Explorer, version 6.0 or higher
- ♦ Netscape Navigator, version 6.0 or higher
- ♦ Mozilla, version 1.5 or higher
- ♦ Firefox 2.0 or higher

1.2.2 Logging in to the Orchestrator Server

You can log into the ZENworks Orchestrator by:

- 1 Open a command terminal.
- 2 Change to the `/opt/novell/zenworks/zos/server/bin/` directory.
- 3 Enter `./zoc`.
- 4 Click on your allowed managed grid.
- 5 Enter your username and password.

Using the ZENworks Orchestrator User Portal

2

The Novell® ZENworks® Orchestrator User Portal is a graphical management tool that makes it easy to use the server for Orchestrator to deploy and manage jobs. This section provides an introduction and overview of the portal and includes the following information:

- ♦ [Section 2.1, “Starting the Portal,” on page 11](#)
- ♦ [Section 2.2, “Getting Started Using the Portal,” on page 11](#)

2.1 Starting the Portal

The ZENworks Orchestrator User Portal is a Web application that is hosted by the server for Orchestrator. A network or grid management administrator provides users with a URL (the server name for Orchestrator) to launch the portal application.

The information in this section includes the following content:

- ♦ [Section 2.1.1, “Browser Requirements,” on page 11](#)
- ♦ [Section 2.1.2, “Logging in to the ZENworks Orchestraotr Server,” on page 11](#)

2.1.1 Browser Requirements

The following browsers support the User Portal:

- ♦ Internet Explorer, version 6.0 or higher
- ♦ Netscape Navigator, version 6.0 or higher
- ♦ Mozilla, version 1.5 or higher
- ♦ Firefox 2.0 or higher

2.1.2 Logging in to the ZENworks Orchestraotr Server

Before using the portal, users need to log in to the ZENworks Orchestrator Server (ZOS Server) where jobs are managed. For more information about logging in to the ZOS Server, see .

2.2 Getting Started Using the Portal

This section describes how to perform some basic tasks with the portal for ZENworks Orchestrator. The following content is included:

- ♦ [Section 2.2.1, “Starting the Portal,” on page 12](#)
- ♦ [Section 2.2.2, “Identifying Available Jobs,” on page 14](#)
- ♦ [Section 2.2.3, “Starting a Local Job from the Server,” on page 14](#)
- ♦ [Section 2.2.4, “Monitoring a Job,” on page 20](#)
- ♦ [Section 2.2.5, “Viewing the Job Log,” on page 21](#)

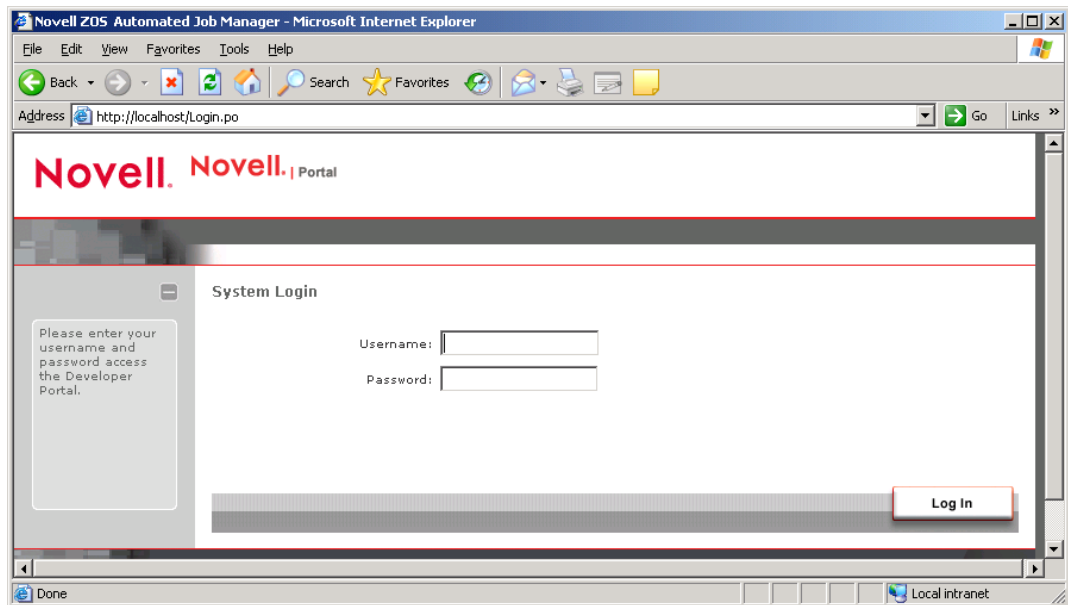
- ◆ Section 2.2.6, “Pausing and Resuming a Running Job,” on page 21
- ◆ Section 2.2.7, “Cancelling a Running Job,” on page 22
- ◆ Section 2.2.8, “Logging Out of the Portal,” on page 22

2.2.1 Starting the Portal

The ZOS Server includes a Web server that runs the User Portal in HTML. This portal is an administration tool that helps users manage the jobs that are deployed to managed computing resources, but it does not provide a facility to run jobs, even if you are logged in as an administrator.

Use the following steps to run the portal and log in at a workstation:

- 1 Start any Internet browser.
- 2 Enter the URL (server name) of the server for Orchestrator to display a page where the user can log in to the server:

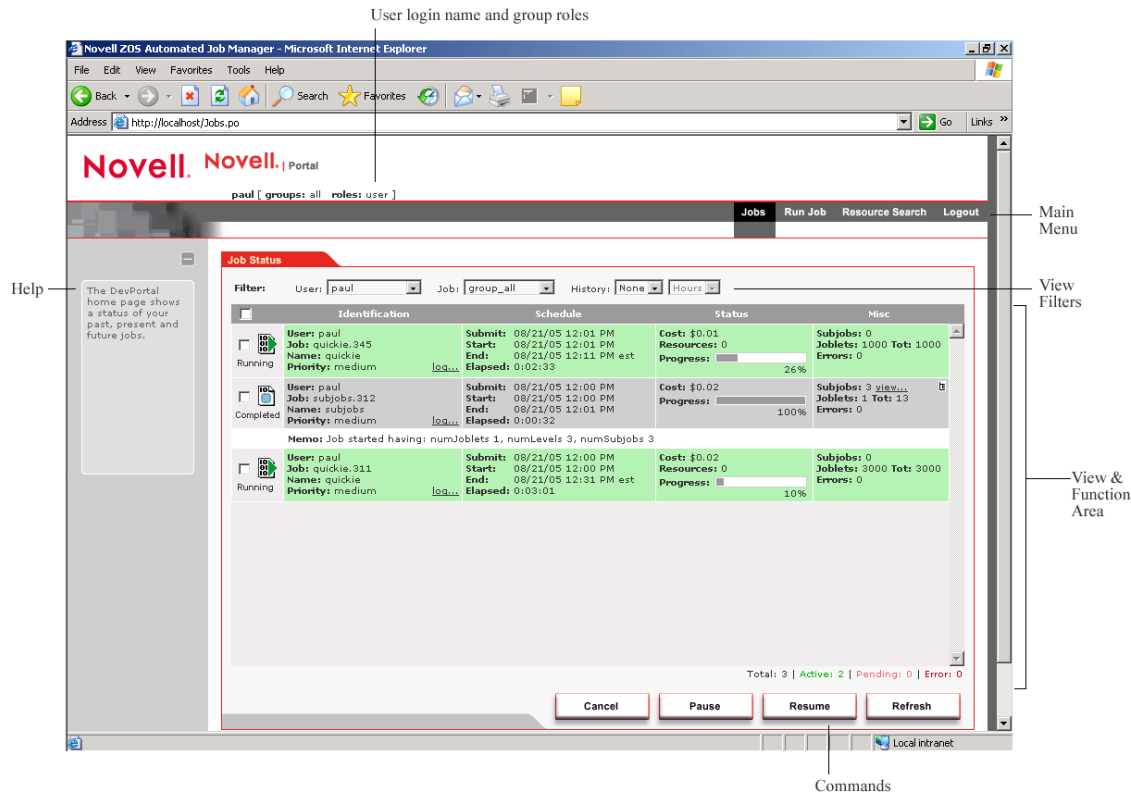


- 3 Enter the same default username and password you used for logging into the console for ZENworks Orchestrator.
- 4 Click *Log In* to log into the portal.

The User Portal Interface

After you log in to the portal, you will see the following interface displayed in the browser:

Figure 2-1 The User Portal Interface



The table below provides details about the various labeled areas of the interface.

Table 2-1 Descriptions of the Areas of the User Portal

Area	Description
User Login and Group Roles	This area indicate the user and group logged-in.
Main Menu	Provides operations categorized under the menus: <ul style="list-style-type: none"> ♦ <i>Jobs</i> include a list of active jobs running on the ZOS and their correlating job run information and status. The Jobs facility also includes the ability for users to control the running jobs with various functions. ♦ <i>Run Job</i> provides a facility to submit a job to the ZOS with a selection of parameters for the job and policies; the conditions for which the job is to run. ♦ <i>Resource Search</i> provides a facility to discover the resources that are available for use, including jobs, policies, and computing resources available for running jobs. ♦ <i>Logout</i> provides a facility to end a user session.
View Filters	The View Filters area provides a way to filter the view of current running jobs. Depending on the function view, this panel may vary.

Area	Description
View and Function	The View and Function area displays jobs. Other functions use this area for displaying computing resources and reports on the requested information. This is the general area of the portal for interacting with the user's requested data.
Commands	These context sensitive buttons provide a set of commands for the functions that can be performed on the data in the View and Function area.
Help	This area displays context sensitive help information.

2.2.2 Identifying Available Jobs

After you log in, you can use the portal for Orchestrator to identify the available jobs that can be run in the ZENworks Orchestrator system. The portal lists the same runnable jobs that are available for viewing in the console for Orchestrator.

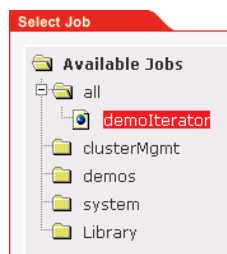
IMPORTANT: A job must exist before you can configure it. Creating a job involves the creation of an application executable and a job file. If you want to create your own jobs, see the ZENworks Orchestration Developer's Toolkit Guide [not available in this release]. The guide has more information about creating and building jobs using the Orchestrator Job Description Language (JDL) and job policies.

ZENworks Orchestrator includes prepackaged jobs for demonstration. Prepackaged jobs are automatically deployed with each new server each time it is started. These jobs include *demoInterator*, *failover*, *quickie*, and *subjobs*.

To see runnable jobs from the portal command menus, click *Run Job* in the menu bar to open the *Run Job* menu screen.

The *Select Job* tab is listed in the *View and Function* section of the portal interface. The *Available Jobs* explorer is located on the left side of this interface. The *Job Parameters* section is located on the right side. The *Job Parameters* section is associated with job selections in the *Available Jobs* explorer. To view all jobs, expand the explorer under *Available Jobs > all*.

Figure 2-2 The Available Jobs Explorer View Expanded to Show All Jobs



2.2.3 Starting a Local Job from the Server

There are three stages to running an Orchestrator job on managed computing resources. This section uses the example job, “quickie,” to illustrate these stages:

- ◆ “Selecting the Job You Want to Run” on page 15

- ♦ “Selecting Computing Resources to Run the Job” on page 15
- ♦ “Scheduling a Time to Run the Job” on page 18
- ♦ “Running the Job” on page 19

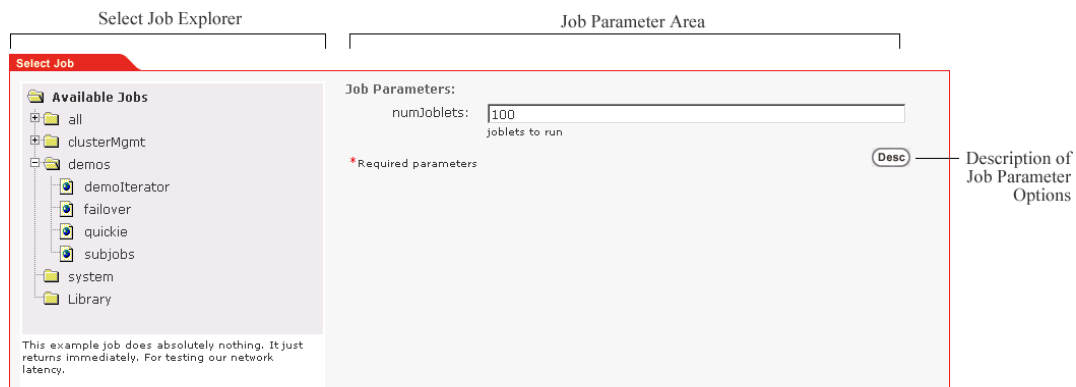
NOTE: Quickie is a demonstration job used primarily to show computing resource utilization. Its purpose is to simply start its executable program on an appropriate computing resource in the grid, wait for a period of time, and then stop to free the computing resource again.

Selecting the Job You Want to Run

When you want to start a job locally from the server, you need to first select the job and its parameters. The portal interface labels this as the *What* stage of running a job.

Use the following steps to select the job and job parameters you want to run:

- 1 From the *Select Job* Explorer, expand the *Available Jobs* directory tree to *demос*.



This lists the demo jobs highlighted above that come with the server for Orchestrator.

- 2 Select the *quickie* job to display its job parameters in the *Job Parameters* area of the interface.

NOTE: If you need more information about a job parameter, click *Desc* to expand the description of the fields of the *Job Parameters* area.

- 3 Accept the default values for the parameters.

NOTE: The *numjoblets* parameter is an argument for the associated job. The default parameter value for the *quickie* job is 100.

- 4 Click *Lock Job* (located at the bottom-right of the Function Area of the Portal) to select the job.

The *What* stage on the interface now lists the *quickie* job selection. If you want to change the job selection or its parameters, you can click *Edit* on this stage anytime before you actually run the job.

Selecting Computing Resources to Run the Job

After the job is selected, users need to identify the computing resources where the job is to be run. The portal interface labels this as the *Where* stage.

Searching for computing resources is possible using the *Resource Search* facility accessed from the main menu. When the Resource Search page is displayed, users can select computing resource search criteria to use as operators to search for the computing resources they want.

IMPORTANT: The *Resource Search* tab area shows a set of filters for identifying computing resources. A resource is a computing resource that can run jobs. Your grid might have different kinds of computing resources, identified by a set of facts about that computing resource. A resource fact includes many attributes about that computing resource such as its name, hardware configuration, operating system, and other characteristics. If the grid has uniform resources, the *Where* stage is less useful, but for a grid with multiple resource characteristics, the *Where* stage is very important.

Depending on the resources that you have set up with the server for Orchestrator, you might have multiple computing resource operating system types (operating system fact). For example, you could have both Windows and Linux operating system resources available. Some of the sample demo jobs that come with the server for Orchestrator have executables for both operating systems, and others do not. The *quickie* job can run on both Windows and Linux.

Using Expressions to Identify Appropriate Computing Resources

A simple expression is used for identifying the computing resources appropriate for running a job. Multiple expressions can be applied, but all expressions must be true. Each expression is entered using the expression criteria listed in the following table.

Table 2-2 Expressions Used to Filter Computing Resources Eligible to Run Jobs

Expression	Criteria
Facts	Each computing resource has a set of Facts associated with a resource. A complete list of facts is available from the <i>Select Fact</i> drop-down list for computing resources.
Operator	Each expression requires one operator.
Value	The value criteria indicate the choice associated with the selected fact. The portal generates a list of possible value options based on the selected Fact and the possible values from the current set of resources. When you select a value, it is entered into the criteria edit field to the right of the value selection field. You can edit the value in this field if necessary.

Multiple criteria may be entered in the same fashion. The selection of computing resources must match all criteria entered. You can alter your choice of expressions using the following functions:

- ◆ To remove criteria, click the minus icon located to the right of the criteria.
- ◆ To start a new search or to start over, click *Reset* located below the criteria list.
- ◆ To add more criteria, click on the plus icon.

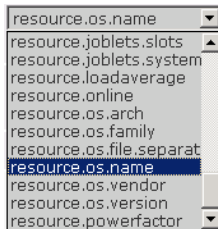
When you have entered all of the search criteria, click *Search*. The matching resources are listed at the lower part of the page under *Search Results*.

Sample Job: Selecting Computing Resources for Running a Job

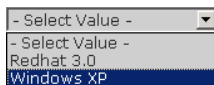
IMPORTANT: If no search facts are selected, the Orchestrator assumes that all computing resources are to be selected.

The following steps demonstrate how the quickie job is configured to run only on Windows XP operating system computing resources:

- 1 From the *Select Fact* drop-down list, select the operating system fact.

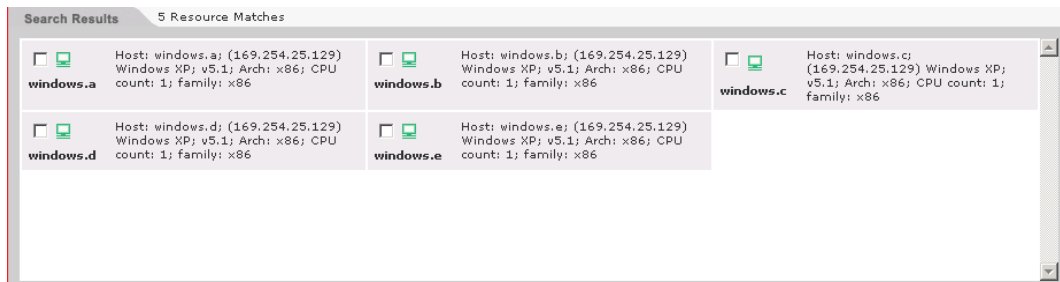


- 2 Specify the operator to use for the fact. In this case the default, *is equal to*, is used.
- 3 Specify the fact value, *Windows XP*.



NOTE: Each of the drop-down items is context sensitive. When a fact is selected, the values update with the valid options for that fact, based on the current set of available computing resources. The values change automatically as computing resources are added or removed in the grid of managed resources.

- 4 Click *Search* to check whether any computing resources match the current search filter criteria. If so, these computing resources are listed at the bottom of the Function area of the portal. See the figure below for an example.



- 5 Click *Lock Criteria* (located at the bottom-right of the Function area of the portal) to select the computing resources.

Notice the *Where* section (located at the top of the Function area of the portal) now shows the *quickie* job resource selections. To change the computing resource selection, click *Edit* before running a job.

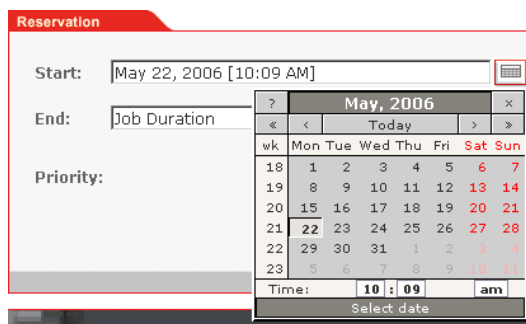
Scheduling a Time to Run the Job

After you have selected a job and the computing resource where you want to run it, you need to select a start time for running the job. The portal labels this as the *When* stage of running a job.

The *Reservation* tab area shows start and end requirements, as well as a priority setting for running the job. You can set up policies associated with the user's profile to control when a job can be run. In the *When* stage, however, the user can add an additional scheduling requirement and job priority for this running this job.

Use the following steps to specify when the job is to be run.

- 1 Click the *Reservation* tab, click the calendar icon located adjacent to the *Start* field to open a calendar scheduling utility, then select a date and time to run the job. You can accept the default in the *Start* field: *As Soon As Possible* if you want to.



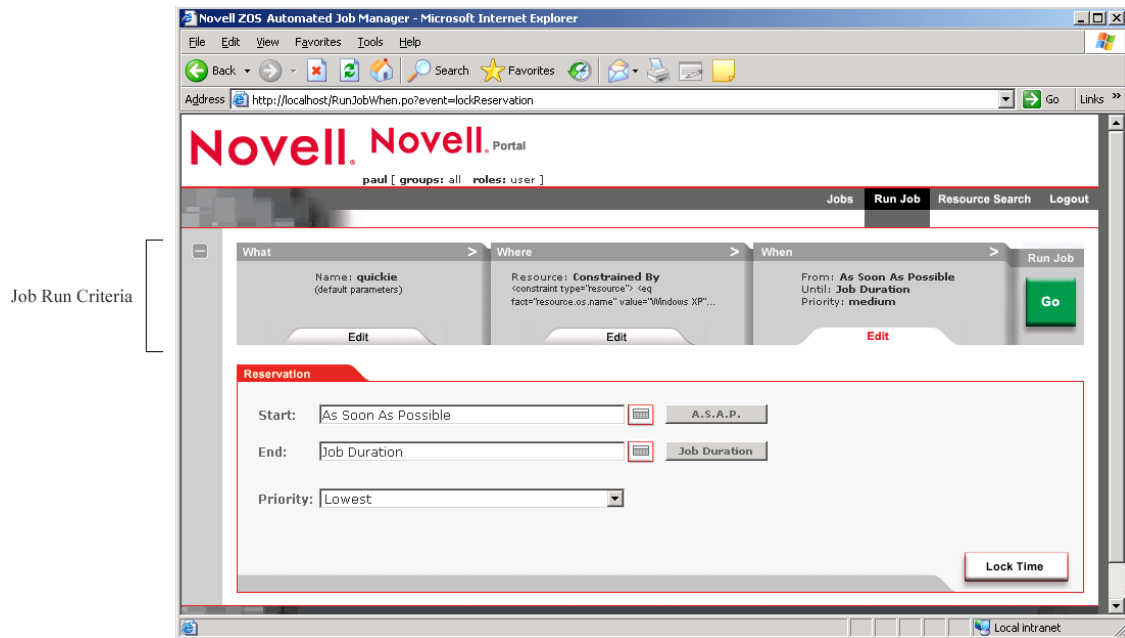
- 2 (Optional) Click the calendar icon located adjacent to the *End* field, then select a date and time to complete the job. You can also accept the default in the *End* field: *Job Duration*.
- 3 (Optional) If the *Priority* field, click the drop-down list to select a priority for the job, or accept the default.

Use the *Priority* parameter to increase or decrease the priority of the job in relation to other jobs that are already running or scheduled to run. This option is used only if computing resources are inadequate at the scheduled job start time to run the job. The options in the drop-down list are context sensitive to the available priorities associated with the user profile.

- 4 Click *Lock Time* (located at the bottom-right of the Function area) to finalize the time to run the job.

The job is now ready to run. The job criteria are displayed in the upper part of the Functional area of the portal.

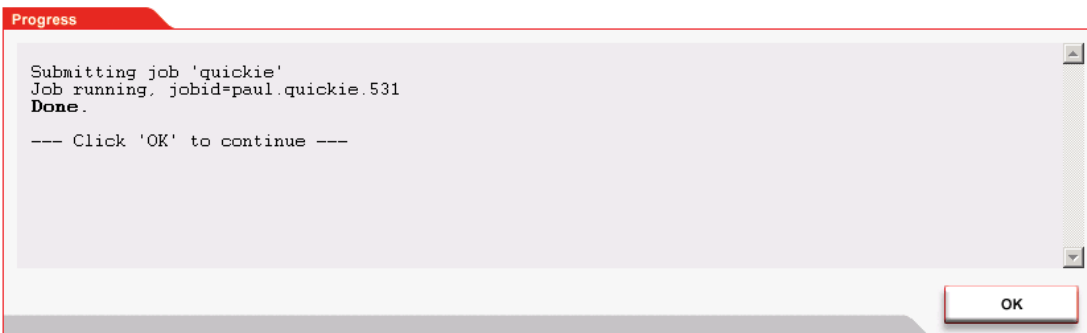
Figure 2-3 The User Portal Displaying the Job Run Criteria Previously Selected



Running the Job

When you have established the job run criteria in the User Portal, click *Go* to schedule the job to run. A progress log is displayed in the *Progress* tab in the Function area of the portal.

Figure 2-4 Progress Message Displayed for the “Quickie” Job



A *Done* message indicates that the job was successfully submitted to the server for ZENworks Orchestrator.

Click *OK* to close the Progress log and return to the Home page to monitor the running job.

NOTE: This same job could also be run using the following zos command:

```
zos run quickie.job -- priority=high -- policy=os.windows/xp
```

For more information about using the zos command line interface, see [Appendix A, “The ZOS Command Line Tool,”](#) on page 23.

2.2.4 Monitoring a Job

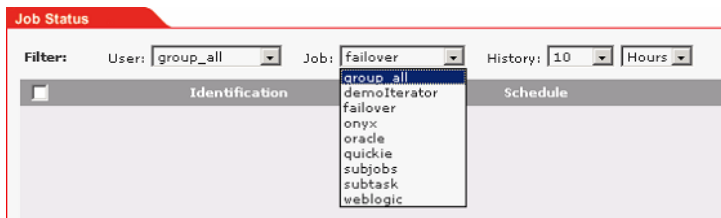
Users can access a list of the Orchestrator jobs that are currently running by clicking the *Jobs* command menu and then the *Jobs* submenu on the User Portal.

Figure 2-5 *Jobs Command Menu on the User Portal*



The *Job Status* page under the *Jobs* submenu lists information about jobs submitted to the Orchestrator. Users can use this information to monitor and manage those jobs.

Figure 2-6 *Job Status Page Showing the Job Filter Drop Down List*




A list of jobs is displayed based on the filter criteria selected. The default filter criteria is all jobs for the current logged-in user. Filter options include:

- ◆ **Users:** Select a user, or one of the built-in group types: *group_all* for all user groups, or *group_system* for all system jobs.
- ◆ **Job:** Select a specific job, or the built-in group types: *group_all* for all jobs.
- ◆ **History:** Select jobs submitted within the last *nn* hours or *nn* days from the current date and time. The minimum time you can set is 1 hour. The maximum time you can set is 24 days. If you select *none* as the history time, all jobs are selected.

When you have selected the filter criteria you want, you can update the current job status. Click *Refresh* (located at the bottom-right of the interface) to show the status.

Figure 2-7 *Status Information for a Current Job*

Identification	Schedule	Status	Misc
 User: paul Job: quickie.645 Name: quickie Priority: medium Completed	Submit: 09/03/05 03:17 AM Start: 09/03/05 03:17 AM End: 09/03/05 03:26 AM Elapsed: 0:09:14	Cost: \$0.05 Progress: <div style="width: 100%;"><div style="width: 100%;"></div></div> 100%	Subjobs: 0 Joblets: 1000 Tot: 1000 Errors: 0

The job status report includes the following columns:

- ◆ The far left column indicates the result for the job status report as sampled at refresh time. The column displays the result with both an icon and text indicators. The job result types are *Scheduled*, *Running*, *Failed*, or *Completed*.
- ◆ **Identification:** This column lists job submission information, including *User*, *Job*, *Name*, and *Priority*.

- ◆ **Schedule:** This column lists information about when the job is scheduled, or its actual start time. It also includes how long the job has been running. If the job is completed, the column also includes the total elapsed run time for job.
- ◆ **Status:** This column lists information about the cost and progress of a job. For jobs that are currently running, the progress indicator bar is partially filled with the percentage of job completion. *Cost* indicates the current cost utilization for a running job, or the total cost for completed and failed jobs. The cost of a job is based on policies set up by administrator for Orchestrator.
- ◆ **Misc:** This column lists other useful information about a job, including its *Subjobs*, its *Joblets* (the number of independent pieces of work running on each computing resource), its *Errors*.
If the number of subjobs is greater than 0, a *view...* link displays in this column. Click this link to show a Job Tree hierarchy under the *Filter* settings for the job. All of the subjobs in the job hierarchy are shown.

Figure 2-8 Job Tree Hierarchy Displayed on the Job Status Page



Click *Root* in the tree to return to the default Job Status display, which is based on the current filter criteria.

Using the Command Line Interface to Monitor Jobs

Use the following command repeatedly if you want to use the command line interface to monitor the job:

```
zos status
```

By default, Orchestrator returns the status of the last job to run.

For more detailed information about using the zos command line interface, see [Appendix A, “The ZOS Command Line Tool,”](#) on page 23.

2.2.5 Viewing the Job Log

From the *Jobs* menu in the portal interface, some jobs might display a *log...* link on the Identification column of the *Job Status* page.

Figure 2-9 Job Identification with an Available Log Link



Selecting this link displays the logging information for the job. The logging display stays current for the job shown running. When the job is completed, its complete log is displayed.

2.2.6 Pausing and Resuming a Running Job

From the *Jobs* menu in the portal, display the job to pause or resume by including it in the *Job Status* list using the filters. Select the running job to pause or resume using the checkbox on the left side.

Then click *Pause* or *Resume* on the bottom of the page. A new page displays indicating the status of the request. Click *OK* to return to the *Job Status* page.

2.2.7 Cancelling a Running Job

Use the following steps to cancel a job that is already running.

- 1 From the *Jobs* menu, display the job to cancel by including it in the *Job Status* list using the filters.
- 2 Select the running Job you want to cancel using the checkbox on the left side.
- 3 Click *Cancel*.

A new page displays indicating the status of the cancel request.

- 4 Click *OK* to return to the *Job Status* page.

2.2.8 Logging Out of the Portal

To log out of the User Portal, click the *Logout* option under the Main Menu.

The ZOS Command Line Tool

A

The zos command line tool can be used by logged-in users. To use the command line tool, each user requires a login name and password that are created by the administrator of Novell® ZENworks® Orchestrator. The zos command line tool can be used to log in to the server for Orchestrator, to submit jobs, monitor jobs, and perform other user functions.

This section includes information about the following:

- ♦ [Section A.1, “List of ZOS Commands,” on page 23](#)
- ♦ [Section A.2, “Getting Started with the ZOS Command,” on page 24](#)
- ♦ [Section A.3, “Details, Usage, and Syntax Examples of ZOS Commands,” on page 27](#)

A.1 List of ZOS Commands

The following table includes a list of zos commands and a description for each command.

Table A-1 Available ZOS Commands and Their Descriptions

Command	Description
login	Log in to the grid
logout	Log out of the grid
password	Change user password on current grid
list	List the available jobs
jobinfo	Display important information for a job
run	Run a job
status	Display the status of a running or previously run job
jobs	List the running queued or historical job
priority	Change the priority of a running job
event	Send an event to a running job
cancel	Cancel a running job
fail	Ask job to end in failure
pause	Pause a running job
resume	Resume a running job
listen	Listen to events generated by a running job
search	Perform grid object search
info	Display information about a grid object
copy	Copy files and directories to and from the datagrid

Command	Description
<code>delete</code>	Delete files and directories in the datagrid
<code>dir</code>	List files and directories in the datagrid
<code>mkdir</code>	Make a new directory in the datagrid
<code>move</code>	Move files and directories in the datagrid
<code>cat</code>	Displays the contents of a datagrid file
<code>log</code>	Displays the log for the specified job
<code>head</code>	Displays the first part of a datagrid file
<code>tail</code>	Displays the end of a datagrid file
<code>help</code>	Displays help for any of the commands in this list

A.2 Getting Started with the ZOS Command

Although many users prefer to use the user Web portal for Orchestrator as they manage jobs, some might prefer a command line interface (CLI). This section is for advanced users or those who prefer CLI.

All zos commands begin with `zos` on the command line. The general format for a zos command is `zos` followed by the command name, followed by command line parameters, if needed:

```
zos command [command_parameters]
```

Before you use the zos CLI, make sure that your path is correctly pointing to the Orchestrator tools directory. On Windows platforms, the path is automatically set when you launch the Orchestrator command prompt window. If the path is set, you then need to log in to the server for Orchestrator before the commands work.

This section includes the following information:

- ◆ [Section A.2.1, “Logging In to the Server for Orchestrator,” on page 24](#)
- ◆ [Section A.2.2, “Running a Job,” on page 25](#)
- ◆ [Section A.2.3, “Monitoring a Job,” on page 25](#)

A.2.1 Logging In to the Server for Orchestrator

The administrator for Orchestrator provides a username, a password, and the name for the server for Orchestrator. The following shows a sample login sequence. Italics are used to indicate variable names.

```
>zos login -user=username Orchestrator_server_name
Please enter current password for 'Orchestrator_server_name': *****
Logged into Orchestrator_server_name_zos> as username
```

The login should now be complete.

Login information is stored in the home directory, so further `zos` commands on this server use the saved login information. To operate on a different server for Orchestrator, run `zos login` again with the new server information.

A.2.2 Running a Job

Use the `run` command to start a job. For example, if you want to run the “quickie” job, you would enter the following command:

```
>zos run quickie
```

If the job submission was successful, a JobID is displayed:

```
JobID: tester.1.quickie
```

To specify parameters when you start a job, add the parameters to the command line. For example, if you wanted to run the quickie job, specifying the `numJoblets` parameter with a value of 100, you would enter the following command:

```
>zos run quickie numJoblets=100
```

```
JobID: tester.2.quickie
```

The server for Orchestrator keeps track of your last started job so that further `zos` commands can default to the last started job. To operate on a different job, specify the Job ID as a command line parameter.

A.2.3 Monitoring a Job

A job can be monitored or managed when it is started or after it is started. To view the status of the last started job, use the `status` command:

```
>zos status
```

```
Running
```

The server returns the status immediately. In the preceding example, the job is `Running`, which means the job is still being processed. Possible status messages include the following:

```
Submitted  
Queued  
Started  
Running  
Paused  
Completing  
Cancelling  
Failing  
Completed  
Cancelled  
Failed
```

You can retrieve detailed status information about a job by using the `-e` option in the command line, as shown in the following example for the last started job:

```
>zos status -e
```

```
Job Status for tester.quickie.2
```

```
-----  
                State: Running (waiting for resources)  
    Resource Count: 0                (0 this job)  
    Percent Complete: 1%  
        Queue Pos: n/a
```

Child Job Count: 0 (0 this job)

Instance Name: quickie
Job Type: quickie
Memo:
Priority: medium
Arguments: numJoblets=100

Submit Time: 5/01/2005 12:20:19
Delayed Start: n/a
Start Time: 5/01/2005 12:20:19
End Time: n/a (estimated)
Elapsed Time: 0:00:05
Queue Time: 0:00:00
Pause Time: 0:00:00

Total CPU Time: 0:00:00 (0:00:00 this job)
Total GCycles: 0:00:00 (0:00:00 this job)
Total Cost: \$0.0000 (\$0.0000 this job)
Burn Rate: \$0.0000/hr (\$0.0000/hr this job)

Cancel Type: <none>
Job Error: <none>

Job Error Count: 0 (0 this job)
Joblet Retry Count: 0 (0 this job)
Node Error Count: 0 (0 this job)

A.3 Details, Usage, and Syntax Examples of ZOS Commands

This section includes a detailed list of the `zos` commands you can use. It also includes examples for using these commands and shows the syntax of typical commands.

NOTE: Items shown in brackets [] are optional. Items shown in *italics* are contextual examples.

The section is organized according to the command names, which include the following:

- ♦ “login” on page 28
- ♦ “logout” on page 29
- ♦ “password” on page 30
- ♦ “list” on page 31
- ♦ “jobinfo” on page 32
- ♦ “run” on page 33
- ♦ “status” on page 35
- ♦ “jobs” on page 36
- ♦ “priority” on page 38
- ♦ “event” on page 39
- ♦ “cancel” on page 40
- ♦ “fail” on page 41
- ♦ “pause” on page 42
- ♦ “resume” on page 43
- ♦ “listen” on page 44
- ♦ “search” on page 45
- ♦ “info” on page 47
- ♦ “copy” on page 48
- ♦ “delete” on page 50
- ♦ “dir” on page 51
- ♦ “mkdir” on page 52
- ♦ “move” on page 53
- ♦ “cat” on page 54
- ♦ “log” on page 55
- ♦ “head” on page 56
- ♦ “tail” on page 57

login

This command logs the user into the server for Orchestrator.

Syntax

```
zos login [servername] [--check] [--user=] [--passwd=] [--port=]
```

Arguments

servername

User must enter the name of the server for Orchestrator he or she is logging into. The default is `localhost`.

Options

-c, --check

Check and report existing login. Ignores other options.

-u=, --user=

Username known by the Orchestrator server. The user must enter a value with this option. The default is `login`.

-p=, --passwd=

User password for authentication. The user must enter a value with this option.

-P=, --port=

The port number used for communication between the user portal and the server for Orchestrator. The user must enter a value with this option.

-C=, --cols=

The specification for the terminal width for formatting. The user must enter a value with this option.

-h, --help

Displays a help message for this operation.

-d, --debug

Turns on debug output during the login operation.

-v, --verbose

Turns on verbose output during the login operation.

Examples

Example 1

To log in to the server *Eng* as *tester20* you would use the following command:

```
zos login Eng --user=tester20
```

logout

This command logs the user out of the server for Orchestrator.

Syntax

```
zos logout
```

Options

-C=, --cols=

Specifies the terminal width for formatting. The user must enter a value with this option.

-h, --help

Displays a help message for this operation.

-d, --debug

Turns on debug output during the logout operation.

-v, --verbose

Turns on verbose output during the logout operation.

password

This command lets the user change his or her password on the current server for Orchestrator.

Syntax

```
zos password [--passwd=] mypassword
```

Arguments

mypassword

The user must enter the current password for authentication.

Options

-p=, --passwd=

Current password for authentication. The user must enter a value with this option.

-C=, --cols=

Specifies the terminal width for formatting. The user must enter a value with this option.

-h, --help

Displays a help message for this operation.

-d, --debug

Turns on debug output during the logout operation.

-v, --verbose

Turns on verbose output during the logout operation.

list

This command lists all of the jobs that have been deployed by the administrator for ZENworks Orchestrator.

Syntax

```
zos list [--detail] [--description]
```

Options

-e, --detail

Displays detailed job information for each job.

-D, --description

Displays a description for each job.

-C=, --cols=

Specifies the terminal width for formatting. The user must enter a value with this option.

-h, --help

Displays a help message for this operation.

-d, --debug

Turns on debug output during this operation.

-v, --verbose

Turns on verbose output during this operation.

Examples

Example 1

To generate a listing of all runnable jobs, you would use the following command:

```
zos list -e
```

jobinfo

This command displays important information about a given job for Orchestrator.

Syntax

```
zos jobinfo [--detail] jobname
```

Arguments

jobname

The user must enter the name of the job for which he or she is seeking information.

Options

-e, --detail

Displays detailed job information for each job.

-C=, --cols=

Specifies the terminal width for formatting. The user must enter a value with this option.

-h, --help

Displays a help message for this operation.

-d, --debug

Turns on debug output during this operation.

-v, --verbose

Turns on verbose output during this operation.

Examples

Example 1

To display a quick list of job parameters for the `quickie` sample job, the user would enter this command:

```
zos jobinfo quickie
```


run

This command runs a job.

Syntax

```
zos run jobname [--priority=] [--as=] [--watch] [--env=] [--start=]
      [--listen] [--policy=] [--trace]
```

Arguments

jobname

The user must enter the name of the job he or she wants to run.

Options

-r=, priority=

Specifies the priority level for a job. The user must enter a value with this option.

-a=, --as=

Specifies the name of the job instance. The user must enter a value with this option.

-W, --watch

Lets user watch job log messages in real time.

-V=, --env=

Lets user use the local environment for running a job or to use an optional list. The user must enter a value with this option.

-s=, start=

Specifies the start time for a job. The user must enter a value with this option.

-l, listen

Lets user listen for job messages until the job is completed.

-P=, policy=

Lets the user specify the policy file to be associated with this job. The user must enter a value with this option.

-t, --trace

Lets the user turn on event tracing for this job.

-C=, --cols=

Specifies the terminal width for formatting. The user must enter a value with this option.

-h, --help

Displays a help message for this operation.

-d, --debug

Turns on debug output during this operation.

-v, --verbose

Turns on verbose output during this operation.

Examples

Example 1

To run the job called *unittest* with the *highest* priority and another parameter, the user would use the following command:

```
zos run unittest --priority=highest testno=test57
```

Example 2

To run the job called *unittest* passing the PATH environment variable only, the user would use the following command:

```
zos run unittest --env=PATH
```

Example 3

To run the job called *unittest* at *5:01 p.m.*, the user would use the following command:

```
zos run unittest --start="5:01 PM"
```

NOTE: The date and time values should be specified as shown in the example—in the format MM/DD/YY hour:minute AM/PM and enclosed in quotes—"11/21/05 4:04 PM".

status

This command displays the status of a running or previously run job.

Syntax

```
zos status [jobid] [--detail] [--errors] [--recurse] [--joblets]
```

Arguments

jobid

The user must enter the jobid that he or she is interested in. The default for this argument is the last job run.

Options

-e, --detail

Displays joblet information for the job.

-E, --errors

Displays job and node error details.

-r, --recurse

Shows the full job hierarchy. This option can be used with or without the `--detail` option.

-J, --joblets

Retrieves individual joblet details. This option can be used only with the `--detail` option.

-C=, --cols=

Specifies the terminal width for formatting. The user must enter a value with this option.

-h, --help

Displays a help message for this operation.

-d, --debug

Turns on debug output during this operation.

-v, --verbose

Turns on verbose output during this operation.

jobs

This command lists the running queued or historical jobs.

Syntax

```
zos jobs [--all] [--detail] [--errors] [--submit=] [--hours=]
        [--active] [--count=] [--parent=]
```

Options

-a, --all

Lists all users' jobs, not just the logged in user's jobs.

-e, --detail

Displays detailed job information for each job.

-E, --errors

Displays details of job and node errors.

-s=, --submit=

Specifies a time when the user wants to see historical jobs. The user must enter a value with this option.

-h=, --hours=

Specifies submit time in the form of last xxx hours. If specified, this option overrides the `submit` option. The user must enter a value with this option.

-A, --active

Lists only the currently active or running jobs.

-c=, --count=

Sets the maximum number of historical jobs retrieved. The user must enter a value with this option.

-p=, --parent=

Specifies the parent job id. If this option is set, only the child jobs of this parent are displayed. The user must enter a value with this option.

-C=, --cols=

Specifies the terminal width for formatting. The user must enter a value with this option.

-h, --help

Displays a help message for this operation.

-d, --debug

Turns on debug output during this operation.

-v, --verbose

Turns on verbose output during this operation.

Examples

Example 1

To display a detailed list of running jobs for all users, the user would enter the following command:

```
zos jobs -a -e
```

Example 2

To display a list of queued, running, and recently run jobs, the user would enter the following command:

```
zos jobs
```

Example 3

To display a list of jobs he or she submitted since 2:20 p.m., the user would enter the following command:

```
zos jobs --submit="2:20 PM"
```

NOTE: The date and time values should be specified as shown in the example—in the format MM/DD/YY hour:minute AM/PM and enclosed in quotes—"11/21/05 4:04 PM".

Example 4

To display a list of all active subjobs for the jobid `barney.load.677`, the user would enter the following command:

```
zos jobs -A --parent=barney.load.677 --all
```

Example 5

To display a list of all his or her jobs run in the past 24 hours, the user would enter the following command:

```
zos jobs -h 24
```

priority

This command changes the priority of a running job.

Syntax

```
zos priority [jobid] [--priority=]
```

Arguments

jobid

The user must enter the jobid that he or she is interested in. The default for this argument is the last job run.

Options

-r=, --priority=

Specifies the priority level for job. The user must enter a value with this option.

-C=, --cols=

Specifies the terminal width for formatting. The user must enter a value with this option.

-h, --help

Displays a help message for this operation.

-d, --debug

Turns on debug output during this operation.

-v, --verbose

Turns on verbose output during this operation.

Examples

Example 1

To change the current job's priority to lowest, the user would enter the following command:

```
zos priority --priority=lowest
```

Example 2

To reset priority back to the user default (*paul*), the user would enter the following command:

```
zos priority paul.quickie.23
```

event

This command sends an event to a running job.

Syntax

```
zos event jobid eventname
```

Arguments

jobid

The user must enter the jobid that he or she is interested in. The default for this argument is the last job run.

event

The user must enter the name of the event he or she wants to send to the job.

Options

-C=, --cols=

Specifies the terminal width for formatting. The user must enter a value with this option.

-h, --help

Displays a help message for this operation.

-d, --debug

Turns on debug output during this operation.

-v, --verbose

Turns on verbose output during this operation.

cancel

This command cancels a running job.

Syntax

```
zos cancel [jobid] [--reason=]
```

Arguments

jobid

The user must enter the jobid that he or she is interested in. The default for this argument is the last job run.

Options

-m=, --reason=

Specifies the reason to log for this action. The user must enter a value with this option.

-C=, --cols=

Specifies the terminal width for formatting. The user must enter a value with this option.

-h, --help

Displays a help message for this operation.

-d, --debug

Turns on debug output during this operation.

-v, --verbose

Turns on verbose output during this operation.

fail

This command directs the job to end in failure.

Syntax

```
zos fail [jobid] [--reason=]
```

Arguments

jobid

The user must enter the jobid that he or she is interested in. The default for this argument is the last job run.

Options

-m=, --reason=

Specifies the reason to log for this action. The user must enter a value with this option.

-C=, --cols=

Specifies the terminal width for formatting. The user must enter a value with this option.

-h, --help

Displays a help message for this operation.

-d, --debug

Turns on debug output during this operation.

-v, --verbose

Turns on verbose output during this operation.

pause

This command directs the running job to pause.

Syntax

```
zos pause [jobid] [--reason=]
```

Arguments

jobid

The user must enter the jobid that he or she is interested in. The default for this argument is the last job run.

Options

-C=, --cols=

Specifies the terminal width for formatting. The user must enter a value with this option.

-h, --help

Displays a help message for this operation.

-d, --debug

Turns on debug output during this operation.

-v, --verbose

Turns on verbose output during this operation.

resume

This command directs the running job to resume.

Syntax

```
zos resume [jobid] [--reason=]
```

Arguments

jobid

The user must enter the jobid that he or she is interested in. The default for this argument is the last job run.

Options

-C=, --cols=

Specifies the terminal width for formatting. The user must enter a value with this option.

-h, --help

Displays a help message for this operation.

-d, --debug

Turns on debug output during this operation.

-v, --verbose

Turns on verbose output during this operation.

listen

This command starts a process to listen to events generated by a running job.

Syntax

```
zos listen [jobid] [--reason=]
```

Arguments

jobid

The user must enter the *jobid* that he or she is interested in. The default for this argument is the last job run.

Options

-e, --detail

Displays detailed job information for each job.

-C=, --cols=

Specifies the terminal width for formatting. The user must enter a value with this option.

-h, --help

Displays a help message for this operation.

-d, --debug

Turns on debug output during this operation.

-v, --verbose

Turns on verbose output during this operation.

search

This command performs a grid object search.

Syntax

```
zos search [--detail] [--facts] [--type=] [--constraint=] [--order=]
```

Arguments

-t=, --type=

The user must specify one of the following grid object types:

- ♦ user
- ♦ resource
- ♦ group
- ♦ job
- ♦ jobinstance
- ♦ vmhost

Options

-e, --detail

Displays detailed information for each grid object rather than just a name.

-f, --facts

Displays facts for each grid object.

-c=, --constraint=

Specifies a constraint string to restrict search. The command defaults to "" to match everything. The user must enter a value with this option.

-o=, --order=

Specifies returned order (*fact/a|d, fact/a|d,..*).

-C=, --cols=

Specifies the terminal width for formatting. The user must enter a value with this option.

-h, --help

Displays a help message for this operation.

-d, --debug

Turns on debug output during this operation.

-v, --verbose

Turns on verbose output during this operation.

Examples

Example 1

To generate a detailed listing of all users in group *eng*, the user would enter the following command:

```
zos search -t user -c '<contains fact="user.groups" value="eng"/>' -o  
"user.location/d"
```

info

This command displays information about a grid object.

Syntax

```
zos info objectName --type= [--detail]
```

Arguments

objectName

The user must enter the name of the object that he or she is interested in.

-t=, --type=

The user must specify one of the following grid object types:

- ♦ user
- ♦ resource
- ♦ group
- ♦ job
- ♦ jobinstance
- ♦ vmhost

Options

-e, --detail

Displays detailed job information for each job.

-C=, --cols=

Specifies the terminal width for formatting. The user must enter a value with this option.

-h, --help

Displays a help message for this operation.

-d, --debug

Turns on debug output during this operation.

-v, --verbose

Turns on verbose output during this operation.

Examples

Example 1

To get detailed information on user *paul*, use the following command:

```
zos info paul -t user
```

copy

This command copies files and directories to and from the data zos.

Syntax

```
zos copy source dest [--recursive]
```

Arguments

source

The datagrid source from which the files or directories are to be copied.

dest

The datagrid destination to which the files or directories are to be copied.

Options

-r, --recurse

Shows the full job hierarchy. This option can be used with or without the `--detail` option.

-C=, --cols=

Specifies the terminal width for formatting. The user must enter a value with this option.

-h, --help

Displays a help message for this operation.

-d, --debug

Turns on debug output during this operation.

-v, --verbose

Turns on verbose output during this operation.

Examples

Example 1

To copy a file from the job instance directory to a local file, use the following command:

```
zos copy grid:///^/results.txt local.txt
```

Example 2

To copy a file from a named job instance directory to a local file, use the following command:

```
zos copy grid:///^user.myjob.1024/results.txt local.txt
```

Example 3

To copy a local file into the job directory for job *myjob*, use the following command:

```
zos copy local.txt grid:///!myjob
```


Example 4

To copy a local file into a subdirectory of the job directory, use the following command:

```
zos copy local.txt grid:///!myjob/subdir
```

Example 5

To copy a local file into the current job instance directory, use the following command:

```
zos copy local.txt grid:///^/
```

delete

This command deletes files and directories in the data grid.

Syntax

```
zos delete path [--recursive]
```

Arguments

path

The path on the data grid.

Options

-r, --recurse

Shows the full job hierarchy. This option can be used with or without the `--detail` option.

-C=, --cols=

Specifies the terminal width for formatting. The user must enter a value with this option.

-h, --help

Displays a help message for this operation.

-d, --debug

Turns on debug output during this operation.

-v, --verbose

Turns on verbose output during this operation.

dir

This command lists files and directories in the data grid.

Syntax

```
zos dir path
```

Arguments

path

The path on the data grid.

Options

-C=, --cols=

Specifies the terminal width for formatting. The user must enter a value with this option.

-h, --help

Displays a help message for this operation.

-d, --debug

Turns on debug output during this operation.

-v, --verbose

Turns on verbose output during this operation.

Examples

Example 1

To list files for the current job instance, use the following command:

```
zos dir grid:///^
```

Example 2

To list the deployed files for the job named *animation*, use the following command:

```
zos dir grid:///!\!animation
```

mkdir

This command makes a new directory in the data grid.

Syntax

```
zos mkdir path [--parents]
```

Arguments

path

The path on the data grid.

Options

-p, --parents

Creates parent directories if needed.

-C=, --cols=

Specifies the terminal width for formatting. The user must enter a value with this option.

-h, --help

Displays a help message for this operation.

-d, --debug

Turns on debug output during this operation.

-v, --verbose

Turns on verbose output during this operation.

move

This command moves files and directories in the data grid.

Syntax

```
zos move source dest
```

Arguments

source

The datagrid source from which the files or directories are to be moved.

dest

The datagrid destination to which the files or directories are to be moved.

Options

-C=, --cols=

Specifies the terminal width for formatting. The user must enter a value with this option.

-h, --help

Displays a help message for this operation.

-d, --debug

Turns on debug output during this operation.

-v, --verbose

Turns on verbose output during this operation.

Examples

Example 1

To display the JDL file for the benchmark job, use the following command:

```
zos cat grid:///!benchmark/benchmark.jdl
```

Example 2

To display the `job.log` file for the active or last run job, use the following command:

```
zos cat grid:///^/job.log
```

cat

This command displays the contents of a data grid file.

Syntax

```
zos cat path
```

Arguments

path

The path on the data grid.

Options

-C=, --cols=

Specifies the terminal width for formatting. The user must enter a value with this option.

-h, --help

Displays a help message for this operation.

-d, --debug

Turns on debug output during this operation.

-v, --verbose

Turns on verbose output during this operation.

log

This command displays the log for the specified job.

Syntax

```
zos log [jobid]
```

Arguments

jobid

The user must enter the jobid that he or she is interested in. The default for this argument is the last job run.

Options

-C=, --cols=

Specifies the terminal width for formatting. The user must enter a value with this option.

-h, --help

Displays a help message for this operation.

-d, --debug

Turns on debug output during this operation.

-v, --verbose

Turns on verbose output during this operation.

Examples

Example 1

To display the log for the last job, use the following command:

```
zos log
```

Example 2

To display the log for job 3 on the list returned by `zos jobs`, use the following command:

```
zos log ^3
```

Example 3

To display the log for job id *itsajob*, use the following command

```
zos log itsajob
```

head

This command displays the first part of a data grid file.

Syntax

```
zos head [--bytes=] [--lines=] [--quiet] path
```

Arguments

path

The path on the data grid.

Options

-c=, --bytes=

Generate file output of the specified number of bytes only. The user must enter a value with this option.

-n=, --lines=

Generate file output of the specified number of lines only. The user must enter a value with this option.

-q, --quiet

Suppress all warning and informational messages.

-C=, --cols=

Specifies the terminal width for formatting. The user must enter a value with this option.

-h, --help

Displays a help message for this operation.

-d, --debug

Turns on debug output during this operation.

-v, --verbose

Turns on verbose output during this operation.

tail

This command displays the end of a data grid file.

Syntax

```
zos tail [--retry] [--bytes=] [--follow] [--by-name] [--follow-retry]
[--lines=] [--max-unchanged=] [--jobid=] [--sleep-interval=] [--
refresh=] [--count=] [--timeout=] [--quiet] path
```

Arguments

path

The path on the data grid.

Options

-r, --retry

Continues trying to open the file if it is not yet accessible.

-c=, --bytes=

Generate file output of the specified number of bytes only. The user must enter a value with this option.

-f, --follow

Follow the file, watching for newly appended data.

-N, --by-name

If following a file, periodically reopen the file.

-F, --follow-retry

The same as `--follow --by-name --retry`

-n=, --lines=

Generate file output of the specified number of lines only. The user must enter a value with this option.

-U=, max-unchanged=

The number of size checks between reopening a followed file (used with `--by-name` only). The user must enter a value with this option.

-J=, jobid=

If following a file, follow until the specified job terminates. The user must enter a value with this option.

-s=, --sleep-interval=

The frequency of file size checks in seconds when following a file. The user must enter a value with this option.

-R, refresh=

Server connection refresh rate. The user must enter a value with this option.

-X, --count=

Maximum number of lines/bytes to display.

-t, --timeout=

The maximum number of seconds to follow a file.

-q, --quiet

Suppress all warning and informational messages.

-C, --cols=

Specifies the terminal width for formatting. The user must enter a value with this option.

-h, --help

Displays a help message for this operation.

-d, --debug

Turns on debug output during this operation.

-v, --verbose

Turns on verbose output during this operation.

Examples

Example 1

To display the last 5 lines of `grid:///benchmark.jdl`, use the following command:

```
zos tail -n 5 grid:///!benchmark/benchmark.jdl
```

Example 2

To display the last 5 lines of `grid:///benchmark.jdl` in an alternative way, use the following command:

```
zos tail -5 grid:///!benchmark/benchmark.jdl
```

Example 3

To follow a file in (approximately) real time, use the following command:

```
zos tail -f grid:///^myjob/job.log
```

Example 4

To continually watch the job log of the latest job, use the following command:

```
zos tail --follow --by-name --retry grid:///!lastjob/job.log
```

Example 5

To watch the job log of job `myname.myjob.123` while it runs, use the following command:

```
zos tail -F -J myname.myjob.123 grid:///^myname.myjob.123/job.log
```