Altair Access 2020.4

Command Line Interface Guide

Updated: 12/07/2020

Intellectual Property Rights Notice

Copyrights, trademarks, trade secrets, patents and third party software licenses.

Altair PBS Works™ v. 2020.4 Copyright © 1994-2020.

High-performance Innovation

Altair Engineering Inc. Copyright © 1986-2020. All Rights Reserved.

Altair HyperWorksX™ Copyright © 2019-2020

> **Notice:** Pre-release versions of Altair software are provided 'as is', without warranty of any kind. Usage of pre-release versions is strictly limited to non-production purposes.

Altair PBSWorks™ - Accelerating Innovation in the Cloud™

Altair® PBS Professional® ©1994-2020

Altair Control™ ©2008-2020; (formerly **PBS Control**)

Altair Access™ ©2008- 2020; (formerly **PBS Access**)

Altair Accelerator™ ©1995- 2020; (formerly **NetworkComputer**)

Altair Accelerator Plus™ ©1995- 2020; (formerly **WorkloadXelerator**)

Altair FlowTracer™ ©1995- 2020; (formerly **FlowTracer**)

Altair Allocator™ ©1995- 2020; (formerly **LicenseAllocator**)

Altair Monitor™ ©1995- 2020; (formerly **LicenseMonitor**)

Altair Hero™ ©1995- 2020; (formerly **HERO**)

Altair Software Asset Optimization™ (SAO) ©2007- 2020

> **Note:**
>
> **Compute Manager™** ©2012-2017 is now part of **Altair Access**
>
> **Display Manager™** ©2013-2017 is now part of **Altair Access**
>
> **PBS Application Services™** ©2008-2017 is now part of **Altair Access**
>
> **PBS Analytics™** ©2008-2017 is now part of **Altair Control**
>
> **PBS Desktop™** ©2008-2012 is now part of **Altair Access**, specifically **Altair Access desktop,** which also has **Altair Access web** and **Altair Access mobile**
>
> **e-Compute™** ©2000-2010 was replaced by "**Compute Manager**" which is now **Altair Access**

Altair HyperWorks™ - The Platform for Innovation™

Altair AcuConsole™ ©2006-2020

Altair AcuSolve™ ©1997-2020

Altair ElectroFlo™ ©1992-2020

Altair ESAComp™ ©1992-2020

Altair Feko™ ©1999-2014 Altair Development S.A. (Pty) Ltd.; ©2014-2020 Altair Engineering Inc.

Altair Flux™ ©1983-2020

Altair FluxMotor™ ©2017-2020

Altair HyperCrash™ ©2001-2020

Altair HyperGraph™ ©1995-2020

Altair HyperLife™ ©1990-2020

Altair HyperMesh™ ©1990-2020

Altair HyperStudy™ ©1999-2020

Altair HyperView™ ©1999-2020

Altair Virtual Wind Tunnel™ ©2012-2020

Altair HyperXtrude™ ©1999-2020

Altair Manufacturing Solver™ ©2011-2020

Altair MotionSolve™ ©2002-2020

Altair MotionView™ ©1993-2020

Altair Multiscale Designer™ ©2011-2020

Altair OptiStruct™ ©1996-2020

Altair Radioss™ ©1986-2020

Altair Seam™ ©1985-2019 Cambridge Collaborative, Inc., © 2019-2020 Altair Engineering Inc.

Altair SimLab™ ©2004-2020

Altair SimSolid™ ©2015-2020

Altair nanoFluidX™ © 2013-2018 Fluidyna GmbH, © 2018-2020 Altair Engineering Inc.

Altair ultraFluidX™ © 2010-2018 Fluidyna GmbH, © 2018-2020 Altair Engineering Inc.

Altair WinProp™ ©2000-2020

Altair ConnectMe™ ©2014-2020

Plus other products from the Altair solidThinking Platform.

Altair Packaged Solution Offerings (PSOs)

Altair Automated Reporting Director™ ©2008-2020

Altair GeoMechanics Director™ ©2011-2020

Altair Impact Simulation Director™ ©2010-2020

Altair Model Mesher Director™ ©2010-2020

Altair NVH Director™ ©2010-2020

Altair Squeak and Rattle Director™ ©2012-2020

Altair Virtual Gauge Director™ ©2012-2020

Altair Weight Analytics™ ©2013-2020

Altair Weld Certification Director™ ©2014-2020

Altair Multi-Disciplinary Optimization Director™ ©2012-2020.

Altair solidThinking - Where Innovation Takes Shape™

Altair Inspire™ ©2009-2020 including Altair Inspire Motion and Altair Inspire Structures

Altair Inspire Extrude Metal ©1996-2020 (formerly Click2Extrude®-Metal)

Altair Inspire Extrude Polymer ©1996-2020 (formerly Click2Extrude®-Polymer)

Altair Inspire Cast ©2011-2020 (formerly Click2Cast®)

Altair Inspire Form ©1998-2020 (formerly Click2Form®)

Altair Inspire Mold ©2009-2020 (initial release-Q2 2019)

Altair Inspire Render ©1993-2016 Solid Iris Technologies Software Development One PLLC, © 2016-2020 Altair Engineering Inc (initial release-Q3 2019, formerly Thea Studio)

Altair Inspire Studio ©1993-2020 (formerly 'Evolve')

Altair Compose™ ©2007-2020 (formerly solidThinking Compose®)

Altair Activate™ ©1989-2020 (formerly solidThinking Activate®)

Altair Embed™ ©1989-2020 (formerly solidThinking Embed®)

- **Altair Embed SE** ©1989-2020 (formerly solidThinking Embed® SE)
- **Altair Embed/Digital Power Designer** ©2012-2020

Altair SimLab™ ©2004-2020

Altair 365™ ©1994-2020

Altair SmartWorks™ - Innovation Intelligence®

Altair intellectual property rights are protected under U.S. and international laws and treaties. Additionally, Altair software is protected under patent #6,859,792 and other patents pending. All other marks are the property of their respective owners.

ALTAIR ENGINEERING INC. Proprietary and Confidential. Contains Trade Secret Information.

Not for use or disclosure outside of Altair and its licensed clients. Information contained in Altair software shall not be decompiled, disassembled, "unlocked", reverse translated, reverse engineered, or publicly displayed or publicly performed in any manner. Usage of the software is only as explicitly permitted in the end user software license agreement. Copyright notice does not imply publication.

Third party software licenses

AcuConsole contains material licensed from Intelligent Light (www.ilight.com) and used by permission.

Software Security Measures:

Altair Engineering Inc. and its subsidiaries and affiliates reserve the right to embed software security mechanisms in the Software for the purpose of detecting the installation and/or use of illegal copies of the Software. The Software may collect and transmit non-proprietary data about those illegal copies. Data collected will not include any customer data created by or used in connection with the Software and will not be provided to any third party, except as may be required by law or legal process or to enforce our rights with respect to the use of any illegal copies of the Software. By using the Software, each user consents to such detection and collection of data, as well as its transmission and use if an illegal copy of the Software is detected. No steps may be taken to avoid or detect the purpose of any such security mechanisms.

Technical Support

Altair provides comprehensive software support via telephone and e-mail.

Telephone and E-mail

When contacting Altair support, please specify the product and version number you are using along with a detailed description of the problem. Many times, it is very beneficial for the support engineer to know what type of workstation, operating system, RAM, and graphics board you have, so please have that information ready. If you send an e-mail, please specify the workstation type, operating system, RAM, and graphics board information in the e-mail.

To contact an Altair support representative, reference the following table.

Location	Telephone	E-mail
Australia	+61 3 9866 5557 +61 4 1486 0829	anz-pbssupport@altair.com
China	+86 21 6117 1666	pbs@altair.com.cn
France	+33 (0)1 4133 0992	pbssupport@europe.altair.com
Germany	+49 (0)7031 6208 22	pbssupport@europe.altair.com
India	+91 80 66 29 4500 +1 800 208 9234 (Toll Free)	pbs-support@india.altair.com
Italy	+39 800 905595	pbssupport@europe.altair.com
Japan	+81 3 6225 5821	pbs@altairjp.co.jp
Korea	+82 70 4050 9200	support@altair.co.kr
Malaysia	+91 80 66 29 4500 +1 800 208 9234 (Toll Free)	pbs-support@india.altair.com
North America	+1 248 614 2425	pbssupport@altair.com
Russia	+49 7031 6208 22	pbssupport@europe.altair.com
Scandinavia	+46 (0) 46 460 2828	pbssupport@europe.altair.com
Singapore	+91 80 66 29 4500 +1 800 208 9234 (Toll Free)	pbs-support@india.altair.com

Location	Telephone	E-mail
South Africa	+27 21 831 1500	pbssupport@europe.altair.com
South America	+55 11 3884 0414	br_support@altair.com
United Kingdom	+44 (0)1926 468 600	pbssupport@europe.altair.com

Contents

What's New

1

Latest features available with Altair Access™ Command Line Interface.

There are no new features in this release.

Get Started

2

Use the Command Line Interface for job submission and monitoring, file operations, and managing servers.

This chapter covers the following:

The Access Command Line Interface (CLI) allows users to access the power of PBS Professional from the command-line without using a graphical user interface, while leveraging the capabilities of Access and application definitions. The CLI is available with Access 2020.4 and later releases.

A variety of commands are available to assist user's with job submission and monitoring activities, as well as file operations. Additionally, server and administrative commands are available.

2.1 Document Conventions

Common typographical conventions for Altair Access Command Line Interface™ technical publications.

PA_HOME
: The Access Command Line Interface home directory which contains configuration, data, and logging files. Default location is:

  - on Linux: `/var/spool/pbsworks/2020.4/access/home`
  - on Windows Docker: `/opt/altair/access/home`

  However, this can be overridden during the installation of Access Command Line Interface.

PA_EXEC
: The Access Command Line Interface execution directory which contains binaries and scripts. Default location is:

  - on Linux: `/opt/altair/pbsworks/2020.4/access/exec`
  - on Windows Docker: `/opt/altair/access/exec`

  However, this can be overridden during the installation of Access Command Line Interface.

PBS_HOME
: The location where the PBS Professional daemon/service configuration files, accounting logs, etc. are installed. Default is:

  `/var/spool/pbs`

PBS_EXEC
: The location where the PBS Professional executable programs are installed. Default is:

  `/opt/pbs/`

PAS_HOME
: The Windows PAS home directory which contains configuration, data, and logging files. Default location is: `C:\Program Files\altair\pas\2020.4\PAS\home`, however this can be overridden during the installation of PAS on Windows.

PAS_EXEC
: The Windows PAS execution directory which contains binaries and scripts. Default location is: `C:\Program Files\altair\pas\2020.4\PAS\exec`, however this can be overridden during the installation of PAS on Windows.

2.2 Prerequisites

Prerequisites and planning to run Command Line Interface.

Access Desktop

Before you can use the CLI to submit and monitor jobs, administer a cluster, or manage files you must establish a connection to an HPC cluster. Establish the connection by using the Access user interface or by issuing the `pas-add-server` command. The first cluster that is registered is considered the default server. Any CLI commands issued after adding the cluster, are made against this cluster unless otherwise explicitly specified. Most commands provide an option to specify a non-default server.

- Access Desktop must be installed and running.
- A connection to an HPC cluster must be established by registering the cluster via Access Desktop or by using the `pas-server-add` command.
- The first time a CLI command is issued after installation, you will be asked for your credentials. You can use the `pas-passwd` command to set the password credentials.
- Users of the CLI will need to add the path to the CLI to their search path.

Access Web

- Access Web must be installed and running.
- A connection to an HPC cluster must be established by registering the cluster via Access Web UI or by using the `pas-server-add` command.
- The first time a CLI command is issued after installation, you will be asked for your credentials. You can use the `pas-passwd` command to set the password credentials.
- Users of the CLI will need to add the path to the CLI to their search path.

2.3 Command Line Interface Path

Default location of Command Line Interface path.

Access Desktop

`C:\Users\<user name>\AppData\Local\Altair\Altair Access\2020.4\exec\bin\commandline\`

Access Web

`/opt/altair/pbsworks/2020.4/access/exec/bin/commandline/`

2.4 Commands Quick Reference

List of access commands.

The following is a list of all commands and a link to each of the commands.

Table 1: Administration Commands

Command	Description
pas-ping	Checks if the server is up or down.
pas-getsrv	Displays information about the server.
pas-ver	Displays server version and build information.
pas-server-add	Add a server in the Access Desktop application.
pas-server-list	Deletes a server from the Access Desktop application.
pas-server-delete	Returns a list of all the server added in the Access Desktop application.
pas-admin	Determines if the user has administrative privileges on the server.
pas-getsroot	Returns the root directory of the staging area on the server.
pas-passwd	Set the user’s password.

Table 2: Job Submission and Monitoring Commands

Command	Description
pas-getapps	Returns a list of all applications known to the server.
pas-getapp	Returns a list of all parameters available for the specified application as defined by the application definition on the server.
pas-submit	Submits a job to the server.
pas-stat	Displays the status of a job. You can display the status of a particular job or the status of all jobs.
pas-sum	Displays a summary of the job information. You can displays a summary of job information for a specific job, or all jobs on the server.

Command	Description
pas-getres	Downloads the files related to the given job identifier.
pas-del	Deletes a job from the Workload Management System.
pas-custom-action-list	Get a list of available custom action of a job.
pas-execute-custom-action	Execute a custom action for a job.

Table 3: File Management Commands

Command	Description
pas-fmkdir	Creates a directory on the server.
pas-fup	Uploads a file.
pas-fdown	Downloads a file.
pas-fdel	Deletes a file from the server.
pas-fcompr	Compresses a file on the server.
pas-flist	Displays a list of files on the server.

Administration Commands

3

Commands to help with the administrative tasks.

This chapter covers the following:

All commands will connect to the first server (cluster) registered after installation (this server is considered the default server). Most commands provide an option to specify a non-default PAS server.

3.1 Determine if Server is Up and Running

Check if the PAS server (cluster) is up or down.

Name

`pas-ping`

Description

Use the `pas-ping` command to check if the PAS server (cluster) is up or down. This command will return a message indicating the status of the server:

`PAS server at machine name is up/down.`

Syntax

`pas-ping` [*options*]

Options

-h

Display help for the command.

-s SERVERNAME

Ping the PAS server specified by SERVERNAME. SERVERNAME is the name of server (cluster) provided while adding the server (cluster) using Access. If this option is not specified, then the default PAS server will be pinged.

> **Note:** The *-s SERVERNAME* option is not supported for Access Desktop.

-F JSON

Print the output format in JSON.

Command Examples

Ping the default PAS server

```
pas-ping
```

Ping a non-default PAS server

```
pas-ping -s 192.168.4.123
```

Ping a PAS server and get the output in JSON format

```
pas-ping -s 192.168.4.123 -F JSON
```

Command Output Example

```
pas-ping
```

```
    PAS server at localhost is up
```

```
pas-ping -s 192.168.4.123 -F JSON
    {u'data': {u'host': u'localhost', u'id': u'localhost', u'isDefault': False},
    u'success': True, u'exitCode': u'0'}
```

3.2 Get Server Information

Display the hostname and port number of a server.

Name

`pas-getsrv`

Description

Use the `pas-getsrv` command to display the hostname and port number of a server.

Syntax

`pas-getsrv` [*options*]

Options

-h

Display help for the command.

-s SERVERNAME

Returns the hostname and port number of the PAS server specified by ***SERVERNAME***. If this option is not specified, then the default server and its port information is displayed.

> **Note:** The ***-s SERVERNAME*** option is not supported for Access Desktop.

Command Example

Get information about the default server

```
pas-getsrv
```

Command Output Example

```
pas-getsrv
    Server=localhost
    Port=17084
```

3.3 Get Server Version and Build Information

Display server version and build information.

Name

`pas-ver`

Description

Use the `pas-ver` command to display server version and build information.

Syntax

`pas-ver` [*options*]

Options

-h

Display help for the command.

-s SERVERNAME

Display version and build information for the PAS server specified by ***SERVERNAME***. If this option is not specified, then the default server and its port information is displayed.

> **Note:** The ***-s SERVERNAME*** option is not supported for Access Desktop.

-F JSON

Print the output format in JSON.

Command Examples

Display the server version and build information for the default server

```
pas-ver
```

Display the server version and build information for a non-default server

```
pas-ver -s 192.168.4.123
```

Display the server version in JSON format

```
pas-ver -s 192.168.4.123 -F JSON
```

Command Output Example

```
pas-ver
```

```
    majorVersionNumber :: 2018
    minorVersionNumber :: 2
    patchNumber :: 0
    buildId :: 20180624
```

```
    platform :: Linux-x86_64
    arch :: amd64
```

```
pas-ver -s 192.168.4.123 -F JSON
    {"majorVersionNumber": 2018, "patchNumber": 0, "buildId": "20180624", "arch":
    "amd64", "platform": "Linux-x86_64", "minorVersionNumber": 2}
```

3.4 Add a Server

Add a service cluster using SSH or an Access Web server using HTTPS.

Name

`pas-server-add`

Description

Use the `pas-server-add` command to add a service cluster (i.e. a PBS Professional complex) using SSH or an Access Web server using HTTPS.

Use the `pas-server-add` command to add a service cluster (i.e. a PBS Professional complex) using SSH by providing:

- ***SERVERNAME***, the name by which the server will be known within Altair Access. The special characters allowed while providing the ***SERVERNAME*** are dot, underscore, hyphen, and space. The ***SERVERNAME*** should not end with white space.
- ***HOSTNAME*** or ***IPADDRESS*** of the server.
- ***Port Number*** of the server (Optional). This is the gateway port which is by default 4443.
- ***USERNAME*** and either a ***PASSWORD*** or an ***SSHKEYPATH***. The credentials of the user should be available in the server.

Use the `pas-server-add` command to add an Access Web server using HTTPS by providing:

- ***SERVERNAME***, the name by which the server will be known within Altair Access. The special characters allowed while providing the ***SERVERNAME*** are dot, underscore, hyphen, and space. The ***SERVERNAME*** should not end with white space.
- ***Access Web URL***, the Access Web server URL should be provided in the format `https://<HOSTNAME>:<PORT>` and the port number is the gateway port.
- ***USERNAME*** and a ***PASSWORD*** . The credentials of the user should be available in the server.
- **-a *PBS_COMMAND_LINE_ADAPTOR*** - The Adaptor type option used for adding Access Web server.

> **Note:** This command is valid for Access Desktop only.

By default, the user who issues this command is logged into the ***SERVER***.

Syntax

`pas-server-add -s` *SERVERNAME* `-n` { *HOSTNAME:port* | *IP ADDRESS:port* } `-u` *USERNAME* `-p` { *PASSWORD* | `-l` *SSHKEYPATH* }

Syntax for adding a service cluster using SSH

`pas-server-add -s` *SERVERNAME* `-n` { *https:// HOSTNAME : port* | *IP ADDRESS : port* } `-u` *USERNAME* `-p` *PASSWORD* `-a PBS_COMMAND_LINE_ADAPTOR`

Syntax for adding an Access Web server using HTTPS

Options

-h

Display help for the command.

-L

Skip logging in while adding a server.

-r

Set the stage root directory. By default, the stage root directory is set to `/stage/$USER/AltairAccess`

The staging directory is where the necessary job files are transferred after job submission via a client for execution.

-c URL

Set the central repository URL using this parameter.
If your site has chosen to store application definitions in a central repository rather than on the local workstation, enter the URL for accessing the central repository in the format `https://<HOSTNAME>:<PORT>`. *<HOSTNAME>* is the hostname of the machine where Access Web is installed and *<PORT>* is the gateway port which is by default 4443.

-a

The Adaptor type option specifies how Access Desktop should communicate with the server. The adaptor types supported are:

- `PBS_SSH_ADAPTOR` - used to communicate to the server using SSH. This is the default option.
- `PBS_COMMAND_LINE_ADAPTOR` - used to communicate to the server using HTTPS.
- `LOCAL_DESKTOP_QUEUE_ADAPTOR` - used to communicate for local queue, mostly used in case of local job submissions (Inspire).

-P

Provide the server properties. The information provided is updated in the `serverData.xml` file. You can provide the server properties like Copy Backup, Scheduler Cycle Time, and others. The valid input format is `<key>=<value>[,<key>=<value>...]`.

-g

Use this option to prevent an SSH key from being generated. By default, the command generates a SSH key so that on subsequent logins, a username and password do not need to be provided. This option is applicable only for Access Desktop.

Command Examples

Adding a service cluster with username and password

```
pas-server-add -s cluster -n 192.168.4.123 -u dave -p evad&234
```

Adding an Access Web server with username and password

```
pas-server-add -s web_cluster -n https://172.16.82.23:17034 -u dave -p evad&234 -
a PBS_COMMAND_LINE_ADAPTOR
```

Adding a server using SSH key

```
pas-server-add -s cluster -n 192.168.4.123 -u dave -l “C:\Users\Privatekey”
```

Adding a server using a non-default port number

```
pas-server-add -s cluster -n 192.168.2.23:17034 -u dave -l “C:\Users\Privatekey”
```

Adding a server with the properties

```
pas-server-add -s cluster -n 192.168.4.123 -u dave -p evad&234 -P
COPY_BACK_SCHEDULER_CYCLE_TIME=300,COPY_BACK_THREAD_POOL_SIZE=5,
JOB_S_FILE_OVERRIDE=true
```

Adding a server with Central Repository

```
pas-server-add -s cluster -n 192.168.4.123 -u dave -p evad&234
-c https://192.168.4.124:17084
```

Command Output Example

```
pas-server-add -s cluster -n 192.168.4.123 -u dave -p evad&234
    Server: cluster added and User logged in successfully
```

```
pas-server-add -s web_cluster -n https://172.16.82.23:17034 -u dave -p evad&234 -
a PBS_COMMAND_LINE_ADAPTOR
    Server: web_cluster added and User logged in successfully
```

3.5 List Servers

Display the list of servers or clusters added.

Name

`pas-server-list`

Description

Use the `pas-server-list` command to fetch the list of servers (clusters) that have either been added via the Access UI or the `pas-server-add` command.

> **Note:** This command is valid for Access Desktop only.

Syntax

`pas-server-list` [*options*]

Options

-h
: Display help for the command.

-d
: Return the details for the default server (cluster).

-F JSON
: Print the output format in JSON.

Command Examples

List the available servers

```
pas-server-list
```

To get the default server details

```
pas-server-list -d
```

Command Output Example

```
pas-server-list
```

```
    ServerId : cluster Host : altvm10vm1
    ServerId : web_cluster Host : localhost
    ServerId : cluster_qa Host : altvm10vm14
    ServerId : cluster_rm Host : altvm10vm3
```

```
pas-server-list -d
```

```
    ServerId : cluster Host : altvm10vm1
```

3.6 Delete a Server

Delete a server (cluster) by specifying its SERVERNAME

Name

`pas-server-delete`

Description

Use the `pas-server-delete` command to delete a server (cluster) by specifying its ***SERVERNAME***. ***SERVERNAME*** is the name of the cluster as it is known within Access.

> **Note:** This command is valid for Access Desktop only.

Syntax

`pas-server-delete -s` *SERVERNAME* [*options*]

Options

-h

Display help for the command.

Command Examples

Delete a server (cluster)

```
pas-server-delete -s cluster
```

Command Output Examples

```
pas-server-delete -s cluster
    Server: cluster deleted Successfully
```

3.7 Determine if a User is an Administrator

Determine if the effective user has administrative privileges for a server.

Name

`pas-admin`

Description

Use the `pas-admin` command to determine if the effective user has administrative privileges for a server. The command will return *true* when the user is an administrator and *false* when the user is not a server administrator.

Syntax

`pas-admin` [*options*]

Options

-h

Display help for the command.

-s SERVERNAME

Determine if the effective user has administrative privileges for PAS server specified by ***SERVERNAME***. If this option is not specified, then the default server will be assumed.

> **Note:** The **-s *SERVERNAME*** option is not supported for Access Desktop.

Command Examples

Determine if the effective user has administrative privileges for the default server

```
pas-admin
```

Determine if the effective user has administrative privileges for a non-default server

```
pas-admin -s 192.168.4.123
```

Command Output Examples

```
pas-admin
    true
```

3.8 Getting the Staging Area Root Directory

Return the root directory of the staging area set.

Name

`pas-getsroot`

Description

Use the `pas-getsroot` command to return the root directory of the staging area set during the installation.

Syntax

`pas-getsroot` [*options*]

Options

-h

Display help for the command.

-s SERVERNAME

Returns the staging area root directory of the PAS server specified by ***SERVERNAME***. If this option is not specified, then the default server will be assumed.

Command Examples

Get the staging area root directory of the default server

```
pas-getsroot
```

Get the staging area root directory of a non-default server

```
pas-getsroot -s 192.168.4.123
```

Command Output Examples

```
pas-getsroot
    /stage
```

3.9 Set the User's Password

Set your CLI password.

Name

`pas-passwd`

Description

Use the `pas-passwd` command to set your CLI password. Once this command is issued, you will be asked to enter a password, and then re-enter it for verification. Once the password is set, it will be used for all subsequent CLI requests.

Syntax in Access Desktop

`pas-passwd` [*option*]

Syntax in Access Web

`pas-passwd` `-s` *SERVERNAME* `-u` *USERNAME* [*option*]

Options

-h
Display help for the command.

-s SERVERNAME
Set the CLI password for the PAS server specified by ***SERVERNAME***. If this option is not specified, then the default server will be assumed.

-u USERNAME
Set the password for the specified ***USERNAME***. By default, it is the current user.

-p PASSWORD
Specify the user's current password.

-l PATHNAME
Location of the SSH public key file for the user. This can be used if you do not want to specify the password. The SSH key file path should be provided in quotes.

> **Note:** The **`-l PATHNAME`** option is not supported for Access Web.

-F JSON
Print the output format in JSON.

-g
Use this option to disable generating SSH Key. By default, the command will generate a SSH Key which enables the user not to provide username and password for subsequent logins. This option is applicable only for Access Desktop.

Command Examples

Set the user's password

```
pas-passwd
```

Set the user's password by providing the username and password

```
pas-passwd -s 192.168.4.123 -u dave -p evad&123
```

Set the user's password by providing the SSH key location

```
pas-passwd -s 192.168.4.123 -u dave -l "C:\Users\Privatekey"
```

Command Output Examples

```
pas-passwd
```

```
    Enter user's password:xxxxxxxx
    Retype password:xxxxxxxx
    Changed user root's password on server <localhost>
```

Job Submission and Monitoring Commands

4

Commands for job submission, job monitoring, and obtaining job results.

This chapter covers the following:

All commands will connect to the first server (cluster) registered after installation (this server is considered the default server). Most commands provide an option to specify a non-default PAS server.

A simple workflow for submitting a job is:

1. Get a list of the available applications on the server using the `pas-getapps` command.
2. Once you have determined which application to run, get a list of application parameters using the `pas-getapp` command.
3. Upload the job input files to the server using the `pas-fup` command.
4. Submit a job to the server using the `pas-submit` command.
5. Check the status of your job using the `pas-stat` command.
6. Download the job result files using the `pas-getres` command.

4.1 List of all Applications

Display a list of all applications.

Name

`pas-getapps`

Description

Use the `pas-getapps` command to return a list of all applications known to the PAS server.

Syntax

`pas-getapps` [*option*]

Options

-h

Display help for the command.

-s SERVERNAME

Retrieve a list of all applications known to the PAS server specified by ***SERVERNAME***. If this option is not specified, then the default server will be assumed.

-F JSON

Print the output format in JSON. This will also provide the timestamp of when the application definition was last updated in the cluster. The timestamp that is displayed is in Hash format.

Command Examples

Get a list of all application known to the default server

```
pas-getapps
```

Get a list of all applications known to a non-default server

```
pas-getapps -s 192.168.4.123
```

Get a list of all applications from Access Web server

```
pas-getapps -s web_cluster
```

Print the output in JSON format

```
pas-getapps -F JSON
```

Command Output Examples

```
pas-getapps
    ShellScript2
    RADIOSS
```

```
    Sleeper
    JobArray
    ShellScript
    OptiStruct
    Signal Trap
    ShellScript
```

```
pas-getapps -s 192.168.4.123
```

```
    RADIOSS
    ShellScript
    OptiStruct
```

```
pas-getapps -F JSON
```


```
{"time_stamp":
 "96f44fd54d5f8178d5b4c0d06ebd0a3d1377db97c04cb4b12007da64a4bd74a3",
"data": [{"definitions": {"file": {"required": ["value"], "type": "object",
"properties": {"dname": {"type": "string"}, "required": {"type": "boolean"},
"upload": {"type": "boolean"}, "value": {"type": "string"}}}},
"$schema": "http://json-schema.org/draft-04/schema#",
"required": ["NCPUS", "MEMORY", "PRIMARY_FILE"], "type": "object",
"properties": {"FILES": {"items": {"$ref": "#/definitions/file"},
"DisplayName": "Job Files", "type": "array", "Displayable": true,
"Description": "Files to be included with the Job Script."},
"MEMORY_PLACEMENT": {"DisplayName": "Memory Per Chunk",
"Description": "The physical memory requested per chunk.",
"enum": ["true", "false"], "Displayable": true, "value": "true",
"type": "string"}, "NCPUS": {"DisplayName": "Number of Processors",
"Description": "The total number of processors.",
"Displayable": true, "value": 1, "minimum": 1, "type": "integer"},
"JOB_ARGS": {"DisplayName": "Job Script Arguments", "type": "string",
"Displayable": true, "Description": "Pass specific arguments to the
Job Script."}, "QUEUE": {"DisplayName": "Queue", "type": "string",
"Displayable": true, "Description": "Queue to which job must be submitted"},
"SUBMISSION_DIRECTORY": {"Description": "Result files will be placed
here by PBS Desktop.", "isDirectory": true, "DisplayName": "Output Directory",
"Displayable": true, "$ref": "#/definitions/file"},
"PRIMARY_FILE": {"Description": "The Job Script to run.",
"DisplayName": "Job Script", "Displayable": true,
"$ref": "#/definitions/file"}, "CHUNK_PLACEMENT": {"DisplayName": "Placement
of Chunks", "Description": "The placement of job chunks.",
"enum": ["pack", "free", "scatter"], "Displayable": true, "value": "free",
"type": "string"}, "MEMORY": {"DisplayName": "Amount of Memory (MB)",
"Description": "The total physical memory requested.",
"Displayable": true, "value": 10, "minimum": 10, "type": "integer"},
"CHUNKS": {"DisplayName": "Number of Chunks", "Description": "The total
number of job chunks.", "Displayable": true, "value": 1, "minimum": 1,
"type": "integer"}, "ApplicationId": {"type": "string", "Displayable": false,
"value": "ShellScript"}, "JOB_NAME": {"DisplayName": "Job Name",
"type": "string", "Displayable": true, "Description": "The name of the job."},
"ApplicationFileExtension": {"items": {"type": "string"}, "type": "array",
"Displayable": false, "value": [".sh", ".py", ".csh"]}}}],
"success": true, "exitCode": "0"}
```

4.2 List of Application Parameters

Display the list of application parameters.

Name

`pas-getapp`

Description

Use the `pas-getapp` command to return a list of all parameters available for the specified ***APPNAME***. This command will also provide information on whether the application parameter is mandatory or optional. The output of this command can be redirected to a file.

Syntax

`pas-getapp -a` *APPNAME* [*option*]

Options

-h
Display help for the command.

-s SERVERNAME
Retrieve application parameters for ***APPNAME*** for the server specified by ***SERVERNAME***. If the option is not specified, then the default server will be assumed.

-F JSON
Print the output format in JSON.

Command Examples

Get the parameters available for the application "optistruct" on the default server

```
pas-getapp -a optistruct
```

Get the parameters available for the application "optistruct" on a non-default server

```
pas-getapp -a optistruct -s 192.168.4.123
```

Get the parameters available for the application "optistruct" on Access Web server

```
pas-getapp -a optistruct -s web_cluster
```

Get the parameters available for the application "optistruct" and redirect the output to a file called /tmp/optistruct_params

```
pas-getapp -a optistruct > /tmp/optistruct_params
```

Command Output Examples

```
pas-getapp -a ShellScript
```

```
    ApplicationId=ShellScript
    ApplicationName=ShellScript
    FILES(false)(FileNameMulti)
    MEMORY_PLACEMENT(false)(StringEnumerated)(Value=true)
        [Options:true,false]
    NCPUS(true)(Int)(Min:1)(Value=1)
    JOB_ARGS(false)(String)
    QUEUE(false)(String)
    SUBMISSION_DIRECTORY(false)(DirectoryName)
    PRIMARY_FILE(true)(File)
    CHUNK_PLACEMENT(false)(StringEnumerated)(Value=free)
        [Options:pack,free,scatter]
    MEMORY(true)(Int)(Min:10)(Value=10)
    CHUNKS(false)(Int)(Min:1)(Value=1)
    JOB_NAME(false)(String)
```

4.3 Submit a Job

Submit a job to PBS Professional.

Name

`pas-submit`

Description

Use the `pas-submit` command to submit a job. Upon successful submission of the job, this command will return the PBS Professional job identifier. The `pas-submit` command accepts input file paths relative to the PAS server location, therefore you must upload job input files to the server before submitting a job. Use the `pas-fup` command to upload any job input files to the server in Access Web.

There are two ways to submit a job using this command:

- The application parameters can be entered manually on the command line.
- The application parameters can be stored in a file. Use the `pas-getapp` command to generate a list of the application parameters, and then edit this list to fill in the appropriate parameter values.

Syntax

`pas-submit` [*option*]

Options

-h
Display help for the command.

-f FILEPATH
Specify the path of the file that contains the application parameters. Use the `pas-getapp` command to generate this file.

-s SERVERNAME
Submit a job to the server specified by ***SERVERNAME***. If the server is not specified, then the default server will be assumed.

-w
Use this option to wait until the job is completed and result files are downloaded.

> **Note:** The **-w** option is valid for Access Web only.

-F JSON
Print the output format in JSON.

Command Examples

These examples assume that the job input files have already been uploaded to the server.

Submit a job to the server `blrec166` by entering all application parameters at the command line

```
pas-submit ApplicationId=ShellScript ApplicationName=ShellScript
```

```
JOB_NAME=hello_world PRIMARY_FILE=/home/tsmith/hello_world.sh MEMORY=100 NCPUS=1
```

Submit a job by specifying the path to the file containing the application parameters

```
pas-submit -f /tmp/parm_file
```

Submit a job to a non-default server

```
pas-submit -f /tmp/parm_file -s 192.168.4.123
```

Submit a job to the Access Web server

```
pas-submit -f /tmp/parm_file -s web_cluster
```

Submit a job to the server hwesuse113-01 passing an application argument of datetime

```
pas-submit ApplicationId=MyApp ApplicationName=MyApp JOB_NAME=MyApp_test
EXECUTION_TIME='2014-07-30;13:40:00' SHELL_SCRIPT=/home/user1/hello_world.sh
SUBMISSION_DIRECTORY=/home/user1
```

Submit a job to the Server hwesuse113-01 passing an application argument of boolean

```
pas-submit ApplicationId=MyApp ApplicationName=MyApp JOB_NAME=MyApp_test
LOGGING=true SHELL_SCRIPT=/home/user1/hello_world.sh
SUBMISSION_DIRECTORY=/home/user1
```

Command Output Examples

```
pas-submit ApplicationId=ShellScript ApplicationName=ShellScript
JOB_NAME=ShellScript_test MEMORY=10 NCPUS=1 JOB_SCRIPT=/tmp/hello_world.sh
SUBMISSION_DIRECTORY=/tmp/PAS_output
```

```
    Response from server
    149.hwesuse113-01
```

4.4 Check the Status of a Job

Display the status of a job.

Name

`pas-stat`

Description

Use the `pas-stat` command to display the status of a job (similar to the PBS Professional command `qstat -f`). You can display the status of a particular job or the status of all jobs.

Syntax

`pas-stat` [*option*]

Options

-h
: Display help for the command.

-j JOBID
: Get the status of ***JOBID***. If this option is not specified, then the status of all jobs will be returned.

-s SERVERNAME
: Get the status of a job on the server specified by ***SERVERNAME***. If this option is not specified, then the default server will be assumed.

-F JSON
: Print the output format in JSON.

Command Examples

Check the status of a specific job on the default server

```
pas-stat -j 248.blrec166
```

Check the status of all jobs on the default server

```
pas-stat
```

Check the status of all jobs on a non-default server

```
pas-stat –s 192.168.4.90
```

Check the status of a specific job on a non-default server

```
pas-stat -j 248.blrec166 –s 192.168.4.90
```

Command Output Examples

```
pas-stat -j 129.hwesuse113-01
```

```
    Job Id: 129.hwesuse113-01
    pbs_job_state : Q
    pbs_server : hwesuse113-01
    pbs_Account_Name : n/a
    pbs_Job_Name : test1
    pbs_Job_Owner : ebleicher@hwesuse113-01.prog.altair.com
    pbs_Checkpoint : u
    pbs_ctime : Wed Mar 12 13:51:37 2014
    pbs_Error_Path : hwesuse113-01.prog.altair.com:/var/spool/pas/temp/test1.er
    pbs_exec_host : n/a
    pbs_exec_vnode : n/a
    pbs_Hold_Types : n
    pbs_Join_Path : n
    pbs_Keep_Files : oe
    pbs_Mail_Points : a
    pbs_mtime : Wed Mar 12 13:51:38 2014
    pbs_Output_Path : hwesuse113-01.prog.altair.com:/var/spool/pas/temp/test1.ou
    pbs_Priority : 0
    pbs_qtime : Wed Mar 12 13:51:38 2014
    pbs_Rerunable : True
    pbs_stime : n/a
    pbs_session_id : 0
    pbs_sandbox : private
    pbs_jobdir : n/a
    pbs_stagein : runtime@hwesuse113-01:/var/spool/pas/repository/applications/
    ShellScript/runtime,pbs_spawn@hwesuse113-01:/opt/altair/pas/2020.4/pas/
    pbs_spawn,hello_world.sh@hwesuse113-01:/users/ebleicher/hello_world.sh
    pbs_stageout : *@hwesuse113-01:/stage/ebleicher/test1
    pbs_Variable_List :
    PAS_APPLICATION=ShellScript,PAS_CHUNKS=1,PAS_CLIENT_HOST=127.0.0.2,
    PYTHONPATH=/opt/pbs/default/bin/pbs_python,
    PAS_SUBMISSION_DIRECTORY=pbscp://hwesuse113-01/stage/ebleicher/test1,
    JOB_ARGS=,AIF_MODE=enterprise,
    PAS_JOB_SCRIPT=pbscp://hwesuse113-01/users/ebleicher/hello_world.sh,
    PAS_SERVER_PORT=17084,PAS_MEMORY=1000,PAS_CHUNK_PLACEMENT=free,
    JOB_SCRIPT=pbscp://hwesuse113-01/users/ebleicher/hello_world.sh,
    PAS_MEMORY_PLACEMENT=true,PBS_HOME=/var/spool/PBS,
    PAS_HPCBP_ARGUMENTS=runtime/start.py,PAS_JOB_NAME=test1,
    PAS_HPCBP_EXECUTABLE=/opt/pbs/default/bin/pbs_python,
    PAS_PYTHON_PATH=/opt/pbs/default/bin/pbs_python,PAS_JOB_NAME=test1,
    PAS_NCPUS=10,PAS_SERVER_VERSION=2020.4,AIF_USER=ebleicher,
    PBS_O_QUEUE=workq,PBS_O_HOST=hwesuse113-01.prog.altair.com
    pbs_comment : Can Never Run: Insufficient amount of resource
    ncpus (R: 10 A: 1 T: 1)
    pbs_etime : Wed Mar 12 13:51:38 2014
    pbs_SubStatus : 11
    pbs_array : n/a
    pbs_array_state_count : n/a
    pbs_array_indices_submitted : n/a
    pbs_array_indices_remaining : n/a
```

4.5 Summary of Job Information

Display a summary of the job information.

Name

`pas-sum`

Description

Use the `pas-sum` command to display a summary of the job information (similar to the PBS Professional command `qstat`). You can displays a summary of job information for a specific job, or all jobs on the server.

Syntax

`pas-sum` [*option*]

Options

-h
: Display help for the command.

-j JOBID
: Display a summary of job information for ***JOBID***. If this option is not specified, then a summary for all jobs will be returned.

-s SERVERNAME
: Display a summary of job information for a job on the server specified by ***SERVERNAME***. If the server is not specified, then the default server will be assumed.

-F JSON
: Print the output format in JSON.

Command Examples

Get a summary of the job information for a given job

```
pas-sum -j 243.blrec166
```

Get a summary of job information for all jobs on the default server

```
pas-sum
```

Get a summary of job information for all jobs on a non-default server

```
pas-sum -s 192.168.4.123
```

Command Output Examples

```
pas-sum -j 129.hwesusue113-01
    pbs_jobid : 129.hwesuse113-01
```

```
    pbs_Job_Name : test1
    pbs_Job_Owner : ebleicher@hwesuse113-01.prog.altair.com
    pbs_job_state : Q
    pbs_queue : workq
```

4.6 Download Job Results

Download the files related to the given ***JOBID*** to the ***TARGET_LOCATION***.

Name

`pas-getres`

Description

Use the `pas-getres` command to download the files related to the given ***JOBID*** to the ***TARGET_LOCATION***. The job must be in a running or finished state to execute this command. If the job was submitted through Altair Access the job submission directory will be zipped and downloaded. Otherwise the standard output file, standard error file, and any other files specified in the stage-out list will be zipped.

> **Note:** This command is valid for Access Web only.

Syntax

`pas-getres` `-j` *JOBID* `-o` *TARGET_LOCATION* [*option*]

> **Note:** ***TARGET_LOCATION*** must have an ending slash or backslash:
>
> `c:\temp\`
>
> `/home/user1/temp/`

Options

-h

Display help for the command.

-s SERVERNAME

Download job results for a job on the server specified by ***SERVERNAME***. If the server is not specified, then the default server will be assumed.

Command Examples

> **Note:** Executing these commands will download the job results files and create a zip file called `C:\download\results\176-blrec166.zip` on the client host.

Download the job results of a given job to the specified target location

```
pas-getres -j 176.blrec166 -o C:\download\results\
```

Download the job results of a given job on a non-default server

```
pas-getres -j 176.blrec166 -o C:\download\results\ -s 192.168.4.123
```

Command Output Examples

```
pas-getres –j 176.blrec166 -o C:\download\results\
    Job Id 176.blrec166 downloaded successfully
```

4.7 Delete a Job

Delete a given ***JOBID*** from the Workload Management System.

Name

`pas-del`

Description

Use the `pas-del` command to delete a given ***JOBID*** from the Workload Management System. The job can be in any of the following states:

Table 4: Valid job states for deleting a job

Job State	Description
B	Job arrays only: job array has begun.
H	Job is held. A job is put into a held state by the server or by a user or administrator. A job stays in a held state until it is released by a user or administrator.
Q	Job is queued, eligible to run or be routed.
R	Job is running.
W	Job is waiting for its requested execution time to be reached or job specified a staging request which failed for some reason.

Syntax

`pas-del -j` *JOBID* [*option*]

Options

-h
: Display help for the command.

-s SERVERNAME
: Delete a job from the server specified by ***SERVERNAME***. If the server is not specified, then the default server will be assumed.

Command Examples

Delete a job from the default server

```
pas-del -j 243.blrec166
```

Delete a job from a non-default server

```
pas-del -j 243.blrec166 -s 192.168.4.123
```

Command Output Examples

```
pas-del -j 151.hwesuse113-01
    Job(151.hwesuse113-01) deleted successfully
```

4.8 List of Custom Actions

Get the list of available custom action.

Name

`pas-custom-action-list`

Description

Use the `pas-custom-action-list` command to get the list of available custom actions for the ***JOBID***.

> **Note:** This command is valid for Access Desktop only.

Syntax

`pas-custom-action-list -j` *JOBID* [*option*]

Options

-h
Display help for the command.

-s SERVERNAME
Get a list of custom actions associated with a job from the PAS server specified by ***SERVERNAME***. If the server is not specified, then the default server will be assumed.

-F JSON
Print the output format in JSON.

Command Examples

Get a list of custom actions associated with a job from the default server

```
pas-custom-action-list -j 243.blrec166
```

Get a list of custom actions associated with a job from a non-default server

```
pas-custom-action-list -j 243.blrec166 -s 192.168.4.123
```

Command Output Examples

```
pas-custom-action-list -j 151.hwesuse113-01
    ApplicationId : CustomActionApp
    CustomActions:
        CustomAction:
            Executable :
            Name : SendSignal.py
            Language : PYTHON
        DisplayName : Send Signal
        Name : SEND_SIGNAL
        Arguments :
```

```
            SIGNAL :
                DisplayName : SIGNAL
                type : string
            required : [SIGNAL]
        ShowOutput : True
        JobStates : [RUNNING, RUNNING, WAITING, FINISHED, QUEUED, SUCCEEDED]
        Description : Send Signal
```

4.9 Execute a Custom Action for a Job

Execute a custom action for a job.

Name

`pas-execute-custom-action`

Description

Use the `pas-execute-custom-action` command to execute a ***CUSTOMACTION*** for a ***JOBID***.

> **Note:** This command is valid for Access Desktop only.

Syntax

`pas-execute-custom-action -j` *JOBID* `-N` *CUSTOMACTION* [*argument_name=value, argument_name=value ....*] [*option*]

> **Note:** The ***argument_name=value*** are the arguments that are being passed to the custom action, for example, ***SEND_SIGNAL=STOP***. Not every custom action will have arguments, so they are optional.

Options

-h
: Display help for the command.

-s SERVERNAME
: Execute the custom action for a job that has been submitted to the server specified by ***SERVERNAME***. If the server is not specified, then the default server will be assumed.

-F JSON
: Print the output format in JSON.

Command Examples

Execute custom action list for a job from the default server

```
pas-execute-custom-action -j 243.blrec166
-N SEND_SIGNAL SIGNAL=STOP
```

Execute custom action for a job from a non-default server

```
pas-execute-custom-action -j 243.blrec166 -s 192.168.4.123
-N SEND_SIGNAL SIGNAL=STOP
```

Command Output Examples

```
pas-execute-custom-action -j 4461 -N SEND_SIGNAL SIGNAL=STOP -s 192.168.4.123
```

```
    isStdErrTruncated : False
    isStdOutTruncated : False
    stdOut:RXhlY3V0aW5nIHNjcmlwdDogL3Zhci9zcG9vbC9wYnN3b3Jrcy9wYXMvMjAxOC4y
    L2hvbWUvZGF0YS9wYXMvdGFyZ2V0cy9sb2NhbGhvc3QvcmVwb3NpdG9yeS9wcml2YXRlL2dlbm
    VyYXRlZC9qc29uLWFwcGxpY2F0aW9ucy9DdXN0b21BY3Rpb25BcHAvc3VibWl0dGltZS9TZW5k
    U2lnbmFsLnB5IHdpdGggc2lnbmFsIDog U1RPUAo=
    exitCode : 0
```

File Management Commands

5

Commands for Altair Access file operations.

This chapter covers the following:

All commands will connect to the first server (cluster) registered after installation (this server is considered the default server). Most commands provide an option to specify a nondefault PAS server.

5.1 Create a Directory

Create a directory on a server.

Name

`pas-fmkdir`

Description

Use the `pas-fmkdir` command to create a ***DIRECTORY*** on a server.

Syntax

`pas-fmkdir -d` *DIRECTORY* [*option*]

Options

-h
Display help for the command.

-j JOBID
Make the ***DIRECTORY*** in the ***JOBID***'s execution directory. The ***JOBID*** must be in a running state.

-s SERVERNAME
Create a directory on the server specified by ***SERVERNAME***. If this option is not specified, then the default server will be assumed.

-F JSON
Print the output format in JSON.

Command Examples

Create a directory on the default server

```
pas-fmkdir -d /users/tsmith/ShellScript_output
```

Create a directory on a non-default server

```
pas-fmkdir -d /users/tsmith/ShellScript_output -s 192.168.4.123
```

Create a directory in the running job's execution directory

```
pas-fmkdir -d ShellScript_output -j 347.hwesuse113-01
```

Command Output Examples

```
pas-fmkdir -d C:\testDir
    Directory <C:\testDir> Created successfully
```

5.2 Upload a File

Upload a file to a directory.

Name

`pas-fup`

Description

Use the `pas-fup` command to upload a ***FILE*** to a ***DEST*** on a server.

Syntax

`pas-fup -f` *FILE* `-o` *DEST* [*option*]

Options

-h
: Display help for the command.

-j JOBID
: Upload a ***FILE*** to the ***JOBID***'s execution directory.

-s SERVERNAME
: Upload a ***FILE*** to the server specified by ***SERVERNAME***. If this option is not specified, then the default server will be assumed.

Command Examples

Upload a file to the default server

```
pas-fup -f C:\foobar.txt -o /users/tsmith/foobar.txt
```

Upload a file to a non-default server

```
pas-fup -f C:\foobar.txt -o /users/tsmith/foobar.txt -s 192.168.4.123
```

Upload a file to the job's execution directory

```
pas-fup -j 87.blrec3vm12 -f /stage/vis.zip -o vis.zip
```

Command Output Examples

```
pas-fup -f C:\foobar.txt -o /users/tsmith/foobar.txt
    Uploaded successfully
```

5.3 Download a File

Download a file from a server.

Name

`pas-fdown`

Description

Use the `pas-fdown` command to download a ***FILE*** from a server.

Syntax

`pas-fdown -f` *FILE* [*option*]

Options

-h
Display help for the command.

-j JOBID
Download a ***FILE*** from the ***JOBID***'s execution directory. The job must be in a running state.

-o DEST
Download a ***FILE*** from the server to the target location ***DEST***. If this option is not specified, the ***FILE*** will be downloaded to the current working directory.

-s SERVERNAME
Download a ***FILE*** from the server specified by ***SERVERNAME***. If this option is not specified, then the default server will be assumed.

Command Examples

Download a file from the default server

```
pas-fdown -f /users/tsmith/results.out -o c:\temp\results.out
```

Download a file from a non-default server

```
pas-fdown -f /users/tsmith/results.out -o c:\temp\results.out
-s 192.168.4.123
```

Download a file from a running job's execution directory

```
pas-fdown -f ShellScript_test.o325 -o c:\temp\ShellScript_test.o325
-j 325.blrec166
```

Command Output Examples

```
pas-fdown -f /users/tsmith/results.out -o c:\temp\results.out
```

```
    Downloaded successfully
```

5.4 Delete a File

Delete a file from a server.

Name

`pas-fdel`

Description

Use the `pas-fdel` command to delete a ***FILE*** located on the server.

Syntax

`pas-fdel -f` *FILE* [*option*]

Options

-h
Display help for the command.

-j JOBID
Delete a ***FILE*** located in the ***JOBID***'s execution directory. The job must be in a running state.

-s SERVERNAME
Delete a ***FILE*** located on the server specified by ***SERVERNAME***. If this option is not specified, then the default server will be assumed.

Command Examples

Delete a file located on the default server

```
pas-fdel -f /users/tsmith/bartest.fem
```

Delete a file located on the non-default server

```
pas-fdel -f /users/tsmith/bartest.fem -s 192.168.4.123
```

Delete a file located in a running job's execution directory

```
pas-fdel -f hello_world.sh -j 329.hwesuse113-01
```

Command Output Examples

```
pas-fdel -f /tmp/PAS_output/ShellScript_test/ShellScript_test.e149
    File(/tmp/PAS_output/ShellScript_test/ShellScript_test.e149)
    deleted successfully
```

5.5 Compress a File

Compress a file located on a server and save the compressed file to a specified destination.

Name

`pas-fcompr`

Description

Use the `pas-fcompr` command to compress a ***FILE*** located on a server to ***DEST***.

Syntax

`pas-fcompr -f` *FILE* `-o` *DEST* [*option*]

Options

-h
Display help for the command.

-j JOBID
Compress a ***FILE*** located in the ***JOBID***'s execution directory. The job must be in a running state.

-s SERVERNAME
Compress a ***FILE*** located on the server specified by ***SERVERNAME***. If this option is not specified, then the default server will be assumed.

Command Examples

Compress a file located on the default server and create a zip file on the default server

```
pas-fcompr -f /users/tsmith/bar.fem -o /users/tsmith/bar.zip
```

Compress a file located on a non-default server and create a zip file on that server

```
pas-fcompr -f /users/tsmith/bar.fem -o /users/tsmith/bar.zip
-s 192.168.4.123
```

Compress a file located in a running job's execution directory and created a zip file on the default server

```
pas-fcompr -f ShellScriptTest.o332 -o /users/tsmith/ShellScriptTest.zip
-j 332.blrec166
```

Command Output Examples

```
pas-fcompr -f /tmp/PAS_output/ShellScript_test/ShellScript_test.o149
-o /tmp/j149.zip
```

```
    File(/tmp/PAS_output/ShellScript_test/ShellScript_test.o149)
    compressed successfully
```

5.6 List of Files

Display a list of files from the server.

Name

`pas-flist`

Description

Use the `pas-flist` command to display a list of ***FILE***s located on the server.

Syntax

`pas-flist -f` *FILE* [*option*]

Options

-h
Display help for the command.

-j JOBID
List ***FILE***s in the job's execution directory. The job must be in a running state.

-s SERVERNAME
List ***FILE***s located on the server specified by ***SERVERNAME***. If the option is not specified, then the default server will be assumed.

-F JSON
Print the output format in JSON.

Command Examples

List files in a directory located on the default server

```
pas-flist -f /stage/tsmith/test1
```

List files in a directory located on a non-default server

```
pas-flist -f /stage/tsmith/test1 -s 192.168.4.123
```

List a specific file located on the default server

```
pas-flist -f /stage/tsmith/test1/test1.o2
```

List a specific file in the job's execution directory

```
pas-flist -f dirlister.e361 -j 248.blrec166
```

List all files in the job's execution directory

```
pas-flist -f . -j 248.blrec166
```

Command Output Examples

```
pas-flist -f . -j 335.hwesuse113-01
    hello_world.sh
    pbsjob.env
    pbs_spawn
    ShellScript_test.e335
    runtime
    ShellScript_test.o335
```