

PBS Access 2018.2

# Command Line Interface Guide



 Altair | PBS Works™

PBS Works is a brand of  Altair

## Intellectual Property Rights Notice: Copyrights, Trademarks, and Third Party Licenses

Updated: July 18, 2018.

### Altair® PBS Works® v.2018.2

Accelerating Innovation in the Cloud™

Copyright© 1994-2018 Altair Engineering Inc. All Rights Reserved.

Special Notice: Pre-release versions of Altair software are provided 'as is', without warranty of any kind. Usage of pre-release versions.

### Altair PBS Works™:

PBS Professional® ©1994-2018; PBS Control ©2008-2018; PBS Access ©2008- 2018;

Compute Manager™ ©2012-2017; Display Manager™ ©2013-2017; PBS Pro ©1994-2017;

PBS Application Services ©2008-2017; PBS Analytics ©2008-2017; PBS Access WebPBS Desktop ©2008-2012; and

e-Compute™ ©2000-2010.

**Runtime 2017** ©1995 – 2018.

### HyperWorks® Products:

HyperMesh® ©1990-2018; HyperCrash® ©2001-2018; OptiStruct® ©1996-2018; RADIOSS® ©1986-2018; HyperView® ©1999-2018; HyperView Player® ©2001-2018; HyperMath® ©2007-2018; HyperStudy® ©1999-2018; HyperGraph® ©1995-2018; MotionView® ©1993-2018; MotionSolve® ©2002-2018; HyperForm® ©1998-2018; HyperXtrude® ©1999- 2018; Process Manager™ ©2003-2018; Templex™ ©1990-2018; TextView™ ©1996-2018; MediaView™ ©1999-2018; TableView™ ©2013-2018; BatchMesher™ ©2003-2018; HyperWeld® ©2009-2018; HyperMold® ©2009-2018; Manufacturing Solutions™ ©2005-2018; solidThinking Inspire® 2018 ©2009-2018; solidThinking Evolve®2017 ©1993-2017; Durability Director™ ©2009-2018; Suspension Director™ ©2009-2018; AcuSolve® ©1997-2018; AcuConsole® ©2006-2018; SimLab® ©2004-2018; Virtual Wind Tunnel™ ©2012-2018; FEKO® (©1999-2014 Altair Development S.A. (Pty) Ltd.; ©2014-2018 Altair Engineering, Inc.); ConnectMe™ ©2014-2018; Click2Extrude™ Polymer 2017 ©1996-2018; Click2Extrude™ Metal 2018 ©1996-2018; Click2Form™ 2018 ©1998-2018; Flux® 2018 ©1983-2018, FluxMotor® 2018 ©2017-2018; WinProp v.2018 ©2000-2018.

### Additional Altair Products:

Multiscale Designer™ ©2011-2017;

ESAComp™ v.4.7 ©1992-2018;

**Altair Packaged Solution Offerings (PSOs):**

Automated Reporting Director™ ©2008-2018; GeoMechanics Director ©2011-2018; Impact Simulation Director™ ©2010-2018; Model Mesher Director™ ©2010-2018; NVH Director™ ©2010-2017; Squeak and Rattle Director™ ©2012-2018; Virtual Gauge Director™ ©2012-2018; Weight Analytics™ ©2013-2017; Weld Certification Director™ ©2014-2018; Multi-Disciplinary Optimization Director™ ©2012-2018.

**Altair Simulation Cloud Suite:**

Simulation Manager™ ©2003-2017; Compute Manager™ ©2003-2017; Display Manager™ ©2003-2017; and Process Manager™ ©2003-2016.

**Software products of solidThinking, Inc., a wholly owned subsidiary of Altair Engineering:**

solidThinking Inspire® 2018 ©2009-2018; solidThinking Evolve®2017 ©1993-2018; solidThinking Compose® 2018 ©2007-2018, solidThinking Activate® 2018 ©1989-2018, solidThinking Embed® 2018 ©1989-2018, solidThinking Embed® SE 2018 ©1989-2018; Click2Extrude™ Metal 2018 ©1996-2018; Click2Extrude™ Polymer 2018 ©1996-2018; Click2Cast® 4.1 ©2011-2018; Click2Form™ 2018 ©1998-2018.

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](http://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

Location	Telephone	e-mail
Australia	+1 800 174 396	anz-pbssupport@india.altair.com
China	+86 21 6117 1666	es@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 +1 800 425 0234 (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 +1 800 425 0234 (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 +1 800 425 0234 (Toll Free)	pbs-support@india.altair.com
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

This document is proprietary information of Altair Engineering, Inc.

# Contents

<b>About this Guide</b>	<b>ix</b>
Audience .....	ix
Related Documentation .....	ix
Document Conventions .....	x
Acronyms .....	xii
<b>Chapter 1: Introduction</b>	<b>1</b>
What's New .....	1
PBS Access Command Line Interface Path .....	2
Prerequisites .....	2
<b>Chapter 2: Administering PBS Access</b>	<b>3</b>
Determining if Server is Up and Running .....	4
Usage .....	4
Options .....	4
Command Examples .....	4
Command Output Example .....	5
Getting Server Information .....	6
Usage .....	6
Options .....	6
Command Examples .....	6
Command Output Example .....	6
Displaying Server Version & Build Information .....	7
Usage .....	7
Options .....	7
Command Examples .....	7
Command Output Example .....	8
Adding a Server .....	9
Usage .....	9
Options .....	9
Command Examples .....	10
Command Output Example .....	10
Listing Servers .....	11
Usage .....	11
Options .....	11

Command Examples .....	11
Command Output Example .....	12
Deleting a Server .....	13
Usage .....	13
Options .....	13
Command Examples .....	13
Command Output Example .....	13
Determining if the User is an Administrator .....	14
Usage .....	14
Options .....	14
Command Examples .....	14
Command Output Example .....	14
Getting the Staging Area Root Directory .....	15
Usage .....	15
Options .....	15
Command Examples .....	15
Command Output Example .....	15
Setting the User's Password .....	16
Usage in PBS Access Web .....	16
Usage in PBS Access Desktop .....	16
Options .....	16
Command Examples .....	17
Command Output Example .....	17

---

## **Chapter 3: Submitting and Monitoring Jobs** **19**

Getting a List of all Applications .....	20
Usage .....	20
Options .....	20
Command Examples .....	20
Command Output Example .....	21
Getting a List of Application Parameters .....	22
Usage .....	22
Options .....	22
Command Examples .....	22
Command Output Example .....	23
Submitting a Job .....	24
Usage .....	24
Options .....	24

Command Examples .....	25
Command Output Example .....	25
Checking the Status of Jobs .....	26
Usage .....	26
Options .....	26
Command Examples .....	26
Command Output Example .....	27
Getting a Summary of Job Information .....	29
Usage .....	29
Options .....	29
Command Examples .....	29
Command Output Example .....	30
Downloading Job Results .....	31
Usage .....	31
Options .....	31
Command Examples .....	32
Command Output Example .....	32
Deleting a Job .....	33
Usage .....	33
Options .....	33
Command Examples .....	33
Command Output Example .....	33
Getting a List of Custom Actions Defined for a Job .....	34
Usage .....	34
Options .....	34
Command Examples .....	34
Command Output Example .....	35
Executing a Custom Action for a Job .....	36
Usage .....	36
Options .....	36
Command Examples .....	36
Command Output Example .....	37

---

## **Chapter 4: Maintaining Files and Directories** **39**

Creating a Directory .....	40
Usage .....	40
Options .....	40
Command Examples .....	40

Command Output Example .....	40
Uploading a File .....	41
Usage .....	41
Options .....	41
Command Examples .....	41
Command Output Example .....	41
Downloading a File .....	42
Usage .....	42
Options .....	42
Command Examples .....	42
Command Output Example .....	43
Deleting a File .....	44
Usage .....	44
Options .....	44
Command Examples .....	44
Command Output Example .....	44
Compressing a File .....	45
Usage .....	45
Options .....	45
Command Examples .....	45
Command Output Example .....	45
Displaying a List of Files .....	46
Usage .....	46
Options .....	46
Command Examples .....	46
Command Output Example .....	47
<b>Chapter 5: PBS Access Commands Quick Reference</b> .....	<b>49</b>
<b>Appendix: Index</b> .....	<b>51</b>



# About this Guide

---

This document provides information about using the PBS Access Command Line Interface (CLI) for job submission and monitoring, file operations, and managing servers.

This chapter contains the following topics:

- [Audience](#)
- [Related Documentation](#)
- [Document Conventions](#)
- [Acronyms](#)

## Audience

This document is intended for users who need to:

- Submit a job to a PBS Professional complex
- Monitor their jobs
- Upload and download job files

## Related Documentation

The other PBS Access related documents are as follows:

- [PBS Access Web Administrator's Guide](#)
- [PBS Access Desktop Administrator's Guide](#)
- [PBS Access Web Release Notes](#)
- [PBS Access Desktop Release Notes](#)
- [PAS Administrator's Guide](#)
- [Diving Into Application Definitions](#)

Documentation can be found at the following URL:

<http://www.pbsworks.com/SupportGT.aspx?d=PBS-Professional,-Documentation>

## Document Conventions

The following typographical conventions are used in this document:

### **PAD\_HOME**

The PBS Access home directory which contains configuration and logging files. Default location for PBS Access Desktop is `C:\Users\\PBS_access\home`. This location can be overridden during the installation of PBS Access.



The home directory folder will be available only if the PBS Access Desktop is started after installation.

### **PAD\_EXEC**

The PBS Access execution directory which contains binaries and scripts. Default location for PBS Access Desktop is `C:\Program Files\altair\PBS Access\2018.2\exec`. This location can be overridden during the installation of PBS Access.

### **PAW\_HOME**

The PBS Access home directory which contains configuration and logging files. Default location for PBS Access Web is `/var/spool/PBS_access/2018.2/home`. This location can be overridden during the installation of PBS Access.

### **PAW\_EXEC**

The PBS Access execution directory which contains binaries and scripts. Default location for PBS Access Web is `/opt/altair/PBS_access/2018.2/exec`. This location can be overridden during the installation of PBS Access.

### **File Pathname**

Names of a file or a directory are displayed as follows:

```
/home/user1/submission_dir/PBSJobApp/174.hwesuse113.oe
```

### **Keyboard Inputs**

Keyboard inputs, for example commands that are entered are displayed as follows:

```
./etc/init.d/pbsworks-pa start
```

## Code Example and File Contents

Examples of programming code or the contents of a file are displayed as follows:

```
if userInputs['ACCOUNT']:  
    job.attr_accounting_label =userInputs['ACCOUNT']
```

## Parameters, Environment Variables, XML tags

Environment variables, XML tags, parameters, and references to a snippet of a file are displayed as follows:

Set the value of the `MaxBackupIndex` parameter to the maximum number of log files to keep.

## UI Controls

User Interface controls such as command buttons, tab names, menu options, application fields, or items in a list are displayed as follows:

Click the **Settings** button.

## Document References

References to an external document are displayed as follows:

For more information see the *PBS Access Web 2018.2 Administrator's Guide*.

## Messages

Log file messages or messages returned from a command are displayed as follows:

Please make sure PBSA engine and Monetdb Server are not running.

## Notes


Information or notes that supplement or emphasize important points are displayed as follows:



The default TOC type for a result file is set in the `plugin_def.xml` file


## Tips

Tips that makes your work easier are displayed as follows:

 If you get an error during registration of the HyperView Player, right click on the shortcut and choose the **Run as Administrator** option from the context menu.

## Cautions

Cautions draw special attention to any action that could cause damage to the product or cause the loss of data. For example:

 Do not turn the monitor off by unplugging it from the computer or wall socket. Severe damage to the monitor may result. Turn the monitor off before unplugging it.

## Acronyms

The following acronyms are used in this guide.

Table 1. Acronyms

Acronym	Description
API	Application Programming Interface
HPC	High Performance Computing
PBS	Portable Batch System

# 1. Introduction

---

The PBS 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 PBS Access and application definitions.

The CLI is available with PBS Access 2018.2 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. Please see the following chapters for a description of each command, usage information, and examples.

- [Administering PBS Access on page 3](#)
- [Submitting and Monitoring Jobs on page 19](#)
- [Maintaining Files and Directories on page 39](#)

## What's New

This release of CLI 2018.2 is in support of the PBS Access Web 2018.2 and PBS Access Desktop 2018.2 release. The CLI 2018.2 is installed automatically with PBS Access 2018.2. The installation of CLI is no longer available as part of the PAS installer.

The following are the new features:

- Automatically installed with PBS Access Web 2018.2 and PBS Access Desktop 2018.2
- New commands available with PBS Access Desktop only:
  - pas-server-add
  - pas-server-list
  - pas-server-delete
  - pas-custom-action-list
  - pas-execute-custom-action
- New options are added for some of the commands:
  - -F JSON option can be used to print the command output in JSON format
  - -p <port number> option is added to the pas-ping command to ping a server using a non-default port number.

- `-P <Protocol>` option is added to the `pas-ping` command to ping the server using the specified protocol

## PBS Access Command Line Interface Path

The default installation location of the CLI scripts is:

PBS Access Desktop:

```
C:\Program Files\altair\PBS Access\2018.2\exec\bin\commandline\
```

PBS Access Web:

```
/opt/altair/PBS_Access/2018.2/exec/bin/commandline/
```

## Prerequisites

Following are the prerequisites to run the CLI:

### PBS Access Desktop

- PBS Access Desktop must be installed and running
- Establish a connection to the HPC cluster by registering it using PBS Access Desktop UI or using the `pas-server-add` command, so that you may begin using the CLI commands.
- 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.
- The PBS Access CLI commands work only from the following path:

```
PAD_EXEC\bin\commandline\
```

### PBS Access Web

- PBS Access Web must be installed and running
- Establish a connection to the HPC cluster by adding a service cluster using PBS Access Desktop UI, so that you may begin using the CLI commands.
- 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. Administering PBS Access

---

The following administrator's commands are available to help with the following administrative tasks. 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.

- [Determining if Server is Up and Running](#)
- [Getting Server Information](#)
- [Displaying Server Version & Build Information](#)
- [Adding a Server](#)
- [Listing Servers](#)
- [Deleting a Server](#)
- [Determining if the User is an Administrator](#)
- [Getting the Staging Area Root Directory](#)
- [Setting the User's Password](#)

## Determining if Server is Up and Running

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

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

### Usage

```
pas-ping [options]
```

### Options

`-h`

Get help information for the command.

`-H` or `-s SERVERNAME`

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



The `-H` or `-s SERVERNAME` option is not supported for PBS Access Desktop.

`-p PORT`

Use this option when PAS is listening on a non-default port. The default port is 17084.

`-P PROTOCOL`

Protocol to be used to ping the server. The valid options are `http` and `https`. The default protocol is `https` for PBS Access Web and `http` for PBS 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 the default PAS server that is listening on a non-default port:

```
pas-ping -p 17085
```

Ping a PAS server and get the output in JSON format:

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

Ping the default PAS server by providing the protocol:

```
pas-ping -P http
```

## 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'}
```

## Getting Server Information

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

### Usage

```
pas-getsrv [options]
```

### Options

`-h`

Get help information 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.



The `-s SERVERNAME` option is not supported for PBS Access Desktop.

## Command Examples

Get information about the default server:

```
pas-getsrv
```

## Command Output Example

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

## Displaying Server Version & Build Information

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

### Usage

```
pas-ver [options]
```

### Options

`-h`

Get help information 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 will be assumed.



The `-s SERVERNAME` option is not supported for PBS 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 the 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}
```

## Adding a Server

Use the `pas-server-add` command to add a server (cluster) by providing a `SERVERNAME`, machine details in the form of a `HOSTNAME` or `IPADDRESS`, and credentials of a user. Credentials are provided by a `USERNAME` and either a `PASSWORD` or an `SSHKEYPATH`.

`SERVERNAME` is the name of the cluster as it will be known within PBS Access. The special characters allowed while providing the `SERVERNAME` are dot, underscore, hyphen, and space. The `SERVERNAME` should not end with white space.



This command is valid for PBS Access Desktop only.

By default, the user who issues this command is logged into the `SERVER`. You can use `-L` option to skip logging in while adding a server.

## Usage

```
pas-server-add -s SERVERNAME -n HOSTNAME|IPADDRESS -u USERNAME -p PASSWORD | -l  
SSHKEY [options]
```

## Options

`-h`

Get help information for the command.

`-L`

If you have chosen to enter your password, then use this option so that you do not have to enter your credentials again. An SSH key is generated and stored allowing you to login to the cluster without having to enter your credentials. The SSH key file path should be provided in quotes.

`-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>: 4443 where <HOSTNAME> is the hostname of the machine where PBS Access Web is installed.

-a

The Adaptor type option is used in Desktop Embedded mode (Inspire Integration). The information provided is updated in the serverData.xml file. The default value is PBS\_SSH\_ADAPTOR.

-P

Provide the server properties. This option is used in Desktop Embedded mode (Inspire Integration). 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>...].

## Command Examples

Adding a server with username and password:

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

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 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
```

## Listing Servers

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



This command is valid for PBS Access Desktop only.

### Usage

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

### Options

`-h`

Get help information 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 : cluster_qa Host : altvm10vm14
```

```
ServerId : cluster_rm Host : altvm10vm3
```

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

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



## Deleting a Server

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



This command is valid for PBS Access Desktop only.

### Usage

```
pas-server-delete -s SERVERNAME
```

### Options

`-h`

Get help information for the command.

### Command Examples

Delete a specified server (cluster):

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

### Command Output Example

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

## Determining if the User is an Administrator

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.

### Usage

```
pas-admin [options]
```

### Options

`-h`

Get help information 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.



The `-s SERVERNAME` option is not supported for PBS 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 Example

```
pas-admin  
true
```

## Getting the Staging Area Root Directory

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

### Usage

```
pas-getsroot [options]
```

### Options

`-h`

Get help information 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 Example

```
pas-getsroot  
/stage
```

## Setting the User's Password

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.

### Usage in PBS Access Web

```
pas-passwd [options]
```

### Usage in PBS Access Desktop

```
pas-passwd -s SERVER -u USERNAME [options]
```

### Options

`-h`

Get help information 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 logged in 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.

`-F JSON`

Print the output format in JSON.

## 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 Example

```
pas-passwd
Enter user's password:xxxxxxx
Retype password:xxxxxxx
Changed user root's password on server <localhost>
```



## 3. Submitting and Monitoring Jobs

---

Seven commands are available for job submission, job monitoring, and obtaining job results. 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.

- [Getting a List of all Applications](#)
- [Getting a List of Application Parameters](#)
- [Submitting a Job](#)
- [Checking the Status of Jobs](#)
- [Getting a Summary of Job Information](#)
- [Downloading Job Results](#)
- [Deleting a Job](#)
- [Getting a List of Custom Actions Defined for a Job](#)
- [Executing a Custom Action for a Job](#)

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.

## Getting a List of all Applications

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

### Usage

```
pas-getapps [options]
```

### Options

`-h`

Get help information 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.

`-a APPNAME`

Use this option to return a list of all parameters available for the specified `APPNAME`.

`-F JSON`

Print the output format in JSON.

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

```
pas-getapps -a optistruct -s 192.168.4.123
```



## Command Output Example

```
pas-getapps
ShellScript2
RADIOSS
Sleeper
JobArray
PBSJobApp
OptiStruct
Signal Trap
ShellScript
```

```
pas-getapps -s 192.168.4.123 -a PBSJobApp
ApplicationId=PBSJobApp
ApplicationName=PBSJobApp
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)
```

## Getting a List of Application Parameters

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.

### Usage

```
pas-getapp -a APPNAME [options]
```

### Options

`-h`

Get help information 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" and redirect the output to a file called `/tmp/optistruct_params`:

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

## Command Output Example

```
pas-getapp -a PBSJobApp
ApplicationId=PBSJobApp
ApplicationName=PBSJobApp
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)
```

## Submitting a Job

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 PBS Access Web. For more information about uploading job files see [Uploading a File on page 41](#).

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

1. The application parameters can be entered manually on the command line.
2. 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.

## Usage

```
pas-submit [options]
```

## Options

`-h`

Get help information 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.

`-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=PBSTestApp ApplicationName=PBSTestApp 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 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 Example

```
pas-submit ApplicationId=PBSTestApp ApplicationName=PBSTestApp JOB_NAME=PBSTestApp_test
MEMORY=100 NCPUS=1 JOB_SCRIPT=/tmp/hello_world.sh
SUBMISSION_DIRECTORY=/tmp/PAS_output
Response from server
149.hwesuse113-01
```

## Checking the Status of Jobs

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.

### Usage

```
pas-stat [options]
```

### Options

`-h`

Get help information 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 Example

```
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/
PBSJobApp/runtime,pbs_spawn@hwesuse113-01:/opt/altair/pas/2018.2/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=PBSJobApp,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_SC
RIPT=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=2018.2.1,AIF_USER=eble
icher,PBS_0_QUEUE=workq,PBS_0_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



## Getting a Summary of Job Information

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

### Usage

```
pas-sum [options]
```

### Options

`-h`

Get help information 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 Example

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

## Downloading Job Results

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 PBS 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.



TARGET\_LOCATION must have an ending slash or backslash:

`c:\temp\  
/home/user1/temp/`



This command is valid for PBS Access Web only.

## Usage

```
pas-getres -j JOBID -o TARGET_LOCATION [options]
```

## Options

`-h`

Get help information 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



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 Example

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

## Deleting a Job

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 1. 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.

## Usage

```
pas-del -j JOBID [options]
```

## Options

-h

Get help information 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 Example

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

## Getting a List of Custom Actions Defined for a Job

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



This command is valid for PBS Access Desktop only.

### Usage

```
pas-custom-action-list -j JOBID [options]
```

### Options

-h

Get help information 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-del -j 243.blrec166 -s 192.168.4.123
```

## Command Output Example

```
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
```

## Executing a Custom Action for a Job

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



This command is valid for PBS Access Desktop only.

### Usage

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



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`

Get help information 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=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=STOP
```



## Command Output Example

```
pas-execute-custom-action -j 4461 -N SEND_SIGNAL SIGNAL=STOP -s 192.168.4.123
  isStdErrTruncated : False
  isStdOutTruncated : False
  stdout:RXh1Y3V0aW5nIHNjcm1wdDogL3Zhci9zcG9vbC9wYnN3b3Jrcy9wYXNmVmJAX0C4yL2hvbWUvZGF0
YS9wYXNvdGFyZ2V0cy9sb2NhbGhvc3QvcmlvbnNpdG9yeS9wcm12YXR1L2d1bmVpYXR1ZC9qc29uLWFwcGxp
Y2F0aW9ucy9DdXN0b21BY3Rpb25BcHAv3VibW10dG1tZS9TZW5kU21nbmFsLnB5IHdpdGggc21nbmFsIDog
U1RPUAo=
  exitCode : 0
```



## 4. Maintaining Files and Directories

---

The following commands are available for PBS Access file operations. All commands will connect to the PAS server that is registered after the installation. Most commands provides an option to specify a non-default PAS server.

- [Creating a Directory](#)
- [Uploading a File](#)
- [Downloading a File](#)
- [Deleting a File](#)
- [Compressing a File](#)
- [Displaying a List of Files](#)

## Creating a Directory

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

### Usage

```
pas-fmkdir -d DIRECTORY [options]
```

### Options

`-h`

Get help information 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/PBSJobApp_output
```

Create a directory on a non-default server:

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

Create a directory in the running job's execution directory:

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

### Command Output Example

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

## Uploading a File

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

### Usage

```
pas-fup -f FILE -o DEST [options]
```

### Options

`-h`

Get help information 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 /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 Example

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

## Downloading a File

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

### Usage

```
pas-fdown -f FILE [options]
```

### Options

`-h`

Get help information 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 PBSJobApp_test.o325 -o c:\temp\PBSJobApp_test.o325 -j 325.blrec166
```

## Command Output Example

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

## Deleting a File

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

### Usage

```
pas-fdel -f FILE [options]
```

### Options

`-h`

Get help information 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 Example

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



## Compressing a File

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

### Usage

```
pas-fcompr -f FILE -o DEST [options]
```

### Options

`-h`

Get help information 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 PBSJobAppTest.o332 -o /users/tsmith/PBSJobAppTest.zip -j 332.blrec166
```

## Command Output Example

```
pas-fcompr -f /tmp/PAS_output/PBSJobApp_test/PBSJobApp_test.o149 -o /tmp/j149.zip  
File(/tmp/PAS_output/PBSJobApp_test/PBSJobApp_test.o149) compressed successfully
```

## Displaying a List of Files

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

### Usage

```
pas-flist -f FILE [options]
```

### Options

`-h`

Get help information for the command.

`-j JOBID`

List FILES in the job's execution directory. The job must be in a running state.

`-s SERVERNAME`

List FILES 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 dir1ister.e361 -j 248.blrec166
```

List all files in the job's execution directory:

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

## Command Output Example

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



## 5. PBS Access Commands Quick Reference

---

The following is a list of all PBS Access Commands and a link to each command's documentation page.

Table 1. Administering PBS Access

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

Table 2. Submitting and Monitoring Jobs

Command	Description
<a href="#">pas-getapps</a>	Returns a list of all applications known to the server.
<a href="#">pas-getapp</a>	Returns a list of all parameters available for the specified application as defined by the application definition on the server.
<a href="#">pas-submit</a>	Submits a job to the server.
<a href="#">pas-stat</a>	Displays the status of a job. You can display the status of a particular job or the status of all jobs.
<a href="#">pas-sum</a>	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.
<a href="#">pas-getres</a>	Downloads the files related to the given job identifier.
<a href="#">pas-del</a>	Deletes a job from the Workload Management System.

Table 2. Submitting and Monitoring Jobs

Command	Description
<a href="#">pas-custom-action-list</a>	Get a list of available custom action of a job.
<a href="#">pas-execute-custom-action</a>	Execute a custom action for a job.

Table 3. Maintaining Files and Directories

Command	Description
<a href="#">pas-fmkdir</a>	Creates a directory on the server.
<a href="#">pas-fup</a>	Uploads a file.
<a href="#">pas-fdown</a>	Downloads a file.
<a href="#">pas-fdel</a>	Deletes a file from the server.
<a href="#">pas-fcompr</a>	Compresses a file on the server.
<a href="#">pas-flist</a>	Displays a list of files on the server.

# Index

---

## A

administrative tasks 1, 3  
application definitions 1  
application parameters 19, 22, 24  
applications 19, 20

## F

file

*compress* 45  
*delete* 44  
*directory* 40  
*download* 19, 31, 42  
*list* 46  
*operations* 1, 39  
*upload* 19, 24, 41

## J

job

*custom action* 34, 36  
*delete* 33  
*execute custom action* 36  
*monitoring* 1, 19  
*results* 19, 31  
*status* 26  
*submission* 1, 19, 24  
*summary* 29

## P

PBS Access 31

*administrator* 14  
*build* 7  
*password* 16  
*port* 6  
*staging area* 15  
*status* 4  
*version* 7

PBS Access Commands

*pas-admin* 14  
*pas-custom-action-list* 34  
*pas-del* 33  
*pas-execute-custom-action* 36  
*pas-fcompr* 45  
*pas-fdel* 44  
*pas-fdown* 42

*pas-flist* 46  
*pas-fmkdir* 40  
*pas-fup* 19, 24, 41  
*pas-getapp* 19, 22  
*pas-getapps* 19, 20  
*pas-getres* 19, 31  
*pas-getsroot* 15  
*pas-getsrv* 6  
*pas-passwd* 16  
*pas-ping* 4  
*pas-server-add* 9  
*pas-server-delete* 13  
*pas-server-list* 11  
*pas-stat* 19, 26  
*pas-submit* 19, 24  
*pas-sum* 29  
*pas-ver* 7  
PBS Professional 1

## Q

*qstat* 26, 29

## S

Server

*hostname* 6