Altair Access Web 2018.4

Administrator's Guide

Altair | PBS Works™

PBS Works is a brand of Altair

Intellectual Property Rights Notice: Copyrights, Trademarks, Trade Secrets, Patents and Third Party Software Licenses

Updated: January 3, 2019.

Special 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.

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

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

Contents

1 What's New 5

2 Get Started 7

2.1 PBS Application Services 8
2.2 Document Conventions 9
2.3 System Requirements 10
2.3.1 System Requirements for Access Web 10
2.3.2 System Requirements for Remote Session Component 11
2.4 Hardware Requirements 12
2.5 Ports Used by Access Web 13
2.6 Supported Product Configurations 14
2.7 Prerequisites for Installation 15
2.7.1 Prerequisites for Installing Access Web 15
2.7.2 Prerequisites for Installing Remote Session Components 16
2.8 Roles in Access Web 18

3 Install Access Web and the Remote Session Component 19

3.1 Install Access Web 20
3.2 Install Remote Session Components 22
3.2.1 Install the Remote Session Component on the PBS Professional Server and on PAS 22
3.2.2 Install the Remote Session Component on the PBS MoMs 24
3.2.3 Install the Interactive Proxy on the Access Web Server 26

4 Configure Access Web After Installation 27

4.1 Copy Application Definitions and Site Configuration File 28
4.2 Configure the License Server 29
4.3 Log into Access Web 30
4.4 Add a Service Cluster 31
4.5 Onboard an Application Definition 35
4.5.1 Integrate Right Click Context Menu of Access Web 35
4.5.2 Master File Analyzer 38

5 Upgrade and Downgrade Access Web 39

5.1 Upgrade Access Web 40
5.2 Upgrade PBS Application Services 42

5.3 Downgrade Access Web....44
5.4 Downgrade PBS Application Services....45

6 Uninstall Access Web and Remote Session Component....46

6.1 Uninstall Access Web....47
6.2 Uninstall Remote Session....48
6.2.1 Unconfigure PBS Professional and PBS Application Services....48
6.2.2 Uninstall Remote Session Components....49

7 Access Web Service Commands....50

8 Manage Service Clusters....51

8.1 Add a Service Cluster....52
8.2 Edit a Service Cluster....56
8.3 Delete a Service Cluster....58

9 Advanced Configurations....59

9.1 Configure the Access Web Component....60
9.1.1 Change Port Numbers....60
9.1.2 Set the Double-Click Delay Time....67
9.1.3 Change the Database Password....67
9.1.4 Configure Default File Viewer....68
9.1.5 Configure Default Columns in Job List View....69
9.1.6 Change the Maximum File Upload Size....69
9.1.7 Change the File Opening behavior of a Remote Session Application....70
9.1.8 Configure Notifications for a Job State Change....70
9.1.9 Disable to View all Jobs....72
9.1.10 Add a Generic Action for a PAS server....73
9.1.11 Map a File Extensions to an Icon....76
9.1.12 Set Maximum Page Size for Files....77
9.1.13 Install and Configure SSL Certificate....77
9.2 Configure PBS Application Services....79
9.2.1 Enable PBS Job History....79
9.2.2 Verify the Installation of PBS Application Services....79
9.2.3 Configure System Zip Utility....80
9.2.4 Configure System Unzip Utility....80
9.2.5 Configurable Parameters....81
9.2.6 Configure PBS Resources....85
9.2.7 Configuring PAS Server Logging Behavior....85
9.2.8 Configuring JVM Performance....92
9.3 Configure the Remote Session Component....93
9.3.1 Change the Session Timeout Value for Interactive Applications....93

9.3.2 Change the Job Submission Host for Interactive Applications 93
9.3.3 Change the Job Submission Port for Interactive Applications 94
9.3.4 Configure GPU Limits when the Number of GPUs Change 94
9.4 Configure Results Visualization Service 97
9.4.1 Configure HyperWorks Location 97
9.4.2 Configure HyperWorks Licenses 97
9.4.3 Configure Solver File Readers 98
9.4.4 Supported Result File Types 100
9.4.5 Configure Data Directory 105
9.4.6 Set a Data Cleanup 105
9.4.7 Configure RVS Parameters 106

10 Application Definitions 109

10.1 Application Definition Components 110
10.1.1 Application Input File 110
10.1.2 Application Converter File (HPCBP Converter) 110
10.1.3 Application Runtime Scripts 110
10.1.4 Site Configuration File 111
10.1.5 Server Configuration File 111
10.2 Sample Application Definition ShellScript 112
10.3 Map Icons to an Application 113
10.4 Administration of Application Definition 114
10.4.1 Add a New Application Definitions 114
10.4.2 Application Definition Validation 114
10.4.3 Maintenance of Existing Application Definition 115
10.5 Sitewide Settings 116
10.5.1 Site Configuration File Content 116
10.5.2 Initial Site Configuration File after Installation of PAS 116
10.5.3 Sample of a Site Configuration File 117
10.5.4 Use Site Configuration Information in an Application Definition 118
10.5.5 Site Configuration File Backup 118
10.5.6 Site Configuration File Validation 118
10.6 Interactive Application Definitions 119
10.6.1 Mandatory Interactive Application Definitions Changes 119
10.6.2 Special Interactive Application Arguments 119
10.6.3 Add a New Interactive Application 121

11 Log Files 124

11.1 PAS Log File 125
11.2 Other Log Files 126

12 Troubleshoot Access Web 127

12.1 Run the Access Web Diagnosis Script 128

12.2 Troubleshoot PBS Application Services 129
12.2.1 Verify PAS Status 129
12.2.2 Troubleshoot PAS Job Submission Issues 130
12.2.3 Troubleshoot Error Messages during Installation 131
12.3 Troubleshoot Remote Session Components 134
12.3.1 Run the Remote Session Diagnosis Script 134
12.3.2 Remote Session Checklist 135
12.3.3 Display Session is shown but not the Application that was Started 136
12.3.4 Application Definition working in VirtualGL Environment Issue 137
12.3.5 Desktop Manager not Displaying 137
12.3.6 Graphic Card Compatibility Issues 138
12.4 Troubleshoot Result Visualization Services 140
12.4.1 Troubleshoot Checklist 140
12.4.2 Locate Log Files 140
12.4.3 Common Issues 141

What's New

1

Latest features available with Altair Access™ Web.

Improved Port Detection by Installer

To avoid potential conflicts the installer will detect if the default ports in the port range are busy and use new ports.

Consistent Files and Jobs Views

Standardization of job and file list appearance and behavior to improve consistency.

Persistence View

The last navigated folder path in the Files tab, the last viewed job status in the Jobs tab and the last viewed job file in the Output and Running Folder tabs are retained and displayed even if you switch tabs when you turn enable the preference Persistence View.

Custom File Icons

Custom icons can be configured for every file type in the `fileextensions.json` file located at `<PA_HOME>/config/pa/`

Display Hidden Files

Display hidden files and folders by enabling (name of Preference) located in the Preferences panel.

Copy and Paste File Paths

Navigate to different locations by copying and pasting the file path into the breadcrumbs at the top of the List view.

Filter Subjobs View

Subjobs can now be filtered by job status: queued, running, completed, and failed. The filter can be saved as favorite for reuse.

Reset Preferences to the Default

Preferences can be reset to their default value. The following preferences can be reset to the default value:

- Saved filters
- Custom columns
- Default job actions
- Persistence
- Notifications

File Upload Progress

While uploading files, a file upload widget displays the upload progress in the file upload notification.

Generic or Server Actions

Perform a action on a server. These actions are server specific and can be executed on single or multiple jobs without being specific to an application or job state.

Visualize Results from the Files Tab

All Results Visualization operations can now be performed from the Files and Jobs tabs.

Save and Open Plots (.rvs) Files

Save plots as .rvs files. These .rvs files can be opened later without a plot operation.

Overlay Plots

Compare plots by overlaying them. Overlay multiple plots, hide and show overlaid plots and save the resulting view.

Hiding a Plot

Improved control over display and hiding of plots by clicking the plot name.

Shared File System for RVS

Access Web supports shared file systems. Results visualization performance can now be enhanced by mounting the stage directory and scratch directory on the machine where Access Web is installed. This is applicable when Access Web and PAS are installed on separate machines.

Screen Capture in Remote Session

Grab a screen capture of a remote session and save it as an image file on your local system.

Get Started

2

Use Access Web to submit jobs to a Workload Manager.

This chapter covers the following:

- 2.1 PBS Application Services (p. 8)
- 2.2 Document Conventions (p. 9)
- 2.3 System Requirements (p. 10)
- 2.4 Hardware Requirements (p. 12)
- 2.5 Ports Used by Access Web (p. 13)
- 2.6 Supported Product Configurations (p. 14)
- 2.7 Prerequisites for Installation (p. 15)
- 2.8 Roles in Access Web (p. 18)

Altair's new Access Web provides a simple, powerful, and consistent interface for submitting and monitoring jobs on remote clusters, clouds, or other resources. Engineers and researchers can now focus on core activities and spend less time learning how to run applications or moving data around. The Access Web remote visualization and collaboration capabilities bring access to an expensive, highend 3D visualization datacenter hardware right to the user. Access Web provides an ability to visualize the results by extracting plot and animation data. You can view plots for running jobs as well as for jobs which have been successfully completed. You can download and analyze animations using the Altair HyperView Player. Results Visualization Service allows you to compare two or more plots in the result viewer, save the plot you generated with the data as .rvs file and directly view your saved plots with the selected data.

Access Web Features:

- **Novice to Expert:** simple and powerful
- **Same UX:** desktop and web
- **Secure:** protected access to HPC resources
- **End-to-end:** submit, monitor progress, steer, fix, and rerun jobs
- **Collaborate:** shared 3D analysis
- **3D Remote Visualization**
- **Save time:** Simplify job submission and management thanks to a powerful GUI with smart, simplified interfaces
- **Be more productive:** Spend more time focused on work and not IT tasks - for example, monitor jobs graphically without having to download huge job files
- **Increase ROI:** Consolidate access to applications and optimize license availability
- **Reduce errors and improve consistency:** Embed your company's best-practice "know how" directly into Application Definitions used for job submission

2.1 PBS Application Services

PBS Application Services (PAS) allows users to quickly create application portals that boost productivity and accelerate innovation.

Today, doing more with less is simply business as usual. Enterprises and institutions are placing higher demands upon their high performance computing infrastructure – larger and more complex applications and dynamic end-user work loads. Optimizing end-user productivity and getting the most from limited resources and complex application work loads can seem to be an impossible challenge.

PAS is a middleware component that communicates with the PBS Professional complex and the operating system, and provides services for the front end client applications such as graphical user interfaces. This middleware component is used as the preferred method of communication between the integration of client applications and the PBS Professional complex. PAS allows users to quickly create application portals that boost productivity and accelerate innovation. Application administrators can use PAS to create custom, solver specific application definitions. Leveraging key industry standards like the Open Grid Forum High Performance Computing Basic Profile, PAS makes it easy to respond to dynamic changes to your users and applications.

Benefits of PBS Application Services:

- Stable WS-based application programming interface (API) for integration into bigger systems
- Streamlined APIs for integration of applications into PBS Professional for the independent software vendors
- Front end graphical user interface application building is made easier
- Independent software vendors and the community can define new application definitions and exchange them within the PBS Professional complex in a turn-key fashion
- Platform neutrality allowing PBS Application Services to run on Linux platforms
- Customized application definition files can be shared across all supported platforms
- Transactional manner of operations (e.g. eliminates out of sync issues with resources not included in scheduling, etc.)
- Grid-friendly as it uses the same extensible markup language/web services (XML/WS) as large-scale grid solutions
- Easier integration into Enterprise SOA and business process management
- Standards-based implementation using Web Services, WS-Security, high performance computing basic profile (HPCBP)

PBS Application Services provides the following major groups of functionality:

- Installation and management of application definitions by administrators
- Site configuration such as billing accounts and MPROC by administrators
- User submission and management of PBS Professional jobs
- File operations services

2.2 Document Conventions

Common typographical conventions for Access Web technical publications.

PA_HOME
: The Access Web home directory which contains configuration, data, and logging files. Default location is `/var/spool/pbsworks/2018.4/access/home`, however this can be overridden during the installation of Access Web.

PA_EXEC
: The Access Web execution directory which contains binaries and scripts. Default location is `/opt/altair/pbsworks/2018.4/access/exec`, however this can be overridden during the installation of Access Web.

PBS_HOME
: The PBS Professional home directory which contains configuration and logging files. Default location is `/var/spool/pbs`, however this can be overridden during the installation of PBS Professional.

PBS_EXEC
: The PBS Professional execution directory which contains binaries and scripts. Default location is `/opt/pbs`, however this can be overridden during the installation of PBS Professional.

2.3 System Requirements

Supported platforms and browsers, hardware requirements, required ports, and security.

2.3.1 System Requirements for Access Web

Supported platforms and browsers, hardware requirements, and required ports.

Supported Platforms

Access Web is supported on the following Linux 64-bit platforms:

- Red Hat Enterprise Linux 7.4
- Cent OS 7.4
- SLES 12 SP2

Supported Server Platforms

Access Web supports only the 64-bit (x86_64) architecture platforms for visualization of results.

Supported Browsers

Access Web is supported on the following browsers:

Windows

- Firefox latest ESR (Only Extended Support Release is supported. Please refer to https://www.mozilla.org/en-US/firefox/organizations/)
- Google Chrome latest
- Microsoft Internet Explorer 11
- Microsoft Edge (limited testing)

Linux

- Not supported at this time.

OSX

- Firefox latest ESR (Only Extended Support Release is supported. Please refer to https://www.mozilla.org/en-US/firefox/organizations/)
- Google Chrome latest
- Safari Latest

See Also

Change Port Numbers

2.3.2 System Requirements for Remote Session Component

Hardware and system requirements necessary for running an interactive application.

Graphics

- Supported only on Nvidia and ATI (AMD) graphics cards.
- Install the 3D adapter drivers provided by the manufacturer.
- Full 3D acceleration or Pixel Buffer support should be enabled by the Linux drivers.

GPU Cores and Memory

Specific to the interactive applications being run.

CPU Cores and Physical Memory

Specific to the interactive applications being run.

Other

- For running interactive sessions, X Server and application on local display must be configured and working.
- X Server must be configured to export True Color (24 bit or 32 bit) visuals.
- Use Virtual Private Networking or secured channels for communication between clients and interactive server if encryption is required.
- The PBS Professional execution host must be able to access and run the applications available through Access Web

2.4 Hardware Requirements

Minimum and recommended hardware requirements necessary to install and run Access Web.

Hardware Requirements for Access Web

Minimum hardware configuration required for Access Web are:

Hardware	Minimum Requirement	Recommended
CPU	2 CPU cores with a minimum speed of 2.5 GHz	4 CPU cores with a minimum speed of 2.5 GHz
Memory (Physical)	4 GB	8 GB
Disk Space	4 GB	10 GB

Hardware Requirements for Results Visualization Service

Minimum hardware configuration required for Results Visualization Service (RVS) are:

Hardware	Minimum Requirement	Recommended
CPU	A Quad Core Processor of 2.5 GHz	A Quad Core Processor of 3.1 GHz
Memory (Physical)	16 GB	32 GB
Disk Space	500 GB	500 GB

Note: Result Visualization of results requires good network connectivity to all the connected file servers such as PAS server and job execution hosts. A minimum speed of 100mbps is required while a speed of 1gbps is recommended.

You can install Access Web and HyperWorks Desktop software separately. The HyperWorks Desktop path can be made accessible from Access Web by creating a mounting point for the HyperWorks installed directory on Access Web installed machine.

Tip: For a better performance of RVS, mount the stage directory and scratch directory on the Access Web installed machine if Access Web and PAS are installed separately.

2.5 Ports Used by Access Web

A list of ports used by Access Web services and components.

The Access Web installer has the auto-port detection logic in place and ports will be picked up by each service within the specified range.

If the required port is not available during the installation, then Access Web will increment the port number and try again. This will continue until a free port is found, or until 100 consecutive ports are refused at which point the installer will ask for a port.

Below is a table of all ports and the port range range used by Access Web.

Table 1: Ports Used by Access Web

Port	Port Range	Service Using the Port
4443	4443 - 4542	Gateway
4543	4543 - 4642	Access Web Server
4643	4643 - 4742	Postgres Database
4743	4743 - 4842	Message Broker (ActiveMQ)
4843	4843 - 4942	Remote Session Services Webserver
4943	4943 - 5042	Remote Session Services Job Update
5043	5043 - 5142	Result Manager Services
5143	5143 - 5242	Result Core Services
5243	5243 - 5342	PBS Application Services
5343	5343 - 5442	Job Profiles Services
5443	5443 - 5542	Remote Session Services Proxy
5901	5901 to 59XX	Turbo VNC Server Port for Remote Session

Note: It is recommended to have firewall on installation node and block all Access Web ports from outside world except Gateway port (4443) and Remote Session Services Job Update port (4943).

2.6 Supported Product Configurations

Supported product configurations for using Access Web.

Access Web	PAS	PBS Professional	Hyperworks
2018.4	2018.4	18.2.3 18.2.2 18.2.1 18.1.3 (OSS) 14.2.4 14.1.0 (OSS)	14.0 2017.2

Note: HyperWorks Desktop application is required to visualize CAE results.

2.7 Prerequisites for Installation

Prerequisites and planning for installing Access Web, PBS Application Services, Remote Session Components, and for Enabling Visualization of CAE Results.

2.7.1 Prerequisites for Installing Access Web

Mandatory requirements for PBS Professional and the Service User.

Service User

The Service User is provided by the administrator during installation and it should be pre-existing in the system. This user will owns Access Web services and the files in PA_HOME and PA_EXEC. The default user is "pbsworks" for installing Access Web.

PBS Professional

Ensure that a supported version of PBS Professional is installed on the HPC cluster.

PAS

To take full advantage of all the new features available with Access Web, the cluster must be updated to the latest version of PAS.

PBS Application Services Staging Directory

During the installation of the PAS Server, you will be prompted to enter a value for the PAS staging directory. The staging directory is where the necessary job files are transferred after job submission via a client, for example Access Web, for transfer to PBS Professional for execution.

The default location of the staging directory is: `/stage`

However, you have the option to choose a custom stage directory during installation.

Following are considerations for selecting and creating the staging directory:

- The stage area can grow quite large, depending on the size of the average job. Give careful consideration to the disk usage and disk capacity.
- The pathname for the staging directory should not contain spaces.
- PAS implements an automatic staging directory cleanup such that any data in the staging directory that is not generated by PAS is at risk of being purged without notice. It is recommended to select a staging directory where only PAS data files will be stored. Use of a directory that contains important data is not recommended. The automatic staging directory cleanup does not delete any files in the user's home directory. When the staging directory is set to `$USER_HOME` (for example: `$USER_HOME/stage`), the files in that directory are not automatically cleaned up and the responsibility for maintaining the staging directory falls on the individual user.

Altair HyperWorks Desktop

A supported version of Altair HyperWorks Desktop must be installed on the HPC cluster.

> **Tip:** Install HyperWorks Desktop by following the Linux installation instructions in the HyperWorks 2017 Installation Guide.

The HyperWorks Desktop installation should not be in `/root` or any user's home directory and in a location where all users have read and write permission.

> **Troubleshooting:** If HyperWorks Desktop is not installed, only plotting of CSV files is supported.

2.7.2 Prerequisites for Installing Remote Session Components

Mandatory requirements for Remote Session components.

Access Web

Access Web must be installed prior to installing the Remote Session components.

Host Name and Port

It is recommended to configure hostname resolution, so that the PBS MoMs can connect to the Access Web server using its hostname rather than IP address and the configured Remote Session Service Job Update port. After installing the Remote Session component, view the value of the `jobsub.monitor.host` variable in the file `PA_HOME/config/displaymanager/dmrest.properties` to confirm the configured hostname. If there is no access through hostname, then a slight delay may occur while opening a remote session for the first time.

The Remote Session Proxy should be able to connect to interactive execution node through hostname and Turbo VNC Port.

Refer to `Ports Used by Access Web` section in System Requirements for more information on ports.

Prerequisites for Installing on the PBS MoM

For the Access Web Remote Session installer to set the custom resource at each execution host, the root user of all execution hosts must be granted operator access. Prior to starting the Remote Session installer, issue the following command to grant this access:

```
qmgr: s s operators+=root@*
```

If root operator permission is set, number of ngpus available will be set by the remote session installer while installing agent.

> **Note:** After installation, remove the root user of the PBS MOMs from the operators list on the PBS server.

After installing the Remote Session components, you can remove the root user of the PBS MOMs from the operators list on the PBS server using the command:

```
qmgr: s s operators-=root@*
```

If this access is not granted, then you will manually have to set the custom resource by issuing a `qmgr` set command for each execution host after installation of Remote Session. For example:

```
set node <node> resources_available.ngpus = 4
```

> **Warning:** Restricting it to a specific subdomain still allows anyone running Linux on the subdomain access to the PBS Server.

Prerequisite Resource Libraries for Interactive Application

The following libraries must be installed on the machine hosting the Access Web interactive proxy (these libraries do not need to be installed on the PBS Server or the PBS MoM) before attempting to install the Interactive Application component. Use the appropriate system tool (e.g. RPM, YUM, YAST etc.) to install them.

- libc.so.6()(64bit)
- libc.so.6(GLIBC_2.2.5)(64bit)
- libc.so.6(GLIBC_2.3)(64bit)
- libcrypt.so.1()(64bit)
- libdl.so.2()(64bit)
- libdl.so.2(GLIBC_2.2.5)(64bit)
- libGL.so.1()(64bit)
- libGLU.so.1()(64bit)
- libICE.so.6()(64bit)
- libm.so.6()(64bit)
- libm.so.6(GLIBC_2.2.5)(64bit)
- libpam.so.0()(64bit)
- libpthread.so.0()(64bit)
- libpthread.so.0(GLIBC_2.2.5)(64bit)
- libpthread.so.0(GLIBC_2.3.2)(64bit)
- libSM.so.6()(64bit)
- libX11.so.6()(64bit)
- libXaw.so.7()(64bit)
- libXcursor.so.1()(64bit)
- libXext.so.6()(64bit)
- libXmu.so.6()(64bit)
- libXt.so.6()(64bit)
- libXv.so.1()(64bit)
- libz.so.1()(64bit)

2.8 Roles in Access Web

Roles defined in Access Web.

Service User

The Service User is provided by the administrator during installation and it should be pre-existing in the system. This user will own Access Web services and the files in PA_HOME and PA_EXEC. The default user is "pbsworks" for installing Access Web.

Portal Administrator

The first user to login to Access Web after installation is automatically designated as the portal administrator. The portal administrator is the only user who can add, edit, or delete service clusters.

Install Access Web and the Remote Session Component

3

Instructions for installing Access Web, PBS Application Services, and Remote Session components, so that interactive and non-interactive jobs can be submitted.

This chapter covers the following:

- 3.1 Install Access Web (p. 20)
- 3.2 Install Remote Session Components (p. 22)

3.1 Install Access Web

Install Access Web component, so that you can submit non-interactive jobs to the Workload Manager.

- Review System Requirement of Access Web
- Review Prerequisites of Access Web
- Uninstall previous versions of Access Web.

Ensure you having the following information before you start your installation:

- License server details
- HyperWorks location

A binary or executable needs to be downloaded or obtained using your usual Altair support channels.

The Access Web 2018.4 installer provides an option for installing Access Web, PBS Application Services, or both.

1. Login to the machine as root where you want to install Access Web.
2. Enter the command:
   ```
   ./AltairAccessWeb_<Version>_<Build ID>_<YYYYMMDD>_<Timestamp>.bin -i console
   ```
3. Read the introduction and press `ENTER`.
4. Page through the license agreement by pressing `ENTER` until you are asked to accept its terms and conditions.
5. Accept the license agreement by entering `Y` and pressing `ENTER`.
6. Choose one of the following options:
   - Enter **1** and press **ENTER** to install Access Web Services.
   - Enter **2** and press **ENTER** to install PBS Application Services.
   - Enter **3** and press **ENTER** to install Access Web Services and PBS Application Services. This is the default option.

   > **Note:** The following steps will change based on your choice.

7. Enter the license server details and press **ENTER**.
   The license server details should have the port and hostname in the format `port@hostname`.

   > **Note:** You can provide multiple license server details separated by a colon in Linux and semi-colon in Windows.

   > **Tip:** It is not mandatory to provide the license server at installation time. You will however be required to configure the license server information prior to logging into Access Web.

8. Enter a staging directory for PBS Application Services and press **ENTER**.
   The staging directory is where user's job files will be staged prior to execution. This directory must exist prior to the installation of PAS.

> **Tip:** The stage area can grow quite large depending on the size of the average job. It's not recommended the use `/tmp` or user's `/home` as the staging directory, as they will get periodically purge.

9. Enter the service user and press **ENTER**.

> **Note:** The default service user is 'pbsworks' and the Access Web service is registered under this user.

10. Enter the location where the Access Web binaries are to be installed.
    You may choose to install in the default location.
11. Enter the location where the Access Web configuration and logs files are to be installed.
    You may choose to install in the default location.
12. Optional: Enter the HyperWorks location for visualizing your results.
    The Access Web 2018.4 supports HyperWorks 2017.2.0.16.
13. Review the installation summary and press **ENTER**.
    The installation starts. It may take a few minutes for the installation to complete.
14. Start Access Web manually by entering the following command:
    ```
    service pbsworks-pa start
    ```
15. Login to the machine hosting the PBS Server.
16. As a PBS Operator or Manager enter the following command:
    ```
    qmgr -c 'set server job_history_enable=True'
    ```

If you want to run interactive jobs, you must now install the Interactive Application component of Access Web.

3.2 Install Remote Session Components

Install components necessary to enable the remote session capabilities of Access Web.

Review the system requirements and prerequisites for installation.

A binary that installs the Remote Session components needs to be downloaded or obtained using your usual Altair support channels.

The installer will need to be run multiple times across several machines and must be performed in the following sequence:

1. Install the Remote Session component on the PBS Professional headnode and in the machine where PBS Application Services is installed. This installation will:
   - add a custom resource to PBS Professional called "ngpus"
   - creates an interactive queue called "iworkq"
   - add a new application definition "GlxSpheres" to PAS
   - restart PBS Professional and Access Web
2. Install the Remote Session component on all PBS MoMs on which you want to run interactive jobs.
3. Install proxy software necessary for running interactive applications on the machine hosting Access Web.

3.2.1 Install the Remote Session Component on the PBS Professional Server and on PAS

Install binaries and configure PBS Professional and PAS to support interactive applications.

Review the system requirements and prerequisites for installation.

Install the remote session component on the PBS Professional headnode and in the machine where PAS is installed.

This installation will:

- add a custom resource to PBS Professional called "ngpus"
- add an interactive queue called "iworkq"
- restart Access Web
- restart PBS Professional

1. Login to the machine as root where the PBS Professional Server and PAS is installed.
2. Enter the command:

   ```
   ./AltairRemoteSessionAgent_<Version>_<Build ID>_<YYYYMMDD>_<Timestamp>.bin -i
    console
   ```

3. If you are installing the remote session component for the first time, then you will see the below message, enter `1` and press **ENTER**.

   ```
   Manage Instances
   ----------------
   ->1- Install a new instance
     2- Modify an existing instance
   ```

4. Read the introduction and press **ENTER**.
5. Page through the license agreement by pressing **ENTER** until you are asked to accept its terms and conditions.
6. Accept the license agreement by entering **Y** and pressing **ENTER**.
   Four options are displayed.
7. Enter `1` to configure the PBS Professional and PAS servers and press **ENTER**.
8. PBS Professional and PAS is restarted during the installation process, choose whether you want to proceed:
   - Choose **No** to exit and run the installer at a more suitable time.
   - Choose **Yes** to run the installer.

   > **CAUTION:** It is advisable that you run the installer when critical jobs are not running.
9. Enter the number of GPUs that are available in the cluster and press **ENTER**.
   If you have a cluster with 10 execution hosts and only two of those execution hosts have associated GPUs, then add up the number of GPUs for both execution hosts and enter this number.
10. Review the installation summary and press **ENTER**.
    The installation starts. It may take a few minutes for the installation to complete.
11. Press **ENTER** to complete the installation process.

Verify that iworkq is created and a GPU resource is configured for PBS Professional.

Verifying the Existence of the Interactive Queue

Verify that a PBS Professional interactive queue has been created.

After running the interactive installer on the PBS Professional Server, a queue called "iworkq" should exist. All interactive jobs are submitted to this queue. The attribute *max_queued_res.ngpus* denotes the number of GPUs available in the cluster and should reflect the number entered during installation. Other attributes used internally by Access Web are *resouce_max.ngpus* and *resouce_min.ngpus*. These two attributes are always to set the value of "1".

Enter the command:

```
qmgr -c "p q iworkq"
```

Output similar to the following is displayed:

```
#
# Create queues and set their attributes.
#
#
# Create and define queue iworkq
#
create queue iworkq
set queue iworkq queue_type = Execution
set queue iworkq Priority = 150
set queue iworkq max_queued_res.ngpus = [o:PBS_ALL=7]
set queue iworkq resources_max.ngpus = 1
set queue iworkq resources_min.ngpus = 1
set queue iworkq resources_default.arch = linux
set queue iworkq resources_default.place = free
```

```
set queue iworkq default_chunk.mem = 512mb
set queue iworkq default_chunk.ncpus = 2
set queue iworkq enabled = True
set queue iworkq started = True
```

Verifying the Existence of the Interactive Custom Resource

Verify that a new resources called ngpus has been added to PBS Professional.

After running the interactive installer on the PBS Professional headnode, a new custom resource called ngpus is added to PBS Professional. This resource is necessary to run interactive jobs. You can verify the existence of this custom resource by viewing the contents of the PBS Professional resource definition file and the scheduler's configuration file.

1. Navigate to the location `PBS_HOME/server_priv`.
2. Enter the command:

   ```
   grep -A3 "DM STATIC" resourcedef
   ```

   The resource ngpus should be defined in this file.

   ```
   # ************ BEGINNING OF DM STATIC RESOURCES SECTION.DO NOT EDIT BY HAND ******
   ngpus type=long flag=nh
   # ************ END OF DM STATIC RESOURCES SECTION.DO NOT EDIT BY HAND ************
   ```
3. Navigate to the location `PBS_HOME/sched_priv`.
4. Enter the command:

   ```
   grep ngpus sched_config
   ```

   The resource ngpus should be displayed in the list of resources defined in this file.

   ```
   resources: "ncpus, mem, arch, host, vnode, netwins, aoe, ngpus"
   ```

Verifying the Installation of the Interactive Application Definition

Verify that a new application definition called GlxSpheres is installed.

1. Login to the machine hosting the PAS Server as root or a user with sudo permissions.
2. Navigate to the directory `PA_HOME/data/pas/targets/localhost/repository/applications`.
3. Verify that a new application definition has been placed in this location called GlxSpheres.

3.2.2 Install the Remote Session Component on the PBS MoMs

Install binaries and configure the PBS Mom to support interactive applications.

Review the system requirements and prerequisites for installation.

Install necessary components on the PBS MoM to support interactive applications, including TurboVNC and Virtual GL. The installer also configures a new resource called "ngpus".

Note: This installation process must be repeated for all execution hosts where you want to run interactive jobs.

1. Login to the machine as root where the PBS Professional MoM is installed.
2. Enter the command:

```
./AltairRemoteSessionAgent_<Version>_<Build ID>_<YYYYMMDD>_<Timestamp>.bin -i
 console
```

3. If you are installing the PBS MoM interactive component on a machine hosting both the PBS Professional Server and the MoM you will see the below message, enter `1` and press **ENTER**.

```
Manage Instances
----------------
->1- Install a new instance
  2- Modify an existing instance
```

4. Read the introduction and press `ENTER`.
5. Page through the license agreement by pressing `ENTER` until you are asked to accept its terms and conditions.
6. Accept the license agreement by entering `Y` and pressing `ENTER`.
   Four options are displayed.
7. Enter `2` and press **ENTER**.
8. Enter the location where the binaries are to be installed and press **ENTER**.
   You may choose to install in the default location.
9. Enter the location where the configuration and logs files are to be installed and press **ENTER**.
   You may choose to install in the default location.
10. Enter the number of GPUs available on the execution host and press **ENTER**.
11. Review the installation summary and press **ENTER**.
    The installation starts. It may take a few minutes for the installation to complete.
12. Press **ENTER** to complete the installation process.
    The TurboVNC and Virtual GL is installed to support interactive applications by configuring XServer.
13. Restart the XServer.

Warning: Restarting the XServer might affect the running X application. Please make sure there are no critical X application is running.

Verify that "ngpus" resource has been added to the execution host by executing the command:

```
pbsnodes -av
```

The new resource should be listed and it should be assigned the value entered for the available GPUs entered during installation.

```
resources_available.ngpus = <NGPUS>
```

3.2.3 Install the Interactive Proxy on the Access Web Server

Install the Guacomole proxy server on the machine hosting Access Web to support interactive applications.

Review the system requirements and prerequisites for installation.

1. Login to the machine as root where Access Web is installed.
2. Enter the command:

   ```
   ./AltairRemoteSessionAgent_<Version>_<Build ID>_<YYYYMMDD>_<Timestamp>.bin -i
    console
   ```
3. If you are installing the Interactive Proxy server on a machine hosting either the PBS Professional Server or the MoM you will see the below message, enter `1` and press **ENTER**.

   ```
   Manage Instances
   ----------------
   ->1- Install a new instance
     2- Modify an existing instance
   ```
4. Read the introduction and press **ENTER**.
5. Page through the license agreement by pressing **ENTER** until you are asked to accept its terms and conditions.
6. Accept the license agreement by entering **Y** and pressing **ENTER**.
   Four options are displayed.
7. Enter `4` and press **ENTER**.
8. Enter the location where the binaries are to be installed and press **ENTER**.

   You may choose to install in the default location.
9. Enter the location where the configuration and logs files are to be installed and press **ENTER**.

   You may choose to install in the default location.

   The installation starts. It may take a few minutes for the installation to complete.
10. Press **ENTER** to complete the installation process.

Verify that the Guacamole proxy daemon is installed and running by issuing the following command:

```
service guacd status
```

Configure Access Web After Installation

4

Mandatory configuration steps that must be completed before Access Web can be started.

This chapter covers the following:

- 4.1 Copy Application Definitions and Site Configuration File (p. 28)
- 4.2 Configure the License Server (p. 29)
- 4.3 Log into Access Web (p. 30)
- 4.4 Add a Service Cluster (p. 31)
- 4.5 Onboard an Application Definition (p. 35)

Before you start the Access Web service, you must copy over default application definitions provided by Altair and a corresponding PAS site configuration file.

4.1 Copy Application Definitions and Site Configuration File

Copy default application definitions and a PAS site configuration file.

Default application definitions and a site configuration file are provided to get you up and running quickly. Obtain them through your usual Altair support channels.

1. Login to the machine where PAS is installed.
2. Copy any default application definitions required for your site to the location: `PA_HOME/data/pas/targets/localhost/repository/applications`
3. Copy the default `site-config.xml` file to `PA_HOME/data/pas/targets/localhost/repository`.
4. Edit the `site-config.xml` file.
5. For each application, update the value of the XML element `<Executable>` to the location of the application's executable.

```
<Application id="Abaqus">
   <ApplicationVersions>
       <ApplicationVersion>
          <Option>13.0</Option>
          <Executable>/opt/scripts/abaqus</Executable>
       </ApplicationVersion>
   </ApplicationVersions>
</Application>
```

6. Remove `time_stamp.txt` from `PA_HOME/data/pas/`.

4.2 Configure the License Server

Configure the license server after Access Web installation in console mode.

During the installation Access Web, you are prompted to provide a license server in the format `port@hostname`. If this information is not provided at that time, then the license server must be configured post-installation.

> **Note:** You must have administrative privileges to configure the Access Web license server.

1. Login to the machine as root where Access Web is installed.
2. Open the file `app.properties` located at `PA_HOME/config/license/`.
3. Change the value of *pbsworks.license.server* to the port and hostname of the license server in the format `port@hostname`.

```
Licensed application name
pbsworks.license.application=PBSAccess

#server location for license server
pbsworks.license.server=6200@cntrlicsrv03
```

4. Restart Access Web for these changes to take effect by entering the following command:

```
service pbsworks-pa restart
```

4.3 Log into Access Web

Log into Access Web so that you can submit and monitor jobs.

Review the supported browsers before logging into Access Web.

Before you can submit a job, a service cluster must be added. Only the portal administrator can add service clusters. The first person to login to Access Web after installation is considered the portal administrator.

> **Note:** A localhost cluster is added by default, if Access Web is installed with PBS Application Services (PAS).

1. Enter the URL `https://<hostname>:4443/pbsworks` in the address bar of a supported browser.
   where <hostname> is the IP address or hostname of the machine where Access Web is installed.
   The Access Web login screen is displayed.

   > **Troubleshooting:** Contact an Altair Application Engineer if you are unable to access the login screen.

2. Enter your username and password.
3. Click **Log In.**

If Access Web is not installed with PAS, then you must add a service cluster.

4.4 Add a Service Cluster

Establish a connection to an HPC cluster so that you may begin submitting jobs.

Before you can configure a cluster, you must know the hostname of the PAS Server installed on the PBS Professional headnode.

The first person to login to Access Web after installation is considered the portal administrator. The portal administrator is the only user who can add or delete service clusters. A service cluster must be added before jobs can be submitted to the Workload Manager.

Note: A localhost cluster is added by default, if Access Web is installed with PBS Application Services (PAS).

1. Choose one of the following options:
   - If no service clusters have been configured, click the **Configure one or more services** link.
   - Click and then click **Add**.

   The Add Service Cluster screen is displayed.

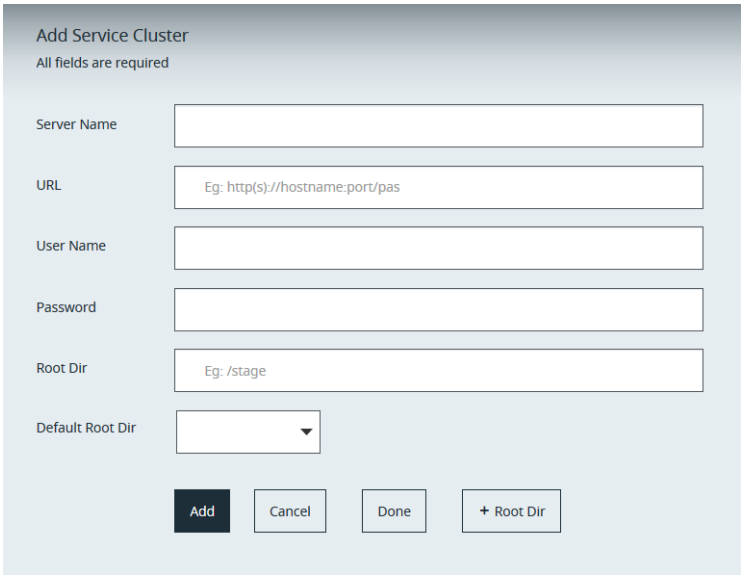

Figure 1: Add Service Cluster

2. For **Server Name**, enter a short name describing the cluster.
3. For **URL**, enter the URL for connecting to the Workload Manager.

   The URL is in the format `https://<hostname>:5243/pas` where <hostname> is the hostname of the machine where the PAS Server is installed (typically the PBS Professional headnode).

   Note: It is recommended to add a cluster using the hostname of the PAS Server rather than the IP address. This prevents connectivity issues from arising when an IP address is changed (DHCP, etc.).

4. For **User Name** and **Password**, enter your login credentials.
   The user name and password must be available in PAS.
5. For **Root Dir**, enter the pathname where user job input and result files are stored.
   Ex: `/home`, `/users`, `/stage`
6. Click + Root Dir if you want to add another **Root Dir** and enter the pathname.

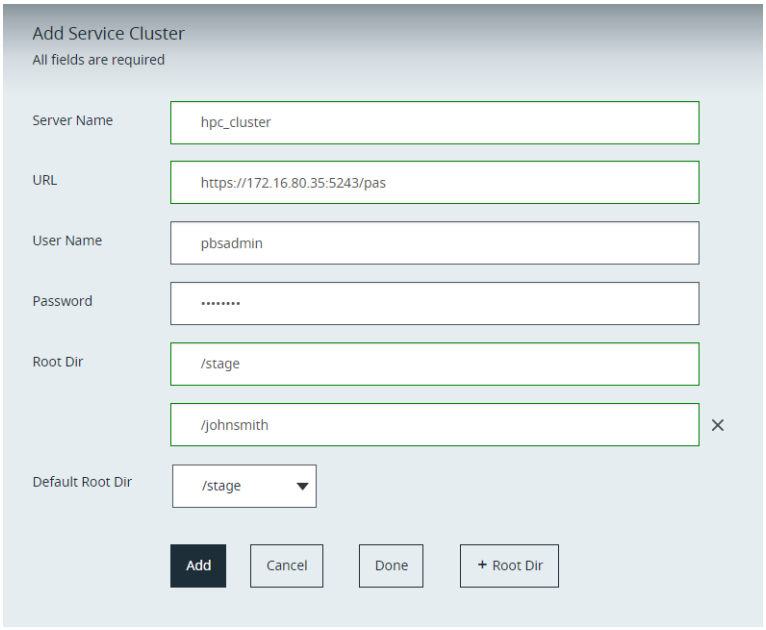

Figure 2: Multiple Root Directory Entry

7. Select the default root directory to be displayed from the **Default Root Dir** drop-down menu.

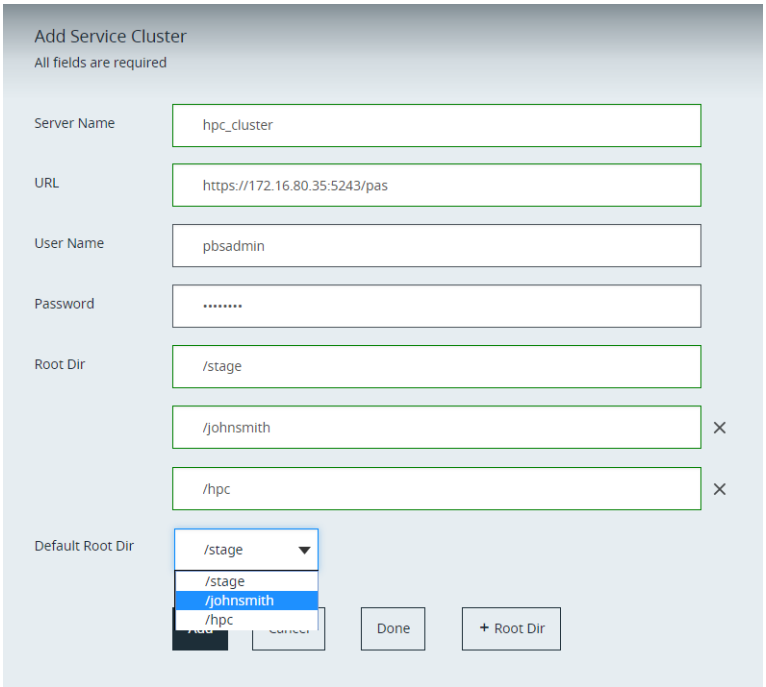

Figure 3: Default Root Directory

8. Click **Add**.
   If the service cluster is added successfully, then a notification is displayed.

   Note: A notification is displayed to all users logged into Access Web when a service cluster gets added, edited, deleted, if it goes down or if it is unreachable.

9. Repeat steps 2 through 8 to add additional service clusters.
10. Click **Done**.
    A list of service clusters that have been added is displayed.

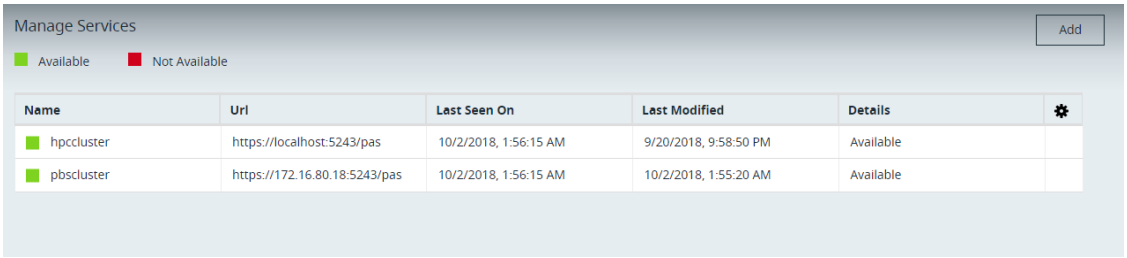

Figure 4: Service Clusters List

The green color next to the service cluster indicates that it is available to use. The red color indicates that the service cluster is not available.

The **Details** column provides the reason when a service cluster is not available. Mouse hover the **Details** column of a service cluster to view the error message.

Figure 5: Service Cluster Details

4.5 Onboard an Application Definition

Port an application definition from a legacy version of PBS Works so that it can be used by Access Web.

Access Web allows onboarding the legacy application definitions in a very easy way and it can be done at any time. The following are the ways we suggest onboarding legacy application definitions:

- Testing your legacy application definitions without making any change – The focus here is to use Job Submission form to submit jobs to a service cluster and verify if Access Web can render and work with your application definition correctly.
- Enriching your application definitions to use Access Web features - Access Web allows you to enhance your application definition by making some modification for using the following features of Access Web:
  - File System Right Click Context Menu Integration
  - Master File Analyzer

4.5.1 Integrate Right Click Context Menu of Access Web

Enrich your legacy application definition to use the right click context menu integration feature of Access Web.

Access Web allows you to enhance your application definition by making some modification to use the right click context menu integration feature. You can use this feature by defining:

- PRIMARY_FILE and QUEUE argument in application definition

  Application definitions must have a PRIMARY_FILE argument defined in the application definition input file that represents the primary input file for the solver. If a legacy application definition calls the primary input file something other than PRIMARY_FILE, then a mapping file must be updated to port the application definition. Additionally, if the legacy application definition contains an application argument that represents the queue to which the job is submitted, the name of the application argument must be QUEUE. If it is not, the mapping file must be updated.
- Update a solver's application definition to link it to a specific file extension.

  Update a solver's application definition to link it to a specific file extension. Access Web links a job input file to a specific application or solver via the file's extension. For example, the solver Optistruct is a structural analysis solver and can process input files with a .fem extension. The association between the file extension and the solver is done through the application definition and must be set up so that Access Web can determine which solvers are available for a file extension.

1. Edit the solver's application input file `app-inp-application.xml`.
2. Link the file extension to the solver by adding the following XML:

   ```
   <ApplicationExtension>file_extension</ApplicationExtension>
   ```

   The below example links a file with the extension of .fem to the Optistruct solver.

   ```
   <ApplicationId>Optistruct</ApplicationId>
   <ApplicationName>Optistruct</ApplicationName>
   <ApplicationExtension>.fem</ApplicationExtension>
   ```
3. Save the application input file.

4. Update the site configuration file `site-config.xml` with the appropriate application information such as versions and policies:

```
<Applications>
    <Application id="Optistruct">
        <ApplicationVersions>
            <ApplicationVersion>
                <Option>11.0</Option>
                <Executable>/opt/hyperworks/11.0/altair/scripts/optistruct</
Executable>
            <ApplicationVersion>

            <ApplicationVersion>
                <Option>12.0</Option>
                <Executable>/opt/hyperworks/12.0/altair/scripts/optistruct</
Executable>
            <ApplicationVersion>
        </ApplicationVersions>
</Application>
</Applications>
```

5. Edit the file `PA_HOME/config/pa/appmapping/applicationmapping.json`.
6. Add the following JSON between the bracket []

```
{
    "serverName": "server-1","version": "13.1","applications":
    [
        {
        "applicationName": "RADIOSS-SMP",
        "primaryFile": "MASTER_FILE",
        "queue": "Queues"
        }
    ]
}
```

7. Change the value of serverName to the name of the server provided when adding the service cluster to Access Web.

```
"serverName": "server-1",
```

8. Change the value of version to the legacy version of PBS Works that you are porting from.

```
"version": "13.1",
```

9. Change the value of applicationName to the name of the application that you want to port.

   Denoted by the XML element `<ApplicationName>` in the legacy application definition. The legacy XML looks like this:

```
<ApplicationName>Optistruct</ApplicationName>
```

   The JSON should look like this:

```
"applicationName": "Optistruct"
```

10. Change the value of primaryFile to the name of the application argument that represents the application input file for the solver.

    Denoted by the XML element `<Name>` in the legacy application definition. The legacy XML looks like this:

```
<ArgumentChoice>
    <ArgumentFileName>
        <Name>MASTER_FILE</Name>
        <Description>Select your Optistruct Master file.</Description>
        <DisplayName>Master File</DisplayName>
        <InputRequired>true</InputRequired>
```

```
    </ArgumentFileName>
</ArgumentChoice>
```

The JSON should look like this:

```
"primaryFile": "MASTER_FILE"
```

11. Change the value of queue to the name of the application argument that represents the queue to which the job is submitted.

    Denoted by the XML element `<Name>` in the legacy application definition. The legacy XML looks like this:

```
<ArgumentChoice>
    <ArgumentStringEnumerated>
        <Name>BATCH_QUEUE</Name>
        <Description>Select the batch queue you would like to submit to.</
Description>
        <DisplayName>Batch Queue</DisplayName>
        <InputRequired>false</InputRequired>
        <Option>workq</Option>
        <Option>testq</Option>
        <DefaultValue>workq</DefaultValue>
    </ArgumentStringEnumerated>
</ArgumentChoice>
```

    The JSON should look like this:

```
"queue": "BATCH_QUEUE"
```

12. Add additional applications by repeating previous step 9-11 making sure that when you add the next application to the JSON mapping file you separate the applications using a comma.

```
"applications":
[
    {
        "applicationName": "ShellScript",
        "primaryFile": "JOB_SCRIPT"
    },
    {
        "applicationName": "Optistruct",
        "primaryFile": "MASTER"
    }
]
```

13. Save the file.
14. Copy your legacy application definitions to the PAS repository.
15. Restart PAS by issuing the command:

```
service pbsworks-pa restart
```

    The following notification is displayed to any users that are logged into Access Web:

```
There is a change in configuration data.  Application will reload.
```

    Once Access Web reloads, the new application definition is available.

16. Right-click on a job input file that has the file extension that was just added to the application definition.
17. Verify that the correct solver and job profiles are being displayed in the context menu.

4.5.2 Master File Analyzer

The Master File Analyzer identifies the list of include files from the input or master file that is required to submit a job.

This feature must be enabled in the application definition to dynamically identify the include files. The default application definitions with Master File Analyzer capabilities are provided to get you up and running quickly. Obtain them through your usual Altair support channels.

You can refer to `Diving Into Application Definitions` guide and the samples provided to convert or write application definitions to support Master File Analyzer.

Upgrade and Downgrade Access Web

5

Upgrade Access Web previous versions 2018.2 or 2018.3 to 2018.4 and downgrade Access Web 2018.4.

This chapter covers the following:

- 5.1 Upgrade Access Web (p. 40)
- 5.2 Upgrade PBS Application Services (p. 42)
- 5.3 Downgrade Access Web (p. 44)
- 5.4 Downgrade PBS Application Services (p. 45)

The upgrade and downgrade of Access Web is not supported by the installer. You have to manually perform the steps to upgrade or downgrade.

The Access Web 2018.4 supports only PAS 2018.4. So, if you are upgrading Access Web, then you have to upgrade PAS.

5.1 Upgrade Access Web

Upgrade Access Web previous versions 2018.2 or 2018.3 to 2018.4.

- Review System Requirement and Prerequisites of Access Web
- For installing Access Web 2018.4, refer to Install Access Web.
- Refer Upgrading PAS to upgrade PAS.

> **Note:** Do not uninstall Access Web previous version.

A binary or executable of Access Web 2018.4 needs to be downloaded or obtained using your usual Altair support channels.

Perform these steps to upgrade:

- Access Web and PAS (single and different machine setup) 2018.3 to 2018.4
- Access Web 2018.2 to 2018.4

  > **Note:** To upgrade PAS 2018.2 to 2018.4, follow the steps in Upgrade PBS Application Services topic.

The upgrade script will perform the following:

- Migrate application definitions, the site configuration file, and the server configuration file
- Migrate the user preference and configuration related files from Home and Exec folders of previous versions 2018.2 or 2018.3 to 2018.4
- Automatically start Access Web 2018.4

1. Stop Access Web previous version using the following command:
   ```
   service pbsworks-pa stop
   ```
2. Create a backup of `/var/.com.zerog.registry.xml` file.
3. Remove `/var/.com.zerog.registry.xml` file.
4. Create a backup of `/etc/pbsworks-pa.conf` as `/etc/pbsworks-pa.conf.<PreviousVersion>`

   > **Note:** Upgrade is supported for 2018.2 or 2018.3 versions.

5. Install Access Web 2018.4.

   > **Note:** Do not start the Access Web 2018.4.

   > **Note:** The Step 6 is not required if you are upgrading from 2018.3 to 2018.4.

6. Edit `/etc/pbsworks-pa.conf` file and make sure you have set `PA_SERVER=1` and `PA_PAS=0`.
7. Navigate to `PA_EXEC/init/`.
8. Execute the `pa-upgrade.sh` script as follows:
   ```
   pa-upgrade.sh <PreviousVersion_PA_HOME> <PreviousVersion_PA_EXEC>
   ```

where <PreviousVersion_PA_HOME> is the Access Web 2018.2 or 2018.3 home folder and <PreviousVersion_PA_EXEC> is the Access Web 2018.2 or 2018.3 execution folder.

After upgrading, the Access Web 2018.4 will be up and the following message will be displayed:

```
"Access Web upgraded successfully"
```

9. Enter the URL `https://<hostname>:4443/pbsworks` in the address bar of a supported browser.

   where <hostname> is the hostname of the machine where Access Web is installed.

   The Access Web login screen is displayed.

   > **Troubleshooting:** Contact an Altair Application Engineer if you are unable to access the login screen.

10. Enter your username and password.
11. Click **Log In.**

To upgrade PAS 2018.2 to 2018.4, follow the steps in Upgrade PBS Application Services.

To configure Access Web, refer to Configuring Access Web After Installation.

5.2 Upgrade PBS Application Services

Upgrade PAS previous versions 2018.2 to 2018.4.

- Review System Requirement and Prerequisites of Access Web
- For installing Access Web 2018.4, refer to Install Access Web.
- Refer to Upgrading Access Web to upgrade Access Web.

> **Note:** Do not uninstall PAS previous version 2018.2.

A binary or executable of Access Web 2018.4 needs to be downloaded or obtained using your usual Altair support channels.

Perform these steps to upgrade PBS Application Services 2018.2 to 2018.4.

The upgrade script will perform the following:

- Migrate application definitions, the site configuration file, and the server configuration file
- Migrate the user preference and configuration related files from Home and Exec folders of 2018.2 to 2018.4
- Automatically start Access Web 2018.4

1. Stop PAS 2018.2 using the following command:
   ```
   service pas stop
   ```
2. Create a backup of `/var/.com.zerog.registry.xml` file.
3. Remove `/var/.com.zerog.registry.xml` file.
4. Create a backup of `/etc/pas.conf` as `/etc/pas.conf.PreviousVersion`
5. Install Access Web 2018.4.

   > **Note:** Do not start the Access Web 2018.4.

6. Edit `/etc/pbsworks-pa.conf` file and make sure you have set `PA_SERVER=0` and `PA_PAS=1`.
7. Navigate to `PA_EXEC/init/`.
8. Execute the `pa-upgrade.sh` script as follows:
   ```
   pa-upgrade.sh <2018.2_PBSWORKS_HOME> <2018.2_PBSWORKS__EXEC>
   ```
   where <2018.2_PBSWORKS__HOME> is the PAS 2018.2 home folder and <2018.2_PBSWORKS__EXEC> is the PAS 2018.2 execution folder.

   After upgrading, the Access Web 2018.4 will be up and the following message will be displayed:
   ```
   "Access Web upgraded successfully"
   ```
9. To verify PAS instance, enter the URL `https://<hostname>:5243/pas` in the address bar of a supported browser.

   where <hostname> is the hostname of the machine where PAS is installed.

   The browser will display the PAS information.

   For example, if you type https://<hostname>:5243/pas the following information will be displayed:

PBSWorks Application Services

Version: 2018.4.0
Build: 20181209

REST Services

Troubleshooting: Contact an Altair Application Engineer if the browser does not provide the information.

To configure Access Web, refer to Configuring Access Web After Installation.

5.3 Downgrade Access Web

Downgrade Access Web 2018.4 to Access Web 2018.2 or 2018.3.

Perform these steps to downgrade:

- Access Web and PAS (single and different machine setup) 2018.4 to 2018.3
- Access Web 2018.4 to 2018.2

> **Note:** To downgrade PAS 2018.4 to 2018.2, follow the steps in Downgrade PBS Application Services topic.

1. Stop Access Web 2018.4 using the following command:
   ```
   service pbsworks-pa stop
   ```
2. Create a backup of `/etc/pbsworks-pa.conf` as `/etc/pbsworks-pa.conf.2018.4`
3. Rename `/etc/pbsworks-pa.conf.PreviousVersion` to `/etc/pbsworks-pa.conf`
4. Copy `PreviousVersion_PA_EXEC/init/pbsworks-pa` script to `/etc/init.d/`
   You are successfully downgraded to Access Web 2018.2 or 2018.3.
5. Start Access Web using the following command:
   ```
   /etc/init.d/pbsworks-pa start
   ```
6. Enter the URL `https://<hostname>:4443/pbsworks` in the address bar of a supported browser.
   where <hostname> is the IP address or hostname of the machine where Access Web is installed.
   The Access Web login screen is displayed.

   > **Troubleshooting:** Contact an Altair Application Engineer if you are unable to access the login screen.
7. Enter your username and password.
8. Click **Log In.**

To downgrade PAS 2018.4 to 2018.2, follow the steps in Downgrade PBS Application Services.

5.4 Downgrade PBS Application Services

Downgrade PAS 2018.4 to PAS 2018.2.

Perform these steps to downgrade PBS Application Services 2018.4 to 2018.2.

1. Stop PAS 2018.4 using the following command:
   ```
   service pbsworks-pa stop
   ```
2. Start PAS using the following command:
   ```
   /etc/init.d/pas start
   ```
3. To verify PAS instance, enter the URL `http://<hostname>:17084/pas` in the address bar of a supported browser.

   where <hostname> is the hostname of the machine where PAS is installed.

   The browser will display the PAS information.

   > **Troubleshooting:** Contact an Altair Application Engineer if the browser does not provide the information.

Uninstall Access Web and Remote Session Component

6

Instructions for uninstalling previous version of Access Web and Remote Session component.

This chapter covers the following:

- 6.1 Uninstall Access Web (p. 47)
- 6.2 Uninstall Remote Session (p. 48)

6.1 Uninstall Access Web

Remove a previous version of Access Web.

You must stop Access Web before uninstalling. For more information about stopping Access Web, see Access Web Service Commands.

> **Note:** Uninstalling Access Web will not remove `PA_HOME`.

1. Login as root to the machine where Access Web is installed.
2. Navigate to the `/opt/altair/pbsworks/2018.4/access/Altair_Access_installation` directory.
3. Execute the uninstall script by entering the following command:

   ```
   ./Change_Altair_Access_Installation -i console
   ```

   The command must contain spaces with escape characters.
4. Follow the instructions provided by the uninstaller.

6.2 Uninstall Remote Session

Unconfigure PBS Professional and PBS Application Services and uninstall remote session components to disable the remote session capabilities of Access Web.

The uninstalling Remote Session Components must be run multiple times across several machines and must be performed in the following sequence:

1. Unconfigure the PBS Professional and PAS in PBS Professional headnode and in the machine where PBS Application Services is installed. This will remove:
   - Deletes the interactive queue
   - Unconfigure GPUs as a custom resource
   - Only Glxsphere which is installed as part PAS configuration
2. Uninstall Remote Session Components on all PBS MoMs and proxy software

6.2.1 Unconfigure PBS Professional and PBS Application Services

Unconfigure PBS Professional and PAS before uninstalling Remote Session Components.

You must stop Access Web before uninstalling. For more information about stopping Access Web, see Access Web Service Commands.

> **CAUTION:** It is advisable that you run the installer when critical jobs are not running.

Run the installer to unconfigure PBS Professional and PAS on the PBS Professional headnode and in the machine where PAS is installed.

The components that will be removed are:

- Deletes the interactive queue
- Unconfigure GPUs as a custom resource
- Only Glxsphere which is installed as part PAS configuration

1. Login to the machine as root where the PBS Professional Server and PAS is installed.
2. Navigate to the folder where you have the Remote Session Agent installer.
3. Enter the command:
   ```
   ./AltairRemoteSessionAgent_<Version>_<Build ID>_<YYYYMMDD>_<Timestamp>.bin -i
    console
   ```
4. If you are installing the remote session component for the first time, then you will see the below message, enter `1` and press **ENTER**.
   ```
   Manage Instances
   ----------------
   ->1- Install a new instance
     2- Modify an existing instance
   ```
5. Read the introduction and press `ENTER`.

6. Page through the license agreement by pressing `ENTER` until you are asked to accept its terms and conditions.
7. Accept the license agreement by entering `Y` and pressing `ENTER`.
   Four Options are displayed.
8. Enter `3` to unconfigure the PBS Professional and PAS servers and press **ENTER**.
9. PBS Professional and PAS is restarted during the installation process, choose whether you want to proceed:
   - Choose **No** to exit and run the installer at a more suitable time.
   - Choose **Yes** to run the installer.

   CAUTION: It is advisable that you run the installer when critical jobs are not running.

Perform Uninstalling Remote Session components.

6.2.2 Uninstall Remote Session Components

Uninstall previous version of remote session components.

You must stop Access Web before uninstalling. For more information about stopping Access Web, see Access Web Service Commands.

CAUTION: It is advisable that you run the installer when critical jobs are not running.

If it is a distributed deployment, then perform the following:

- Login to each of the PBS MoM and uninstall Remote Session Components
- Login to the machine hosting Access Web and uninstall Remote Session Components to remove the proxy

1. Login as root to the machine where Access Web is installed.
2. Navigate to the `/opt/altair/pbsworks/2018.4/dmagent/_AltairRemoteSessionAgent_Installer_<Version>_installation` directory.
3. Execute the uninstall script by entering the following command:
   ```
   ./Change\ AltairRemoteSessionAgent_Installer_<Version>\ Installation -i console
   ```
   The command must contain spaces with escape characters.
4. Follow the instructions provided by the uninstaller.

   Note: This will uninstall the Remote Session component and Interactive Proxy only if you installed Access Web on the same machine as the PBS Professional Server.

Access Web Service Commands 7

Commands for starting, stopping, restarting and checking the status of Access Web.

The below commands should be executed by the root as defined in `/etc/pbsworks-pa.conf`.

Start Access Web
```
service pbsworks-pa start
```

Stop Access Web
```
service pbsworks-pa stop
```

Restart Access Web
```
service pbsworks-pa restart
```

Determine the Status of Access Web
```
service pbsworks-pa status
```

Manage Service Clusters

8

Add, edit, and delete service clusters.

This chapter covers the following:

- 8.1 Add a Service Cluster (p. 52)
- 8.2 Edit a Service Cluster (p. 56)
- 8.3 Delete a Service Cluster (p. 58)

Only the portal administrator can add, edit, and remove service clusters. Regular users are unable to access the Manage Services page.

8.1 Add a Service Cluster

Establish a connection to an HPC cluster so that you may begin submitting jobs.

Before you can configure a cluster, you must know the hostname of the PAS Server installed on the PBS Professional headnode.

The first person to login to Access Web after installation is considered the portal administrator. The portal administrator is the only user who can add or delete service clusters. A service cluster must be added before jobs can be submitted to the Workload Manager.

Note: A localhost cluster is added by default, if Access Web is installed with PBS Application Services (PAS).

1. Choose one of the following options:
   - If no service clusters have been configured, click the **Configure one or more services** link.
   - Click and then click **Add**.

   The Add Service Cluster screen is displayed.

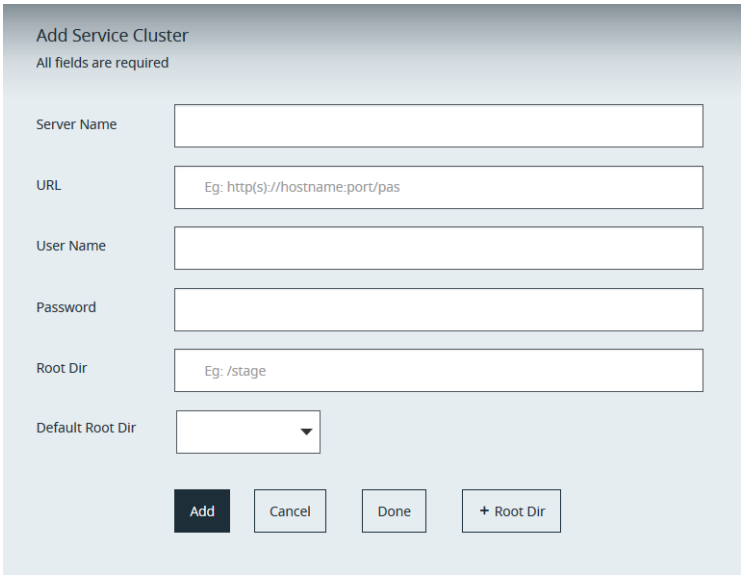

Figure 6: Add Service Cluster

2. For **Server Name**, enter a short name describing the cluster.
3. For **URL**, enter the URL for connecting to the Workload Manager.

   The URL is in the format `https://<hostname>:5243/pas` where <hostname> is the hostname of the machine where the PAS Server is installed (typically the PBS Professional headnode).

   Note: It is recommended to add a cluster using the hostname of the PAS Server rather than the IP address. This prevents connectivity issues from arising when an IP address is changed (DHCP, etc.).

4. For **User Name** and **Password**, enter your login credentials.
   The user name and password must be available in PAS.
5. For **Root Dir**, enter the pathname where user job input and result files are stored.
   Ex: `/home`, `/users`, `/stage`
6. Click + Root Dir if you want to add another **Root Dir** and enter the pathname.

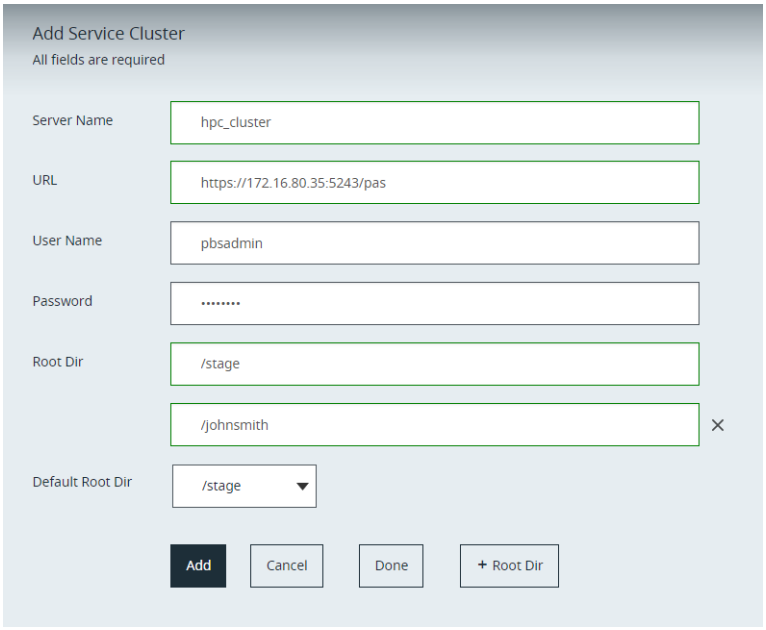

Figure 7: Multiple Root Directory Entry

7. Select the default root directory to be displayed from the **Default Root Dir** drop-down menu.

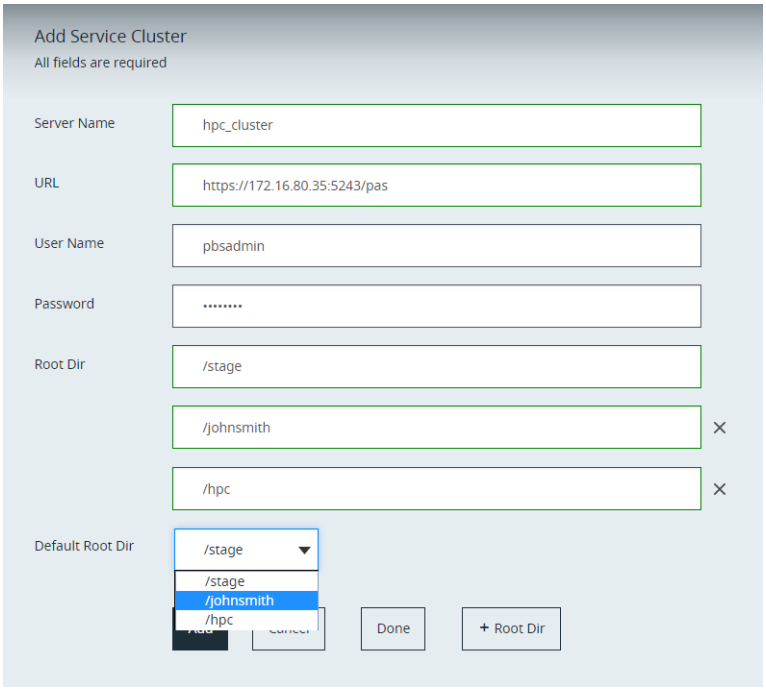

Figure 8: Default Root Directory

8. Click **Add**.
   If the service cluster is added successfully, then a notification is displayed.

   Note: A notification is displayed to all users logged into Access Web when a service cluster gets added, edited, deleted, if it goes down or if it is unreachable.

9. Repeat steps 2 through 8 to add additional service clusters.
10. Click **Done**.
    A list of service clusters that have been added is displayed.

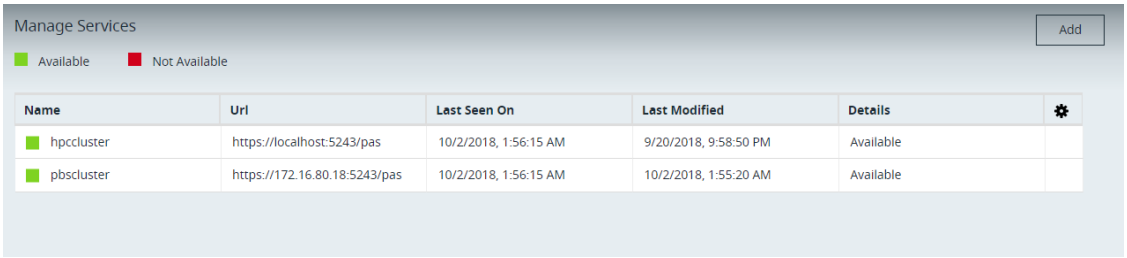

Figure 9: Service Clusters List

The green color next to the service cluster indicates that it is available to use. The red color indicates that the service cluster is not available.

The **Details** column provides the reason when a service cluster is not available. Mouse hover the **Details** column of a service cluster to view the error message.

Figure 10: Service Cluster Details

8.2 Edit a Service Cluster

Update a service cluster password or root directory so that you can continue to submit your jobs.

Note: Only the portal administrator can edit a service cluster.

1. Click .
   A list of service clusters that have been previously added is displayed.

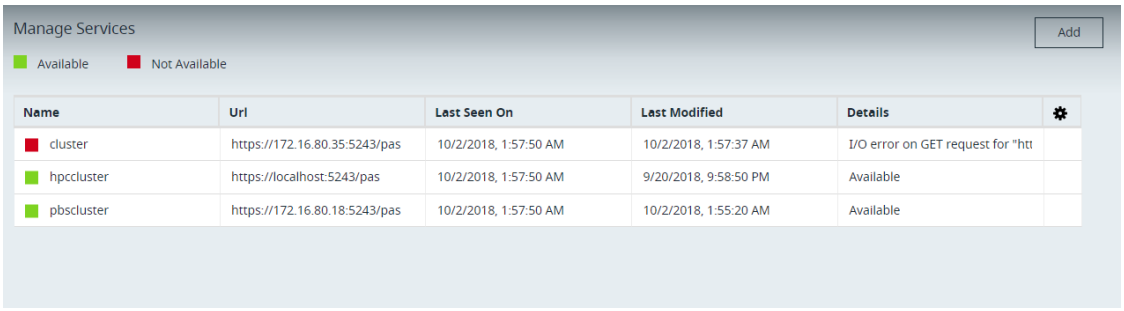

Figure 11: Manage Services

2. Right-click the cluster that you want to edit.
3. Click **Edit** from the context menu.

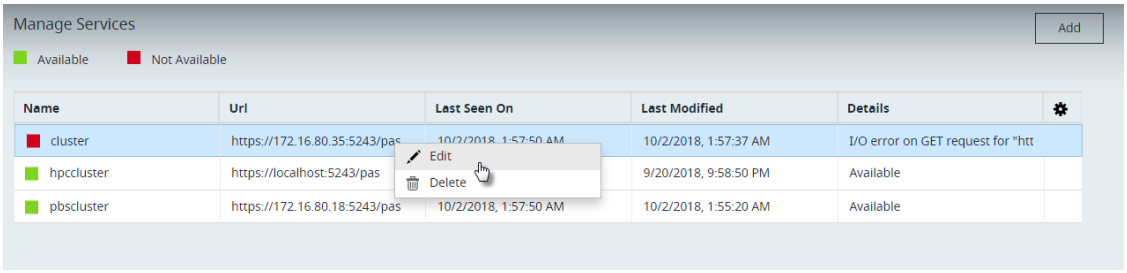

Figure 12: Cluster Edit Option

The Edit Service Cluster screen is displayed.

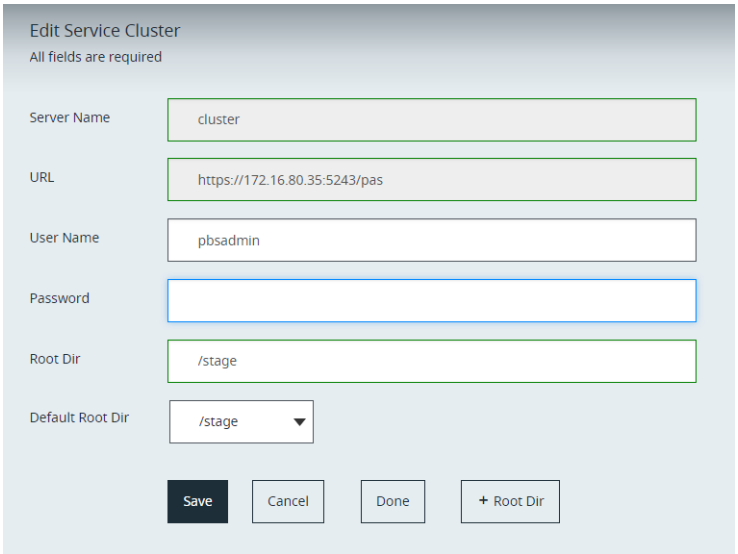

Figure 13: Edit Service Cluster

4. Update the service cluster information.
5. Click **Save**.

> **Note:** A notification is displayed to all users logged into Access Web when a service cluster gets added, edited, deleted, if it goes down or if it is unreachable.

6. Click **Done**.

8.3 Delete a Service Cluster

Remove a service cluster when you no longer want to submit and manage jobs on that cluster.

> **Note:** Only the portal administrator can delete a service cluster.

1. Click .
   A list of service clusters that have been previously added is displayed.

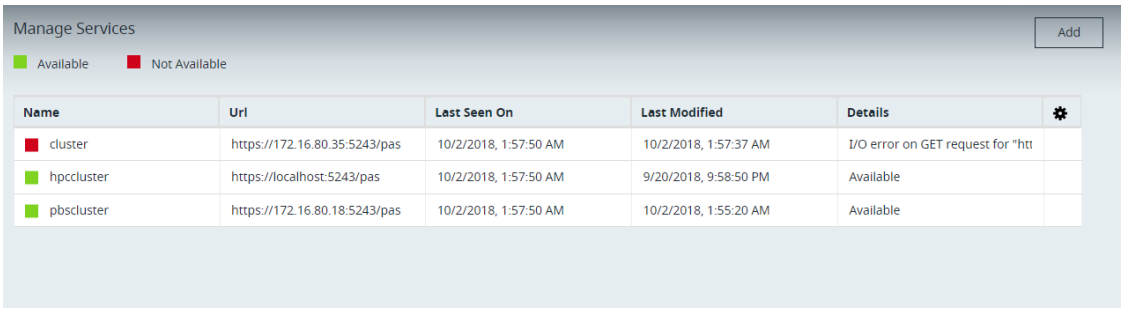

Figure 14: Manage Services

2. Right-click the cluster that you want to remove.
3. Click **Delete** from the context menu.

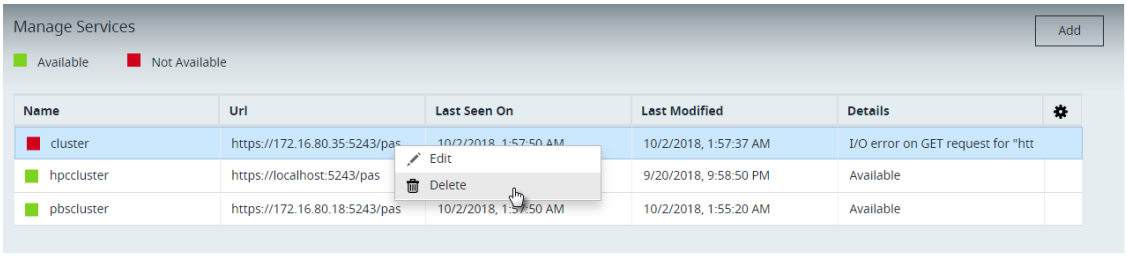

Figure 15: Server Cluster Delete

4. Click **Yes**.
   The files and jobs from this cluster will no longer be accessible.

> **Note:** A notification is displayed to all users logged into Access Web when a service cluster gets added, edited, deleted, if it goes down or if it is unreachable.

5. Click **Done**.

Advanced Configurations

9

Advanced configurations for Access Web and its services.

This chapter covers the following:

- 9.1 Configure the Access Web Component (p. 60)
- 9.2 Configure PBS Application Services (p. 79)
- 9.3 Configure the Remote Session Component (p. 93)
- 9.4 Configure Results Visualization Service (p. 97)

9.1 Configure the Access Web Component

Configurations required for Access Web component.

9.1.1 Change Port Numbers

Change the default port numbers used by Access Web.

You must stop Access Web before changing the port number. For more information about stopping Access Web, see Access Web Service Commands.

The Access Web installer has the auto-port detection logic in place and ports will be picked up by each service within the specified range. Refer to ports used by Access Web for more information.

To allocate specific port to each of the service, then follow the steps mentioned in the topics.

Changing the Gateway Port Number

Change the port that the Gateway service listens on.

You must stop Access Web before changing the port number. For more information about stopping Access Web, see Access Web Service Commands.

The gateway port number has to be updated in the files `nginx.conf` and `rm_servers.xml`.

> **Note:** The default gateway port number is 4443 and the port range is 4443 to 4542.

1. Login to the machine where Access Web is installed as root.
2. Navigate to `PA_HOME/config/api_gateway/`.
3. Edit the `nginx.conf` file and update the default value of 4443 of *`listen`* of *`server`*:

```
server {
        listen       4443;
        server_name  localhost;
        add_header X-Frame-Options "SAMEORIGIN";
```

4. Navigate to `PA_HOME/config/resultmanager/`.
5. Edit the `rm_servers.xml` file and update the default value of 4443 in `PAServerURL`.

```
<PAServerURL>https://localhost:4443</PAServerURL>
```

6. Start Access Web by entering the command:

```
service pbsworks-pa start
```

Changing the Web Server Port Number

Change the port that the Access Web server listens on.

You must stop Access Web before changing the port number. For more information about stopping Access Web, see Access Web Service Commands.

The web server port number has to be updated in the following files:

- `server.xml`
- `nginx.conf`
- `dmrest.properties`
- `dmrest.properties.template`
- `app.properties`
- `ServiceRegistry.json`

Note: The default web server port number is 4543 and the port range is 4543 to 4642.

1. Login to the machine where Access Web is installed as root.
2. Navigate to `PA_HOME/config/pa/tomcat/conf/`.
3. Edit the `server.xml` file and update the port number.

   Note: Search for the Connector port and replace 4543 with the new port number.
4. Navigate to `PA_HOME/config/api_gateway/`.
5. Edit the `nginx.conf` file and update the port number.

   ```
   env STORAGE_SERVICE_PORT=4543;

   upstream pbsaccess {
       server localhost:4543;
   }
   ```
6. Navigate to `PA_HOME/config/displaymanager/`.
7. Edit the `dmrest.properties` file and update the port number.

   ```
   pbsaccess.storage.service.host=https://localhost:4543/storage
   ```
8. Edit the `dmrest.properties.template` file and update the port number.

   ```
   pbsaccess.storage.service.host=https://localhost:4543/storage
   ```
9. Navigate to `PA_HOME/config/shared/`.
10. Edit the `app.properties` file and update the port number.

    ```
    pbsworks.ams.url = https://localhost:4543/AAService/aaservice/authn/oauth2
    ```
11. Edit the `ServiceRegistry.json` file and update the port number.

    ```
    {"service":
    [{"name":"ams","host":"localhost","port":"4543","service_name":"AAService","scheme":"https
    ```
12. Start Access Web by entering the command:

    ```
    service pbsworks-pa start
    ```

Changing the Postgres Port Number

Change the port that Postgres listen on.

You must stop Access Web before changing the port number. For more information about stopping Access Web, see Access Web Service Commands.

> **CAUTION:** Changing the Postgres port requires the removal and recreation of the Postgres database. The script that performs this work also removes all log files located at `PA_HOME/logs`. Additionally, this script allows the Service User who owns the Postgres database and the files in PA_HOME and PA_EXEC to be changed. If you do not want to change the Service User, then provide the username of the current Service User when executing the script. The current Service User can be determined by viewing the contents of `/etc/pbsworks-pa.conf`.

The Postgres database port number has to be updated in the files `configure.sh` and `app.properties`

> **Note:** The default Postgres database port number is 4643 and the port range is 4643 to 4742.

1. Login to the machine where Access Web is installed as root.
2. Navigate to `PA_EXEC/database/scripts/`.
3. Edit the `configure.sh` file and update the port number.
   ```
   PG_PORT=4643;
   ```
4. Navigate to `PA_HOME/config/shared/`.
5. Edit the `app.properties` file and update the port number.
   ```
   spring.datasource.url=jdbc:postgresql://localhost:4643/pbsworks
   ```
6. Navigate to `PA_EXEC/init/`
7. Run the command:
   ```
   ./reconfigure-pa.sh
   ```

Changing the Message Broker Port Number

Change the port that the message broker (ActiveMQ) listens on.

You must stop Access Web before changing the port number. For more information about stopping Access Web, see Access Web Service Commands.

The Message Broker port number has to be updated in the files `message-app.properties`, `env` and `activemq.xml`.

> **Note:** The default Message Broker port number is 4743 and the port range is 4743 to 4842.

1. Login to the machine where Access Web is installed as root.
2. Navigate to `PA_HOME/config/shared/`.
3. Edit the `message-app.properties` file and update the port number.
   ```
   pbsworks.messaging.broker.url=tcp://localhost:4743
   ```

4. Navigate to `PA_EXEC/shared/thirdparty/apache/activemq/bin/`.
5. Edit the `env` file and update the port number.
   ```
   ACTIVEMQ_QUEUEMANAGERURL="--amqurl tcp://localhost:4743"
   ```
6. Navigate to `PA_EXEC/shared/thirdparty/apache/activemq/conf/`
7. Edit the `activemq.xml` file and update the port number.
   ```
   <transportConnector name="openwire"
   uri="tcp://127.0.0.1:4743?
   maximumConnections=1000&wireFormat.maxFrameSize=104857600>
   ```
8. Start Access Web by entering the command:
   ```
   service pbsworks-pa start
   ```

Changing the Remote Session Proxy Port Number

Change the port that the Remote Session Proxy (GUACD) listens on.

You must stop Access Web before changing the port number. For more information about stopping Access Web, see Access Web Service Commands.

The Interactive Proxy port number has to be updated in the following files:

- `guacd.conf`
- `guacamole.properties`
- `guacamole.properties.template`

> **Note:** The default Interactive Proxy port number is 5443 and the port range is 5443 to 5542.

1. Login to the machine where Access Web is installed as root.
2. Navigate to `/etc/guacamole/`.
3. Edit the `guacd.conf` file and update the port number.
   ```
   bind_port = 5443
   ```
4. Navigate to `PA_HOME/config/displaymanager/`.
5. Edit the `guacamole.properties` file and update the port number.
   ```
   guacd-port:     5443
   ```
6. Edit the `guacamole.properties.template` file and update the port number.
   ```
   guacd-port:     5443
   ```
7. Start Access Web by entering the command:
   ```
   service pbsworks-pa start
   ```
8. Restart the Interactive Proxy by entering the command:
   ```
   /etc/init.d/guacd start
   ```

Changing the Remote Session Web Server Port Number

Change the port that the Remote Session Web Server listens on.

You must stop Access Web before changing the port number. For more information about stopping Access Web, see Access Web Service Commands.

The Interactive Application web server port number has to be updated in the following files:

- `server.xml`
- `nginx.conf`
- `guacamole.properties`
- `guacamole.properties.template`

> **Note:** The default Interactive Application web server port number is 4843 and the port range is 4843 to 4942.

1. Login to the machine where Access Web is installed as root.
2. Navigate to `PA_HOME/config/displaymanager/tomcat/conf/`.
3. Edit the `server.xml` file and update the port number.

   > **Note:** Search for the Connector port and replace 4843 with the new port number.

4. Navigate to `PA_HOME/config/api_gateway/`.
5. Edit the `nginx.conf` file and update the port number.

   ```
   upstream pbsaccess {
       server localhost:4843;
   }
   ```

6. Navigate to `PA_HOME/config/displaymanager/`.
7. Edit the `guacamole.properties` file and update the port number.

   ```
   dm-host: https://localhost:4843/displaymanager
   ```

8. Edit the `guacamole.properties.template` file and update the port number.

   ```
   dm-host: https://localhost:4843/displaymanager
   ```

9. Start Access Web by entering the command:

   ```
   service pbsworks-pa start
   ```

Changing the Remote Session Job Update Port Number

Change the port that the Remote Session Job Update listens on.

You must stop Access Web before changing the port number. For more information about stopping Access Web, see Access Web Service Commands.

The Interactive Application Job Update port number has to be updated in the files `dmrest.properties` and `dmrest.properties.template`.

> **Note:** The default Interactive Application Job Update port number is 4943 and the port range is 4943 to 5042.

1. Login to the machine where Access Web is installed as root.
2. Navigate to `PA_HOME/config/displaymanager/`.
3. Edit the `dmrest.properties` file and update the port number.
   ```
   jobsub.monitor.port=4943
   ```
4. Edit the `dmrest.properties.template` file and update the port number.
   ```
   jobsub.monitor.port=4943
   ```
5. Start Access Web by entering the command:
   ```
   service pbsworks-pa start
   ```

Changing the Result Manager Services Port Number

Change the port that the Result Manager Services listens on.

You must stop Access Web before changing the port number. For more information about stopping Access Web, see Access Web Service Commands.

The Result Manager Services port number has to be updated in the files `server.xml` and `nginx.conf`.

> **Note:** The default Result Manager Services port number is 5043 and the port range is 5043 to 5142.

1. Login to the machine where Access Web is installed as root.
2. Navigate to `PA_HOME/config/resultmanager/tomcat/conf/`.
3. Edit the `server.xml` file and update the port number.

   > **Note:** Search for the Connector port and replace 5043 with the new port number.

4. Navigate to `PA_HOME/config/api_gateway/`.
5. Edit the `nginx.conf` file and update the port number.
   ```
   upstream resultmanager {
       server localhost:5043;
   }
   ```
6. Start Access Web by entering the command:
   ```
   service pbsworks-pa start
   ```

Changing the Result Core Services Port Number

Change the port that the Result Core Services listens on.

You must stop Access Web before changing the port number. For more information about stopping Access Web, see Access Web Service Commands.

The Result Core Services port number has to be updated in the files `server.xml`, `rm.servers.xml`, and `resultmanager.conf`.

> **Note:** The default Result Core Services port number is 5143 and the port range is 5143 to 5242.

1. Login to the machine where Access Web is installed as root.
2. Navigate to `PA_HOME/config/resultservice/tomcat/conf/`.
3. Edit the `server.xml` file and update the port number.

   > **Note:** Search for the Connector port and replace 5143 with the new port number.

4. Navigate to `PA_HOME/config/resultmanager/`.
5. Edit the `rm.servers.xml` file and update the port number in `RVSServerURL`.

   ```
   <RVSServerURL>https://localhost:5143</RVSServerURL>
   ```

6. Navigate to `PA_HOME/config/api_gateway/default.d/`.
7. Edit the `resultmanager.conf` file and update the port number:

   ```
   proxy_pass https://localhost:5143/resultservice;
   ```

8. Start Access Web by entering the command:

   ```
   service pbsworks-pa start
   ```

Changing the PBS Application Services Port Number

Change the port that the PAS listens on.

You must stop Access Web before changing the port number. For more information about stopping Access Web, see Access Web Service Commands.

The PAS port number has to be updated in the files `server.xml` and `nginx.conf`.

> **Note:** The default PAS port number is 5243 and the port range is 5243 to 5342.

1. Login to the machine where Access Web is installed as root.
2. Navigate to `PA_HOME/config/pas/tomcat/conf/`.
3. Edit the `server.xml` file and update the port number.

   > **Note:** Search for the Connector port and replace 5243 with the new port number.

4. Navigate to `PA_HOME/config/api_gateway/`
5. Edit the `nginx.conf` file and update the port number.

   ```
   env PAS_SERVICE_PORT=5243;
   ```

6. Start Access Web by entering the command:

   ```
   service pbsworks-pa start
   ```

Changing the Job Profile Services Port Number

Change the port that the Job Profile Services listens on.

You must stop Access Web before changing the port number. For more information about stopping Access Web, see Access Web Service Commands.

The job profile services port number has to be updated in the files `server.xml` and `nginx.conf`.

> **Note:** The default job profile services port number is 5343 and the port range is 5343 to 5442.

1. Login to the machine where Access Web is installed as root.
2. Navigate to `PA_HOME/config/jobprofiles/tomcat/conf/`.
3. Edit the `server.xml` file and update the port number.

   > **Note:** Search for the Connector port and replace 5343 with the new port number.

4. Navigate to `PA_HOME/config/api_gateway/`
5. Edit the `nginx.conf` file and update the port number.

   ```
   upstream jobprofiles {
           server localhost:5343;
   }
   ```

6. Start Access Web by entering the command:

   ```
   service pbsworks-pa start
   ```

9.1.2 Set the Double-Click Delay Time

Change the delay time required between two consecutive clicks for a double-click.

The default delay time between two consecutive clicks is set to 500ms (500 millisecond)

> **Note:** You must have administrative privileges to change the double-click delay time.

1. Login to the machine as root where Access Web is installed.
2. Navigate to `PA_HOME/config/pa/`.
3. Open the `configuration.json` file and change the value of the `doubleClickDelay`.

9.1.3 Change the Database Password

Change the database password to encrypted text.

If the Postgres database password is changed (via passwd, yppasswd, etc.), then Access Web will need to be updated with the new password.

The Postgres database password is stored in the `app.properties` file. The location of the file for a typical installation of Access Web is: `PA_HOME/config/shared/`

1. Login to the machine as root where Access Web is installed.
2. Navigate to `PA_EXEC/shared/thirdparty/postgresql/bin/`.
3. Execute the following command:

   ```
   ./psql -p <DB_PORT> -d pbsworks -U <DB_USER>)
   ```

   DB_PORT is database port, and DB_USER is database username

   For example,

   ```
   ./psql -p 4643 -d pbsworks -U pbsworks
   ```
4. Enter the existing password. The default password is `postgres`.
5. Enter `\password` in Postgres prompt.
6. Enter the new password.
7. Navigate to `PA_EXEC/init/`.
8. Execute the following command:

   ```
   ./pa-encrypt.sh
   ```
9. Enter the new password given in step 7.

   This command will output the password in its encrypted format.
10. Navigate to `PA_HOME/config/shared/`
11. Edit the `app.properties` file and update the value of *`spring.datasource.password`* to the encrypted format of the new password.
12. Restart Access Web by entering the command:

    ```
    service pbsworks-pa restart
    ```

9.1.4 Configure Default File Viewer

Configure default file viewer to open the file based on the file extension.

By default, the file extension .out, .Log, .stat, .rad, .fem will open in the file viewer by double-clicking the files.

> **Note:** You must have administrative privileges to change the double-click delay time.

1. Login to the machine as root where Access Web is installed.
2. Navigate to `PA_HOME/config/pa`/.
3. Open the `nativeviewer.json` file.
4. Add the file extensions under `ApplicationFileExtension` value.

```
"ApplicationFileExtension": {
    "type": "array",
    "items": {
        "type": "string"
            },
    "value": [".sh",
            ".fem",
            ".py",
            ".env",
            ".txt",
            ".Log",
```

```
            ".stat",
            ".rad",
            ".out"
            ],
    "Displayable": false
}
```

The file extensions mentioned in `ApplicationFileExtension` value will open with the default file viewer.

> **Note:** If you double-click or open a file with unknown file extension, then an application list dialog box is displayed to choose the desired application to view the file.

9.1.5 Configure Default Columns in Job List View

Configure the job properties columns that are displayed in the job list view.

By default, the job properties columns displayed in the job list view in Jobs tab are Job ID, Job Name, Job State, Creation Time and User Name. You can add or remove the `defaultGridColumns` property value in the `jobpropertiesmap.json` file.

> **Note:** You must have administrative privileges to change the default job properties value.

1. Login to the machine as root where Access Web is installed.
2. Navigate to `PA_HOME/config/pa/`.
3. Open the `jobpropertiesmap.json` file.
4. Update the `defaultGridColumns` value.

   ```
   "defaultGridColumns": ["jobId", "jobName", "jobState", "creationTime", "userName"]
   ```

   The updated job properties value will be displayed in the job list view in Jobs tab.

9.1.6 Change the Maximum File Upload Size

Change the default file upload size based on your requirement.

You must stop Access Web before changing the maximum file upload size. For more information about stopping Access Web, see Access Web Service Commands.

1. Login to the machine as root where Access Web is installed.
2. Navigate to `PA_HOME/config/api_gateway/`.
3. Open the `nginx.conf` file.
4. Update the `client_max_body_size` value in MB.

   ```
   #set max file upload size to 4GB
       client_max_body_size 4096m;
   ```

5. Start Access Web using the following command:

   ```
   service pbsworks-pa start
   ```

9.1.7 Change the File Opening behavior of a Remote Session Application

Change the file opening behavior of a remote session application from cross mounted file system to non-shared file system.

The default application definition provided with Access Web is configured such that the selected file is not copied to the execution node. The remote session will open the file in execution node with the assumption that the file name and file path is available in execution node. This option is the cross mounted file system where the file system is available on the execution node and head node.

In the case of non-shared file system, there is no shared file system between the execution node and head node. The Access Web will copy the file to execution node job directory and remote application will open with the copied file from job directory.

The default interactive application definition provided with Access Web provides the flexibility to work in both the environment. By default, the sample application definition is configured to work in cross mount file system environment.

Perform on the following choices to change the behavior in Access Web:

- Uncheck the “Run from job directory” field from Job Submission Form while submitting.

  Note: The “Run from job directory” field is displayed only if you select “All Fields” option in Job Submission form. The behavior will be changed for that session only.

- Update the Application Definition Input file and change the default value of "Run from job directory" to true.

  Note: This change will be implemented for all the sessions.

9.1.8 Configure Notifications for a Job State Change

Configure email notifications for a job state change.

Users of Access Web can configure email notifications when a job's state changes, including who will receive the email notification, and when the email will be sent. Currently, Access Web supports sending email notifications for the following job state changes:

- job is aborted
- job begins execution
- job finishes execution

However, the email will not be sent unless the application associated with the job has been configured accordingly. This is done through the application definition.

Each Access Web user must set their email preferences through the Access Web application.

1. Click .
2. Select **Profile**.

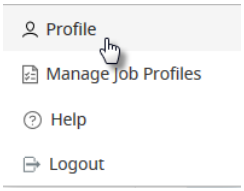

Figure 16: Profile

Access WebPreferences is displayed.

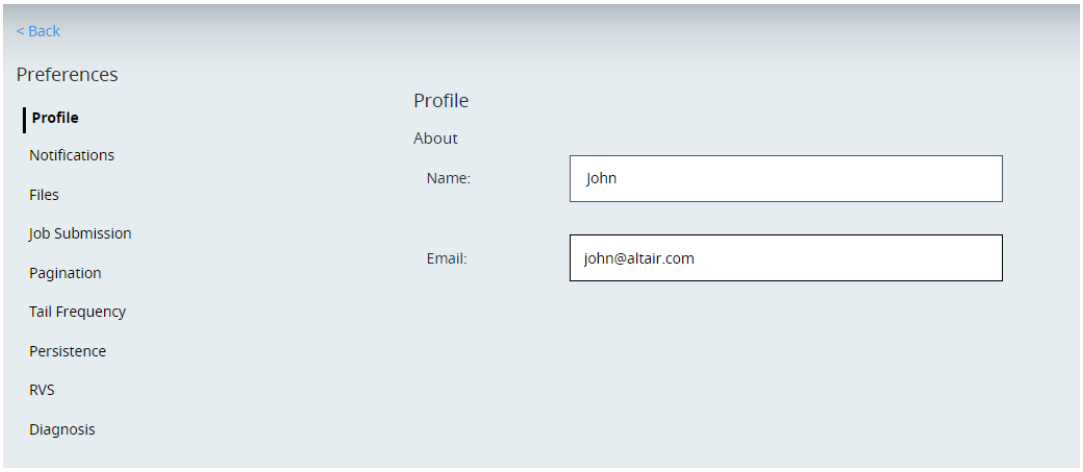

Figure 17: Preferences

3. Double-click the Email field to enter additional email ID and press **Enter**.
   You can enter multiple email IDs separated by semi-colon (;).

   Tip: The job state change notifications will be sent to this email ID.

4. Click **Job Submission** in the left panel.
   The Job Submission panel is displayed.

   Note: By default, all the options of job submission notification is disabled.

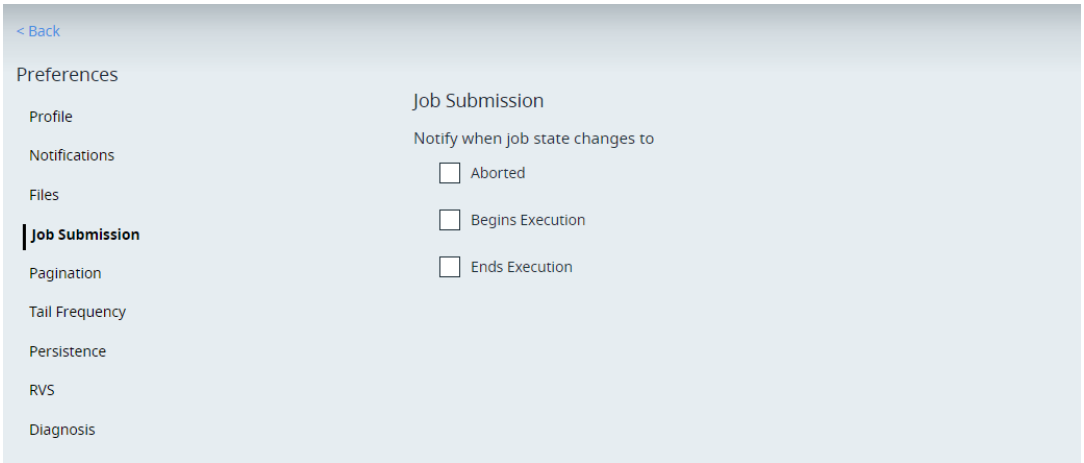

Figure 18: Job Submission

Note: Click < Back to go back to the previous page.

5. Check any or all of the options in the **Notify when job state changes to** list to indicate when the email will be sent.
6. The application definition must be modified to support the email notification. This is accomplished by modifying the application definition submittime script, `presubmit.py`, with the following lines of code:

```
import re

''' Mail Options '''

if userInputs['MAIL_USERS'].replace(';',',')]:
    job.attr_mail_list = userInputs['MAIL_USERS']

if userInputs['MAIL_POINTS']:
    mail_points = userInputs['MAIL_POINTS']

    if re.match(r"[abe]", mail_points):
        job.attr_mail_options = mail_points
```

This code captures the mail preferences entered through Access Web, and sets the job's mail options, so that PBS Professional knows to send an email when a particular job state is reached.

9.1.9 Disable to View all Jobs

You can configure to restrict the user from viewing all jobs.

By default, users can view all jobs.

1. Login to the machine as root where Access Web is installed.
2. Navigate to `PA_HOME/config/pa/`.
3. Open the `configuration.json` file.
4. Change the value of `restrictOthersJobs` to `true`.

> **Note:** By default the value is set to `false`.

9.1.10 Add a Generic Action for a PAS server

Perform a generic action on a job.

A generic action is similar to an application definition's application action. The main difference is that an application action is associated with particular application, while generic actions are simple python scripts that run on a PAS server irrespective of the job or application.

A JSON file is used to define generic actions. You have to specify the PAS server name and the generic actions that has to be performed on that server. For a typical installation of Access Web, the generic action JSON file is `genericactions.json` and it is located at `PA_HOME/config/pa/`

> **Note:** Refer to https://jsonlint.com/ to validate the JSON file and refer to https://jsonformatter.org/ to format the code.

An example of a generic action for the server "hpccluster", stored in a JSON file called `genericactions.json`. A job can be selected from the Job Monitoring page of Access Web, and an action can be executed. This generic action allows a command and command options to be entered. Access Web will then run the command.

```
{
 "hpccluster": [{
   "Name": "qstat",
   "DisplayName": "Qstat of Job",
   "Description": "Qstat of Job",
   "ScriptLocation": "/stage/GenericActions/qstat.py",
   "Arguments": {
    "jobid": {
     "type": "string",
     "DisplayName": "Job ID"
    },
    "required": ["jobid"]
   }
  },
  {
   "Name": "Tracejob",
   "DisplayName": "Tracejob of Job",
   "Description": "Tracejob of Job",
   "ScriptLocation": "/stage/GenericActions/tracejob.py",
   "Arguments": {
    "jobid": {
     "type": "string",
     "DisplayName": "Job ID"
    },
    "required": ["jobid"]
   }
  },
  {
   "Name": "Distributed Tracejob",
   "DisplayName": "Distributed Tracejob of Job",
   "Description": "Distributed Tracejob of Job",
   "ScriptLocation": "/stage/GenericActions/dtj.py",
```

```
  "Arguments": {
   "jobid": {
    "type": "string",
    "DisplayName": "Job ID"
   },
   "required": ["jobid"]
  }
 },
 {
  "Name": "Node Status",
  "DisplayName": "Node status of cluster",
  "Description": "Node status of cluster",
  "ScriptLocation": "/stage/GenericActions/nodestatus.py",
  "Arguments": {
   "jobid": {
    "type": "string",
    "DisplayName": "Job ID"
   },
   "required": [ ]
  }
 },
 {
  "Name": "Queue Hold",
  "DisplayName": "Queue hold a job",
  "Description": "Queue hold a job",
  "ScriptLocation": "/stage/GenericActions/qhold.py",
  "Arguments": {
   "jobid": {
    "type": "string",
    "DisplayName": "Job ID"
   },
   "required": ["jobid"]
  }
 },
 {
  "Name": "Queue Release",
  "DisplayName": "Queue release a job",
  "Description": "Queue release a job",
  "ScriptLocation": "/stage/GenericActions/qrls.py",
  "Arguments": {
   "jobid": {
    "type": "string",
    "DisplayName": "Job ID"
   },
   "required": ["jobid"]
  }
 },
 {
  "Name": "Suspend a job ",
  "DisplayName": "Suspend a job",
  "Description": "Suspend a job",
  "ScriptLocation": "/stage/GenericActions/qsig.py",
  "Arguments": {
   "jobid": {
    "type": "string",
    "DisplayName": "Job ID"
   },
   "required": ["jobid"]
  }
 },
 {
  "Name": "Delete jobs ",
  "DisplayName": "Delete job(s)",
```

```
   "Description": "Delete job(s)",
   "ScriptLocation": "/stage/GenericActions/qdel.py",
   "Arguments": {
    "jobid": {
     "type": "string",
     "DisplayName": "Job ID"
    },
    "required": ["jobid"]
   }
  }
 ]
}
```

Define a Generic Action

Define a generic action JSON file.

The following JSON elements provide a way of identifying and describing the generic action:

- Name - is the internal name of the generic action
- DisplayName - is the name of the generic action that will be displayed to the user
- Description - is the description of the generic action
- ScriptLocation - is the location of the execution script

  ```
  "Name": "qstat",
  "DisplayName": "Qstat of Job",
  "Description": "Qstat of Job",
  "ScriptLocation": "/stage/qstat.py",
  ```

- Arguments - defines arguments (input fields) that a user will enter prior to executing the generic action.

  ```
  "Arguments":
              {
                  "jobid":
                  {
                      "type": "string",
                      "DisplayName": "Job ID"
                  },
                  "required": ["jobid"]
  ```

  The following elements define an argument:

  - type - defines the type of the argument
  - DisplayName - is the name of the argument that will be displayed
  - required - indicates a required field

Define a Generic Action Execution Script

Define a generic action execution script file.

The generic action script is a python script that is responsible for executing the generic action, using the information entered by the user (defined by the generic action JSON file). You can directly edit this script, taking full advantage of Python to add further inspection and complexity to the execution of the generic action. In this example, the python script will create a subprocess to execute the command entered by the user.

```
import subprocess, sys, os

def execcmd(cmd):
            try:
                        p = subprocess.Popen(cmd, shell=True, stdout=subprocess.PIPE,
stderr=subprocess.STDOUT)
                        data = ""
                        for line in p.stdout.readlines():
                                    data = data + line
            except:
                        errTtype = sys.exc_type
                        try:
                                    errName = errTtype.__name__
                        except AttributeError:
                                    errName = errType
                        data = "Error: " + str(errName) + " --- " + str(sys.exc_value)
            return data
cmd = "/opt/pbs/bin/qstat -fx " + sys.argv[2]
print (cmd)
print execcmd(cmd)
sys.stdout.flush()
```

9.1.11 Map a File Extensions to an Icon

Map icons to a file extension in the Access Web application.

The file icons are mapped to a file extension in the JSON file, `fileextensions.json` and the location of this file is at `PA_HOME/config/pa/`

The file icon image file must be placed at the location `PA_HOME/config/pa/fileicons/`

This is an example of a file extension and icon that has been mapped in `fileextensions.json` located at `PA_HOME/config/pa/`:

```
{
    "fileextension":
    [
        {
            "name": "pdf",
            "displayText": "Acrobat",
            "icon": "icon_pdf.png"
        },
        {
            "name": "zip",
            "displayText": "Archive",
            "icon": "icon_zip.png"
        }
        {
            "name": "h3d",
            "displayText": "H3D",
            "icon": "hvpctrl-32.gif"
        }
    [
}
```

The XML attributes of the file extension mapping file is as follows:

- name - The file extension.
- displayText - Access Web will display this as the files type.

- icon - Icon image file located at `PA_HOME/config/pa/fileicons/`

This information is used by Access Web to display customized file types and icons when viewing remote files.

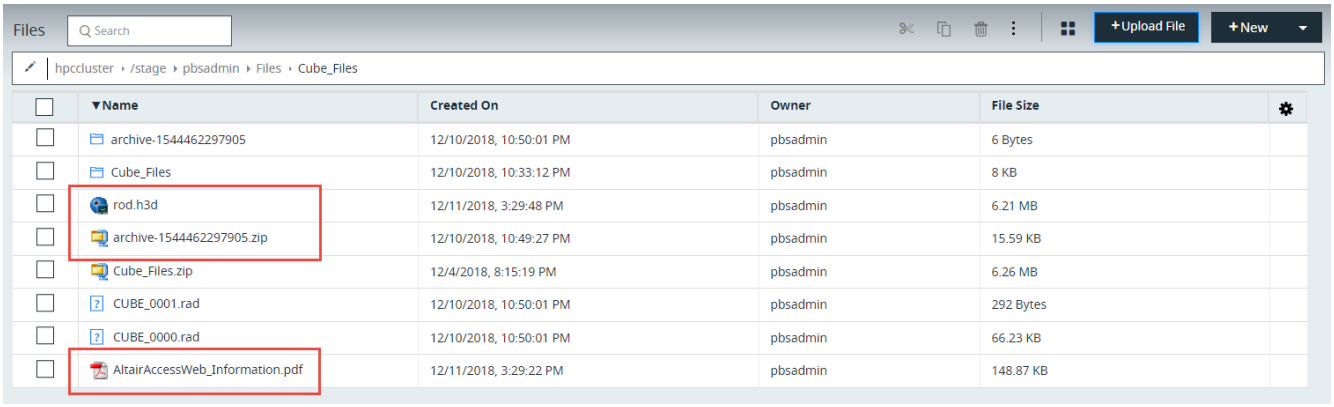

Figure 19: File Icon Mapping

9.1.12 Set Maximum Page Size for Files

Access Web allows the administrator to set the maximum file size (in bytes) for displaying a file in a single page.

The default page size set for the files is 5000 bytes.

Note: You must have administrative privileges to change the file page size.

1. Login to the machine as root where Access Web is installed.
2. Navigate to `PA_HOME/config/pa/`.
3. Open the `configuration.json` file and change the value of the `filechunksize`.

```
"filechunksize": 5000
```

9.1.13 Install and Configure SSL Certificate

Install and configure SSL certificate on the server to secure communication.

If you do not have a valid domain certificate for your site, you need to create a Certificate Signing Request (CSR) and order your certificate.

1. You should have received a `your_domain_name.pem` file from Certificate Authority which contains both your primary certificate and the intermediate certificate. If you have that `.pem` file, you can skip to Step 4.
2. Download the intermediate (`Intermediate.crt`) and your primary certificate (`your_domain_name.crt`) files.
3. Copy these files, along with the `.key` file you generated when creating the CSR, to the directory on the server where the certificate and key files are kept.

4. Concatenate the primary certificate file (`your_domain_name.crt`) and the intermediate certificate file (`Intermediate.crt`) into a single `.pem` file by running the following command:

   ```
   cat your_domain_name.crt Intermediate.crt >> bundle.crt
   ```

5. Navigate to `PA_HOME/config/api_gateway/`
6. Open the `nginx.conf` file and update the following line:

   ```
   server
   {
       ssl    on;
       ssl_certificate    /etc/ssl/your_domain_name.pem; (or bundle.crt)
       ssl_certificate_key    /etc/ssl/your_domain_name.key;
   }
   ```

   - `ssl_certificate` should be your primary certificate combined with the intermediate certificate (`your_domain_name.crt`).
   - `ssl_certificate_key` should be the `.key` file generated when you created the CSR.

7. Restart Access Web by entering the following command:

   ```
   service pbsworks-pa restart
   ```

9.2 Configure PBS Application Services

Information about configuration files, verification of your PAS installation, post-installation configuration and tasks.

9.2.1 Enable PBS Job History

Enable the PBS Professional job history status.

PAS does not automatically set the PBS Professional *job_history_enable* attribute to true. Therefore, to enable job history use the PBS Professional command: `qmgr -c 'set server job_history_enable=True'`

9.2.2 Verify the Installation of PBS Application Services

Verify PAS installation and server status.

1. Open any of the supported browsers.
2. Enter PAS URL (https://<hostname>:<port>/pas)

   Note: The valid PAS port is 5243.

   The browser will display the PAS information.

   For example, if you type https://<hostname>:5243/pas the following information will be displayed:

9.2.3 Configure System Zip Utility

Configure PAS to use the system compression utility.

By default, PAS uses zip 3.0 utility to compress the file.

Configure System Zip Utility to Compress by Updating Python Script

Enable the system zip utility to compress the files by updating the Python Script.

1. As root, open the `FileCompress.py` file located at `PA_EXEC/pas/scripts/`
2. Before `zipcmd = [zip_full_path_normalized] + ['-rq'] + [compressionSpeed]` line, add the following: `zip_full_path_normalized = '/usr/bin/zip'`

   The `/usr/bin/zip` is the path where the system zip utility is located.

   > **Note:** If you do not specify the system zip utility path, then PAS zip utility will be used to compress the files.

Configure System Zip Utility to Compress without Updating Python Script

Enable the system zip utility to compress the files without updating the python script.

1. As root, navigate to `PA_EXEC/pas/bin/Linux-x86_64`
2. Rename the existing zip utility to `zip_backup`.
3. Create a soft link for zip with the system zip utility: `ln -s /usr/bin/zip zip`

   The `/usr/bin/zip` mentioned, is the path where the system zip utility is located.

   > **Note:** Restore the permission of the changed zip utility.

9.2.4 Configure System Unzip Utility

Configure PAS to use the system uncompression utility.

By default, PAS uses unzip 6.0 utility to uncompress the file.

Configure System Unzip Utility to Uncompress by Updating Python Script

Enable the system unzip utility to uncompress the files by updating the Python Script.

1. As root, open the `FileUncompress.py` file located at `PA_EXEC/pas/scripts/`
2. Before `files_list_cmd = [unzip_full_path_normalized] + ['-lqq'] + [archive]` line, add the following: `unzip_full_path_normalized = '/usr/bin/unzip'`

   The `/usr/bin/unzip` is the path where the system unzip utility is located.

   Note: If you do not specify the system unzip utility path, then PAS unzip utility will be used to uncompress the files.

Configure System Unzip Utility to Uncompress without Updating Python Script

Enable the system unzip utility to uncompress the files without updating the python script.

1. As root, navigate to `PA_EXEC/pas/bin/Linux-x86_64`
2. Rename the existing unzip utility to `unzip_backup`.
3. Create a soft link for unzip with the system unzip utility: `ln -s /usr/bin/unzip unzip`

   The `/usr/bin/unzip` mentioned, is the path where the system unzip utility is located.

   Note: Restore the permission of the changed unzip utility.

9.2.5 Configurable Parameters

PAS stores its configuration data in a file called `server.conf`.

The default location of this file is: `PA_HOME/config/pas/conf`

All parameters are stored in `key=value` pairs. Spaces and tabs are preserved (as spaces) and providing quotes around values is optional. As spaces are preserved, do not use `key = value` (notice the space surrounding the =) unless spaces are intentional. Empty lines beginning with a comment symbol (i.e., #) are ignored.

Standard Configuration Parameters

Standard configuration parameters and its default values in PAS Server.

The following table describes the standard configuration parameters:

Table 2: Standard Configuration Parameters

Parameter	Description	Default Value
LOGGING_CONFIG_FILE_RELATIV	The PAS logging configuration file (Log4J) which is used to configure logging levels. Since the PAS uses Log4J as its logging and tracing API, you can access additional information on Log4J via the Apache Log4J website at http://logging.apache.org/log4j	`server-log.xml`
ADMIN_USER	The PAS administration account (for example, pbsadmin) that has complete control over the PAS Server. This account is required for any remote PAS administration tasks. On Linux, the administration account can be a non-privileged user account that has been granted manager status for PBS Professional, and can be changed after installation.	pbsadmin
LOGIN_MODULE_NAME	The PAS authentication module. PAS uses the Java Authentication and Authorization Service (JAAS) as its authentication abstraction layer.	UnixLogin
STAGE_ROOT	The PAS file staging location. This directory must exist prior to the installation of PAS. The staging directory is where the necessary files are transferred after job submission (via a client for example, PBS Portals or Access Web), but prior to the portal submitting the job to PAS for transfer to PBS Professional for execution. When PBS Professional selects the	`/stage`

Parameter	Description	Default Value
	job for execution, the files are stagedin to the execution host from the staging directory. Upon job completion, any files the job wants to retain are staged-out by PBS Professional to the staging directory, and the client retrieves them from there when it is ready to do so. Customer written portals may choose to do the same, but are not required to do so. The value of STAGE_ROOT can be set to point to the user's home directory by using the variable `$USER_HOME` (for example: `$USER_HOME/stage`) after installation.	
FILE_TRANSFER_PROTOCOL	The PAS file transfer mechanism.	http
FILE_EXPIRATION_TIME	The PAS stage area file expiration time. That is, the age in days after which data is removed from the staging directory (as defined by the STAGE_ROOT parameter) and its subdirectories with the exception of the profiles directory. Files in the profiles directory are never removed by PAS.	14
ZIP_COMPRESSION_SPEED	Compression speed for the default PAS and system compression utility. The compression utility uses this value to compress the files. The zip compression value -1 being fastest and -9 being slowest.	-1
MAX_LIST_FILES_COUNT	The number of files to be returned by the FileList API. Allows sites to impose limits on	1000

Parameter	Description	Default Value
	how many files are returned to improve the response time of Access Web remote file operations.	
BUFFER_SIZE	Maximum buffer size for file operations in bytes.	65536

Advanced Configuration Parameters

Optional parameters available to you that are appropriate for advanced configurations in PAS Server.

The following table describes the advanced configuration parameters:

Table 3: Advanced Configuration Parameters

Parameter	Description	Default Value
PYTHON_PATH	The PAS Python binary. PAS requires that each PBS execution host provide a standard Python binary. The full path to your systems Python binary can be changed here. The PBS Professional daemon pbs_mom will look for a binary called pbs_python in `PBS_EXEC/bin`. If using custom actions or any other operations that involve the pbs_mom accessing this pbs_python (aka python binary) there may be problems if the python libraries are not located in the standard locations relative to `PBS_EXEC/bin/pbs_python`. If Python is installed elsewhere a link to (Linux)the actual python binary must exist as `PBS_EXEC/bin/pbs_python`.	`$PBS_EXEC/bin/pbs_python`

Parameter	Description	Default Value
PBS_DATA_REFRESH_TIME	Time interval in minutes to fetch PBS related data like qmgr, queues etc	15
SSH_LOGGER_ENABLE	SSH Log enabling	False
STAGE_ROOT_TEMP_DIR	Temporary folder for zip/download API	`/tmp` The default path can be changed to `$STAGE_ROOT/tmp`
JSON_SCHEMA_VALIDATION	JSON Schema Validation	False

9.2.6 Configure PBS Resources

Configure PBS resources in PAS.

You can install PAS on a separate machine from the PBS Professional Server as a failover server. The PBS Professional *`flatuid`* attribute must be set to True. This attribute specifies whether, for each user, the username at the submission host must be the same as the one at the Server host. The username at the Server host must always be the same as the username at the execution host. When *`flatuid`* is set to True, the Server assumes that UserA@host1 is the same as UserA@host2. Therefore, if *`flatuid`* is True, UserA@host2 can operate on UserA@host1's job. Follow these steps to set *`flatuid`* to True. These steps must be performed either by root or a PBS manager.

1. At the command line, enter the command: `qmgr -c "print server"`
2. If the attribute *`flatuid`* is equal to False or is not set (you do not see it in the output from the print server command), then set the value to True by issuing the command: `qmgr -c "set server flatuid=true"`

9.2.7 Configuring PAS Server Logging Behavior

Configure logging behavior for certain functional aspects of PAS, such as file operations or job submission to troubleshoot errors.

The PAS logging behavior can be defined in `server-log.xml` file located at `PA_HOME/config/pas/conf/`.

The functional areas are defined by the *`<category>`* element and the existing categories in the `server-log.xml` are useful as a basic configuration, are more course-grained, and provide the default PAS

logging. The default logging level is set to `info` providing informational messages that highlight the progress of the application at a coarse-grained level.

To configure the level of logging, edit the `server-log.xml` file and set the *<priority value>* attribute of a particular *<category>* to one of the following values:

- `off` - has the highest possible rank and is intended to turn off logging.
- `fatal` - very severe error events that will presumably lead the application to abort.
- `error` - error events that might still allow the application to continue running.
- `warn` - potentially harmful situations.
- `info` - informational messages highlighting the progress of the application at a coarse-grained level.
- `debug` - fine-grained informational events that are most useful to debug an application.
- `trace` - finer-grained informational events than the DEBUG.
- `all` - the lowest possible rank and is intended to turn on all logging.

```
<category name="category">
    <priority value="<logging_level>" />
</category>
```

Remote File operations, Job submission, and Job Status

Configure logging behavior for remote file operations, job submission, and job status.

The logging for this category is relevant for the communication with the EIFL server.

1. Navigate to `PA_HOME/config/conf/`.
2. Update `server-log.xml` with the following XML:

   ```
   <category name=" com.altair.gw.aif.cli.implementation.BasicCommandsImpl">
       <priority value="debug" />
   </category>
   ```
3. Update `server-log.xml` to get detailed troubleshooting information for the job status:

   ```
   <category name="com.altair.gw.aif.pbs.ifl.implementations.PbsEiflWs">
       <priority value="debug" />
   </category>
   ```

   The following are examples of the type of messages that will be logged if debugging is enabled for the job status operation:

   Table 4: Logging Messages for Communication with the EIFL Server

Event	Message
Before a web service API call to the EIFL server	“Success getting eifl server port <port>”
After an EIFL web service API call to the EIFL server	“Success returning from eifl.waitExit(), port is <port>”

Event	Message
For a job status request, before the web service API call to the EIFL server	"qstatJobs(): Success getting eifl server port <port>"
After an EIFL web service API call to the EIFL server	"qstatJobs(): Success returning from eifl.waitExit(), port is <port>"
For the get detailed job status operation, if an error occurs, the error code will be logged as returned by PBS	"PbsEiflWs.java getErrorCode(): pbs error code is <pbsErrorCodeStr>"

File Operations (local and remote)

Configure logging behavior for file operations (local and remote).

1. Navigate to `PA_HOME/config/conf/`.
2. Update `server-log.xml` to change the `priority value`:

```
<category name="com.altair.gw.aif.fileop.implementation.FileOperations">
    <priority value="debug" />
</category>

<category name="com.altair.gw.aif.fileop.implementation.FileOperationsHelper">
    <priority value="debug" />
</category>
```

Job Submission

Configure logging behavior for job submission.

1. Navigate to `PA_HOME/config/conf/`.
2. Update `server-log.xml` to get detailed logging information about user inputs and to see how long it takes to execute a submission request by adding the following XML:

```
<category name=" com.altair.gw.aif.rest. RESTJobsPortImpl">
    <priority value="debug" />
</category>
```

3. Update `server-log.xml` to get information about the process of creating PBS job attributes from user inputs and the application definition by adding the following XML:

```
<category name=" com.altair.gw.aif.rest.util.PASNextGenJobUtils">
    <priority value="debug" />
</category>
```

   PAS supports multiple adapters to communicate with the workload manager.
4. If the SSH adapter is enabled for the communication with the PBS cluster, add the following XML to the `serverlog.xml` file to troubleshoot job submission:

```
<category name=" com.altair.gw.aif.ssh.implementation. SSHImplementation”>
    <priority value="debug" />
</category>
```

The following are examples of the type of messages that will be logged if debugging is enabled for this class:

Table 5: Logging Messages for Job Submission

Event	Message
Before job submission, PAS will record the job name and the resources requested.	“PbsJobFactory.getPbsJob():job name = <job name>, resourceList to String: <job resources>
Once the job is submitted, PAS will record the job id.	“JobSubmit success. Job id: <jobId>”
For job status requests, PAS will log any empty responses.	"HpcpPbsAdapter.jobList(): qstat is empty"

Job Submission and Status

Configure logging behavior for job submission and status.

1. Navigate to `PA_HOME/config/conf/`.
2. Update `server-log.xml` with the following XML:

```
<category name="com.altair.gw.aif.pbs.ifl.responses.JobsStatus_Response">
    <priority value="debug" />
</category>

<category name=" com.altair.gw.aif.pbs.ifl.responses.PBSObjectStatus">
    <priority value="debug" />
</category>
```

Enabling a debugging level for the “PbsJobsPortImpl” class also provides the ability to determine the IP address of a client request for PAS job submission services.

For each job submission API call, a log record is created having the API name, requesting user name, and remote host. For example,

```
Entered getJobs(). User: <username>, client host: '<clientHost>'
```

Application Definitions

Configure logging behavior for application definitions.

1. Navigate to `PA_HOME/config/conf/`.

2. Update `server-log.xml` to get detailed logging information about application definitions by adding the following XML:

```
<category name="com.altair.gw.aif.converter">
    <priority value="debug" />
</category>
```

Dynamic Application Refresh Script

Configure logging behavior for dynamic application refresh script.

1. Navigate to `PA_HOME/config/conf/`.
2. Update `server-log.xml` to get detailed logging information about dynamic application refresh script by adding the following XML:

```
<category name="com.altair.gw.aif.applications.utils.DynamicApplicationBuilder">
    <priority value="debug" />
</category>
```

During the Job Status Retrieval

Configure logging behavior to troubleshoot errors during the job status retrieval.

1. Navigate to `PA_HOME/config/conf/`.
2. Update `server-log.xml` to get user information and the total time of the PAS server operation by adding the following XML:

```
<category name="com.altair.gw.aif.rest. RESTJobsPortImpl">
    <priority value="debug" />
</category>
```

3. To get job details information coming from the Workload Manager to PAS and details about response from PBS, add the following XML to the `server-log.xml` file:

```
<category name="com.altair.gw.aif.pbs.ifl.responses.JobsStatus_Response">
    <priority value="debug" />
</category>

<category name=" com.altair.gw.aif.pbs.ifl.responses.PBSObjectStatus">
    <priority value="debug" />
</category>
```

4. If the SSH adapter is enabled for the communication with the PBS cluster, add the following XML to the `server-log.xml` file to troubleshoot job submission:

```
<category name="com.altair.gw.aif.ssh.implementation. SSHImplementation”>
    <priority value="debug" />
</category>
```

   This will log the job status attributes as returned to the PAS Server from the script executed through the SSH channel.

File Operations Execution (Local and Remote) and File Download

Configure logging behavior to troubleshoot errors during the file operations execution (local and remote) and file download.

1. Navigate to `PA_HOME/config/conf/`.
2. Update the `server-log.xml` to get more details on user information and the total time of the PAS server operation by adding the following XML:

```
<category name="com.altair.gw.aif.rest.RESTFilePortImpl">
    <priority value="debug" />
</category>
```

3. Update the `server-log.xml` file to get details about the local and remote file operation (parameters, current working directory, the operation result) at the PAS server level by adding the following XML:

```
<category name="com.altair.gw.aif.fileop.implementation. FileOperations">
    <priority value="debug" />
</category>
```

4. Update the `server-log.xml` file to get details about the local file operation by adding the following XML:

```
<category name="com.altair.gw.aif.fileop.implementation. FileOperationsHelper">
    <priority value="debug" />
</category>
```

5. Update the `server-log.xml` file to get details about the remote file operations by adding the following XML:

```
<category name=" com.altair.gw.aif.process.ProcessWrapper">
    <priority value="debug" />
</category>
```

   With this category we'll see more details related to user impersonation when we create the process as specific user.

6. If the SSH adapter is enabled for the communication with PBS cluster, it is possible to enable the logging of additional information related to the SSH tunnel. For the troubleshooting and debugging of remote file operations add the following XML to the `server-log.xml` file:

```
<category name="com.altair.gw.aif.fileop.implementation.cli. RemoteFileOpsTunnel-
Based">
    <priority value="debug" />
</category>
```

   It will show the command and parameters passed to the SSH tunnel and what is the result of the communication with the remote host coming back to the PAS Server.

7. Update the `server-log.xml` file to see more information about the file download by adding the following XML:

```
<category name=" com.altair.gw.aif.fileop.implementation. FileDownloadModel">
    <priority value="debug" />
</category>
```

Application Definition Related Errors

Configure logging behavior to troubleshoot application definition related errors.

1. Navigate to `PA_HOME/config/conf/`.
2. Update the `server-log.xml` to get more details on user information and the total time of the PAS server operation by adding the following XML:

```
<category name="com.altair.gw.aif.rest. ApplicationDefinitionService">
    <priority value="debug" />
</category>
```

3. If any errors indicate the failure of communication with the file system, add the following XML to the `server-log.xml` file to log more details about application definition reading and writing from the file system:

```
<category name=" com.altair.gw.aif.applications.utils. ApplicationsRepository">
    <priority value="debug" />
</category>
```

   Application definitions are cached for better performance. If any discrepancy between the application definition in the upstream products and the file system is observed, the cache debug logging should be enabled to make sure the PAS Server is providing the most current application definition.

4. Update the `server-log.xml` to get more information on cache debug logging by adding the following XML:

```
<category name=" com.altair.gw.aif.utils.generic. CacheManager">
    <priority value="debug" />
</category>
```

User Profile

Configure logging behavior to troubleshoot user profile errors.

1. Navigate to `PA_HOME/config/conf/`.
2. Update the `server-log.xml` to get more details on user information and the total time of the PAS Server operation by adding the following XML:

```
<category name="com.altair.gw.aif.rest. ProfileService">
    <priority value="debug" />
</category>
```

Server Registration

Configure logging behavior to troubleshoot server registration errors.

1. Navigate to `PA_HOME/config/conf/`.
2. Update the `server-log.xml` to get more details on user information and the total time of the PAS Server operation by adding the following XML:

```
<category name="com.altair.gw.aif.rest. ServerService">
    <priority value="debug" />
</category>
```

User Account

Configure logging behavior to troubleshoot user account related errors.

1. Navigate to `PA_HOME/config/conf/`.
2. Update the `server-log.xml` to get more details on user information and the total time of the PAS Server operation by adding the following XML:

```
<category name="com.altair.gw.aif.rest. UserService">
    <priority value="debug" />
</category>
```

Session Errors

Configure logging behavior to troubleshoot session related errors.

1. Navigate to `PA_HOME/config/conf/`.
2. Update the `server-log.xml` to get more details on user information and the total time of the PAS Server operation by adding the following XML:

```
<category name="com.altair.gw.aif.rest. SessionService">
    <priority value="debug" />
</category>
```

9.2.8 Configuring JVM Performance

The Java Virtual Machine (JVM) heap size of PAS may need to be adjusted. Currently, the default JVM heap size is 1024 MB. If OutOfMemory errors or warnings are reported in the PAS log file, this value should be increased. It can be increased in multiples of 256 MB.

On Linux, the Java heap size default is stored in a script file called `setenv.sh`. The location of this file for a typical installation of PAS is: `PA_EXEC/pas/scripts`

Modify the JVM argument -Xmx1024m. Increase the 1024 value.

This is the area of `setenv.sh` that will need to be modified (this file is too large to show in its entirety):

```
JAVA_OPTS="$JAVA_OPTS -Xmx1024m -XX:PermSize=128m -XX:MaxPermSize=128m"
```

The Access Web service will need to be restarted for the changes to be reflected.

9.3 Configure the Remote Session Component

Configurations when you install remote session component.

9.3.1 Change the Session Timeout Value for Interactive Applications

Change the duration of a remote session so that it remains active.

The default remote session timeout is 30 minutes. The session is killed if the user does not access the remote session within this time.

1. Login to the machine where Access Web is installed as root.
2. Navigate to `PA_HOME/config/displaymanager/`.
3. Edit the file `dmrest.properties`.
4. Update the value of *`jobaction.expiry_time`* in seconds.

```
#session expiry time(seconds)
    jobaction.expiry_time=1800
```

9.3.2 Change the Job Submission Host for Interactive Applications

Change the hostname or IP address of the machine from which jobs are being submitted.

Interactive jobs connect to the remote session service to provide job status updates. When the hostname or IP address of the machine where Access Web is installed changes, the remote session service must be updated with the new value.

> **Note:** This hostname or IP address must be accessible from all execution hosts.

1. Login to the machine where Access Web is installed as root.
2. Navigate to `PA_HOME/config/displaymanager/`.
3. Edit the file `dmrest.properties`.
4. Update the value of *`jobsub.monitor.host`* to the new hostname or IP address.

```
#Enables handling asynchronous job updates.
    jobsub.monitor.host=dm-05
    jobsub.monitor.port=4909
```

9.3.3 Change the Job Submission Port for Interactive Applications

Change the port that the remote session component listens on for job status updates.

Interactive jobs connect to the remote session service to provide job status updates. Modify the default port that the remote session component listens on for these updates if the default port is already being used by a different process.

> **Note:** The default port that the Interactive component listens on for job status updates is 4909.

1. Login to the machine where Access Web is installed as root.
2. Navigate to `PA_HOME/config/displaymanager/`.
3. Edit the file `dmrest.properties`.
4. Update the value of *`jobsub.monitor.port`* to the new port number.

```
#Enables handling asynchronous job updates.
    jobsub.monitor.host=blrvm9vm14
    jobsub.monitor.port=4909
```

9.3.4 Configure GPU Limits when the Number of GPUs Change

Reconfigure the remote session component when GPUs are added or removed from a cluster.

The installer configures all the node and cluster limits for GPUs. Follow the below procedure in situations where these limits might change (e.g. a node going down or being removed, adding new graphical nodes post-installation, etc.)

Modify the Cluster GPU Limit

Modify the available number of GPUs in a cluster when graphical nodes or devices are added or removed.

A queue (iworkq) is created when the remote session installer is run on the PBS Server. This queue exclusively handles interactive job requests. An attribute is set on the queue which limits the number of GPUs that can be allocated to jobs queued in or running from this queue. Modify the value of this attribute to accommodate any changes in the limit due to the addition or removal of graphical nodes or devices.

1. Login to the machine hosting the PBS Server as root or a user with sudo permissions.
2. Enter the command:

```
qmgr -c "p q iworkq"
```

Output similar to the below is displayed.

```
#
# Create queues and set their attributes.
#
```

```
#
# Create and define queue iworkq
#
create queue iworkq
set queue iworkq queue_type = Execution
set queue iworkq Priority = 150
set queue iworkq max_queued_res.ngpus = [o:PBS_ALL=5*]
set queue iworkq resources_max.ngpus = 1
set queue iworkq resources_min.ngpus = 1
set queue iworkq enabled = True
set queue iworkq started = True
```

The value of *max_queued_res.ngpus* is the available GPU limit for the cluster. This is the cumulative number of all the GPUs available in the cluster managed by the PBS server.

3. Change the value of *max_queued_res.ngpus* using the command:

```
qmgr -c "set queue iworkq max_queued_res.ngpus= [o:PBS_ALL=<GPUS>]"
```

where <GPUS> is the new number of GPUs available in the cluster.

Modify the Node GPU Limit

Modify the available number of GPUs for any execution hosts when graphical nodes or devices are added or removed.

When the remote session installer is run on the PBS MoM, an attribute is set on each graphical node which defines the number of available GPUs for the node. Modify the value of this attribute to accommodate any changes in the limit due to the addition or removal of GPUs.

Note: This must be done for any execution hosts that have had graphical nodes or devices added or removed.

1. Enter the command:

```
pbsnodes -a
```

Output similar to the below is displayed.

```
interactive-05
     Mom = interactive-05.cad.company.com
     Port = 15002
     pbs_version = PBSPro_13.1.3.170747
     ntype = PBS
     state = free
     pcpus = 32
     resources_available.arch = linux
     resources_available.host = interactive-05
     resources_available.mem = 131727204kb
     resources_available.ncpus = 32
     resources_available.ngpus = 3
     resources_available.vnode = interactive-05
     resources_assigned.accelerator_memory = 0kb
     resources_assigned.mem = 0kb
     resources_assigned.naccelerators = 0
     resources_assigned.ncpus = 0
     resources_assigned.netwins = 0
     resources_assigned.ngpus = 0
     resources_assigned.vmem = 0kb
     resv_enable = True
     sharing = default_shared
```

The value of *resources_available.ngpus* is the available GPU limit for the node, this is the cumulative number of all the graphical devices available for a particular node.

2. Change the value of *resources_available.ngpus* using the command:

```
qmgr -c "set node <VNODENAME> resources_available.ngpus=<NGPUS>"
```

where <VNODENAME> is the name of the node and <NGPUS> is the new number of GPUs available on this execution host.

9.4 Configure Results Visualization Service

This sections provides relevant information for the administrator in configuring Results Visualization Service (RVS).

9.4.1 Configure HyperWorks Location

Configure Access Web with HyperWorks location, to enable results visualisation capabilities.

Follow these steps when the HyperWorks location was not specified while installing Access Web 2018.4.

1. Edit the file at `PA_HOME/config/resultservice/config/site_config.xml`
2. Change the value of the `@location` to the location of HyperWorks.

```
<Products>
   <Product id="ALTAIR_HYPERWORKS" defaultVersion="2017.2">
        <Version id="2017.2" location="HW_EXEC/altair/" />
   </Product>
</Products>
```

3. Restart Access Web Service.

9.4.2 Configure HyperWorks Licenses

In the license server that we specify during the Access Web installation should include HWHyperMath and HWHyperViewTrans features. This enables to visualize the supported result files.

You can configure the RVS license in these two ways:

- To add the HyperWorks license in Access Web license file. Refer Add HyperWorks Licenses to the Access Web License File.
- To point RVS license to HyperWorks license server. Refer Point RVS to the HyperWorks License Server.

Add HyperWorks Licenses to the Access Web License File

Add HyperWorks licenses to the Access Web license file to enable RVS capabilities.

Open your HyperWorks license file to copy the HWHyperMath and HWHyperViewTrans features to the Access Web license file.

1. Login to the machine hosting Access Web.
2. Open the Access Web license configuration file at `PA_HOME/config/license/app.properties`.
3. Locate the following line.

   `pbsworks.license.server=port@`*`LICENSE_SERVER_HOSTNAME`*

   Where *LICENSE_SERVER_HOSTNAME* is the hostname of the license server.
4. Login to the Access Web license server specified in the previous step.
5. Open the Access Web license file at `/usr/local/altair/licensing14.0/altair_lic.dat`.

6. Copy the HWHyperMath and HWHyperViewTrans features from the HyperWorks license file to the Access Web license file `altair_lic.dat`.
7. Restart the LMX license server.
   ```
   /etc/init.d/altairlmxd restart
   ```
8. Kill any HyperMath sessions that exists in the Access Web machine.
   ```
   kill -9 <process id>
   ```

This will enable the RVS capabilities of Access Web.

Point RVS to the HyperWorks License Server

To enable RVS capabilities without merging the HyperWorks licenses with the Access Web license so that you can keep these licenses separate, point RVS to the Hyperworks License Server.

1. Open the file at `PA_EXEC/resultservice/scripts/setenv.sh file`.
2. Update the Altair license path to point to the HyperWorks license server in this format `port@hostname`.
   ```
   export ALTAIR_LICENSE_PATH=port@hostname
   ```
3. Kill any HyperMath sessions that exists in the Access Web machine.
   ```
   kill -9 <process id>
   ```

This will enable the RVS capabilities of Access Web.

Licensing System of HyperWorks Units

RVS use Altair patented licensing system of HyperWorks Units (HWU).

Animation request will checkout 6 HWU and Plot request will checkout 10 HWU on the server. Units are leveled for the same user but stacked for different users. They are checked out only during the results extraction on the server. As soon as the results are extracted, units are immediately returned to the licensing pool. The client side rendering of plot and animation results are covered by PBSA licensing and does not require any extra units.

9.4.3 Configure Solver File Readers

You can activate the result data files such as Abaqus, CFX, Fluent, and STAR-CCM+ that are not supported by default.

Solver log file readers are already present in the standard installation of HyperWorks, but they are not activated by default. Enable them by activating a HyperWorks plotting preference file.

Activate the Abaqus Reader

Activate the Abaqus reader so that Abaqus data files can be read by HyperWorks.

1. Edit the file at `HW_EXEC/hw/prefinc/preferences_common_plot.mvw`.

2. Locate the following line of code:

```
*RegisterExternalColumnReader({external_readers_dir + "/hgtextcolumn.exe"})
```

3. Add the following line of code directly before the code referenced in the previous step to activate the Fluent reader:

```
*RegisterExternalReader({external_readers_dir + "/hgabaqus.exe"}, "", "", ascii)
```

   The code looks like this when you are done.

```
*RegisterExternalReader({external_readers_dir + "/hgabaqus.exe"}, "", "", ascii)
*RegisterExternalColumnReader({external_readers_dir + "/hgtextcolumn.exe"})
```

Activate the CFX Reader

Activate the CFX reader so that CFX data files can be processed by HyperWorks.

1. Edit the file at `HW_EXEC/hw/prefinc/preferences_common_plot.mvw`.
2. Locate the following line of code:

```
*RegisterExternalColumnReader({external_readers_dir + "/hgtextcolumn.exe"})
```

3. Add the following line of code directly before the code referenced in the previous step to activate the Fluent reader:

```
*RegisterExternalReader({external_readers_dir + "/hgCFX.exe"}, "", "", ascii)
```

   The code looks like this when you are done.

```
*RegisterExternalReader({external_readers_dir + "/hgCFX.exe"}, "", "", ascii)
*RegisterExternalColumnReader({external_readers_dir + "/hgtextcolumn.exe"})
```

Activate the Fluent Reader

Activate the Fluent reader so that Fluent data files can be processed by HyperWorks.

1. Edit the file at `HW_EXEC/hw/prefinc/preferences_common_plot.mvw`.
2. Locate the following line of code:

```
*RegisterExternalColumnReader({external_readers_dir + "/hgtextcolumn.exe"})
```

3. Add the following line of code directly before the code referenced in the previous step to activate the Fluent reader:

```
*RegisterExternalReader({external_readers_dir + "/hgfluent.exe"}, "", "", ascii)
```

   The code looks like this when you are done.

```
*RegisterExternalReader({external_readers_dir + "/hgfluent.exe"}, "", "", ascii)
*RegisterExternalColumnReader({external_readers_dir + "/hgtextcolumn.exe"})
```

Activate the STAR-CCM+ Reader

Activate the STAR-CCM reader so that STAR-CCM data files can be processed by HyperWorks.

1. Edit the file at `HW_EXEC/hw/prefinc/preferences_common_plot.mvw`.
2. Locate the following line of code:

```
*RegisterExternalColumnReader({external_readers_dir + "/hgtextcolumn.exe"})
```

3. Add the following line of code directly before the code referenced in the previous step to activate the Fluent reader:

   ```
   *RegisterExternalReader({external_readers_dir + "/hgStarCCM.exe"}, "", "", ascii)
   ```

   The code looks like this when you are done.

   ```
   *RegisterExternalReader({external_readers_dir + "/hgStarCCM.exe"}, "", "", ascii)
   *RegisterExternalColumnReader({external_readers_dir + "/hgtextcolumn.exe"})
   ```

4. Copy the hgstarccm executable from the user download area to `HW_EXEC/io/abf_readers/bin/linux64`.

9.4.4 Supported Result File Types

The supported result file types supported by Access Web for visualizing Plot and Animation.

Table 6: Supported Result File Types for Plotting

Results File Format	Plot Data	Animation Data	Default TOC Type
Radioss Bulk			
*.op2	Yes	Yes	Animation
*.h3d	Yes	Yes	Animation
*.res	Yes	NA	Plot
*.pch	Yes	NA	Plot
*.gz	Yes	Yes	Animation
Radioss Block			
*A00#	Yes	Yes	Animation
*.T##	Yes	NA	Plot
*.gz	Yes	Yes	Animation
Optistruct			
*.op2	Yes	Yes	Animation
*.h3d	Yes	Yes	Animation
*.res	Yes	NA	Plot
*.pch	Yes	NA	Plot
*.hgdata	Yes	NA	Plot

Results File Format	Plot Data	Animation Data	Default TOC Type
MotionSolve			
*.mrf	Yes	NA	Plot
*.plt	Yes	NA	Plot
*.h3d	Yes	Yes	Animation
*.maf	Yes	NA	Plot
Abaqus			
*.odb	Yes	Yes	Animation
*.dat	Yes	NA	Plot
*.out	Yes	NA	Plot
ADAMS			
*.req	Yes	NA	Plot
*.res	Yes	NA	Plot
*.nam	Yes	NA	Plot
*.rsp	Yes	NA	Plot
*.shl	Yes	NA	Plot
*.sta	Yes	NA	Plot
Ansys			
*.rst	Yes	Yes	Animation
*.rth	Yes	Yes	Animation
*.rth	Yes	Yes	Animation
CFX			
*.out	Yes	NA	Plot
Fluent			
*.out	Yes	NA	Plot
output.*	Yes	NA	Plot

Results File Format	Plot Data	Animation Data	Default TOC Type
*.trn	Yes	NA	Plot
.txt	Yes	NA	Plot
Ls-Dyna			
d3plot	Yes	Yes	Animation
*dynain	Yes	Yes	Animation
*.fz	Yes	Yes	Animation
Intfor	Yes	Yes	Animation
Ptf	Yes	Yes	Animation
ABSTAT	Yes	NA	Plot
BINOUT	Yes	Yes	Animation
BNDOUT	Yes	Yes	Animation
DBFSI	Yes	NA	Plot
DEFORC	Yes	NA	Plot
*.dyn	Yes	NA	Plot
ELOUT	Yes	NA	Plot
GLSTAT	Yes	NA	Plot
GECOUT	Yes	NA	Plot
JNTFORC	Yes	NA	Plot
MATSUM	Yes	NA	Plot
NCFORC	Yes	NA	Plot
NODFOR	Yes	NA	Plot
NODOUT	Yes	NA	Plot
RBDOUT	Yes	NA	Plot
RCFORC	Yes	NA	Plot
RWFOC	Yes	NA	Plot

Results File Format	Plot Data	Animation Data	Default TOC Type
SBTOUT	Yes	NA	Plot
SECFORC	Yes	NA	Plot
SLEOUT	Yes	NA	Plot
SPHOUT	Yes	NA	Plot
SWFORC	Yes	NA	Plot
NanoFluidX			
nFX	Yes	No	NA
UltraFluidX			
uFX	Yes	No	NA
HyperXtrude			
*.h3d	Yes	Yes	Animation
*.out	Yes	NA	Plot
FEMZIP			
*.fz	Yes	Yes	Animation
d3plot	Yes	Yes	Animation
HW ASCII			
*.hwascii	Yes	Yes	Animation
MADYMO			
*.fai	Yes	NA	Plot
*.kn3	Yes	Yes	Animation
Marc			
*.t16	Yes	Yes	Animation
Nastran			
*.op2	Yes	Yes	Animation
*.pch	Yes	NA	Plot

Results File Format	Plot Data	Animation Data	Default TOC Type
NIKE3D			
*n3plot	Yes	Yes	Animation
Pamcrash			
*.DSY	Yes	Yes	Animation
*.erfh5	Yes	Yes	Animation
*.THP	Yes	NA	Plot
*.fz	Yes	Yes	Animation
*.h3d	Yes	Yes	Animation
Permas			
*A##	Yes	Yes	Animation
Other Ascii formats			
*.xgr	Yes	NA	Plot
*.dat	Yes	NA	Plot
*.col	Yes	NA	Plot
*.csv	Yes	NA	Plot
*.rvp	Yes	NA	Plot

Note: The default TOC type (plot or animation) will be identified depending on the file type registration and the parameter `isDefault` in the `plugin_def.xml` file.

Tip: To set the Default TOC Type, change the value of `isDefault` to true in the server configuration (`PA_HOME/config/resultservice/plugins/plot_toc_data_provider/plugin_def.xml`). Any new file formats other than the ones mentioned in the table which are supported by HyperWorks can be configured in `plugin_def.xml` file.

9.4.5 Configure Data Directory

You can configure RVS data directory for storing the RVS data such as result files, temporary files and cache data.

1. Open the file at `PA_HOME/config/resultservice/config/site_config.xml`.
2. Configure the folders to store your RVS data in the following line.
   `<HWE_RM_DATA_LOC><folder_name</HWE_RM_DATA_LOC`.

   Note: By default, the RVS data is stored in the temp folder.
3. Restart Access Web Service.

9.4.6 Set a Data Cleanup

To improve the performance of visualizing the result files you can configure the RVS data cleanup for the RVS cache data.

To configure the RVS data files refer, Configure Data Directory

You can configure the RVS data cleanup criteria settings in `site_config.xml` file. The recurring cleanup can be configured in these formats: Daily, Weekly or Monthly.

Note: By default, the RVS data cleanup is set for Weekly Sun 1:00 AM.

If you wish to setup the cleanup criteria on a daily basis, you need to specify the time as `DAILY 21:45` which represents everyday at 9:45 PM.

If you wish to setup the cleanup criteria weekly, you need to specify the days and the time. Here are a few examples:

`WEEKLY SUN 14:30` represents every Sunday in a week at 2:30 PM.

`WEEKLY MON-FRI 1:00` represents the range of days in a week. Every Monday to Friday at 1 AM.

`WEEKLY MON,WED 3:00` represents the days Monday and Wednesday in a week at 3 AM

If you wish to setup the cleanup criteria monthly, you would need to specify the days and time. Here are a few examples:

`MONTHLY 15 15:30` represents the every 15th day in a month at 3:30 PM.

`MONTHLY 10-20 1:30` represents every 10th day to 20th day in a month at 1:30 AM.

`MONTHLY 1, 11, 21 8:00` represents every 1st day, 11th day and 21st day in a month at 8 AM.

1. Open the file at `PA_HOME/config/resultservice/config/site_config.xml/`.
2. Set the value of cleanup time `<CleanupTime>` indicate when the cleanup should occur.
   The below example will set the cleanup daily at 1AM.
   ```
   <CleanupTime>DAILY 01:00</CleanupTime>
   ```
3. Enter the cleanup time in 24 hours format.
4. Choose one of the cleanup criteria to cleanup the untouched RVS data.

- Specify the value of cleanup criteria `<noofDays>`. The RVS data that is not accessed for the specified number of days is considered for the cleanup.

```
<Criterion id="LAST_ACCESSED_TIME_BASED" noOfDays="30"
class="com.altair.hwe.publish.resultsmanager.defaultimpl.LastAccessed
TimeBasedCriterion"/>
```

- Specify the value of cleanup criteria `<noofDays>`. The RVS data that is not modified for the specified number of days is considered for the cleanup.

```
<Criterion id="LAST_MODIFIED_TIME_BASED" noOfDays="100"
class="com.altair.hwe.publish.resultsmanager.defaultimpl.LastModified
TimeBasedCriterion"/>
</CleanupCriteria>
```

5. Replace the criterion id with the framework element value to configure the RVS data cleanup. `<Criterion id="FRAMEWORK_CACHE_CLEANUP_CRITERIA">` .
   The following snippet is set with Daily cleanup time and the chosen cleanup crietria is, the last modified time to clean the RVS data.

```
<CleanupTime>DAILY 01:00</CleanupTime>
<CleanupCriteria>
    <Criterion id="LAST_ACCESSED_TIME_BASED" noOfDays="30"
        class="com.altair.hwe.publish.resultsmanager.defaultimpl.LastAccessed
        TimeBasedCriterion"/>
    <Criterion id="FRAMEWORK_CACHE_CLEANUP_CRITERIA" noOfDays="100"
        class="com.altair.hwe.publish.resultsmanager.defaultimpl.LastModified
        TimeBasedCriterion"/>
</CleanupCriteria>
<FrameworkCacheCleanupCriteria>FRAMEWORK_CACHE_CLEANUP_CRITERIA</
FrameworkCacheCleanupCriteria>
```

6. Save the file and restart Access Web Service.

9.4.7 Configure RVS Parameters

Main parameters related to RVS in Access Web

The default value of the parameters and the location of the files in which these settings are stored are listed here:

TOC Size

Specify the maximum TOC size limit in bytes. It helps to send the partial TOC (bytes).

Default value is 2097152 (In bytes)

Configure the maximum TOC size using `<<SizeLimit toc="2097152" />>` and it is located at `PA_HOME/config/resultservice/config/site_config.xml/`

AIF Impersonation

The RVS result extractions and license check out are processed using the portal user.

Default value is set to true.

Configure the RVS impersonation using `< <AIFImpersonation enabled ="true">>` and it is located at `PA_HOME/config/resultservice/config/site_config.xml`

> **Note:** If the parameter value is set to false then all the RVS result extractions and license check out are processed as an administrator.

Cache enabled

Enable or disable data caching for RVS on the server.

Default value is set to true.

Configure the RVS data caching using `<<Cache enabled="true">>` and it is located at `PA_HOME/config/resultservice/config/site_config.xml`

Session defaultTimeout

Is the maximum amount of time the server should wait for a response from another application before disconnecting.

The default value is 6000 milliseconds.

Configure the session default timeout using `<<Session defaultTimeout="6000" />>` and it is located at `PA_HOME/config/resultservice/config/site_config.xml`

Operation defaultTimeout

Is the maximum amount of time the server should wait before closing an old connection and creating a new connection.

The default value is 6000 milliseconds.

Configure the operation default timeout using `<<Operation defaultTimeout="6000"/>>` and it is located at `PA_EXEC/resultservice/binaries/rvs_lib/config/site_config.xml`

Socket timeout for HMath

It is the maximum amount of time that the HMath server should wait to setup a connection with RVS.

The default value is 6000 milliseconds.

Configure the socket timeout for HMath in the line `<<Application id="HYPERMATH_APPLICATION">` and enter the `<socketTimeout="6000">` value.

The parameter is located at `PA_HOME/config/resultservice/plugins/hypermath_application/plugin_def.xml`

Connection timeout for HMath

It is the maximum amount of time that the HMath server should wait to respond for the data query from RVS.

The default value is 6000 milliseconds.

Configure the connection timeout for HMath in the line `<<Application id="HYPERMATH_APPLICATION">>` and enter the `<connectionTimeout="6000">` value.

The parameter is located at `PA_HOME/config/resultservice/plugins/hypermath_application/plugin_def.xml`

Request timeout for PBS datasource

It is the maximum amount of time that the RVS server will wait for a request from PAS.

The default value is 6000 milliseconds.

Configure the request timeout for PAS in the line `<<DataSourceHandler id="PBS_DATA_SOURCE_HANDLER">>` and enter the `<requestTimeout="6000">` value.

The parameter is located at `PA_HOME/config/resultservice/plugins/pbs_datasource_handler/plugin_def.xml`

Connection timeout for PBS datasource

It is the maximum amount of time that the PAS server should wait to respond for the data query from RVS.

The default value is 6000 milliseconds.

Configure the connection timeout for PAS in the line `<<DataSourceHandler id="PBS_DATA_SOURCE_HANDLER">>` and enter the `<connectionTimeout="6000">` value.

The parameter is located at `PA_HOME/config/resultservice/plugins/pbs_datasource_handler/plugin_def.xml`

Application Definitions

10

PAS requires a predefined set of instructions, called application definitions, to describe your application parameters to users, store their responses, and prepare those responses for job execution via PBS Professional.

This chapter covers the following:

- 10.1 Application Definition Components (p. 110)
- 10.2 Sample Application Definition ShellScript (p. 112)
- 10.3 Map Icons to an Application (p. 113)
- 10.4 Administration of Application Definition (p. 114)
- 10.5 Sitewide Settings (p. 116)
- 10.6 Interactive Application Definitions (p. 119)

An application definition provides a flexible set of instructions that can be manipulated to allow for precise control over all aspects of application-specific parameters and job execution. These application definitions are stored in a central repository and will make PBS Professional aware of each of them upon server startup. The location of the application directory for a typical installation of PAS is:`PA_HOME/data/pas/targets/localhost/repository/applications`.

10.1 Application Definition Components

Overview of application definition components.

An application definition is comprised of a set of Python scripts and two XML files which could include references to site wide configuration settings. A separate application definition with application-specific details is required for each application that will be integrated into PAS. Application definitions are compliant with the Open Grid Forum High Performance Computing Basic Profile (HPCBP), Business Process Execution Language (BPEL), and Job Submission Description Language (JSDL) standards.

Application definitions are stored in the PAS application directory. The default location of this directory is: `PA_HOME/data/pas/targets/localhost/repository/applications/`.

Diving Into Application Definitions explains what an application definition is and how to use an application definition. If you are installing PAS for the first time and are not familiar with application definitions, then Diving Into Application Definitions is a good place to start. A set of tutorials is provided starting with how to create the simplest of application definitions and progressing to more advanced topics. Recipes are also available demonstrating advanced techniques. For more comprehensive information about application definitions and their associated files see *Diving Into Application Definitions*.

10.1.1 Application Input File

The application input file is where administrators can define the allowed arguments for a given application. This file is also used by graphical, web-based and even command-line tools to display these arguments to users for job submission.

The mandatory naming convention for the application input file is `app-inp-applicationname.xml` where `applicationname` is whatever name you choose to give your application.

10.1.2 Application Converter File (HPCBP Converter)

The application converter file is where administrators take the values received by the user via the input file and communicates this information to the PAS and PBS Professional. This file allows the PAS administrator to configure the job submission environment.

The mandatory naming convention for the application converter file is `app-conv-applicationname.xml` where `applicationname` is whatever name you decide to give to your application.

10.1.3 Application Runtime Scripts

The application runtime scripts are what really get executed on the execution hosts. The runtime script, `start.py`, is what will be executed as the “job script”. This script file is responsible for executing

the application associated with your application definition, using the information entered by the user (defined by the application input file) and converted (via the application converter file). Additional scripts can also be included that support and enhance the runtime script. Administrators can directly edit these runtime script(s), taking full advantage of Python to add further inspection and complexity to job submission and finally execution of the application itself. This adds tremendous flexibility as nearly infinite possibilities for job control exist at this phase of job description.

For information about how to use runtime scripts see the tutorials *Enhancing your Application with Runtime Scripts*, *Executing Actions on a Running Job* and the *Recipes* section in *Diving Into Application Definitions*.

10.1.4 Site Configuration File

A default site configuration file, `site-config.xml`, is installed in the PAS configuration directory.

The site configuration file, `site-config.xml`, is meant to make application definitions more portable by consolidating data that may change from cluster to cluster in a central location. It is where administrators can define things like policies, version, and path information for all the available applications, billing account information, etc. The data in this file is used by the application input and converter files.

The location of this file for a typical installation of PAS is: `PA_HOME/data/pas/targets/localhost/repository/`

For information on how to reference the site configuration file in an application definition see the tutorial *Maintaining Multiple Versions of an Application*, recipes *How to Configure & Use Sitewide Billing Accounts*, *How to Configure & Use Sitewide Policies*, and *How to Configure & Use Application Policies* section in *Diving Into Application Definitions*.

10.1.5 Server Configuration File

A server configuration file, `server.conf`, is also installed in the PAS configuration directory.

This file contains all configurable parameters available with PAS. Please review Configurable Parameters on page 31 for an explanation of each parameter. The location of this file for a typical installation of PAS is: `PA_HOME/config/pas/conf`

10.2 Sample Application Definition ShellScript

PAS provides a sample application definition, ShellScript, which is located in the application definition directory.

ShellScript is an example of how an administrator could create an application definition enabling users to use custom job scripts akin to the more common PBS Professional job script. ShellScript allows these job scripts to be written in any language by using its runtime script, `start.py`, to parse the first line (shebang line) to determine the correct interpreter to execute your job script against. ShellScript's versatility allows the following:

- extra arguments to be passed to the job script
- specification of advanced multi-node placement options
- the inclusion of additional files

This application definition in conjunction with the *Diving Into Application Definitions* documentation will assist administrators to create and deploy application definitions.

10.3 Map Icons to an Application

Icons can be mapped to an application so that it gets displayed in the context menu and in the application list.

The application input file is where administrators can map an icon for a given application. The icon mapped in this file is displayed in the context menu and in the application list.

Note: Only Administrators can map an icon to the application.

1. Login to the machine where Access Web is installed as root.
2. Navigate to `PA_HOME/data/pas /targets/localhost/repository/applications/<appname>`.

   where <appname> is the application folder. For example, let's assume that we are adding icon to the ShellScript application definition.
3. Create an `avatar` directory.
4. Navigate to the `avatar` directory.
5. Place the application icon.

   Note: You have to place two different images, one for the context menu (smaller size) and the other for the list menu (bigger size) .
6. Navigate to `PA_HOME/data/pas /targets/localhost/repository/applications/ShellScript`.
7. Edit the application input file and add the `ApplicationIconSmall` xml element to display the icon in context menu and `ApplicationIconMedium` xml element to display the icon in the application list.

   ```
   <ApplicationId>ShellScript</ApplicationId>
   <ApplicationName>ShellScript</ApplicationName>
   <ApplicationExtension>.fem</ApplicationExtension>
   <ApplicationIconSmall>ShellScriptIconSmall.jpg</ApplicationIconSmall>
   <ApplicationIconMedium>ShellScriptIconMedium.jpg</ApplicationIconMedium>
   ```
8. Repeat steps 2 through 7 for all applications.
9. Restart the Access Web by entering the following command:

   ```
   service pbsworks-pa restart
   ```

10.4 Administration of Application Definition

PAS has a central location for storing application definitions - `PA_HOME/data/pas/targets/localhost/repository/applications`.

10.4.1 Add a New Application Definitions

Adding an application definition to the applications directory, followed by a restart of Access Web, will expose your application definition to the user. Follow these steps to add an application definition:

1. Login to the machine hosting the PAS Server.
2. Navigate to the applications directory located at `PA_HOME/data/pas/targets/localhost/repository/applications`.
3. Create a directory called `appname`.

   where `appname` is the name of the application.
4. Place the application definition input file (`app-inp-appname.xml`) and the application definition converter file (`app-conv-appname.xml`) in the application directory.
5. Navigate to the `appname` directory.
6. Create a subdirectory called `runtime` and navigate to that subdirectory.
7. Place any runtime scripts into the runtime directory.
8. Restart the Access Web by entering the following command:

   ```
   service pbsworks-pa restart
   ```

10.4.2 Application Definition Validation

When the Access Web starts, it performs a validation of the existing application definitions. If application definitions fail to meet key criteria, they will be moved to an invalid application directory. This directory will be created if it does not exist. The location of this directory is: `PA_HOME/data/pas/targets/localhost/repository/private/generated/invalid_applications`

In addition, PAS will restore the site configuration file from the last valid backup - `site-config.backup`.

Error messages will be displayed in the PAS log file indicating why the application definition was invalid. Invalid application definitions can be retrieved from the `invalid_applications` directory and modified to meet the criteria necessary to be considered valid. An invalid application definition will not prevent the PAS Server from starting up.

10.4.3 Maintenance of Existing Application Definition

Existing application definitions can be modified or removed. To make PAS aware of the modification or the removal of an application definition, restart the Access Web.

Update an Application Definition

You can easily modify an existing application definition using your favorite XML editor.

1. Login to the machine hosting the PAS Server.
2. Navigate to the applications directory located at `PA_HOME/data/pas/targets/localhost/respository/applications`.
3. Edit and make any modifications necessary to the application definition files.
4. Restart the Access Web by entering the following command:
   ```
   service pbsworks-pa restart
   ```

Remove an Application Definition

Removing an application definition from PAS is also very simple.

Simply remove the application definition directory from the applications folder. You have to restart Access Web to remove your application definition from PAS. Follow these steps to remove an application definition:

1. Login to the machine hosting the PAS Server.
2. Navigate to the applications directory located at: `PA_HOME/data/pas/targets/localhost/respository/applications`.
3. Remove the application directory and all of its content including the `runtime` subdirectory and associated files.
4. Restart the Access Web by entering the following command:
   ```
   service pbsworks-pa restart
   ```

10.5 Sitewide Settings

PAS provides a central repository for site specific information such as executable paths, policies, and billing account information. This information is stored in a site configuration file, `site-config.xml`. By putting some site specific application information in the, `site-config.xml` application definitions can be made more portable and reusable among different PBS Professional complexes. For example, putting binaries locations and version availability information here, makes the rest of the application definition reusable on a different cluster just by modifying that cluster's `site-config.xml`. The location of this file is `PA_HOME/data/pas/targets/localhost/repository`.

10.5.1 Site Configuration File Content

This file can hold virtually any sort of information, since the information stored in the file can be referenced by any application definition XML file. Altair has included certain data in the site configuration file for the integration and support of other products from the PBS Professional family. The example below shows a sample of the site configuration file with the sections currently used by the PBS Professional product suite:

- Applications - This section holds application specific information.
- Application versions - For each application, you can insert site supported versions and for each version its binary pathname.
- Job projects (billing accounts) - This section is for integrating PAS with other products from Altair. Here you can list a set of strings to be used as “accounting” information to be attached to jobs.
- Policies - Site policies are values that can be used in a site’s application definition XML files and/or as values available in job runtime environment (policies are included as environment variables for the jobs).
- Application policies - This section is for setting policies that are specific to an application.

10.5.2 Initial Site Configuration File after Installation of PAS

After installation of PAS the site configuration file will contain references to ShellScript, an application definition that is provided out-of-the-box. Placeholders for billing accounts and policies are provided, but will need to be updated according to your site specifications.

```
<?xml version="1.0" encoding="UTF-8"?>
<SiteConfiguration xmlns="http://schemas.altair.com/pbs/2007/01/site-config"
    xmlns:site-config="http://schemas.altair.com/pbs/2007/01/site-config"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://schemas.altair.com/pbs/2007/01/site-config ../schemas/
site-config.xsd">
    <Applications>
      <Application id="ShellScript">
        <ApplicationVersions/>
```

```
      </Application>
    </Applications>
    <JobProjects id="BILLING_ACCOUNT"/>
    <Policies/>
</SiteConfiguration>
```

10.5.3 Sample of a Site Configuration File

Here is an example of a site configuration file with modifications for site supported application versions, application policies, billing accounts, and sitewide policies:

```
<?xml version="1.0" encoding="UTF-8"?>
<SiteConfiguration xmlns="http://schemas.altair.com/pbs/2007/01/site-config"
   xmlns:site-config="http://schemas.altair.com/pbs/2007/01/site-config"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://schemas.altair.com/pbs/2007/01/site-config ../schemas/
site-config.xsd">
  <Applications>
    <Application id="ShellScript">
      <ApplicationVersions/>
      <Policies>
        <Policy>
          <Option>MAX_CPUS</Option>
          <Value>4</Value>
        </Policy>
      </Policies>
    </Application>
    <Application id="Optistruct">
      <ApplicationVersions>
        <ApplicationVersion>
          <Option>8.0</Option>
          <Executable>/opt/hyperworks/11.0/altair/scripts/optistruct</Executable>
        </ApplicationVersion>
        <ApplicationVersion>
          <Option>9.0</Option>
          <Executable>/sw/optistruct9/optistruct</Executable>
        </ApplicationVersion>
      </ApplicationVersions>
    </Application>
  </Applications>
  <JobProjects id="BILLING_ACCOUNT">
     <Option>Company1</Option>
     <Option>Company2</Option>
  </JobProjects>
  <Policies>
    <Policy>
      <Option>MAX_CPUS</Option>
      <Value>4</Value>
    </Policy>
  </Policies>
</SiteConfiguration>
```

10.5.4 Use Site Configuration Information in an Application Definition

For information on how to reference the site configuration file in an application definition see the tutorial *Maintaining Multiple Versions of an Application*, recipes *How to Configure & Use Sitewide Billing Accounts*, *How to Configure & Use Sitewide Policies*, and *How to Configure & Use Application Policies* in *Diving Into Application Definitions*.

10.5.5 Site Configuration File Backup

Upon server start-up, PAS will validate the content of the site configuration file and the content of application definitions. If the content of both the site configuration file and application definitions is valid, the site configuration file will be backed up to a file called `site-config.backup`. If the content of either the site configuration file or any application definition is invalid, the site configuration file will be backed up to a file called `site-config.YYYYMMDDMMSS` where *YYYYMMDDMMSS* is the file creation timestamp. The site configuration file will be restored from the last valid backup, `site-config.backup`.

Up to five (5) backups will be maintained. All backup files will be maintained in the same directory as the site configuration file.

.

10.5.6 Site Configuration File Validation

Upon server startup, PAS validates the site configuration file. It is validated against its XML schema. An XML schema defines the legal building blocks of a particular XML document. An XML schema:

- defines elements that can appear in a document
- defines attributes that can appear in a document
- defines which elements are child elements
- defines the order of child elements
- defines the number of child elements
- defines whether an element is empty or can include text
- defines data types for elements and attributes
- defines default and fixed values for elements and attributes

The validation process also determines if the content of the site configuration file is well-formed (valid). The content is well-formed if the following criteria is met:

- It must have a root element.
- XML elements must have a closing tag.
- XML tags are case sensitive.
- XML elements must be properly nested.
- XML attribute values must be quoted.

10.6 Interactive Application Definitions

Interactive application definition mandatory and special arguments.

10.6.1 Mandatory Interactive Application Definitions Changes

XML tags that are required for an interactive application definition.

For an application to be identified as interactive, the corresponding application definition must contain the `Interactive` element and its value must be set to true in the application input file.

```
<Interactive>true</Interactive>
```

Additionally, a boolean argument is necessary to run an interactive application which controls how many GPUs are requested at job submission.

```
<ArgumentChoice>
    <ArgumentBooleanWithDescription>
        <Name>GPU</Name>
        <Description>Is GPU required ?</Description>
        <DisplayName>GPU(?)</DisplayName>
        <InputRequired>false</InputRequired>
        <Value>true</Value>
    </ArgumentBooleanWithDescription>
</ArgumentChoice>
```

The `Value` element controls the request for GPUs. For 2D applications set it to false and a request for GPUs will not be made for the application. For 3D applications, `Value` option has to be set to true.

10.6.2 Special Interactive Application Arguments

Arguments that can be added to an interactive application definition.

The following interactive application specific arguments can be added to an application definition to pass arguments and environment variables to the application, define a job working directory, create a backup of job input files, and change the viewing mode from Applet to HTML5.

Arguments

A special string argument having the name `DM_APP_ARGS` can be added to an application definition so that arguments can be passed to the application. Multiple arguments can be passed to the application by separating them by `'\n'`.

```
<ArgumentChoice>
    <ArgumentString>
        <Name>DM_APP_ARGS</Name>
        <Description>'\n' separated args</Description>
        <DisplayName>Arguments</DisplayName>
        <InputRequired>false</InputRequired>
    </ArgumentString>
</ArgumentChoice>
```

Environment

A special string argument having the name `DM_APP_ENVS` can be added to an application definition so that environment variables can be passed to the application. Multiple variables can be passed to the application by separating them by `'\n'`.

```
<ArgumentChoice>
    <ArgumentString>
        <Name>DM_APP_ENVS</Name>
        <Description>'\n' separated envs</Description>
        <DisplayName>Environments</DisplayName>
        <InputRequired>false</InputRequired>
    </ArgumentString>
</ArgumentChoice>
```

WorkDirectory

A special string argument having the name `DM_APP_WDIR` can be added to an application definition so a job working directory is created when the job is submitted.

```
<ArgumentChoice>
    <ArgumentString>
        <Name>DM_APP_WDIR</Name>
        <Description>Working dir</Description>
        <DisplayName>Workding Dir</DisplayName>
        <InputRequired>false</InputRequired>
    </ArgumentString>
</ArgumentChoice>
```

Note: The arguments mentioned above are disabled by default. Enable them by setting the `InputRequired` field to true.

Copy Back Files

A special boolean argument having the name `COPY_BACK_FILES` can be added to an application definition so that job input files are copied to the stageout directory.

```
<ArgumentChoice>
    <ArgumentBooleanWithDescription>
        <Name>COPY_BACK_FILES</Name>
        <Description>Should job file(s) be staged out</Description>
        <DisplayName>Copy back files</DisplayName>
        <InputRequired>true</InputRequired>
        <FeatureEnabled>false</FeatureEnabled>
        <RefreshOnUpdate>true</RefreshOnUpdate>
    </ArgumentBooleanWithDescription>
</ArgumentChoice>
```

Note: The `COPY_BACK_FILES` arguments is disabled by default. Enable this field by setting the `FeatureEnabled` option to true. When `FeatureEnabled` is set to true, the person who is submitting the job can choose whether to stageout job input files.

Client View Mode

A special enumerated list argument having the name `DM_CLIENT_VIEW_MODE` can be added to an application definition so that the person submitting the job can choose the type of mode to view the job results.

```
<ArgumentChoice>
    <ArgumentStringEnumerated>
        <Name>DM_CLIENT_VIEW_MODE</Name>
        <Description>Viewer mode</Description>
        <DisplayName>Viewer Mode</DisplayName>
        <InputRequired>false</InputRequired>
            <Option>Desktop</Option>
            <Option>HTML5</Option>
            <Option>Applet</Option>
        <Value>HTML5</Value>
    </ArgumentStringEnumerated>
</ArgumentChoice>
```

Note: By default, the HTML5 view mode is enabled.

Mandatory/Optional Application Converter File Changes

In the `app-conv-AppName` application definition file, the following section determines which jobs are displayed in Access Web.

```
<jsdl-hpcp:Environment name="DM_JOB">True</jsdl-hpcp:Environment>
```

If this is set to False, apart from the interactive application jobs, all the other jobs belonging to the user will be displayed. Ensure that this is set to True.

10.6.3 Add a New Interactive Application

Create a new interactive application definition by copying a default interactive application definition and making application specific changes.

Any time a new application is added to your HPC, a corresponding application definition needs to be written. Writing a specific interactive application definition is a bit more complex than writing simple application definitions. We recommend the following procedure of copying and modifying the GLXSpheres application definition which is available after installing the Remote Session components of Access Web.

1. Navigate to `PA_HOME/data/pas/targets/localhost/repository/applications`.
2. Copy the GlxSpheres application definition directory and rename it to the name of the new application.

   If the new application is HyperView, then execute the following command:

   ```
   cp -rp GlxSpheres HyperView
   ```
3. Rename the GlxSpheres application definition files to the name of the new application.

   ```
   mv app-actions-GlxSpheres.xml app-actions-HyperView.xml
   mv app-conv-GlxSpheres.xml app-conv-HyperView.xml
   mv app-inp-GlxSpheres.xml app-inp-HyperView.xml
   ```
4. Edit the application input file.
   a) Change the value of the `<ApplicationId>` element to the name of the new application.

      ```
      <ApplicationId>HyperView</ApplicationId>
      ```
   b) Change the value of the `<ApplicationName>` element to the new application name.

      ```
      <ApplicationName>HyperView</ApplicationName>
      ```

c) Locate the application argument `<ArgumentChoice>` called VERSION.

```
<ArgumentChoice>
    <ArgumentStringEnumerated>
        <Name>VERSION</Name>
        <Description>Version of the interactive application you
                        selected to start </Description>
        <DisplayName>Version</DisplayName>
        <xi:include href="site-config.xml" pointer="xpath1
          (//Application[@id='GlxSpheres']/ApplicationVersions//Option)" />
    <ArgumentStringEnumerated>
</ArgumentChoice>
```

d) Change the attribute `@pointer` of the `<xi:include>` element to point to the path of the new application:

```
<xi:include href="site-config.xml" pointer="xpath1
          (//Application[@id='HyperView']/ApplicationVersions//Option)" />
```

5. Edit the application action file.

a) Change the value of the `<ApplicationId>` element to the name of the new application.

```
<ApplicationId>HyperView</ApplicationId>
```

b) Change the value of the `<ApplicationName>` element to the new application name.

```
<ApplicationName>HyperView</ApplicationName>
```

6. Edit the application converter file.

a) Change the value of the `<ApplicationId>` element to the name of the new application.

```
<ApplicationId>HyperView</ApplicationId>
```

b) Change the value of the `<ApplicationName>` element to the new application name.

```
<ApplicationName>HyperView</ApplicationName>
```

c) Set the site specific required environment for the application.

You can set the ALTAIR_LICENSE_PATH, NCPUS, ngpus, MEMORY etc according to the requirements for the application.

```
<jsdl-hpcp:Environment name="ALTAIR_LICENSE_PATH">6200@licsrv</
jsdlhpcp:Environment>
```

7. Navigate to the directory `PA_HOME/data/pas/targets/localhost/repository`.

8. Edit the `site-config.xml` file.

a) Add a new `<Application>` element that points to the new application executable.

```
<Application id="HyperView">
    <ApplicationVersions>
        <ApplicationVersion>
            <Option>13.2</Option>
            <Executable>/altair/hw/13.2/altair/scripts/hv</Executable>
        </ApplicationVersion>
    </ApplicationVersions>
</Application>
```

Tip: You can also define multiple executable versions for the application.

```
<Application id="HyperView">
    <ApplicationVersions>
        <ApplicationVersion>
            <Option>13.1</Option>
            <Executable>/altair/hw/13.1/altair/scripts/hv</Executable>
        </ApplicationVersion>
```

```
        <ApplicationVersion>
            <Option>13.2</Option>
            <Executable>/altair/hw/13.2/altair/scripts/hv</Executable>
        </ApplicationVersion>
    </ApplicationVersions>
</Application>
```

9. Restart the Access Web for these changes to take effect by entering the following command:

```
service pbsworks-pa restart
```

The new application will be available in Access Web after restart.

Log Files

11

Information about the log files.

This chapter covers the following:

- 11.1 PAS Log File (p. 125)
- 11.2 Other Log Files (p. 126)

11.1 PAS Log File

The PAS log file, `pas-server.log`, contains a record of server activities and is useful for identifying issues and problems. The location of the log file for a typical installation of PAS is: `PA_HOME/logs/pas/`

11.2 Other Log Files

The PAS installation log file is located in: `/opt/altair/pbsworks/pas/2018.4/_PAS Services_installation/Logs`

The log files for the Apache Tomcat web server are located in: `PBSWORKS_EXEC/pas/bin/pas-server/logs`

Troubleshoot Access Web

12

Troubleshooting Access Web, PAS, Result Visualization Service, and Remote Session.

This chapter covers the following:

- 12.1 Run the Access Web Diagnosis Script (p. 128)
- 12.2 Troubleshoot PBS Application Services (p. 129)
- 12.3 Troubleshoot Remote Session Components (p. 134)
- 12.4 Troubleshoot Result Visualization Services (p. 140)

The following section provides the troubleshooting information and steps for Access Web and its services.

12.1 Run the Access Web Diagnosis Script

Run the diagnosis script provided in Access Web to get information for troubleshooting Access Web.

The diagnosis script will create a zip file which can be shared with Altair support team for troubleshooting.

1. Login to the machine as root where you have installed Access Web and PBS Application Services.
2. Navigate to `PA_EXEC/init/`
3. Run the following command:

   ```
   pa-diagnosis.py
   ```

   A zip file will be created and stored at:

   ```
   /tmp/pbsworks-pa-diagonsis_20181010161835.zip
   ```

12.2 Troubleshoot PBS Application Services

Troubleshooting information and steps for PAS.

The following section provides the information about troubleshooting information and steps for PAS.

12.2.1 Verify PAS Status

Monitor PAS using the status page.

A status page is available through the following URL to monitor the status of PAS: `https://HOSTNAME:5243/pas/pasStatus`

`HOSTNAME` is the hostname of the machine where the PAS Server is installed.

Green indicates that the system is functioning properly. Red indicates an issue that should be investigated.

PAS Status at time: 6:23:29 PM

JVM Platform OS Name (system property os.name)	Linux - OK
JVM Platform OS Version (system property os.version)	3.10.0-693.el7.x86_64 - OK
JVM Platform Architecture (system property os.arch)	amd64 - OK
JVM Data Model (system property sun.arch.data.model)	64 - OK
Java Version	1.8.0_92 - OK
Python	OK
Dependent shared objects	OK
Available heap memory in MB	186 MB - OK
Memory used by AIF server in MB	101 MB - OK
PAS_EXEC directory disk space.	OK
PAS_HOME directory disk space.	OK
PBS status	PBS server is Running - OK
PBS Version	18.2 - OK
PBS execution speed	OK

Note: For the PBS execution speed, PAS retrieves the PBS version by executing `get_PBS_version.py` located in `PA_EXEC/pas/scripts`. If the response takes more than 5 seconds, the execution speed is considered slow and will be displayed in red.

12.2.2 Troubleshoot PAS Job Submission Issues

Troubleshoot PAS job submission issues by setting the logging level for specific functional areas and by submitting job script directly to PBS.

PAS Job Submission Issues by Setting Logging Level

Troubleshooting PAS job submission issues by setting the logging level for specific functional areas.

To troubleshoot errors during job submission and job monitoring, relevant information must be collected:

- user inputs provided for the job submission
- submission environment
- status (success or failure) of all the dependency
- steps which happen during the job submission
- job submission attributes generated by PAS server for the workload manager (PBS)

Follow these steps to gather relevant data:

1. Check the PAS server log file `pas-server.log` located at `PA_HOME/logs/pas/` for errors.
2. Check the Tomcat `catalina.out` log file located at `PA_HOME/logs/pas/` for network or security errors.
3. Check the system logs.

   Note: For advanced debugging, contact an Altair Application Engineer.

4. Navigate to `PA_HOME/config/pas/conf/`.
5. Update `server-log.xml` file to get detailed logging information about user inputs and to see how long it takes to execute a submission request with the following XML :

   ```
   <category name=" com.altair.gw.aif.rest. RESTJobsPortImpl">
       <priority value="debug" />
   </category>
   ```

6. Update `server-log.xml` file to get information about the process of creating PBS job attributes from user inputs and the application definition with the following XML :

   ```
   <category name=" com.altair.gw.aif.rest.util.PASNextGenJobUtils">
       <priority value="debug" />
   </category>
   ```

   PAS supports multiple adapters to communicate with the workload manager.

7. If the SSH adapter is enabled for the communication with the PBS cluster, add the following XML to the `server-log.xml` file to troubleshoot job submission:

   ```
   <category name=" com.altair.gw.aif.ssh.implementation. SSHImplementation”>
       <priority value="debug" />
   </category>
   ```

8. Restart Access Web by entering the following command:

   ```
   service pbsworks-pa restart
   ```

If this troubleshooting steps do not provide enough information to debug the issue, then submit a job script directly to PBS.

PAS Job Submission Issues by Submitting Job Script

Debug the Job Submission issue by submitting the job script directly to PBS.

1. Enable debug mode for the PAS Server by editing the file `PA_HOME/config/pas/conf/server.conf`.
2. Set `DEBUG` to true.
   ```
   DEBUG=TRUE
   ```
3. Restart Access Web by entering the following command:
   ```
   service pbsworks-pa restart
   ```
   The PAS Server will save the job script to a file in `PA_HOME/data/pas/system/temp`.
4. Use the job script file to submit a job directly to PBS Professional.
5. Enable debug logging in the PAS job script `PA_EXEC/pas/scripts/job.py` by setting `DEBUG` to true.
   ```
   DEBUG=TRUE
   ```
   This provides information about the job process arguments and the job environment at the moment of job execution. This information is available in the job output file.

Note: To gather debugging information for other functional areas of PAS, see Configuring PAS Server Logging Behavior.

12.2.3 Troubleshoot Error Messages during Installation

Error messages, explanation, and a resolution for the error that may occur during the installation of PAS.

Stage Directory is Blank

Condition

During the installation of PAS, I get a Staging Directory is blank warning message.

Cause

Staging Directory path is not provided during installation.

Remedy

The staging directory is where the necessary files are transferred after job submission, but prior to the portal submitting the job to PAS for transfer to PBS Professional for execution. This staging directory

must exist for the installation to complete successfully. Please enter a pathname to an existing directory which will be designated as the PAS staging directory.

PAS Service was unable to start

Condition

During the installation of PAS, I get a PAS Service was unable to start error message.

Cause

This may be due to port required by PAS is unavailable or busy.

Remedy

An error occurred while starting PAS. Review the PAS log file for errors and contact system support.

PAS Log File Contains OutofMemory Errors

Condition

There are OutOfMemoryErrors in the PAS Server log.

Cause

This may be due to Heap Memory size occupied is more than configured memory size.

Remedy

You may need to adjust the Java Virtual Machine (JVM) heap size of PAS. Refer Configuring JVM Performance for more information. Out of memory errors can indicate a underlying problem, therefore it is recommended to report these errors to PAS system support.

Job Submission Fails with the Error: Bad UID for Job Execution

Condition

Our site has installed PAS on a separate machine from the one hosting the PBS Professional Server. When a user submits a job, the job fails with a “Bad UID for job execution” exception.

Cause

This can be due to multiple reasons like the user is root or `flatuid` is not set.

Remedy

The PBS Professional `flatuid` attribute must be set to `True`. This attribute specifies whether, for each user, the username at the submission host must be the same as the one at the Server host. The username at the Server host must always be the same as the username at the execution host. When

`flatuid` is set to `True`, the Server assumes that `UserA@host1` is the same as `UserA@host2`. Therefore, if `flatuid` is `True`, `UserA@host2` can operate on `UserA@host1`'s job. Follow these steps to set `flatuid` to `True`:

1. Log into the machine hosting the PBS Professional Server.
2. At the command line, enter the command:
   ```
   print server
   ```
3. If the attribute `flatuid` is equal to `Flase` or is not set (you do not see it in the output from the print server command), then set the value to `True` by issuing the command:
   ```
   set server flatuid = True
   ```

Job Submission Fails with the Error: Unknown Resource Resource_List.<resource>

Condition

Our site has installed PAS on a separate machine from the one hosting the PBS Professional Server. When a user submits a job, the job fails with a "Unknown resource Resource_List.xxxxxx" exception.

Cause

The resource needed for running jobs are not configured in PBS Professional.

Remedy

Resources required by PAS have not been defined to PBS Professional. Add the required PAS resources to the PBS Professional resource definition file. For more information see Configuring PBS Resources.

New Features of PBS Professional are not working after Installation

Condition

I have upgraded PBS Professional to the newest version and now the new features are not working.

Cause

Whenever PAS starts, it reads PBS configuration and artifacts and keeps into the memory. So after PBS Professional is reinstalled, the new configuration and artifacts will not be available.

Remedy

After installing a new version of PBS Professional, you must restart the PBS Professional and then restart Access Web.

12.3 Troubleshoot Remote Session Components

Troubleshooting information and steps for Remote Session components.

On completing the installation the services relevant to Remote Session should be up and running. This quick troubleshooting checklist assumes services like, dmagent, application services, PBS Professional, and PAS are running and configured.

The following section provides the information about troubleshooting information and steps for Remote Session components.

12.3.1 Run the Remote Session Diagnosis Script

Run the diagnosis script provided in Access Web to get information for troubleshooting Remote Session.

Run diagnosis script on PBS Mom, PBS Professional, PAS and Access Web machine as root and share the output with support team.

1. Login to the machine as root where you have installed Access Web.
2. Enter the command:

   ```
   python <PA_EXEC>/displaymanager/scripts/remotesession-diagnosis.py
   ```

   The following is an example of the command output that is displayed:

   ```
   single_machine_installation
       - guacd
       - Guacd is installed - YES
       - Guacd is running - YES
       - Guacd configuration
       - bind_host = blrvm9vm14
       - bind_port = 4822
       - pbsaccess
       - PBSAccess is installed - YES
       - PBSAccess is running - YES
       - Gucad hostname matched - YES
       - Gucad port matched - YES
       - PBSPro
       - PBSPro is installed - YES
       - PBSPro is running - YES
       - PBSPro iworkq configured: YES
       - PAS
           - PAS is installed - YES
           - PAS is running - YES
           - Interactive Appdef XML:
               - output: Gedit, HVP, HyperMath, HyperMesh, HyperView, HyperWorks, RFJD
           - Interactive Appdef JSON:
               - output: Gedit, HVP, HyperMath, HyperMesh, HyperView, HyperWorks, RFJD
       - execution_node
           - RemoteSession agent: TurboVNC is installed - YES
           - RemoteSession agent: VirtualGL is installed - YES
           - RemoteSession agent: GPU hardware is configured: YES
               - output: OpenGL version string: 2.1 Mesa 17.0.1

           - RemoteSession agent: Direct Rendering: YES
               - output: 300 GLXFBConfigs:
       visual  x   bf lv rg d st  colorbuffer  ax dp st accumbuffer  ms  cav  drw
   ```

```
   id dep cl sp  sz l  ci b ro  r  g  b  a F bf th cl  r  g  b  a ns b eat  typ
   ------------------------------------------------------------------------------
   0x05c 24 tc  0  32  0 r  .  .  8  8  8  8 .  0  0  0  0  0  0  0  0 0 None PXW
   0x05d 24 tc  0  32  0 r  .  .  8  8  8  8 .  0  0  0 16 16 16 16  0 0 Slow PXW
   0x05e 24 tc  0  32  0 r  y  .  8  8  8  8 .  0  0  0  0  0  0  0  0 0 None PXW
   0x05f 24 tc  0  32  0 r  y  .  8  8  8  8 .  0  0  0 16 16 16 16  0 0 Slow PXW
   0x060 24 tc  0  32  0 r  y  .  8  8  8  8 .  0  0  0  0  0  0  0  0 0 None PXW
   0x061 24 tc  0  32  0 r  y  .  8  8  8  8 .  0  0  0 16 16 16 16  0 0 Slow PXW
   0x062 24 tc  0  32  0 r  .  .  8  8  8  8 .  0 16  0  0  0  0  0  0 0 None PXW

       - Desktop Manager Environment installed:
           - output: gnome-classic.desktop, gnome-custom-session.desktop,
gnome.desktop
```

12.3.2 Remote Session Checklist

Checklist for Remote Session components and steps to verify the various components required to run remote session.

Following are the prerequisites for remote session:

- The execution node should be set up to run VNC server
- X server must be configured and running
- Access to X server must be granted for the user. The Remote Session Agent Installer will do this during installation.
- VirtualGL configured and functional

Checklist for the Execution node (or Graphics node):

1. Make sure that all the Remote Session components are able to communicate with each other through hostname.
2. Run the following command to verify if VNC server is set:
   ```
   /opt/TurboVNC/bin/vncserver -noauth
   ```
   a) Connect using any VNC client to the VNC server:
   ```
   $host:$vncwebport (e.g. v01: 1),
   ```
   b) Replace *v01* with the VNC server hostname and *:1* with the display number of the VNC server.

   If you don't see the virtual desktop, please forward the `~/.vnc` folder to the Altair support team for further investigation.
3. Verify if X server is running for display *:0* (usually with process name Xorg or X).
4. If X server is not running, then reconfigure X server with:
   ```
   /usr/bin/nvidia-xconfig --use-display- device=NONE --enable-all-gpus --keyboard=us
   ```
   This will overwrite existing configurations and configure the X server with all available GPUs.
5. Verify if X server access is provided to the user by using the following command in a terminal window in the remote desktop session created in Step 2:
   ```
   /opt/VirtualGL/bin/glxinfo -display :0 -c
   ```
   The command should output a list of visuals and complete with no errors.
6. To verify graphics driver support VirtualGL, run the following command:

```
/opt/VirtualGL/bin/glxinfo -display :0 -c
```

OpenGL renderer string: "should state vendor driver version here"

The output should provide visuals with the following:

```
GLX_NV_multisample_coverage, GLX_NV_robustness_video_memory_purge,
    GLX_NV_swap_group, GLX_NV_video_capture, GLX_SGIX_fbconfig,
    GLX_SGIX_pbuffer, GLX_SGI_swap_control, GLX_SGI_video_sync
OpenGL vendor string: NVIDIA Corporation
OpenGL renderer string: Quadro 4000/PCIe/SSE2
```

7. Verify if VirtualGL is configured and functional by running the following command in a terminal window in the remote desktop session created in Step 2:

   ```
   /opt/VirtualGL/bin/vglrun -d :0.0 -sp /opt/VirtualGL/bin/glxspheres64
   ```

   If GLXSpheres starts, then VirtualGL is properly configured.

   > **Note:** If GLXSpheres does not start, then a detailed investigation is required by the support team.

8. Check for errors of the remote session application definition in the PAS logs.
9. Verify that the pbs_mom logs do not report any error related to the remote session jobs associated with the remote sessions. For example, errors related to file transfers.
10. Verify that PBS Professional jobs submitted as remote session ends up running on the expected host(s) without errors.
    For example, when debugging problems related to the execution host dmhost1, verify that a job like:

    ```
    qsub -l select=1:ncpus=1:mem=8gb:ngpus=1:host=dmhost1 -Wsandbox=PRIVATE -I
    ```

    is correctly running and does not generate any errors in any PBS Professional logs, most importantly in the related pbs_mom logs. Tune the resource requests to match the actual ones remote session uses when generating this jobs. PBS Professional job history will help identify the correct values:

    ```
    qstat - fx <jid of failed dm job>
    ```

12.3.3 Display Session is shown but not the Application that was Started

Condition

After starting an application, you might notice that the display session was shown but not the application.

Cause

Generally, this happens when the user doesn't have access to the 3D X server.

Remedy

Administrator

1. Change the role from Administrator to the User and verify if the user has access to the 3D X server by running the following command in a terminal:

   ```
   xdpyinfo -display :0" or "/opt/VirtualGL/bin/glxinfo - display :0 -c
   ```

   If the user does not have the permission to access the 3D X Server, then the following message is displayed:

   ```
   "unable to open display :0"
   ```
2. Follow the steps provided in http://www.virtualgl.org/vgldoc/2_2_1/#hd005001 to configure VGL to work with X Server
3. Verify whether X Server is configured to export True Color (24 bit or 32 bit) visuals.
   To configure X server for nVidia devices, use the following command:

   ```
   "/usr/bin/nvidia-xconfig --use-display-device=NONE --enable-all-gpus --key-
    board=us"
   ```
4. Choose one of the following options:
   - Perform Step 1 to verify if the user has access to the 3D X server
   - Create a new session from Access Web to check if the session is working fine

12.3.4 Application Definition working in VirtualGL Environment Issue

Condition

Only sutain application is having issue running and other applications are working fine.

Cause

Application having issues working in VirtualGL environment.

Remedy

Please check with application provider if application works in VirtualGL environment.

12.3.5 Desktop Manager not Displaying

Condition

Remote application is showing but not the desktop.

Cause

The possible reasons are:

- Desktop Manager is not installed or configured in application definition
- Not using the GPU hardware for OpenGL call

Remedy

1. Check the `runtime/xstartup.turbovnc` file available inside application definition for configured Desktop Manager.
2. Check if configured Desktop Manager is installed on all the Interactive MOM and executable is correct.
3. Check the Graphic Card Compatibility Issues.
4. In Job running directory check for the `Xvnc.log` to check for any error.
5. Try with other Desktop Manager by installing on Interactive MOM and configuring it on `xstartup.turbovnc` in application definition.

12.3.6 Graphic Card Compatibility Issues

Condition

Remote Session Application is not consuming GPU resource.

Cause

Not using the proper drivers for the graphic card.

> **Note:** Use the drivers provided by the manufacturer of the graphic card. If the manufacturer of your 3D adapter provides proprietary drivers for Linux, then it is recommended that you install these.

Remedy

Administrator

Run the following command and inspect the output:

```
/opt/VirtualGL/bin/glxinfo -display :0 -c
```

For example: if you run the following command:

```
[root@dmsystem-linux ~]# /opt/VirtualGL/bin/glxinfo -display :0 -c
```

The output of the command is displayed as follows:

```
    GLX_ARB_context_flush_control, GLX_ARB_create_context,
    GLX_ARB_create_context_profile, GLX_ARB_create_context_robustness,
    GLX_ARB_fbconfig_float, GLX_ARB_get_proc_address, GLX_ARB_multisample,
    GLX_EXT_buffer_age, GLX_EXT_create_context_es2_profile,
    GLX_EXT_create_context_es_profile, GLX_EXT_framebuffer_sRGB,
    GLX_EXT_import_context, GLX_EXT_stereo_tree, GLX_EXT_swap_control,
    GLX_EXT_swap_control_tear, GLX_EXT_texture_from_pixmap,
    GLX_EXT_visual_info, GLX_EXT_visual_rating, GLX_NV_copy_buffer,
    GLX_NV_copy_image, GLX_NV_delay_before_swap, GLX_NV_float_buffer,
    GLX_NV_multisample_coverage, GLX_NV_robustness_video_memory_purge,
    GLX_NV_swap_group, GLX_NV_video_capture, GLX_SGIX_fbconfig,
    GLX_SGIX_pbuffer, GLX_SGI_swap_control, GLX_SGI_video_sync
OpenGL vendor string: NVIDIA Corporation
OpenGL renderer string: Quadro 4000/PCIe/SSE2
OpenGL core profile version string: 4.3.0 NVIDIA 375.82
OpenGL core profile shading language version string: 4.30 NVIDIA via Cg compiler
OpenGL core profile extensions:
    GL_AMD_multi_draw_indirect, GL_ARB_ES2_compatibility,
    GL_ARB_ES3_1_compatibility, GL_ARB_ES3_2_compatibility,
    GL_ARB_ES3_compatibility, GL_ARB_arrays_of_arrays, GL_ARB_base_instance,
    GL_ARB_blend_func_extended, GL_ARB_buffer_storage, GL_ARB_cl_event,
```

```
name of display: :0
display: :0  screen: 0
direct rendering: Yes
server glx vendor string: NVIDIA Corporation
server glx version string: 1.4
server glx extensions:
    GLX_ARB_context_flush_control, GLX_ARB_create_context,
    GLX_ARB_create_context_profile, GLX_ARB_create_context_robustness,
    GLX_ARB_fbconfig_float, GLX_ARB_multisample, GLX_EXT_buffer_age,
    GLX_EXT_create_context_es2_profile, GLX_EXT_create_context_es_profile,
    GLX_EXT_framebuffer_sRGB, GLX_EXT_import_context, GLX_EXT_libglvnd,
    GLX_EXT_stereo_tree, GLX_EXT_swap_control, GLX_EXT_swap_control_tear,
    GLX_EXT_texture_from_pixmap, GLX_EXT_visual_info, GLX_EXT_visual_rating,
    GLX_NV_copy_buffer, GLX_NV_copy_image, GLX_NV_delay_before_swap,

359 GLXFBConfigs:
    visual  x   bf lv rg d st  colorbuffer  ax dp st accumbuffer  ms  cav  drw
  id dep  cl sp  sz l  ci b ro  r  g  b  a F bf th cl  r  g  b  a ns b eat  typ
----------------------------------------------------------------------------
0x135 24  tc  0  24  0 r  y  .  8  8  8  0 .  4 24  8 16 16 16 16  0 0 None PXW
0x136 24  dc  0  24  0 r  y  .  8  8  8  0 .  4 24  8 16 16 16 16  0 0 None PXW
0x137 24  tc  0  32  0 r  y  .  8  8  8  8 .  4 24  8 16 16 16 16  0 0 None PXW
0x138 24  dc  0  32  0 r  y  .  8  8  8  8 .  4 24  8 16 16 16 16  0 0 None PXW
0x139 24  tc  0  24  0 r  .  .  8  8  8  0 .  4 24  8 16 16 16 16  0 0 None PXW
0x13a 24  dc  0  24  0 r  .  .  8  8  8  0 .  4 24  8 16 16 16 16  0 0 None PXW
0x13b 24  tc  0  32  0 r  .  .  8  8  8  8 .  4 24  8 16 16 16 16  0 0 None PXW
0x13c 24  dc  0  32  0 r  .  .  8  8  8  8 .  4 24  8 16 16 16 16  0 0 None PXW
0x13d 24  tc  0  24  0 r  y  .  8  8  8  0 .  4 24  0 16 16 16 16  0 0 None PXW
0x13e 24  dc  0  24  0 r  y  .  8  8  8  0 .  4 24  0 16 16 16 16  0 0 None PXW
0x13f 24  tc  0  32  0 r  y  .  8  8  8  8 .  4 24  0 16 16 16 16  0 0 None PXW
0x140 24  dc  0  32  0 r  y  .  8  8  8  8 .  4 24  0 16 16 16 16  0 0 None PXW
0x141 24  tc  0  24  0 r  .  .  8  8  8  0 .  4 24  0 16 16 16 16  0 0 None PXW
```

Figure 20: Graphic Card Command Output

Check the following information in the output:

- Look at the OpenGL renderer string and make sure it doesn't contain any variant of mesa or software
- Direct Rendering is Yes
- Look for visuals with a class of tc, a buffer size of 24 or 32, a caveat of None, and have a P in the right-most column

12.4 Troubleshoot Result Visualization Services

Troubleshooting information and steps for RVS.

The following section provides the information about troubleshooting information and steps for RVS.

12.4.1 Troubleshoot Checklist

A quick checklist of pre-requisites for RVS.

1. The RVS server should be installed on a supported operating system and web browsers to visualize the result files. For more information refer System Requirements for Access Web.

   Note: Turn off pop-up blockers to view auto-refresh loading message for all the supported web browsers.

2. For Linux OS, ensure HyperMath is working fine before running RVS.
   a) Verify if the shared library is accessible by running HMathserv at: `/opt/2017.2/altair/scriptsh/hmath`
   b) Install, `"libXScrnSaver-1.2.2-6.1.el7.x86_64"` package to resolve the shared library error: `"error while loading shared libraries:libXss.so.1"`.
3. HyperWorks installation should be accessible to the portal user.
4. The portal user should have read access to results files.
5. Before accessing RVS, ensure the older version of HMath session is killed.

   ```
   Use the command, "ps -ef | grep hmathserv" to get the process ids.
   Use the command, "kill -9 <process id>" to kill any HMath sessions.
   ```

12.4.2 Locate Log Files

The file path details to locate RVS core logs, web server logs, and integration logs are listed.

Locating RVS log files locations:

a) RVS core logs: `PA_HOME/logs/resultservice/resultservicelog/resultservice.log`
b) RVS web server logs: `PA_HOME/logs/resultservice/catalina.out`
c) RVS integration logs: `PA_HOME/logs/resultmanager/resulmanager.log`

12.4.3 Common Issues

This section provides list of common issues occurs in RVS. The causes and the resolution steps are provided to resolve these common issues.

Unable to Use Result Visualization Features

Unable to view or perform any of the RVS options.

Review the following checks to enable RVS features.

- The supported RVS result file types, refer Supported Result File Types
- For RVS Troubleshooting Checklist, refer Troubleshoot Checklist

The causes could be:

- The selected result file format is unsupported.
- RVS service has not started.

Follow the given steps:

1. Use the command, `ps -ef | grep resultservice` to check if RVS is running.
2. If RVS service is running, check for the errors in the RVS core logs at `PA_HOME/logs/resultservice/resultservicelog/resultservice.log`.
3. If the RVS service is not running, check for the error messages in the following path: `PA_HOME/logs/resultservice/catalina.out`.
4. Initiate the RVS service by restarting the Access Web using the command:
   ```
   service pbsworks-pa restart
   ```
5. Use the following link: `url:Access Weburl:4443/resultservice/rest/doc` to check if the RVS port page is accessible.

HyperMath and HVTrans License Errors

Unable to post process the result files due to HyperMath and HVTrans license errors.

Update the Altair license path in the `setenv.sh` file. Refer, Configure HyperWorks Licenses.

The causes could be:

- HWHyperMath and HWHyperViewTrans licenses are not available in the HyperWorkslicense server.
- Due to shortage of licenses.
- The License Server is not reachable.

Follow the given steps:

1. Navigate to RVS license configuration location at `PA_EXEC/resultservice/scripts/setenv.sh file`.
2. Verify if the provided license server is reachable.
3. Verify the HWHyperMath and HWHyperViewTrans licenses are valid and available.
4. You can verify the availability of licenses in the Altair Utility.

5. Use the command "`# ps -ef | grep hmathserv`" to check the old instance of HMath sessions.
6. Use the command, "`kill -9 <process id>`" to kill any running or older HMath sessions. Specify the`< process id>` of HMathserv in the following command, "`kill -9 <process id>`"

Unable to Extract TOC of a Result File

RVS is unable to post process the output files.

Perform the checks listed in the Troubleshoot Checklist

The causes could be:

- Insufficient permissions to read the result file.
- HyperWorks is not installed properly.
- May be an issue with the shared library.

Follow the given step:

If the registered user does not have read permission to open the result file update the user's permissions to read the result files.

HyperMath Execution Error: HW_APP_FAILED Message

HyperMath service is not running or does not have the permissions to execute.

Perform the checks listed in the Troubleshoot Checklist.

Follow the given step:

Check if the HWHyperMath service has started successfully.

```
Use the command, ps -ef | grep hmathserv.
```

The following entries shows the HWHyperMath is running successfully.

```
pbsworks 7226 1 0 Sep26 ? 00:00:00 /bin/sh /opt/hw2017.2/altair/scripts/hmathserv -port
 5678
pbsworks 7227 7226 0 Sep26 ? 00:00:05 /opt/hw2017.2/altair/hwx/bin/linux64/hmathserv -
port 5678
```

Result File Reader is Unrecognized

The error, HW File Error: File not recognized displays when you try to plot or animation.

- To check the supported result file list, refer Supported Result File Types.
- To configure a solver file reader, refer Configure Solver File Readers

The error occurs if a result solver file reader is not registered in HyperWorks.

The causes could be:

- The result file reader is not configured in HyperWorks.

Follow the given steps:

1. Verify if your result file reader is configured.

For the Fluent file reader, the entry should be as following:

```
*RegisterExternalReader({external_readers_dir + "/hgfluent.exe"}, "", "", ascii)
```

2. If not configured, then you need to configure the solver file reader.

HML Script Execution Failed

This error occurs due to Hyperworks license are not configured properly.

- For RVS Troublehsooting Checklist, refer Troubleshoot Checklist.

Follow the steps given here:

1. Resolve any HyperWorks license related issues, verify the license file at HyperWorks at `PA_EXEC/resultservice/scripts/setenv.sh`.
2. If the error still persist, run through the RVS troubleshooting checklist.