Altair Access Web 2022.1.0

Administrator's Guide

Updated: 05/23/2022

altair.com

Intellectual Property Rights Notice

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

Altair PBS Works™ v. 2022.1.0 Copyright © 1994-2022.

Altair Engineering Inc. Copyright © 1986-2022. All Rights Reserved.

Copyrights in the below are held by Altair Engineering, Inc., except where otherwise explicitly stated. This Intellectual Property Rights Notice is exemplary, not exhaustive.

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

Altair HPC & Cloud Products

Altair® PBS Professional® ©1994-2022

Altair Control™ ©2008-2022; (formerly **PBS Control**)

Altair Access™ ©2008- 2022; (formerly **PBS Access**)

Altair Accelerator™ ©1995- 2022; (formerly **NetworkComputer**)

Altair Accelerator Plus™ ©1995- 2022; (formerly **WorkloadXelerator**)

Altair FlowTracer™ ©1995- 2022; (formerly **FlowTracer**)

Altair Allocator™ ©1995- 2022; (formerly **LicenseAllocator**)

Altair Monitor™ ©1995- 2022; (formerly **LicenseMonitor**)

Altair Hero™ ©1995- 2022; (formerly **HERO**)

Altair Software Asset Optimization™ (SAO) ©2007- 2022

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

Altair Simulation Products

Altair AcuConsole™ ©2006-2022

Altair AcuSolve™ ©1997-2022

Altair Activate® ©1989-2022 (formerly **solidThinking Activate**)

Altair Compose® ©2007-2022 (formerly **solidThinking Compose**)

Altair ConnectMe™ ©2014-2022

Altair EDEM™ ©2005-2022 DEM Solutions Ltd, © 2019-2022 Altair Engineering, Inc.

Altair ElectroFlo™ ©1992-2022

Altair Embed® ©1989-2022 (formerly **solidThinking Embed**)

- **Altair Embed SE** ©1989-2022 (formerly **solidThinking Embed SE**)
- **Altair Embed / Digital Power Designer** ©2012-2022
- **Altair Embed Viewer** ©1996-2022

Altair ESAComp™ ©1992-2022

Altair Feko™ ©1999-2014 Altair Development S.A. (Pty) Ltd.; ©2014-2022 Altair Engineering Inc.

Altair Flux™ ©1983-2022

Altair FluxMotor™ ©2017-2022

Altair HyperCrash™ ©2001-2020

Altair HyperGraph™ ©1995-2022

Altair HyperLife™ ©1990-2022

Altair HyperMesh™ ©1990-2022

Altair HyperStudy™ ©1999-2022

Altair HyperView™ ©1999-2022

Altair HyperWorks™ ©1990-2022

Altair HyperXtrude™ ©1999-2022

Altair Inspire™ ©2009-2022 including **Altair Inspire Motion**, **Altair Inspire Structures**, and **Altair Inspire Print3D**

Altair Inspire Cast ©2011-2022 (formerly **Click2Cast**)

Altair Inspire ElectroFlo ©1992-2022

Altair Inspire Extrude Metal ©1996-2022 (formerly **Click2Extrude - Metal**)

Altair Inspire Extrude Polymer ©1996-2022 (formerly **Click2Extrude - Polymer**)

Altair Inspire Form ©1998-2022 (formerly **Click2Form**)

Altair Inspire Friction Stir Welding ©1996-2022

Altair Inspire Mold ©2009-2022

Altair Inspire PolyFoam ©2009-2022

Altair Inspire Play ©2009-2022

Altair Inspire Render ©1993-2016 Solid Iris Technologies Software Development One PLLC, © 2016-2022 Altair Engineering Inc (formerly **Thea Studio**)

Altair Inspire Resin Transfer Molding ©1990-2022

Altair Inspire Studio ©1993-2022 (formerly '**Evolve**')

Altair Manufacturing Solver™ © 2011-2022

Altair Material Data Center ©2019-2022

Altair MotionSolve™ ©2002-2022

Altair MotionView™ ©1993-2022

Altair Multiscale Designer™ ©2011-2022

Altair nanoFluidX™ ©2013-2018 Fluidyna GmbH, © 2018-2022 Altair Engineering Inc.

Altair newFASANT ©2010-2022

Altair OptiStruct™ ©1996-2022

Altair PollEx ©2003-2022

Altair Radioss™ ©1986-2022

Altair Seam™ © 1985-2019 Cambridge Collaborative, Inc., © 2019-2022 Altair Engineering Inc.

Altair SimLab™ ©2004-2022

Altair SimSolid™ ©2015-2022

Altair ultraFluidX™ ©2010-2018 Fluidyna GmbH, © 2018-2022 Altair Engineering Inc.

Altair Virtual Wind Tunnel™ ©2012-2022

Altair WinProp™ ©2000-2022

Altair WRAP ©1998-2020 WRAP International AB, © 2020-2022 Altair Engineering AB

Altair Packaged Solution Offerings (PSOs)

Altair Automated Reporting Director™ ©2008-2022

Altair GeoMechanics Director™ ©2011-2022

Altair Impact Simulation Director™ ©2010-2022

Altair Model Mesher Director™ ©2010-2022

Altair NVH Director™ ©2010-2022

Altair Squeak and Rattle Director™ ©2012-2022

Altair Virtual Gauge Director™ ©2012-2022

Altair Weight Analytics™ ©2013-2022

Altair Weld Certification Director™ ©2014-2022

Altair Multi-Disciplinary Optimization Director™ ©2012-2022

Altair Data Analytics Products

Altair Knowledge Studio® © 1994-2020 Angoss Software Corporation, © 2020-2022 Altair Engineering, Inc.

ALTAIR

Altair Knowledge Studio for Apache Spark © 1994-2020 Angoss Software Corporation, © 2020-2022 Altair Engineering, Inc.

Altair Knowledge Seeker™ © 1994-2020 Angoss Software Corporation, © 2020-2022 Altair Engineering, Inc.

Altair Knowledge Hub™ © 2017-2020 Datawatch Corporation, © 2020-2022 Altair Engineering, Inc.

KnowledgeWorks™ © 2022 Altair Engineering, Inc.

Altair Monarch™ © 1996-2020 Datawatch Corporation, © 2020-2022 Altair Engineering, Inc.

Altair Monarch Server © 1996-2020 Datawatch Corporation, © 2020-2022 Altair Engineering, Inc.

Altair Panopticon™ © 2004-2020 Datawatch Corporation, © 2020-2022 Altair Engineering, Inc.

Altair SmartWorks™

Altair SmartCore™ © 2011-2022

Altair SmartEdge™ © 2011-2022

Altair SmartSight™ © 2011-2022

Altair One™ ©1994-2022

Altair intellectual property rights are protected under U.S. and international laws and treaties. Additionally, Altair software may be protected by patents or other intellectual property rights. All other marks are the property of their respective owners.

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

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

Third party software licenses

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

Software Security Measures:

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

Technical Support

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

Telephone and E-mail

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

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

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

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

Contents

What's New

New features available in Altair Access™ Web.

Next-Generation of Remote Desktops

Remote Desktops is the new and improved version of Remote Sessions. It offers better performance and consumes lesser resources by using H.264, HTML5, Web-assembly, GPU and CPU based encoding and decoding technologies. Remote Desktops can be configured and installed using scripts that are packaged with the Access Web installer.

Remote Desktops Enhancements

You can now further enhance the Remote Desktops experience by adjusting the quality settings, downloading performance statistics and enabling debugging logs for troubleshooting.

Remote Desktops Application Definitions

The new Remote Desktops application definitions require a one-time update of existing Remote Desktops application definitions due to the change in the underlying technology stack. New application definitions can be created and published using the application composer or the application definitions examples. Once the application definitions are created, framework updates can be delivered seamlessly in the future.

Security Vulnerabilities

Log4j is updated to 2.17.1 and the CVE-2021-44228, CVE-2021-45105, CVE-2021-45046, and CVE-2021-44832 vulnerabilities are fixed. Third-party libraries are updated to their latest stable versions to fix library related security issues.

Windows and Linux Compute Nodes

Access Web supports both Linux and Windows execution hosts.

Job submitted to Linux or Windows execution node in PBS Professional is based on the architecture defined in the application definition.

Share Audit Log with Altair

The portal administrator and audit administrator can share audit logs with Altair from Access Web.

Resumable File Upload and Download

File upload and download will resume even if network connection is interrupted.

Enhancement of Site Specific Pages

The configured site-specific web pages will now be displayed at the top-level along with Files, Jobs, and Desktops. The site-specific web pages are opened in a new tab and a More drop-down menu is displayed if there are too many pages configured.

The Administrator has the capability to grant and restrict access to the site-specific pages using the Access Control Management.

Job Submission Page Enhancements

The profile related items are grouped in the Job Submission form. The application definition profiles created by the user are displayed as a drop-down menu in the Job Submission form.

Job Monitoring Enhancements

The job filter in the Job Monitoring page provides an option to filter the jobs based on Interactive Jobs, Batch Jobs and Array Jobs.

Support for HyperWorks Desktop and Compose 2021.2

RVS has now been enhanced to use HyperWorks Desktop and Compose 2021.2 for Visualizations. Since it uses the features of the 2021.2 version of HyperWorks Desktop and Compose, older versions are not supported.

Overview

This chapter covers the following:

2.1 Document Conventions

Common typographical conventions for Altair Access Web™ technical publications.

PA_HOME

> The Access Web home directory which contains configuration, data, and logging files. Default location is: `/var/spool/pbsworks/2022.1.0/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/2022.1.0/access/exec`. However, this can be overridden during the installation of Access Web.

PBS_HOME

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

> `/var/spool/pbs`

PBS_EXEC

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

> `/opt/pbs/`

2.2 About Access Web

Use Access Web to submit jobs to a Workload Manager.

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 view 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

2.3 System Requirements

System requirements for all components of Access Web.

2.3.1 Access Web System Requirements

Supported platforms and hardware requirements.

Supported Platforms

Access Web is supported on the following Linux 64-bit (x86_64) platforms:

- Red Hat Enterprise Linux (RHEL) 7.1 to 7.9, 8.0 to 8.5
- CentOS 7.1 to 7.9
- SUSE Enterprise Linux Server (SLES) 12 SP1 to 12 SP3, 12 SP5, and 15
- Windows only for Execution Nodes
- Ubuntu 18 platform support for Execution Node

Hardware Requirements

The minimum hardware configuration required for Access Web is:

> 📝 **Note:** PBS and PAS should be on the same server. Session Proxy should be a separate server with 10 Gigabit interconnection.

Table 1: Minimum Requirement

Hardware	Minimum Requirement Up to 50 Concurrent Users
CPU	2 CPU cores with a minimum speed of 2.5 GHz
Memory (Physical)	8 GB
Disk Space	100 GB

> 📝 **Note:** Scalability hardware requirement mentioned below is for Access Web. The PBS and PAS server should be installed on a different machine.

Table 2: Hardware Requirement for Scalability
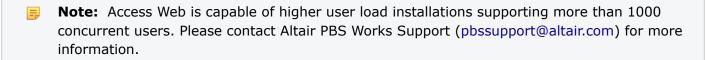
Hardware	300 Users	500 Users	1000 Users
CPU	4 CPU cores with a minimum speed of 2.5 GHz	16 cores with a minimum speed of 2.5 GHz	24 cores with a minimum speed of 2.5 GHz
Memory (Physical)	16 GB	64 GB	128 GB
Disk Space	500 GB	1 TB (Stage area to be considered according to site requirements)	1 TB (Stage area to be considered according to site requirements)
Network Connectivity	10 Gigabit	10 Gigabit	20 Gigabit

Configure your site using the deployment profiles provided with Access Web. Refer to Configure Deployment Profile for more information.

The minimum hardware configuration required for Remote Desktops Proxy is:

Hardware	Recommended
CPU	1 core per remote desktop and a total of 8 cores for scalability
Memory (Physical)	16 to 32 GB
Disk Space	100 GB
Network	15 to 20 Mbps per remote desktop

📝 **Note:** On cloud deployment, the vCPU should be twice the core mentioned above (double the core counts ie., 2 x cores)

📝 **Note:** Access Web is capable of higher user load installations supporting more than 1000 concurrent users. Please contact Altair PBS Works Support (pbssupport@altair.com) for more information.

See Also
Configure the Desktop Manager for Interactive Applications

△ ALTAIR

2.3.2 Supported Browsers

Supported Browsers in Access Web.

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 Edge

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
- Microsoft Edge

2.3.3 Remote Desktops System Requirements

Requirements for the Remote Desktops component that is installed on the execution hosts.

To run interactive applications on the HPC execution hosts ensure that following requirements are met.

Hardware Requirements
It is recommended to size your hardware based on the number of users who will be running remote desktops concurrently. Additional CPUs, GPUs, disk space and memory are required specific to the interactive applications being run.

Hardware	Minimum Requirement
CPU	1 CPU core
Disk Space	1 GB
Memory	1 GB

Desktop Manager

By default, the Remote Desktops component installed on the PBS execution hosts to run interactive applications uses GNOME 3 as a Desktop Manager. The default is GNOME 3 and if GNOME 3 is not installed on the PBS execution hosts as part of the Linux distribution, then it must be installed. If required, user can change Linux Desktop Manager. Refer to Configure the Desktop Manager for Interactive Applications.

Graphics Card

Supported GPU: nvfbc and nvenc to support GRID, Tesla, or Quadro X2000+ (X = K/M/P/RTX etc) hardware products.

If your site chooses to install a graphics card, it must meet these requirements:

- The drivers provided by the manufacturer of the graphics card must be installed.
- NVIDIA Windows display driver 471.41 or newer
- NVIDIA Linux display driver 470.57.02 or newer
- DirectX SDK (Windows only) CUDA 11.0 Toolkit

Encryption

Use Virtual Private Networking or secured channels for communication between clients and the interactive server if encryption is required.

X Server

Accessing a 3D graphics card on Linux requires going through an X server. Therefore:

- For running remote desktops, X Server must be configured and working.
- Target interactive applications must be installed, configured and working (i.e., verify the interactive application is working by plugging a monitor into the machine that the application is running on to verify that is being displayed properly).

2.3.4 Results Visualization System Requirements

System requirements for installing and using the Results Visualization Service.

HyperWorks Desktop

Results Visualization Service is automatically installed with Access Web. To extract animation data Altair HyperWorks Desktop must be installed. Access Web and HyperWorks Desktop can be installed on the same machine or separate machines. If HyperWorks Desktop is installed on a separate machine, it can be made accessible to Access Web by creating a mounting point for the HyperWorks installation directory on the machine where Access Web is installed.

Altair Compose

To extract plot data Compose must be installed. Access Web and Compose can be installed on the same machine or separate machines. If Compose is installed on a separate machine, it can be made accessible to Access Web by creating a mounting point for the Compose installation directory on the machine where Access Web is installed.

> 📝 **Note:** LSDyna templates created in previous versions of Access Web are not supported. Those older templates use older version of Compose, hence you must create new templates.

Network Connectivity

RVS requires good network connectivity to all the connected file servers such as the PAS Server and the PBS execution hosts. A minimum speed of 100mbps is required while a speed of 1gbps is recommended.

Recommendations

When Access Web and PAS are installed separately, it is recommended to mount the PAS staging directory and scratch directory on the Access Web Server to improve performance.

2.4 Supported Product Configurations

Supported product configurations for using Access Web.

When installing Access Web, all components including PAS and Remote Desktops must be the same version.

> 📝 **Note:** Altair License Server 14.5.1 or newer is required for Access Web 2022.1.0.

Linux Support

The currently supported Access Web product configurations for Linux are:

Access Web	PBS Professional	HyperWorks	Compose
2022.1.0	2021.1.2 2021.1.1 2021.1.0 2020.1.4 2020.1.3	2021.2	2021.2

Altair Grid Engine Support

The currently supported Access Web product configurations for Altair Grid Engine (AGE) are:

Access Web	Altair Grid Engine	HyperWorks	Compose
2022.1.0	8.7.0 8.6.17	2021.2	2021.2

Result Visualization

- Altair HyperWorks Desktop and Compose are required to visualize CAE results.

> 📝 **Note:** Compose 2021.2 does not support the templates created for LSDyna files by older version of Compose. While upgrading the Access Web to 2022.1.0, you must delete the older LSDyna files related templates and create a new one.

2.5 Authentication and Authorization

Authentication

HPC administrators spend a lot of time and energy deciding upon and implementing authentication policies for various machines in their HPC. Keeping this in mind, the most straightforward way to setup authentication for web portals, like Access Web, is to authenticate against the local machine. In this way, users who can login to the machine where Access Web is installed can also login to Access Web. This strategy provides a turn-key installation without having to configure Access Web with Active Directory or SSH server details.

Therefore, by default, authentication for Access Web is handled thorough SSH2 on Linux platforms. Access Web assumes that if a user can login to the machine where Access Web is installed, then they should be able to login to Access Web. When managing authentication using services such as NIS or Active Directory, it is recommended to configure these services directly with SSH.

While the above defaults should work for most sites, Access Web also comes with a plug-in based architecture to accommodate more complex authentication use cases and it is possible to extend the default authentication mechanism. Please consult Altair professional support services for more information.

Authorization

The Service User is automatically assigned managerial privileges (is assigned a Manager role) and has unrestricted access to all features of Access Web. Actions that a Manager can do that a regular user cannot:

- add, edit, and delete server clusters (a PAS Server that is connected to an HPC complex).
- restrict other user's access to applications.
- assign managerial privileges to other users.

Regular users (those that are not assigned a Manager role) have access to all features of Access Web except for those detailed above.

Job Submission User Authorization

When a user submits a job from a system other than the one on which the PBS server is running, system-level user authorization is required. This authorization is needed for submitting the job and for PBS to return output files. For more information on how to set up this authorization see *Linux User Authorization* or *Windows User Authorization* in the *PBS Professional Installation & Upgrade Guide*.

See Also
Establish Access Controls

2.6 PBS Works Licensing

Licensing model for Access Web.

Licenses must be installed and available via an Altair License Server version 14.5.1 or newer. Licenses consumed by Access Web are:

PBSWorksUsers

> This is the number of simultaneously logged in users; each user who logs in checks out one PBSWorksUsers license.

> If a user logs into Access Web and then opens a second browser window to access Access Web, a single PBSWorksUsers license is consumed.

> PBSWorksUsers stack, not level, across products. So if 10 users are logged into Access, and 10 users are logged into Control, 20 PBSWorksUsers are checked out, even if they are the same users.

2.7 Architecture

2.7.1 Access Web Components

Components and their associated responsibility.

Access Web consists of a collection of components and their associated services, accessed and invoked via a common interface.

Access Web Service

Access Web is a job submission and monitoring portal which enables users to run, monitor and manage workloads on distributed resources remotely. Access Web uses the High-Performance Computing (HPC) workload management capability of Altair's PBS Application Services (PAS) and Workload Manager (PBS Professional or Altair Grid Engine) for efficient resource utilization, access to the cloud for job submission, and subsequent accounting.

Results Visualization Service

Results Visualization Service (RVS) provides features to access, process and visualize CAE results from anywhere, without installing any post-processing desktop applications through the user friendly web interface of Access Web. CAE analysts can monitor simulations in real-time by tracking and visualizing relevant parameters from solver log files. Meaningful plots and animations can also be created remotely without downloading huge raw results files from compute clusters or remote networks locations. A variety of FEA and MBD solvers are supported - see Supported Result File Types.

RVS is automatically installed with Access Web. To extract plot and post process the plot results Altair Compose must be installed. To extract animation data Altair HyperWorks Desktop must be installed.

PBS Application Services

PBS Application Services (PAS) is a middleware component that communicates with the Workload Manager (PBS Professional or Altair Grid Engine) 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 Workload Manager (PBS Professional or Altair Grid Engine) complex.

PAS can be installed with Access Web or it can be installed separately on the PBS or AGE Server.

Remote Desktops

Remote Desktops provides a platform for collaboration and remote visualization of graphics-intensive applications and data. It enables high performance remote display of applications over the networks using a web browser. Users can share remote desktops fostering teamwork and collaboration.

Depending upon the deployment option selected, Remote Desktops components must be distributed across several machines. Configuration of Remote Desktops components is done through a script provided by Access Web.

Configure Workload Manager PBS to support interactive applications:

- add a custom resource to PBS Professional called *ndesktops*
- add an interactive queue called "iworkq"
- updates the PBS ndesktops resource on provided graphics node by requesting the list of graphics node and number of ndesktops available.

Install binaries and configure the execution hosts (PBS MoM or AGE) to support interactive applications:

- Install Remote Desktops agent.
- The script provides permission to start X server by all the users and extract platform specific agent.

2.7.2 Basic Architecture Overview

Basic architecture of Access Web.

The basic architecture including the services and third party dependencies of Access Web is described as follows:
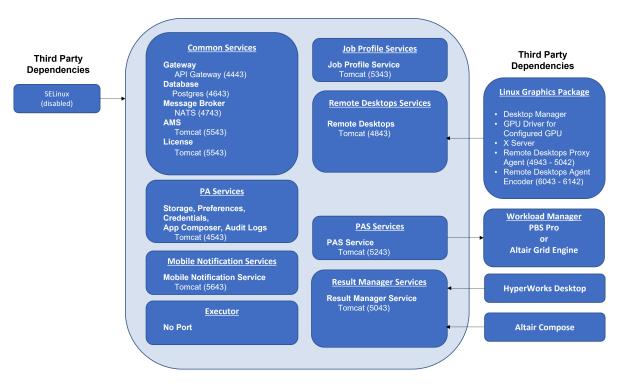
Figure 1: Basic Architecture Overview

2.7.3 File Staging Local Filesystem vs Shared Filesystem

File staging requirements for a local filesystem in comparison to a shared filesystem.

File Staging of Local Files

For a user to submit a job, job input files are required such as scripts, master files, includes files, etc. These files are typically:

- uploaded to the Access Web server and stored in a user directory, for example `/home`.
- located on a remote system, for example stored on a user's laptop or workstation.

When selecting job input files that are located on a user's laptop or workstation during job submission, the job files are copied to the Job File Stage Area. The Job File Stage Area is defined when installing PAS (default is `/stage` on Linux). Once the job is submitted, the job files are copied from the Job File Stage Area to the HPC complex. Once the job completes, the job files are copied back to the Job File Stage Area.
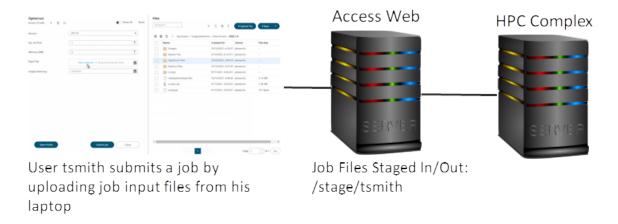
Figure 2: File Staging from User's Laptop or Workstation

Job files that are stored on the Access Web server are copied to the HPC complex from the location where the job files are stored. Once the job completes, the job files are copied back to the same location and placed in a job output directory.
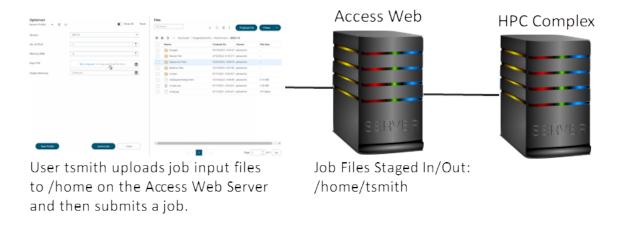
Figure 3: File Staging from User Directory on Access Web Machine

The mechanism for copying the job files between the Access Web server to the HPC complex is defined and configurable at the PBS level. For more information see *Setting File Transfer Mechanism* in the *PBS Professional Administrator's Guide*. By default, the copy mechanism depends on whether the job files are remote or local files. To copy remote files, system-level user authorization is required. Several methods for setting up this authorization are outlined in the *PBS Professional Installation & Upgrade Guide*. For example, through SSH key-based authentication or host-based authentication.

File Staging on a Shared Filesystem
A distributed filesystem can be mounted to share the following directories:

- User directories (ex. `/home`)
- Job File Stage Area (ex. `/stage`)
- Scratch directory (ex. `/scratch`)
- Application directory where solvers are installed (ex. `/software`)

This avoids having to remotely copy job input files, avoids having to distribute solvers across execution hosts, and does not require SSH keys to be generated and distributed across machines when user-ssh key authentication is being used.

> 📝 **Note:** Application definitions must be modified so that staging of job files is done correctly when using a shared filesystem.
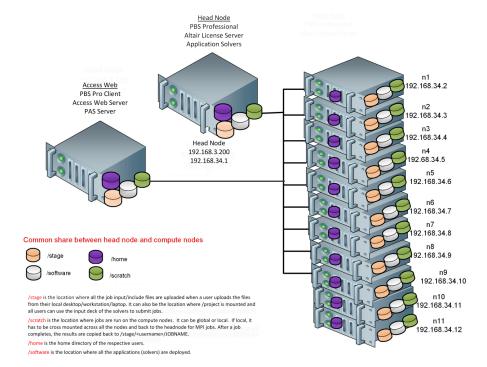
Figure 4: Shared Filesystem

See Also
Shared File System Support

2.7.4 Ports and Service Memory Usage

Overview of ports and memory used by Access Web services.

Ports Used by Access Web

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

The Access Web installer has auto-port detection logic in place and ports are chosen for each service within a 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 used by Access Web and its services.

Ports used by Access Web

Port	Port Range	Service Using the Port	Description
4443	4443 - 4542	Gateway	Orchestrator for all access services.
4543	4543 - 4642	Access Web Server	Service which provides live job updates, application definition composer and audit logs.
4643	4643 - 4742	Postgres Database	Service for data storage.
4743	4743 - 4842	NATS Message Broker	Service to provide async (event base) internal communication.
5343	5343 - 5442	Job Profiles Service	On-demand service to provide Access Desktop central repository for application definitions.
5543	5543 - 5642	AMS Service	Service to provide Authentication and Authorization.
5643	5643 - 5742	Mobile Notification Service	On-demand service to provide push notification support for Access Mobile.

Ports used by Results Visualization Service (RVS)

Port	Port Range	Service Using the Port	Description
5043	5043 - 5142	Result Manager Service	Service to provide integration between Access and RVS.

Ports used by PBS Application Services (PAS)

Port	Port Range	Service Using the Port	Description
5243	5243 - 5342	PAS	Service for Files and HPC operations.

Ports used by Remote Desktops

Port	Port Range	Service Using the Port	Description
4843	4843 - 4942	Remote Desktops Service Webserver	Service to provide Remote Desktops capabilities.
4943	4943 - 5042	Remote Desktops Job agent proxy service.	Service used to stream the remote desktop.
5943		Remote Desktops Edge Gateway for Desktop Remote Desktops	Service used for edge gateway for web Remote Desktops
6043	6043 - 6142	Remote Desktops Job agent encoder service.	Service to capture and encode desktop stream. Used within graphical MoM.

Open Ports

It is recommended to configure a firewall on the Access Web server and block all Access Web ports from the outside world except:

- 4443
- 4743
- 4943 - 5042

See Also
Change Port Numbers

Memory Usage by Service

Overview of default Java heap space memory usage by each service.

The table below provides the amount of memory used by Access Web services and the file where this memory value can be configured post-installation.

The amount of memory can be adjusted by editing the associated configuration file and increasing the default value to a larger value. For example, when OutofMemory errors are noted in the Access Web log files.

Table 3: Memory Usage by Service

Service	Configuration Path and File	Value
Gateway	`PA_HOME/config/shared/deployment-config.yml`	*512 MB*
Access Web Server	`PA_HOME/config/shared/deployment-config.yml`	*1024 MB*
PAS	`PA_HOME/config/shared/deployment-config.yml`	*1024 MB*
Remote Desktops Services Webserver	`PA_HOME/config/shared/deployment-config.yml`	*512 MB*
Result Manager Services	`PA_HOME/config/shared/deployment-config.yml`	*512 MB*
Job Profiles Services	`PA_HOME/config/shared/deployment-config.yml`	*512 MB*
AMS Services	`PA_HOME/config/shared/deployment-config.yml`	*512 MB*
Mobile Notification Service	`PA_HOME/config/shared/deployment-config.yml`	*512 MB*
Executor	`PA_HOME/config/shared/deployment-config.yml`	*512 MB*

See Also
Change Memory used by the Services

Prepare for Installation or Upgrade

<div style="text-align: right">**3**</div>

Deployment options, installation prerequisites, and requirements for the Service User.

This chapter covers the following:

3.1 Deployment Options on Linux

Overview of deployment options for Linux platforms.

Access Web consists of a collection of components and their associated services, accessed and invoked via a common interface.

Information requested during installation of Access Web:

- License server
- Service user
- Installation location for the home (PA_HOME) and execution directories (PA_EXEC)
- HyperWorks installation location (for RVS features)
- Compose installation location (for RVS features)

Information requested during installation of PAS:

- Staging directory
- Installation location for the home (PA_HOME) and execution directories (PA_EXEC)

Information requested during the configuration of Remote Desktops:

- Number of Graphics node and number of ndesktops available on the execution host

3.1.1 Deployment Option 1

Install Access Web and PAS on a single machine and then install the Remote Desktops components on the HPC cluster.

Figure 5:Deployment Option 1

Deployment Option 1

Server - A

Access Web / PAS Server

Access Web/PAS Installation
• Access Web services installed
• PAS services installed
• ShellScript application definition installed

Server - B

Workload Manager Server (PBS or AGE)

Remote Desktops Installation
• Add iworkq queue
• Add custom resources "ndesktops"
• Set resource_available.ndesktops

Server - C

Execution Nodes (PBS Pro or AGE)

Remote Desktops Installation
• Remote Desktops Agent

📝 **Note:** The PBS 2020.1 MoMs can be on Hybrid Linux or Windows.

Deploy Access Web and PAS on the PBS or Altair Grid Engine Server

Install Access Web and PAS on the PBS or Altair Grid Engine Server.

This deployment option is best suited for smaller sites that do not have spare hardware for installing Access Web and PAS on a stand-alone machine, have a small set of users, and do not have high utilization such that Access Web and PAS will place too much of a load on the PBS or Altair Grid Engine Server.

Installation Steps

1. Install Access Web and PAS on the PBS or AGE Server.
    • Installs Access Web and PAS services.
    • Installs ShellScript application definition.
    • Default server cluster is automatically added.
    • Configures PAS for Remote Desktops.
2. Set up Modern Communication.
    • Encapsulates Remote Desktops.
3. Install the Remote Desktops component on the PBS or AGE Server.
    • Configures the PBS or AGE Server and PAS for Remote Desktops.
    • Adds iworkq queue.

△ **ALTAIR**

- Adds custom resource *ndesktops*.
- Sets the number of *ndesktops* available on the node.

4. Configure the Remote Desktops component on the execution hosts (PBS MoM or AGE).

- Configure Remote Desktops Agent.

Deploy Access Web and PAS on a Separate Server

This is the recommended deployment option for larger sites with a single HPC cluster. It insulates the PBS Server so that PBS is not fighting for resources with Access Web and PAS.

This deployment option requires that the PBS Client be installed on the machine hosting Access Web/PAS and the PBS Client must be configured to point to the PBS Server.

Installation Steps

1. Install the PBS Client on Server-A.
2. Install Access Web and PAS on Server-A.

- Installs Access Web and PAS services.
- Installs ShellScript application definition.
- Default server cluster is automatically added.

3. Set up Modern Communication.

- Encapsulates Remote Desktops.

4. Install the Remote Desktops component on the PBS or AGE Server..

- Configures the PBS Server for Remote Desktops.
- Adds iworkq queue.
- Adds custom resource *ndesktops*.
- Sets the number of *ndesktops* available on the node.

5. Configure Remote Desktops component on the execution hosts (PBS MoM or AGE).

- Configure Remote Desktops Agent.

6. Configure the Remote Desktops component on Server-A.

- Configures PAS for Remote Desktops.

3.1.2 Deployment Option 2

Install Access Web on a standalone machine and install PAS on the PBS Server.

This is the recommended deployment option for larger sites with multiple HPC complexes. PAS must be installed on the PBS Server of each HPC complex.

Deployment Option 2

Server - A	**Server - B**	**Server - C**
Access Web Server	**Workload Manager Server (PBS or AGE) / PAS Server**	**Execution Nodes (PBS Pro or AGE)**
Access Web/PAS Installation • Access Web services installed	PAS Installation • PAS services installed • ShellScript application definition installed Remote Desktops Installation • Add iworkq queue • Add custom resources "ndesktops" • Set resource_available.ndesktops	Remote Desktops Installation • Remote Desktops Agent

Figure 6: Deployment Option 2

> 📝 **Note:** The PBS 2020.1 MoMs can be on Hybrid Linux or Windows.

Server Clusters

A server cluster is a PAS Server that is connected to an HPC complex. After installation, the Service User must add server clusters to Access Web so that users can submit jobs to an HPC complex. Access to the local filesystem is also configured through the server cluster. When adding a server cluster, configure access to any directories that users may need access to such as user home directories, for example `/home`.

Installation Steps

1. Install Access Web on Server-A.
   - Installs Access Web services.
2. Install PAS on the PBS Server.
   - Installs PAS services.
   - Installs the ShellScript application definition.
3. Set up Modern Communication.
   - Encapsulates Remote Desktops.
4. Install the Remote Desktops component on the PBS or AGE Server.
   - Configures the PBS or AGE Server and PAS for Remote Desktops.

- Adds iworkq queue.
- Adds custom resource *ndesktops*.
- Sets the number of *ndesktops* available on the node.

5. Configure the Remote Desktops component on the execution hosts (PBS MoM or AGE).

- Configure Remote Desktops Agent.

3.2 Prerequisites for Installation

Prerequisites and planning for installing Access Web, PBS Application Services, and Remote Desktops.

3.2.1 Prerequisites for Installing Access Web

Mandatory requirements for installing Access Web.

The following are the general prerequisites for installation:

- Installation must be done as root or as a user with sudo permissions.
- Installation must be done on a machine running on a supported platform and meet system requirements.
- You will be prompted for a license server during the installation of Access Web, therefore an Altair License Server version 14.5.1 or newer must be installed prior to installing Access Web.
- You will be asked to provide a username that will be the Service User during the installation of Access Web. Review the requirements for this user account before installing Access Web.
- PBSWorksUsers licenses must be purchased.
- Zip and Unzip package must be installed and available.
- The Access Web Server must be reachable and its hostname resolvable from the HPC complex execution hosts.

> 📝 **Note:** A license server and a Service User are not required for installing PAS or the remote desktops components.

PBS Client

For deployment option 1, when installing Access Web and PAS on a machine separate from the PBS Server, the PBS Client must be installed before installing Access Web. For more information see *Install PBS on Client Hosts* in the *PBS Professional Installation and Upgrade Guide*.

Altair Grid Engine Client (AGE)

For deployment option 1, when installing Access Web and PAS on a machine separate from the AGE Server, the AGE Client must be installed before installing Access Web.

Time Synchronization

The clocks on the Access Web server and the PBS or AGE Server must be synchronized using NTP or some other kind of mechanism. When clocks are asynchronous, operations such as getting job updates, files update times, etc. will not work correctly.

PBS Professional (PBS)

Ensure that a supported version of PBS Professional is installed.

△ **ALTAIR**

Altair Grid Engine (AGE)

Ensure that a supported version of Altair Grid Engine is installed.

PBS Application Services (PAS)

When deploying Access Web such that PAS is installed on the PBS Server (Deployment Option 2), the version of PAS installed on the PBS Server must match the version of Access Web that is being installed.

PAS Staging Directory

During the installation of PAS, you will be prompted to enter a value for the staging directory (referred to as the Job File Stage Area by the installer). 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` on Linux.

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

Following are considerations for selecting 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.

Compose

To extract plot data Compose must be installed. Access Web and Compose can be installed on the same machine or separate machines. If Compose is installed on a separate machine, it can be made accessible to Access Web by creating a mounting point for the Compose installation directory on the machine where Access Web is installed.

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

On Linux platform, install these library packages for Compose:

- For SLES 12 and SLES 15, install the library `libgomp` using the command `zypper install libgomp` and the library `libXss` using the command `zypper install libXss`.

Altair HyperWorks Desktop

Results Visualization Service is automatically installed with Access Web. To extract animation data Altair HyperWorks Desktop must be installed. Access Web and HyperWorks Desktop can be installed on the same machine or separate machines. If HyperWorks Desktop is installed on a separate machine, it can be made accessible to Access Web by creating a mounting point for the HyperWorks installation directory on the machine where Access Web is installed.

To install HyperWorks Desktop, follow the instructions in the *Altair HyperWorks Installation Guide*. The installation should be in a location where all users have read and write permission. It should not be in `/root` or any user's home directory.

On Linux platform, install these library packages for Altair HyperWorks Desktop:

- For SLES 12 and SLES 15, install the library `libgomp` using the command `zypper install libgomp` and the library `libXss` using the command `zypper install libXss`.

RVS Network Connectivity

RVS requires good network connectivity to the PAS Server and the HPC execution hosts. A minimum speed of 100mbps is required while a speed of 1gbps is recommended.

> ⓘ   **Tip:** When PAS is installed on a machine separate from Access Web, mount the staging directory and scratch directory on the Access Web machine to improve performance of RVS.

Configure SSH to Allow Username-Password Authentication

SSH may be configured so that username-password authentication is disabled for cloud provider virtual machines such as Azure. Verify that the parameter *PasswordAuthentication* is set to 'yes' in `/etc/ssh/sshd_config`. Changing this parameter requires reloading SSHD (`systemctl reload sshd`).

SELinux Status

The SELinux must be disabled before installing Access Web. Check the SELinux status by using the command:

```
sestatus
```

See Also
System Requirements
Supported Product Configurations
PBS Works Licensing
Service User

3.2.2 Prerequisites for Installing Remote Desktops

Mandatory requirements for Remote Desktops components.

The following are the general prerequisites for installation:

- Installation must be done as root or as a user with sudo permissions.
- Installation must be done on a machine running on a supported platform and meet system requirements.
- Access Web must be installed prior to installing the Remote Desktops components.

Host Name and Port

Access Web communicates with the PBS MoMs through the Remote Desktops Service Job Update 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.

Prerequisites for Installing on the PBS MoM

- By default, the Remote Desktops component assumes that Gnome is being used as the Remote Desktops Desktop Manager. If Gnome is not installed on the PBS MoM as part of the Linux distribution, then it must be installed.
- Graphics cards, if necessary, are installed and configured properly.
- The PBS Professional or Altair Grid Engine execution host must be able to access and run the interactive applications available through Access Web.

Remote Desktops Application Definitions

The new Remote Desktops application definitions require a one-time update of existing Remote Desktops application definitions due to the change in the underlying technology stack. Hence, we are taking a one-time exception to not support any direct migration.

The options to migrate the application definitions to use new Remote Desktops are:

- Create and publish Remote Desktops application definitions using the application composer
- Write new application definition by using one of the example application definition.

Once the application definitions are created, framework updates can be delivered seamlessly in the future.

See Also
System Requirements
Supported Product Configurations
Ports Used by Access Web

3.3 Service User

Information about the Service User.

During the installation of the Access Web Server, you are asked to provide a user account that will be the Service User. This user will own the Access Web services and the files in PA_HOME and PA_EXEC.

Managerial Privileges

The Service User is automatically assigned managerial privileges (is assigned a Manager role) and has unrestricted access to all features of Access Web. Actions that a Manager can do that a regular user cannot:

- add, edit, and delete server clusters (a PAS Server that is connected to an HPC complex).
- restrict other user's access to applications.
- assign managerial privileges to other users.

Prerequisites for the Service User

The user account chosen for the Service User must meet the following requirements:

- Existing user account on the machine were Access Web is being installed.
- Non-root account.
- Account is enabled.
- User account must be able to login to the machine hosting Access Web with a password.

Default Service User

The default user account that Access Web uses as the Service User is "pbsworks". Create this user account or choose a different user account that meets the prerequisites.

Install Access Web

<div style="text-align: right">

4

</div>

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

This chapter covers the following:

Access Web now supports PBS Professional and Altair Grid Engine Workload Managers.

PAS supports clusters where head node is Linux and all execution hosts are Linux, Windows, or both.

> 📝 **Note:** There is no support for PBS Professional Windows Server.

Access Web supports both Linux and Windows execution hosts.

Job submitted to Linux or Windows execution node in PBS is based on the architecture defined in the application definition.

Define *EXECUTION_PLATFORM* parameter in application input file enables user to provide the execution node (Linux or Windows) to submit the job. The *EXECUTION_PLATFORM* parameter with its value can be hardcoded in application converter file as well.

Refer to *Diving Into Application Definitions* guide to set the architecture parameter in application input and application converter file.

4.1 Check the Status of SELinux

Check the status of SELinux and disable SELinux before installing Access Web.

SELinux must be disabled before installing Access Web.

1. Check the status of SELinux using the command:

    ```
    sestatus
    ```

    The message may be:

    ```
    SELinux status: disabled
    or
    SELinux status: enabled
    ```

2. If the status of the SELinux is *enabled*, then follow the steps to disable:

    a) Edit the `/etc/selinux/config` file.

    b) Set the value of SELINUX to *disabled*.

    ```
    SELINUX=disabled
    ```

    c) Reboot the system.

4.2 Install Access Web

Install Access Web to submit jobs to the Workload Manager.

- Review system requirements of Access Web.
- Review prerequisites of Access Web.
- Uninstall previous versions of Access Web.

Ensure that you have the following information before starting the installation:

- license server details
- an user account that will be the Service User
- HyperWorks location
- Compose location

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

Installation must be done as root or as a user with sudo permissions.

The Access Web 2022.1.0 installer provides an option for installing Access Web, PAS, or both. If you are installing both Access Web and PAS on the same machine, then you will only need to run through this installation process once. If you are installing them on a separate machine then you will have to run this installation process once to install Access Web and a second time to install PAS.

1. Login to the machine where Access Web/PAS is to be installed.
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.

    If you have entered 2 or 3 and if PBS Professional is not installed, then the following message is displayed:

    ```
    ==============================================================================
    PBS Professional not installed
    ------------------------------

    PBS Professional is not installed
    Could not detect the installation of PBS Professional.
    Please select "OK" to continue OR "Cancel" to exit installation
    ```

△ **ALTAIR**

```
   ->1- OK
     2- Cancel

ENTER THE NUMBER OF THE DESIRED CHOICE, OR PRESS <ENTER> TO ACCEPT THE
    DEFAULT:
```

7. Choose one of the following options:

    - Enter `1` and press `ENTER` to install Access Web Services. This is the default option.

    - Enter `2` and press `ENTER` to cancel the installation.

    > 📝 **Note:** It is recommended that the Workload Manager (PBS Professional or Altair Grid Engine) is installed before installing Access Web.

8. Enter the license server details and press `ENTER`.

    The license server details should have the port and hostname in the format `port@hostname`. You can provide multiple license server details separated by a colon in Linux.

    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.

9. If you are installing PAS, enter a staging directory and press `ENTER`.

    The staging directory is where user's job files will be staged prior to execution.

10. Enter the Service User and press `ENTER`.

    The default Service User is 'pbsworks' and the Access Web services are registered under this user.

11. Enter the location where the Access Web binaries are to be installed.

    You may choose to install in the default location.

12. Enter the location where the Access Web configuration and logs files are to be installed.

    You may choose to install in the default location.

13. If Access Web is being installed, enter the HyperWorks Desktop location for visualizing results.

    `<Hyperworks_INSTALL_DIR>/altair`

    For example, `/opt/HyperWorks/altair`.

    > 📝 **Note:** You can also configure this path after the installation by editing `PA_HOME/config/resultmanager/site_config.xml`.

14. If Access Web is being installed, enter the Compose location for visualizing plot results.

    `<COMPOSE_INSTALL_DIR>/altair/Compose2021.2`

    For example, `/opt/Compose/altair/Compose2021.2`.

    > 📝 **Note:** You can also configure this path after the installation by editing `PA_HOME/config/resultmanager/site_config.xml`.

15. Review the installation summary and press `ENTER`.

    The installation starts. It may take a few minutes for the installation to complete.

△ **ALTAIR**

> **Note:** Access Web starts a watcher process which will monitor the status of Access
> Services. The watcher process automatically brings up any services that goes down
> abruptly.

4.3 Update Access Web Server Hostname

Update Access Web server hostname when you install Access Web on one system and Workload Manager (PBS Professional or Altair Grid Engine) and PAS together in another system.

> 📝 **Note:** Restart of PAS is required if you are performing the following instructions only to update Access Web server hostname.

> 📝 **Note:** Restart of PAS is not required if you performing the following instructions as part of install Access Web.

1. Login to the machine where PAS is installed as root or as a user with sudo permissions.
2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:

   ```
   source /etc/pbsworks-pa.conf
   ```
3. Edit the file `$PA_HOME/config/shared/app.properties`.

   ```
   vi $PA_HOME/config/shared/app.properties
   ```
4. Update the *nats.broker.url* value with Access Web server hostname.

   ```
   #NATS server connection details
   nats.broker.url=nats://<hostname>:4743
   ```
5. Restart PAS for these changes to take effect by entering the following command:

   ```
   service pbsworks-pa restart
   ```

◁ **ALTAIR**

4.4 Configure PBS Professional

Configure PBS Professional Windows execution hosts and enable job history in PBS.

4.4.1 Configure PBS Professional Windows Setup

Configure the parameters in Access Web to support PBS Windows execution hosts.

> 📝 **Note:** The following steps must be performed to support PBS Windows execution hosts only.

PAS supports clusters where head node is Linux and all execution hosts are Linux, Windows, or both.

> 📝 **Note:** There is no support for PBS Professional Windows Server.

Access Web supports both Linux and Windows execution hosts.

The Modern Communication Module path of PBS Windows execution hosts must be updated in Access Web.

Change the configuration value of `CLUSTER_TYPE` in `PA_HOME/config/pas/conf/server.conf` based on the PBS execution nodes. By default, the value is set to *Linux* indicating that it is PBS Linux execution nodes.

> 📝 **Note:** All the execution nodes should have the same operating system.

1. Login to the machine where Access Web is installed as the root user.
2. Source the Access Web configuration file to set up the environment variable `PA_EXEC`:
   ```
   source /etc/pbsworks-pa.conf
   ```
3. Edit `PA_HOME/config/pas/conf/server.conf`
   ```
   vi $PA_HOME/config/pas/conf/server.conf
   ```
4. Change the `CLUSTER_TYPE` value to *windows*.
   ```
   CLUSTER_TYPE=windows
   ```

   > 📝 **Note:** By default, the value is set to *linux*.

5. Restart the Access Web by entering the command:
   ```
   service pbsworks-pa restart
   ```

△ **ALTAIR**

4.4.2 Enable Job History in PBS

The HPC cluster where simulations will run must have job history enabled to run simulation jobs.

Skip this step if your site has not installed the Simulator component.

The Simulator component requires that job history be enabled on the PBS Server where simulation jobs are run:

1. Login to the PBS Server where simulation jobs will run as a PBS manager.
2. Execute the command:
   ```
   qmgr -c 'set server job_history_enable=True'
   ```

4.5 Configure Altair Grid Engine

Configure Altair Grid Engine (AGE) in Access Web.

The following are the steps that must be configured in Altair Grid Engine and Access Web:

- Configure resource in Altair Grid Engine for Remote Desktops
- Update Scheduler parameter in Altair Grid Engine
- Update job update interval in Altair Grid Engine
- Configure Workload Manager adaptor in Access Web
- Configure parallel environment in Access Web

> 📝 **Note:** Export `SGE_ROOT` and `SGE_CELL` as environment variables in `$PA_EXEC/pas/scripts/setenv.sh`, if they are not set at the system level.

4.5.1 Configure Resources for Remote Desktops

Configure ndesktops, iworkq, and ndesktops resource on each graphical node in the execution host in Altair Grid Engine for Remote Desktops

1. Login to the machine where the AGE is installed.
2. Create ndesktops resource:
   a) Enter the command:
   ```
   qconf -ace ndesktops
   ```
   b) Specify the details of ndesktops as mentioned below:
   ```
   name            ndesktops
   shortcut        ndesktops
   type            INT
   relop           <=
   requestable     YES
   consumable      YES
   default         0
   urgency         0
   aapre           NO
   affinity        0.000000
   ```
3. Create iworkq:
   a) Enter the command:
   ```
   qconf -aq iworkq
   ```
   b) Update the hostlist value to *@allhosts* as mentioned below:
   ```
   qname iworkq
   hostlist @allhosts
   seq_no 0
   load_thresholds np_load_avg=1.75
   suspend_thresholds NONE
   nsuspend 1
   suspend_interval 00:05:00
   ```
4. Associate ndesktops resource on each graphical node:

△ **ALTAIR**

a) Enter the command:

```
qconf -me <execution host name>
```

b) Update the ndesktops resource as mentioned below:

```
hostname accessha41
load_scaling NONE
complex_values m_mem_free=7821.000000M,m_mem_free_n0=7821.000000M, \
ndesktops=1
user_lists NONE
xuser_lists NONE
projects NONE
xprojects NONE
usage_scaling NONE
report_variables NONE
license_constraints NONE
license_oversubscription NONE
```

4.5.2 Update Scheduler Parameter in Altair Grid Engine

Update the scheduler parameter to get the job schedule information.

Enable the *schedd_job_info* parameter of the Scheduler to collect the reasons why jobs could not be scheduled during a scheduler run.

If you enable this *schedd_job_info* parameter, then messages containing the reasons why it was not possible to schedule a job will be collected for the not-scheduled jobs. You can use `qstat` command to retrieve the information for a specific job.

1. Login to the machine where the AGE is installed.
2. Enter the command:

    ```
    qconf -msconf
    ```
3. Enable *schedd_job_info* by changing its value to *true*.

    ```
    schedd_job_info     true
    ```

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

4.5.3 Update Job Update Interval in Altair Grid Engine

Update the job update interval value of the reporting parameters attribute.

The parameter *flush_time* should be set to at least one second (00:00:01). If it is set to 0, buffering of write operations to the accounting and the reporting file is not done. Should the attribute accounting *flush_time* be set, it must either be removed (meaning that *flush_time* will be in effect for the accounting file) or set to at least one second (00:00:01).

1. Login to the machine where the AGE is installed.
2. Enter the command:

    ```
    qconf -mconf
    ```
3. Update the value of *flush_time* of *reporting_params* to *00:00:01*.

```
reporting_params accounting=true reporting=false\
       flush_time=00:00:01 joblog=false sharelog=00:00:00
```

> 📝 **Note:** By default, the value is set to *00:00:15*.

4.5.4 Configure Workload Manager Adaptor in Access Web

Configure the Workload Manager adaptor parameter in Access Web to support Altair Grid Engine (AGE).

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:
   ```
   source /etc/pbsworks-pa.conf
   ```
3. Edit the file `PA_HOME/config/pas/conf/server.conf`.
   ```
   vi $PA_HOME/config/pas/conf/server.conf
   ```
4. Change the value of `WLM_ADAPTOR` to *age*.
   ```
   WLM_ADAPTOR=age
   ```

   > 📝 **Note:** By default, the value of `WLM_ADAPTOR` is set to *pbs*.

5. Edit the file `PA_HOME/config/executor/app-config.xml`.
   ```
   vi $PA_HOME/config/executor/app-config.xml
   ```
6. Change the following code as a comment:
   ```
   <ref bean="priorityJobsProcessingTrigger" />
   ```
7. Edit the file `PA_HOME/config/pa/configuration.json`.
   ```
   vi $PA_HOME/config/pa/configuration.json
   ```
8. Change the *adaptorDetails* to AGE:
   ```
   "adaptorDetails": {
           "id": "AGE",
           "name": "Altair Grid Engine"
         }
   ```

   > 📝 **Note:** By default, the `id` value is set to *PBS* and `name` value is set to *PBS Pro*.

9. Restart Access Web using the following command:
   ```
   service pbsworks-pa restart
   ```

4.5.5 Configure Parallel Environment in Access Web

Configure the parallel environment of Altair Grid Engine in Access Web for the application definitions.

Before you begin, create and configure parallel environment in AGE. Use the command `qconf -spl` to check the existing parallel environment in AGE.

Refer to *Altair Grid Engine Administrator's Guide* for more information.

Configure the parallel environment in Access Web for the application definitions in the `parallel-env-config.json` file located in `PA_HOME/config/pas/conf/`.

If the application ID is not mentioned in the `parallel-env-config.json` file, then the default configuration will be used. The default configuration must be added in the `parallel-env-config.json` file.

If parallel environment is configured in the application definition, then parallel environment will be considered from app-def file instead of from the configuration file.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:

   ```
   source /etc/pbsworks-pa.conf
   ```
3. Edit `PA_HOME/config/pas/conf/parallel-env-config.json`

   ```
   vi $PA_HOME/config/pas/conf/parallel-env-config.json
   ```
4. Add the application ID and the parallel environment.
   For example, set the application ID and the parallel environment as follows:

   ```
   {
     "Optistruct": "smp",
     "Radioss": "mpi",
     "default": "smp"
   }
   ```
5. Restart Access Web using the following command:

   ```
   service pbsworks-pa restart
   ```

△ ALTAIR

4.6 Modern Communication Module

The Modern Communication Module is a set of python libraries that are copied to the HPC execution hosts to improve file operations on running jobs.

Once Access Web is installed, the Modern Communication Module must be distributed to the HPC execution hosts. These libraries improve file operations for job operations on running jobs, such as:

- uploading and downloading files to the running directory.
- listing files from the running directory.
- custom actions on running jobs.
- viewing results visualization animation files from the running directory.
- starting remote desktops.

The Modern Communication Module replaces PBS technology for performing these same types of operations.

> 📋 **Note:** Refer to Configure PBS Professional Windows Setup to set up Modern Communication Module in PBS Windows execution hosts.

The Modern Communication Module can be distributed by:

1. Copying to a shared file system that is accessible to the execution hosts.
2. Running an Access Web script in PBS Linux execution hosts.
3. Copying to the PBS Windows execution hosts.
4. Copying to the Altair Grid Engine execution hosts.

> 📋 **Note:** If you want to change the Desktop Manager for the PBS execution hosts, refer to Configure the Desktop Manager for Interactive Applications

See Also

Setup Modern Communication Module on a Shared File System

Copy the Modern Communication Module to All Execution Hosts

Copy the Modern Communication Module to All Windows PBS Execution Hosts

Copy the Modern Communication Module to All AGE Execution Hosts

4.6.1 Setup Modern Communication Module on a Shared File System

Distribute the Modern Communication Module on a shared file system after completing the installation of Access Web.

> 📋 **Note:** Refer to Configure PBS Professional Windows Setup to set up Modern Communication Module in PBS Windows execution hosts.

Copy the Modern Communication Module located on the PAS Server to the shared file system. The Linux Modern Communication Module is located at: `PA_EXEC/joboperation/binaries/linuxmomclientmodules`.

Before performing the following steps:

- ensure that the shared file system is available and accessible to all the users.
- create a directory on the shared file system to copy the Modern Communication Module. The permissions on this directory need to allow read and execute permissions for all users. For example, `/shared/filesystem/Access-MCM/momclientmodules`.

> 📝 **Note:** All the execution nodes should have the same operating system.

1. Login to the machine where PAS is installed as the root user.
2. Source the Access Web configuration file to set up the environment variable `PA_EXEC`:
   ```
   source /etc/pbsworks-pa.conf
   ```
3. Create a directory on the shared filesystem where the Modern Communication Module will be copied.
   ```
   mkdir -m 755 -p /shared/filesystem/Access-MCM/momclientmodules
   ```
4. Copy the directory `PA_EXEC/joboperation/binaries/linuxmomclientmodules` to the shared file system.
   ```
   cp -rp $PA_EXEC/joboperation/binaries/linuxmomclientmodules /shared/filesystem/
   Access-MCM/momclientmodules
   ```
5. Depending on your Operating System (CentOS, SLES, or Ubuntu), unzip relevant python binary from `$PA_EXEC/shared/thirdparty/python_binaries/` to `/shared/filesystem/Access-MCM/momclientmodules/linuxmomclientmodules`.

   > 📝 **Note:** You can use the CentOS python binary for RHEL Operating System.

6. Give read and execute permissions to all users to the location where the Modern Communication Module was copied.
   ```
   chmod 755 /shared/filesystem/Access-MCM/momclientmodules/linuxmomclientmodules
   ```
7. Edit `PA_HOME/config/pas/conf/server.conf`
   ```
   vi $PA_HOME/config/pas/conf/server.conf
   ```
8. Define the location of the Modern Communication Module by updating the `LINUX_NODE_MODERN_COMMUNICATION_SHARED_LIBS` attribute.
   ```
   LINUX_NODE_MODERN_COMMUNICATION_SHARED_LIBS=/shared/filesystem/Access-MCM/
   momclientmodules/linuxmomclientmodules
   ```
9. Restart the PAS by entering the command:
   ```
   service pbsworks-pa restart
   ```

4.6.2 Copy the Modern Communication Module to All Execution Hosts

Distribute the Modern Communication Module to all execution hosts by running a script.

> 📄 **Note:** Refer to Configure PBS Professional Windows Setup to set up Modern Communication Module in PBS Windows execution hosts.

The script to distribute the Modern Communication Module to all of the PBS MoM hosts is located at `PA_EXEC/joboperation/scripts/distribute_modern_comm_modules.py`.

Before running this script, ensure the following prerequisites have been met:

- The Modern Communication Module distribution script must be run as the root user.
- The script must be run from the system where PAS is installed.
- The distribution script uses PBS Client commands. Hence the PBS Client must be installed on the PAS server.
- The distribution script asks for a location (a directory on the PAS Server) into which it will copy the Modern Communication Module before distributing it to the PBS MoMs. The permissions on this directory need to allow read and execute permissions for all users. This directory must be created before running the script. Once the script is run, the Modern Communication Module is copied to this directory on the PAS Server. The directory is then copied to each execution host and placed in the same location on each execution host.
- The distribution script asks for a user account. This user account will own the directory containing the Modern Communication Module on the PBS MoMs. This user must:
  - exist on the PAS server and all of the PBS MoMs
  - be able to SSH passwordlessly from the PAS Server into each MoM so that the Modern Communication Module can be securely copied to the PBS MoM.

Installation steps to distribute the Modern Communication Module:

1. Login to the machine where PAS is installed as the root user.
2. Source the Access Web configuration file to set up the environment variable `PA_EXEC`:

   `source /etc/pbsworks-pa.conf`
3. Create a directory where the distribution script can copy the Modern Communication Module before distribution to the PBS MoMs:

   `mkdir -m 755 -p /access/mcm`
4. Navigate to the directory `$PA_EXEC/joboperation/scripts`.
5. Execute the script to distribute the Modern Communication Module:

   `python distribute_modern_comm_modules.py`

> 📝 **Note:** Python 3 and above is required to run the modern communication module script.
>
> If your system does not have the latest python, then navigate to `$PA_EXEC/shared/thirdparty/python` and execute the following command:
>
> ```
> bin/python3.8 $PA_EXEC/joboperation/scripts/
> distribute_modern_comm_modules.py
> ```

6. Enter the directory that meets the stated prerequisites.

   This is where the script will copy the Modern Communication Module before distributing it to the PBS MoMs.

   For example, `/access/mcm`

7. Enter a user that meets the stated prerequisites.
   For example, `pbsworks`.
   The following message is displayed:

   ```
   Do you want to override if modules already exist (Y/N). [Default: N]?
   ```

8. Choose one of the following:

   - Enter `N` or press `ENTER` (to accept the default value) to skip copying the Modern Communication Module to the PBS MoM when it already exists on the MoM.
   - Enter `Y` to overwrite the modules. The distribution script will copy the modules even if it detects that Modern Communication Module already exists on the MoM.

   The script will display the inputs provided.

9. Enter `Y` to continue with the inputs provided.

   Messages similar to the below are displayed:

   ```
   Transferring Client Modules to n2.lab.com...
   Successfully transferred to 'n2.lab.com'
   Transferring Client Modules to n1.lab.com...
   Successfully transferred to 'n1.lab.com'
   Updated PAS Configuration file successfully
   ```

10. Restart Access Web by entering the command:

    ```
    service pbsworks-pa restart
    ```

4.6.3 Copy the Modern Communication Module to All Windows PBS Execution Hosts

Distribute the Modern Communication Module on all PBS Windows execution hosts after completing the installation of Access Web.

Copy the Modern Communication Module located on the PAS Server to the PBS Windows execution hosts. The Modern Communication Module is located at: `PA_EXEC/joboperation/binaries/windowsmomclientmodules`.

Before performing the following steps:

- ensure that the PBS Windows execution hosts is available and accessible to all the users.

- create a folder on the PBS Windows execution hosts to copy the Modern Communication Module. The permissions on this folder need to allow read and execute permissions for all users. For example, `/windowsmcm`.

> 📝 **Note:** All the execution nodes should have the same operating system.

1. Login to the machine where PAS is installed as the root user.
2. Source the Access Web configuration file to set up the environment variable `PA_EXEC`:
   ```
   source /etc/pbsworks-pa.conf
   ```
3. Create a folder on the PBS Windows execution hosts where the Modern Communication Module will be copied.

   For example, you can create a folder in *C:/windowsmcm*
4. Copy the directory `PA_EXEC/joboperation/binaries/windowsmomclientmodules` to the folder created in Windows PBS execution hosts.
5. Give read and execute permissions to all users to the folder where the Modern Communication Module was copied.
6. Edit `PA_HOME/config/pas/conf/server.conf`
   ```
   vi $PA_HOME/config/pas/conf/server.conf
   ```
7. Define the location of the Modern Communication Module by updating the `WINDOWS_NODE_MODERN_COMMUNICATION_SHARED_LIBS` attribute.
   ```
   WINDOWS_NODE_MODERN_COMMUNICATION_SHARED_LIBS=/windowsmcm/windowsmomclientmodules
   ```

   > 📝 **Note:** The path should not contain the drive location *C:/*. For example, if you have copied the directory in *C:/windowsmcm/windowsmomclientmodules*, then specify only */windowsmcm/windowsmomclientmodules*.

8. Restart the PAS by entering the command:
   ```
   service pbsworks-pa restart
   ```

4.6.4 Copy the Modern Communication Module to All AGE Execution Hosts

Distribute the Modern Communication Module on all AGE execution hosts after completing the installation of Access Web.

Copy the Modern Communication Module located on the PAS Server to the AGE execution hosts. The Linux Modern Communication Module is located at: `PA_EXEC/joboperation/binaries/linuxmomclientmodules`.

Before performing the following steps:

- ensure that the AGE execution hosts is available and accessible to all the users.
- create a directory on the AGE execution hosts to copy the Modern Communication Module. The permissions on this directory need to allow read and execute permissions for all users. For example, `/agemcm/Access-MCM/momclientmodules`.

> 📝 **Note:** All the execution nodes should have the same operating system.

1. Login to the machine where PAS is installed as the root user.
2. Source the Access Web configuration file to set up the environment variable `PA_EXEC`:
   ```
   source /etc/pbsworks-pa.conf
   ```
3. Create a directory on the AGE execution hosts where the Modern Communication Module will be copied.
   ```
   mkdir -m 755 -p /agemcm/Access-MCM/momclientmodules
   ```
4. Copy the directory `PA_EXEC/joboperation/binaries/linuxmomclientmodules` to the AGE execution hosts.
   ```
   cp -rp $PA_EXEC/joboperation/binaries/linuxmomclientmodules /agemcm/Access-MCM/
   momclientmodules
   ```
5. Depending on your Operating System (CentOS, SLES, or Ubuntu), unzip relevant python binary from `$PA_EXEC/shared/thirdparty/python_binaries/` to `/agemcm/Access-MCM/momclientmodules/linuxmomclientmodules`.

   > 📝 **Note:** You can use the CentOS python binary for RHEL Operating System.

6. Give read and execute permissions to all users to the location where the Modern Communication Module was copied.
   ```
   chmod 755 /agemcm/Access-MCM/momclientmodules/linuxmomclientmodules
   ```
7. Edit `PA_HOME/config/pas/conf/server.conf`
   ```
   vi $PA_HOME/config/pas/conf/server.conf
   ```
8. Define the location of the Modern Communication Module by updating the `LINUX_NODE_MODERN_COMMUNICATION_SHARED_LIBS` attribute.
   ```
   LINUX_NODE_MODERN_COMMUNICATION_SHARED_LIBS=/agemcm/Access-MCM/momclientmodules/
   linuxmomclientmodules
   ```
9. Restart the PAS by entering the command:
   ```
   service pbsworks-pa restart
   ```

△ ALTAIR

4.7 Install Remote Desktops Components on Linux Execution Hosts

Install components necessary to enable the remote visualization capabilities of Access Web on Linux execution hosts.

Ensure that the following prerequisites are met:

- Supported GPU: nvfbc and nvenc to support GRID, Tesla, or Quadro X2000+ (X = K/M/P/RTX etc) hardware products
- Graphics driver installed:
  - NVIDIA Windows display driver 471.41 or newer
  - NVIDIA Linux display driver 470.57.02 or newer
  - DirectX SDK (Windows only) CUDA 11.0 Toolkit
- X server configured with GPU and it is running.
- xsel or xclip library package should be installed on graphical node in case of Linux execution host.
- DPMS must be disabled. It can be done by adding `Option "DPMS" "false"` line under `monitor` section of `/etc/X11/xorg.conf` file.

Access Web provides scripts to run on PBS server machine and install desktop agent on graphical nodes.

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

1. Install the Remote Desktops component on the Workload Manager (PBS Server:
   - adds a custom resource called *ndesktops*.
   - creates an interactive queue called "iworkq".
   - updates the PBS ndesktops resource on provided graphics node by requesting the list of graphics node and number of ndesktops available.
2. Install the Remote Desktops component on all the execution hosts (PBS MoMs or AGE) on which you want to run interactive jobs.

4.7.1 Configure the Workload Manager

Configure Workload Manager PBS to support interactive applications.

Before you begin:

- Review the prerequisites for installation.

Installation must be done as root or as a user with sudo permissions.

On the Workload Manager Server, this installation will:

- add a custom resource to PBS Professional called *ndesktops*
- add an interactive queue called "iworkq"
- updates the PBS ndesktops resource on provided graphics node by requesting the list of graphics node and number of ndesktops available.

The below steps must be run on the PBS Server. If PBS Server is installed on the same machine these steps need to be run once.

1. Login to the machine where the PBS is installed.

2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:
    ```
    source /etc/pbsworks-pa.conf
    ```

3. Navigate to the directory `PA_EXEC/desktop/scripts/`
    ```
    cd $PA_EXEC/desktop/scripts/
    ```

4. Enter the command:
    ```
    bash ./pbs_server_graphics_configure.sh
    ```

    The above script will request you to provide the list of graphical node and number of ndesktops available to update the PBS ndesktops resource on the graphical node.

    The following message is displayed:
    ```
    Note: Please give the list of graphical node(s) having same number of gpus with
     comma separated(,)
    Please provide the list of graphical node(s):
    ```

5. Enter the list of the graphical node(s) that are available on the HPC complex and press `ENTER`.

    The following message is displayed:
    ```
    Please provide the number of gpu(s) that has to be configured for list of graphical
     node(s): 2
    Please provide gpu count:
    ```

6. Enter the number of GPUs that are available on the HPC complex and press `ENTER`.

    The following message is displayed.
    ```
    Do you want to update ndesktops on more graphical node(s)? [Y/N]
    ```

7. Choose one of the following options:

    -   Enter **Y** to update ndesktops on more graphical node(s).
    -   Enter **N** to exit the script.

4.7.2 Configure Multiple GPUs

Configure the execution nodes if it has multiple GPUs.

GPU is selected based on the load of GPU utilization, memory and process.

1. Login to the machine as root or as a user with sudo permissions where the graphical MoM is installed.

2. Navigate to `$MOM_CLIENT_MODULES/desktop/`

    `<MOM_CLIENT_MODULE>` is the location where Modern Communication Module is distributed. For example, `/access/mcm/momclientmodules`.

    > 📑 **Note:** Check the `LINUX_NODE_MODERN_COMMUNICATION_SHARED_LIBS` variable value in the `$PA_HOME/config/pas/conf/server.conf` file to verify the `<MOM_CLIENT_MODULE>` location.

3. Execute the following command:

```
bash ./configure_multigpu.sh
```

> 📝 **Note:** The script must be run on each execution node that has multiple GPUs.

For example, the following message is displayed:

```
[root@ip-172-31-36-239 desktop]# bash configure_multigpu.sh
Using X configuration file: /etc/X11/xorg.conf.
New X configuration file written to /etc/X11/xorg_gpu0.conf

Using X configuration file: /etc/X11/xorg.conf.
New X configuration file written to /etc/X11/xorg_gpu1.conf

Using X configuration file: /etc/X11/xorg.conf.
New X configuration file written to /etc/X11/xorg_gpu2.conf

Using X configuration file: /etc/X11/xorg.conf.
New X configuration file written to /etc/X11/xorg_gpu3.conf
```

4.7.3 Install Remote Desktops Component on Execution Hosts

Install binaries and configure the execution hosts (PBS MoM or AGE) to support interactive applications.

Before you begin:

- Review the Prerequisites for Installing Remote Desktops for installation.
- Run the Remote Desktops precheck diagnosis script on the PBS MoM to check the status of GPU nodes.

> 📝 **Note:** This installation process must be repeated for all graphical MoMs where you want to run interactive jobs.

1. Login to the machine where the graphical MoM is installed.

2. Navigate to `$MOM_CLIENT_MODULES/desktop/`.

`<MOM_CLIENT_MODULE>` is the location where Modern Communication Module is distributed. For example, `/access/mcm/momclientmodules`.

Check the `LINUX_NODE_MODERN_COMMUNICATION_SHARED_LIBS` variable value in the `$PA_HOME/config/pas/conf/server.conf` file to verify the `<MOM_CLIENT_MODULE>` location.

> 📝 **Note:** If you are installing on Windows, then check the `WINDOWS_NODE_MODERN_COMMUNICATION_SHARED_LIBS` variable value in the `$PA_HOME/config/pas/conf/server.conf` file to verify the `<MOM_CLIENT_MODULE>` location.

3. Execute the following command:

```
bash ./install.sh
```

> 📝 **Note:** If you are installing on Windows, then execute the following command:
>
> ```
> <MOM_CLIENT_MODULE_PATH>\python\python.exe <MOM_CLIENT_MODULE_PATH>
> \python\install.py
> ```

For example, the following message is displayed:

```
Allowing all users to start X server
Unzipping the ./packages/centos.tar.gz

Backup of changed configuration files and logs can be found at: /tmp/
desktop_2022-02-16_19-54-23
```

> 📝 **Note:** The `/tmp/desktop_2022-02-16_19-54-23` file must be backed up as this is
> required during uninstallation of Remote Desktops.

The script will provide permission to start X server by all the users and extract platform specific
agent.

4.8 Install Remote Desktops Component on Windows Execution Hosts

Install components necessary to enable the remote visualization capabilities of Access Web on Windows execution hosts.

Before you begin:

- Review the prerequisites for installation.

Access Web provides scripts to run on graphical nodes to install Remote Desktops components.

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

1. Login to the machine where the graphical MoM is installed..

2. Navigate to `<MOM_CLIENT_MODULE_PATH>\python\`.

   `<MOM_CLIENT_MODULE_PATH>` is the path where modern communication module is installed.

3. Open the Windows command prompt.

4. Run the following command:

   ```
   <MOM_CLIENT_MODULE_PATH>\python\python.exe <MOM_CLIENT_MODULE_PATH>\desktop
   \install.py
   ```

5. Once the installation is complete check that the Remote Desktops service is running:

   a) Click **Start** and choose **Run**.

   b) Type `services.msc` to open the Services Management Console.

   c) Locate the **DesktopAgentService**.

      The status should be "Started".

   d) If the service is not running then, right-click the **DesktopAgentService** and click **Start**.

4.9 Open Ports

Open ports for communication with external locations.

It is recommended to configure a firewall on the Access Web server and block all Access Web ports from the outside world except:

- 4443
- 4743
- 4943 - 5042

1. Login to the machine where Access Web is installed.
2. Open the following ports:
   - Gateway port (defaults to 4443)
   - Remote Desktops Job agent proxy service port (defaults to 4943)
   - NATS message broker port (defaults to 4743)

ALTAIR

Upgrade Access Web

5

Instructions for upgrading from a previous version of Access Web.

This chapter covers the following:

The Access Web installer itself does not support upgrades. If you are starting with an old version of Access Web, then all the sections and the steps must be completed exactly.

Only upgrades from Access Web 2021.1.1 or 2021.1.2 to Access Web 2022.1.0 are supported.

Access Web 2022.1.0 supports only PAS 2022.1.0. So, if you are upgrading Access Web, then you have to upgrade PAS.

> 📝 **Note:** Remote Desktops upgrade is not supported as it uses different technology.

Remote Desktops Application Definitions:

The new Remote Desktops application definitions require a one-time update of existing Remote Desktops application definitions due to the change in the underlying technology stack. Hence, we are taking a one-time exception to not support any direct migration.

The options to migrate the application definitions to use new Remote Desktops are:

- Create and publish Remote Desktops application definitions using the application composer
- Write new application definition by using one of the example application definition.

Once the application definitions are created, framework updates can be delivered seamlessly in the future.

5.1 Prepare for an Upgrade

Instructions to prepare for upgrading Access Web.

Before you begin

- Review the prerequisites for installation.

> ⚠️ **Important:** Do not uninstall the previous version of Access Web or PAS.

Create a backup of the Access Web installation registry file and application configuration file. The backup of the configuration file can be used to downgrade back to the currently installed version of Access Web.

1. Login to the machine where the previous version of Access Web is installed as root or as a user with sudo permissions.
2. Stop Access Web:

   ```
   service pbsworks-pa stop
   ```

3. Create a backup of `/var/.com.zerog.registry.xml`.
4. Remove `/var/.com.zerog.registry.xml`.
5. Create a backup of the application configuration file:

   ```
   mv -p /etc/pbsworks-pa.conf /etc/pbsworks-pa.conf.VERSION
   ```

   Where *VERSION* is the currently installed version of Access Web.
6. Remove `/etc/init.d/pbsworks-pa` script.
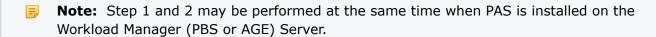
5.2 Uninstall Remote Desktops

Unconfigure Workload Manager (PBS Professional or Altair Grid Engine) and PAS and uninstall the Remote Desktops component installed on the execution hosts.

Uninstalling the previous version (2021.1.1 or 2021.1.2) of Remote Desktops component requires running the Remote Desktops binary multiple times across several machines and must be performed in the following sequence:

1. Unconfigure the Workload Manager (PBS or AGE) and the PAS Server:
   - removes the application definition GlxSpheres which was installed automatically when the Remote Desktops component was installed on the PAS Server
   - deletes the interactive queue
   - removes the custom resource ngpus
2. Uninstall the Remote Desktops component on all execution hosts.

> 📝 **Note:** Step 1 and 2 may be performed at the same time when PAS is installed on the Workload Manager (PBS or AGE) Server.

5.2.1 Unconfigure Workload Manager and PAS

Unconfigure Workload Manager (PBS or AGE) and PAS before uninstalling the Remote Desktops component.

> ⚠️ **CAUTION:** It is advisable that you run the installer when critical jobs are not running as both Workload Manager and PAS are restarted during the uninstallation process.

On the Workload Manager Server, this uninstallation process will:
- delete the custom resource *ngpus*
- remove the interactive queue called "iworkq"
- restart Workload Manager

On the PAS Server, this installation will:
- remove the GlxSpheres application definition.
- restart PAS

The below steps must be run on both the Workload Manager Server and the PAS Server. When the Workload Manager Server and PAS are installed on the same machine, then these steps only need to be run once.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Stop Access Web:
   ```
   service pbsworks-pa stop
   ```
3. Login to the machine where the Workload Manager (PBS or AGE) Server/PAS is installed as root or as a user with sudo permissions.

4. Navigate to the directory where the previous version (2021.1.1 or 2021.1.2) of Remote Desktops installation binary is located.

5. Enter the command:

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

6. If you are uninstalling the Remote Desktops component on a machine hosting both the Workload Manager 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
```

7. Read the introduction and press `ENTER`.

8. Page through the license agreement by pressing `ENTER` until you are asked to accept its terms and conditions.

9. Accept the license agreement by entering `Y` and pressing `ENTER`.
Four options are displayed.

10. Enter `3` (Unconfigure PBS and PAS Server) to unconfigure Workload Manager (PBS or AGE) and PAS and press `ENTER`.

11. The Workload Manager (PBS or AGE)/PAS Server are 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.

12. Stop Access Web:

```
service pbsworks-pa stop
```

5.2.2 Uninstall the Remote Desktops Component from the Execution Hosts

Uninstall the Remote Desktops component from the execution hosts (PBS MoMs or AGE).

> ⚠️ **CAUTION:** It is advisable that you run the installer when critical jobs are not running as Workload Manager (PBS or AGE) is restarted during the uninstallation process.

The below steps must be repeated on every execution hosts where the Remote Desktops component has been installed.

1. Login to the execution hosts (PBS MoM or AGE) machine where the Remote Desktops component is installed as root or as a user with sudo permissions.

2. Choose one of the following options:

- On Ubuntu platforms, navigate to the directory `<MOM_CLIENT_MODULE>/remotesession/ubuntu/`.
- On all other Linux platforms, navigate to the directory: `/opt/altair/pbsworks/<Version>/remotesessionagent/_AltairAccessWeb_RemoteSessionAgent_<Version>_installation`

`<MOM_CLIENT_MODULE>` is the location where Modern Communication Module is distributed. For example, `/access/mcm/momclientmodules`.

3. Choose one of the following options:

   - On Ubuntu platforms, execute the uninstall script by entering the following command:

     ```
     ./uninstall.sh
     ```

   - On all other Linux platforms, execute the uninstall script by entering the following command:

     ```
     ./Change\ AltairAccessWeb_RemoteSessionAgent_<Version>\ Installation -i console
     ```

   The command must contain spaces with escape characters.

4. Follow the instructions provided by the uninstaller.

5.3 Check the Status of SELinux

Check the status of SELinux and disable SELinux before installing Access Web.

SELinux must be disabled before installing Access Web.

1. Check the status of SELinux using the command:

   ```
   sestatus
   ```

   The message may be:

   ```
   SELinux status: disabled
   or
   SELinux status: enabled
   ```

2. If the status of the SELinux is *enabled*, then follow the steps to disable:

   a) Edit the `/etc/selinux/config` file.

   b) Set the value of SELINUX to *disabled*.

   ```
   SELINUX=disabled
   ```

   c) Reboot the system.

5.4 Install the New Version of Access Web

Install the latest version of Access Web when upgrading from a previous version.

- Review system requirements of Access Web.
- Review prerequisites of Access Web.
- Do not uninstall the previous version of Access Web or PAS.

Ensure that you have the following information before starting the installation:

- license server details
- an user account that will be the Service User
- HyperWorks location
- Compose location

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

Installation must be done as root or as a user with sudo permissions.

The Access Web 2022.1.0 installer provides an option for installing Access Web, PAS, or both. If you are installing both Access Web and PAS on the same machine, then you will only need to run through this installation process once. If you are installing them on a separate machine then you will have to run this installation process once to install Access Web and a second time to install PAS.

1. Enter the command:

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

2. Read the introduction and press ENTER.

3. Page through the license agreement by pressing ENTER until you are asked to accept its terms and conditions.

4. Accept the license agreement by entering Y and pressing ENTER.

5. 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.

6. Enter the license server details and press ENTER.

   The license server details should have the port and hostname in the format `port@hostname`. You can provide multiple license server details separated by a colon in Linux.

   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.

7. If you are installing PAS, enter a staging directory and press ENTER.

   The staging directory is where user's job files will be staged prior to execution.

8. Enter the Service User and press ENTER.

△ ALTAIR

The default Service User is 'pbsworks' and the Access Web service is registered under this user.

9. Enter the location where the Access Web binaries are to be installed.

You may choose to install in the default location.

10. Enter the location where the Access Web configuration and logs files are to be installed.

You may choose to install in the default location.

11. If Access Web is being installed, enter the HyperWorks location for visualizing results.

`<Hyperworks_INSTALL_DIR>/altair`

For example, `/opt/HyperWorks/altair`

> 📝 **Note:** You can also configure this path after the installation by editing `PA_HOME/config/resultmanager/site_config.xml`

12. If Access Web is being installed, enter the Compose location for visualizing results.

`<COMPOSE_INSTALL_DIR>/altair/Compose2021.2`

For example, `/opt/Compose/altair/Compose2021.2`

> 📝 **Note:** You can also configure this path after the installation by editing `PA_HOME/config/resultmanager/site_config.xml`.

13. Review the installation summary and press `ENTER`.
The installation starts. It may take a few minutes for the installation to complete.

5.5 Upgrade Access Web

Run an upgrade script to copy artifacts from the previous version of Access Web to Access Web 2022.1.0.

Only upgrades from Access Web 2021.1.1 or 2021.1.2 to Access Web 2022.1.0 are supported.

To extract and visualize plot and animation data using Access Web 2022.1.0, ensure to have Compose and Altair HyperWorks Desktop installed. For more information refer Prerequisites for Installing Access Web.

> ℹ **Tip:** Before installing Altair HyperWorks Desktop 2021.2 we recommend to clear all the older versions of Altair HyperWorks Desktop.

The upgrade script will perform the following:

- Migrate the previous version of application definitions, the site configuration file and the server configuration file to Access Web 2022.1.0.

  > 📝 **Note:** The migration of application definitions and site configuration file will happen when you have not changed the `PAS_REPO` path in the previous version.

  > 📝 **Note:** Application definitions now need to be compatible with Python 3. Python usage in application definitions is usually minimal and straightforward. Validate the updated application definitions before using it.

- Migrate the user preferences and configuration related files from the previous version of Access Web's `PA_HOME` and `PA_EXEC` directories to those same directories of Access Web 2022.1.0.
- Automatically start Access Web 2022.1.0
- Migrate Security Assertion Markup Language (SAML) configuration files.

If you have installed Access Web on a separate machine from PAS, then the following steps will have to be run on both machines.

1. Login to the machine where Access Web/PAS is installed
2. Source the Access Web configuration file to set up the environment variable `PA_EXEC`:
   ```
   source /etc/pbsworks-pa.conf
   ```
3. Navigate to `$PA_EXEC/init/`.
4. 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 2021.1.1 or 2021.1.2 home directory and <PreviousVersion_PA_EXEC> is the Access Web 2021.1.1 or 2021.1.2 execution directory.

   After upgrading, Access Web 2022.1.0 will be up and the following message will be displayed:
   ```
   "Access Web upgraded successfully"
   ```
5. Optional: Update *USER_APPS_REPO* and *PAS_REPO* in the `/etc/pbsworks-pa.conf` file if it was customized in the previous version.

△ **ALTAIR**

6. Restart Access Web by entering the command:

```
service pbsworks-pa restart
```

5.6 Modern Communication Module

The Modern Communication Module is a set of python libraries that are copied to the HPC execution hosts to improve file operations on running jobs.

Once Access Web is installed, the Modern Communication Module must be distributed to the HPC execution hosts. These libraries improve file operations for job operations on running jobs, such as:

- uploading and downloading files to the running directory.
- listing files from the running directory.
- custom actions on running jobs.
- viewing results visualization animation files from the running directory.
- starting remote desktops.

The Modern Communication Module replaces PBS technology for performing these same types of operations.

> 📒 **Note:** Refer to Configure PBS Professional Windows Setup to set up Modern Communication Module in PBS Windows execution hosts.

The Modern Communication Module can be distributed by:

1. Copying to a shared file system that is accessible to the execution hosts.
2. Running an Access Web script in PBS Linux execution hosts.
3. Copying to the PBS Windows execution hosts.
4. Copying to the Altair Grid Engine execution hosts.

> 📒 **Note:** If you want to change the Desktop Manager for the PBS execution hosts, refer to Configure the Desktop Manager for Interactive Applications

See Also

Setup Modern Communication Module on a Shared File System

Copy the Modern Communication Module to All Execution Hosts

Copy the Modern Communication Module to All Windows PBS Execution Hosts

Copy the Modern Communication Module to All AGE Execution Hosts

5.6.1 Setup Modern Communication Module on a Shared File System

Distribute the Modern Communication Module on a shared file system after completing the installation of Access Web.

> 📒 **Note:** Refer to Configure PBS Professional Windows Setup to set up Modern Communication Module in PBS Windows execution hosts.

Copy the Modern Communication Module located on the PAS Server to the shared file system.
The Linux Modern Communication Module is located at: `PA_EXEC/joboperation/binaries/`
`linuxmomclientmodules`.

Before performing the following steps:

- ensure that the shared file system is available and accessible to all the users.
- create a directory on the shared file system to copy the Modern Communication Module. The permissions on this directory need to allow read and execute permissions for all users. For example, `/shared/filesystem/Access-MCM/momclientmodules`.

> 📝 **Note:** All the execution nodes should have the same operating system.

1. Login to the machine where PAS is installed as the root user.
2. Source the Access Web configuration file to set up the environment variable `PA_EXEC`:

   `source /etc/pbsworks-pa.conf`
3. Create a directory on the shared filesystem where the Modern Communication Module will be copied.

   `mkdir -m 755 -p /shared/filesystem/Access-MCM/momclientmodules`
4. Copy the directory `PA_EXEC/joboperation/binaries/linuxmomclientmodules` to the shared file system.

   `cp -rp $PA_EXEC/joboperation/binaries/linuxmomclientmodules /shared/filesystem/`
   `Access-MCM/momclientmodules`
5. Depending on your Operating System (CentOS, SLES, or Ubuntu), unzip relevant python binary from `$PA_EXEC/shared/thirdparty/python_binaries/` to `/shared/filesystem/Access-MCM/`
   `momclientmodules/linuxmomclientmodules`.

   > 📝 **Note:** You can use the CentOS python binary for RHEL Operating System.

6. Give read and execute permissions to all users to the location where the Modern Communication Module was copied.

   `chmod 755 /shared/filesystem/Access-MCM/momclientmodules/linuxmomclientmodules`
7. Edit `PA_HOME/config/pas/conf/server.conf`

   `vi $PA_HOME/config/pas/conf/server.conf`
8. Define the location of the Modern Communication Module by updating the `LINUX_NODE_MODERN_COMMUNICATION_SHARED_LIBS` attribute.

   `LINUX_NODE_MODERN_COMMUNICATION_SHARED_LIBS=/shared/filesystem/Access-MCM/`
   `momclientmodules/linuxmomclientmodules`
9. Restart the PAS by entering the command:

   `service pbsworks-pa restart`

5.6.2 Copy the Modern Communication Module to All Execution Hosts

Distribute the Modern Communication Module to all execution hosts by running a script.

> 📝 **Note:** Refer to Configure PBS Professional Windows Setup to set up Modern Communication Module in PBS Windows execution hosts.

The script to distribute the Modern Communication Module to all of the PBS MoM hosts is located at `PA_EXEC/joboperation/scripts/distribute_modern_comm_modules.py`.

Before running this script, ensure the following prerequisites have been met:

- The Modern Communication Module distribution script must be run as the root user.
- The script must be run from the system where PAS is installed.
- The distribution script uses PBS Client commands. Hence the PBS Client must be installed on the PAS server.
- The distribution script asks for a location (a directory on the PAS Server) into which it will copy the Modern Communication Module before distributing it to the PBS MoMs. The permissions on this directory need to allow read and execute permissions for all users. This directory must be created before running the script. Once the script is run, the Modern Communication Module is copied to this directory on the PAS Server. The directory is then copied to each execution host and placed in the same location on each execution host.
- The distribution script asks for a user account. This user account will own the directory containing the Modern Communication Module on the PBS MoMs. This user must:
    - exist on the PAS server and all of the PBS MoMs
    - be able to SSH passwordlessly from the PAS Server into each MoM so that the Modern Communication Module can be securely copied to the PBS MoM.

Installation steps to distribute the Modern Communication Module:

1. Login to the machine where PAS is installed as the root user.
2. Source the Access Web configuration file to set up the environment variable `PA_EXEC`:

   `source /etc/pbsworks-pa.conf`
3. Create a directory where the distribution script can copy the Modern Communication Module before distribution to the PBS MoMs:

   `mkdir -m 755 -p /access/mcm`
4. Navigate to the directory `$PA_EXEC/joboperation/scripts`.
5. Execute the script to distribute the Modern Communication Module:

   `python distribute_modern_comm_modules.py`

> 📝 **Note:** Python 3 and above is required to run the modern communication module script.
>
> If your system does not have the latest python, then navigate to `$PA_EXEC/shared/thirdparty/python` and execute the following command:
>
> ```
> bin/python3.8 $PA_EXEC/joboperation/scripts/
> distribute_modern_comm_modules.py
> ```

6. Enter the directory that meets the stated prerequisites.

This is where the script will copy the Modern Communication Module before distributing it to the PBS MoMs.

For example, `/access/mcm`

7. Enter a user that meets the stated prerequisites.
For example, `pbsworks`.
The following message is displayed:

```
Do you want to override if modules already exist (Y/N). [Default: N]?
```

8. Choose one of the following:

- Enter `N` or press `ENTER` (to accept the default value) to skip copying the Modern Communication Module to the PBS MoM when it already exists on the MoM.
- Enter `Y` to overwrite the modules. The distribution script will copy the modules even if it detects that Modern Communication Module already exists on the MoM.

The script will display the inputs provided.

9. Enter `Y` to continue with the inputs provided.

Messages similar to the below are displayed:

```
Transferring Client Modules to n2.lab.com...
Successfully transferred to 'n2.lab.com'
Transferring Client Modules to n1.lab.com...
Successfully transferred to 'n1.lab.com'
Updated PAS Configuration file successfully
```

10. Restart Access Web by entering the command:

```
service pbsworks-pa restart
```

5.7 Install Remote Desktops Components on Linux Execution Hosts

Install components necessary to enable the remote visualization capabilities of Access Web on Linux execution hosts.

Ensure that the following prerequisites are met:

- Supported GPU: nvfbc and nvenc to support GRID, Tesla, or Quadro X2000+ (X = K/M/P/RTX etc) hardware products
- Graphics driver installed:
    - NVIDIA Windows display driver 471.41 or newer
    - NVIDIA Linux display driver 470.57.02 or newer
    - DirectX SDK (Windows only) CUDA 11.0 Toolkit
- X server configured with GPU and it is running.
- xsel or xclip library package should be installed on graphical node in case of Linux execution host.
- DPMS must be disabled. It can be done by adding `Option "DPMS" "false"` line under `monitor` section of `/etc/X11/xorg.conf` file.

Access Web provides scripts to run on PBS server machine and install desktop agent on graphical nodes.

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

1. Install the Remote Desktops component on the Workload Manager (PBS Server:
    - adds a custom resource called *ndesktops*.
    - creates an interactive queue called "iworkq".
    - updates the PBS ndesktops resource on provided graphics node by requesting the list of graphics node and number of ndesktops available.
2. Install the Remote Desktops component on all the execution hosts (PBS MoMs or AGE) on which you want to run interactive jobs.

5.7.1 Configure the Workload Manager

Configure Workload Manager PBS to support interactive applications.

Before you begin:

- Review the prerequisites for installation.

Installation must be done as root or as a user with sudo permissions.

On the Workload Manager Server, this installation will:

- add a custom resource to PBS Professional called *ndesktops*
- add an interactive queue called "iworkq"
- updates the PBS ndesktops resource on provided graphics node by requesting the list of graphics node and number of ndesktops available.

The below steps must be run on the PBS Server. If PBS Server is installed on the same machine these steps need to be run once.

1. Login to the machine where the PBS is installed.

2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:

   ```
   source /etc/pbsworks-pa.conf
   ```

3. Navigate to the directory `PA_EXEC/desktop/scripts/`

   ```
   cd $PA_EXEC/desktop/scripts/
   ```

4. Enter the command:

   ```
   bash ./pbs_server_graphics_configure.sh
   ```

   The above script will request you to provide the list of graphical node and number of ndesktops available to update the PBS ndesktops resource on the graphical node.

   The following message is displayed:

   ```
   Note: Please give the list of graphical node(s) having same number of gpus with
    comma separated(,)
   Please provide the list of graphical node(s):
   ```

5. Enter the list of the graphical node(s) that are available on the HPC complex and press `ENTER`.

   The following message is displayed:

   ```
   Please provide the number of gpu(s) that has to be configured for list of graphical
    node(s): 2
   Please provide gpu count:
   ```

6. Enter the number of GPUs that are available on the HPC complex and press `ENTER`.

   The following message is displayed.

   ```
   Do you want to update ndesktops on more graphical node(s)? [Y/N]
   ```

7. Choose one of the following options:

   - Enter **Y** to update ndesktops on more graphical node(s).
   - Enter **N** to exit the script.

5.7.2 Install Remote Desktops Component on Execution Hosts

Install binaries and configure the execution hosts (PBS MoM or AGE) to support interactive applications.

Before you begin:

- Review the Prerequisites for Installing Remote Desktops for installation.
- Run the Remote Desktops precheck diagnosis script on the PBS MoM to check the status of GPU nodes.

> 📝 **Note:** This installation process must be repeated for all graphical MoMs where you want to run interactive jobs.

1. Login to the machine where the graphical MoM is installed.

2. Navigate to `$MOM_CLIENT_MODULES/desktop/`.

`<MOM_CLIENT_MODULE>` is the location where Modern Communication Module is distributed. For example, `/access/mcm/momclientmodules`.

Check the `LINUX_NODE_MODERN_COMMUNICATION_SHARED_LIBS` variable value in the `$PA_HOME/config/pas/conf/server.conf` file to verify the `<MOM_CLIENT_MODULE>` location.

> 📝 **Note:** If you are installing on Windows, then check the `WINDOWS_NODE_MODERN_COMMUNICATION_SHARED_LIBS` variable value in the `$PA_HOME/config/pas/conf/server.conf` file to verify the `<MOM_CLIENT_MODULE>` location.

3. Execute the following command:

```
bash ./install.sh
```

> 📝 **Note:** If you are installing on Windows, then execute the following command: `<MOM_CLIENT_MODULE_PATH>\python\python.exe <MOM_CLIENT_MODULE_PATH>\python\install.py`

For example, the following message is displayed:

```
Allowing all users to start X server
Unzipping the ./packages/centos.tar.gz

Backup of changed configuration files and logs can be found at: /tmp/
desktop_2022-02-16_19-54-23
```

> 📝 **Note:** The `/tmp/desktop_2022-02-16_19-54-23` file must be backed up as this is required during uninstallation of Remote Desktops.

The script will provide permission to start X server by all the users and extract platform specific agent.

△ **ALTAIR**

5.8 Install Remote Desktops Component on Windows Execution Hosts

Install components necessary to enable the remote visualization capabilities of Access Web on Windows execution hosts.

Before you begin:

- Review the prerequisites for installation.

Access Web provides scripts to run on graphical nodes to install Remote Desktops components.

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

1. Login to the PBS MoM machine as root or as a user with sudo permissions.
2. Navigate to `<MOM_CLIENT_MODULE_PATH>\python\`.

   `<MOM_CLIENT_MODULE_PATH>` is the path where modern communication module is installed.
3. Open the Windows command prompt.
4. Run the following command:
   ```
   <MOM_CLIENT_MODULE_PATH>\python\python.exe <MOM_CLIENT_MODULE_PATH>\python
   \install.py
   ```
5. Once the installation is complete check that the Remote Desktops service is running:
   a) Click **Start** and choose **Run**.
   b) Type `services.msc` to open the Services Management Console.
   c) Locate the **DesktopAgentService**.

      The status should be "Started".
   d) If the service is not running then, right-click the **DesktpoAgentService** and click **Start**.

5.9 Enable Job History in PBS

The HPC cluster where simulations will run must have job history enabled to run simulation jobs.

Skip this step if your site has not installed the Simulator component.

The Simulator component requires that job history be enabled on the PBS Server where simulation jobs are run:

1. Login to the PBS Server where simulation jobs will run as a PBS manager.
2. Execute the command:

   ```
   qmgr -c 'set server job_history_enable=True'
   ```

Post-Installation Configuration 6

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

This chapter covers the following:

-
-
-

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

6.1 Application Definitions

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

To run a solver or application through Access Web requires an application definition. 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:

`PA_HOME/data/pas/targets/localhost/repository/applications`.

Application definitions now need to be compatible with Python 3.

> 📄 **Note:** Python usage in application definitions is usually minimal and straightforward. Validate the updated application definitions before using it.

For sites that are installing Access Web for the first time, default application definitions and a site configuration file are provided to get you up and running quickly. Obtain them through your usual Altair PBS Works Support (pbssupport@altair.com).

For sites that are using legacy products such as Compute Manager, legacy application definitions can be ported so that they can be used by Access Web.

Access Web supports application definitions in XML and JSON format.

Users can also have their own personal application definition repositories at `/home/$USER/userapps`.

The path of the user apps folder must be added as a Root Directory while registering a cluster, so that the users can view their application definition in Access Web user interface.

6.1.1 Copy Application Definitions and Site Configuration File

Copy default application definitions and a PAS site configuration file.

Application definitions now need to be compatible with Python 3.

> 📄 **Note:** Python usage in application definitions is usually minimal and straightforward. Validate the updated application definitions before using it.

Default application definitions and a site configuration file are provided to get you up and running quickly. Obtain them through Altair PBS Works Support (pbssupport@altair.com).

1. Login to the machine where PAS is installed as root or as a user with sudo permissions.
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/`

6.1.2 Onboard an Application Definition

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

Application definitions now need to be compatible with Python 3.

> 📄 **Note:** Python usage in application definitions is usually minimal and straightforward. Validate the updated application definitions before using it.

Access Web allows onboarding the legacy application definitions in a very easy way and it can be done at any time.

> 📄 **Note:** The `site-config.xml` file must be updated manually after you onboard the legacy application definitions or you can copy over a site configuration file from a previous installation of Access or Compute Manager.

Recommended techniques for onboarding legacy application definitions:

Test Legacy Application Definitions
Submit jobs to an HPC cluster and verify that Access Web can render and submit the job properly.

Upgrade 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:

Submit a Job using a Right-Click Context Menu
Access Web includes a feature that allows a job to be submitted to an HPC cluster by right-clicking a job input file and choosing a solver. This eliminates the need to submit a job using a job submission form. Below are changes that must be made to the application definition to enable this feature:

PRIMARY_FILE and QUEUE Arguments 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

△ **ALTAIR**

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.

Extracting Include Files from a Master File

Another feature available with Access is the ability to extract the names of include files from a master file and automatically populate a job submission form argument of type FILE_MULTI with the list of include files. A special script called the Master File Analyzer script is required to read the master file and identify the include files.

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.

Refer to the tutorial *Extracting Include Files from a Master File* in *Diving Into Application Definitions* to convert or write application definitions that support the Master File Analyzer script.

Integrate Right Click Context Menu of Access Web

Enrich your legacy application definition so that a job can be submitted by right-clicking a job input file and selecting a solver.

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 server 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.

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.

△ ALTAIR

6.2 Configure the License Server

Configure the license server when it was not provided during the installation of Access Web.

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:** Multiple license servers can be configured in the license configuration using the delimiter ":" in Linux and ";" in Windows.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
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
   ```

   > 📝 **Note:** If you have a different license server for HyperWorks and Compose for RVS, then add the license information in the *pbsworks.license.server*.

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

   ```
   service pbsworks-pa restart
   ```

△ **ALTAIR**

6.3 Configure Results Visualization Service

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

6.3.1 Configure HyperWorks Location

Configure the Altair HyperWorks Desktop location to enable the results visualization capabilities for animation.

Follow these steps when the HyperWorks location was not specified while installing Access Web 2022.1.0.

1. Edit the file at `PA_HOME/config/resultmanager/site_config.xml`.
2. Change the value of the *location* to the location of HyperWorks.
    For example, `<HYPERWORKS_INSTALL_DIR>/altair`

    ```
    <Products>
       <Product id="ALTAIR_HYPERWORKS" defaultVersion="13.0">
           <Version id="13.0" location="/opt/HyperWorks/altair" />
       </Product>
    </Products>
    ```
3. Restart Access Web using the following command:

    ```
    service pbsworks-pa restart
    ```

6.3.2 Configure Compose Location

Configure the Compose location to enable the results visualization capabilities for plot.

Follow these steps when the Compose location was not specified while installing Access Web 2022.1.0

1. Edit the file at `PA_HOME/config/resultmanager/site_config.xml`.
2. Change the value of the *location* to the location of Compose.
    For example, `<COMPOSE_INSTALL_DIR>/altair/Compose2021.2`

    ```
    <Products>
        <Product id="COMPOSE" defaultVersion="2021.2">
              <Version id="2021.2" location="/opt/Compose/altair/Compose2021.2" />
          </Product>
    </Products>
    ```
3. Restart Access Web using the following command:

    ```
    service pbsworks-pa restart
    ```

6.3.3 Configure Environment Variables Set for Compose

Configure the Compose environment variables to enable the results visualization capabilities for plot.

Set the environment variables for the following parameters: `ALTAIR_HOME, HW_ROOTDIR, HW_UNITY_ROOTDIR, PATH,` and `LD_LIBRARY`.

1. Edit the file `/opt/altair/pbsworks/2022.1.0/access/exec/resultmanager/scripts/`
    `setenv.sh`.

2. Update the Compose installation location path: `<COMPOSE_INSTALL_DIR>/altair/`
    `Compose2021.2/`.

    ```
    !/bin/sh
    export ALTAIR_HOME=/opt/Compose/altair/Compose2021.2
    export HW_ROOTDIR=/opt/Compose/altair/Compose2021.2
    export HW_UNITY_ROOTDIR=/opt/Compose/altair/Compose2021.2/hwx
    export HW_PLATFORM=linux64
    export PLATFORM=linux64
    export ALTAIR_PROD_ARCH=linux64
    export PATH=$PATH:/opt/Compose/altair/Compose2021.2/hwx/bin/linux64:/opt/Compose/
    altair/Compose2021.2/hw/bin/linux64
    export LD_LIBRARY_PATH=/opt/Compose/altair/Compose2021.2/hwx/bin/linux64:/opt/
    Compose/altair/Compose2021.2/hw/bin/linux64
    export HW_EDITION=business
    ```

6.3.4 Configure HyperWorks Licenses

Install the HyperWorks Desktop feature, HWHyperViewTrans and Compose in Altair License Server.

These feature licenses must be available via a Altair License Server 14.5.1 or newer license server to enable the visualization of the supported result files. To configure, point RVS to the HyperWorksinstalled license server, refer Configure the License Server.

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

> 🛈 **Tip:**
>
> - Install HyperWorks Desktop by following the Linux installation instructions in the
>   `HyperWorks 2021.2 Installation Guide`.

Licensing System of HyperWorks Units

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

Animation requests checkout 6 HWUs and plot requests checkout 10 HWUs. Units are leveled for the same user but stacked for different users. Licenses are checked out only during the results extraction. 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 Altair Access licensing and does not require any extra units.

6.3.5 Configure Data Directory

Configure the 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/resultmanager/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 using the command, `service pbsworks-pa restart`.

6.3.6 Kill Compose Process Ids

After upgrade or installation, kill any Compose Process ids of your previous version of Access Web.

Follow the given steps:

1. To check if there any of the Compose processes that are running, use the command:

    `ps -ef | grep composeserv`

    The list of Compose process ids is displayed.
2. To kill a process id, specify the `process id` in the following command:

    `kill -9 <process id>`

6.3.7 Allow Pop-up Windows

In the supported web browsers, allow the pop-up windows to view the auto-refresh loading messages for a running job.

Follow the steps given to enable pop-up windows in Google Chrome.

1. Open the supported browser, Google Chrome.
2. Click ⋮ located in the top-right corner of the web page.
3. Choose **Settings**. Click **Advanced** and choose **Site Settings**.
4. Click **Pop-ups and redirects** and change the permissions as follows: Slide the option to **Allowed**. This will allow pop-up windows for all the sites you access.
5. Click **Add** under **Allow** section.
6. On the **Add a site** window, add the Site address. This will allow pop-up windows for the sites you add.

△ALTAIR

Add Servers

Login to Access Web as the Service User to add servers.

This chapter covers the following:

7.1 Log into Access Web as the Service User

Login to Access Web as the Service User to add servers.

1. Open a supported web browser.
2. Enter the URL `https://<hostname>:4443/pbsworks` in the address bar.

    where `<hostname>` is the IP address or hostname of the machine where Access Web is installed.

    The Access Web login screen is displayed.
3. Enter the Service User credentials.
4. Click **Log In**.

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

7.2 Add a Server Cluster

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

A server cluster is a PAS Server that is connected to an HPC complex. Before you can submit a job, a server cluster must be added. Only users that have been assigned a Manager role can add a server cluster. The Service User entered during the installation of Access Web is automatically assigned a Manager role.

When Access Web is installed on the same machine as PAS, a default local server is automatically added, eliminating the need for the Service User to add a server. Users can begin submitting jobs immediately. Access to the local filesystem is also configured through the default server. By default, users have access to the Job File Stage Area directory (default is `/stage`) that is defined at installation. Edit the default server to configure access to additional directories such as user home directories, for example `/home`.

Configure the default home location of users when you register a PAS server. Environment variables can be used to define the user's default home location like `$HOME` representing the PAS home directory and `$USER_HOME` or `%U` representing username. Access Web will display the contents of the user's home directory path when the user logs into Access Web.

The following are examples for setting the user's home directory:

In Linux : `/stage, /home, /stage/$USER, /home/$USER/project`

In Windows: `C:/stage, C:/Users, C:/%HOMEPATH%, C:/%HOMEPATH%/project`

> 📝 **Note:** The user's credentials should be available in the PAS server when you define the home location path with the user (`$USER_HOME` or `%U`).

An **Autocreate Directory** option is provided while adding a server cluster and it is set to true. This will create the root directory specified if it is not available in PAS.

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

    The Add Server Cluster screen is displayed.

Figure 7: Add Server 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.

    > 📒 **Note:** It is recommended to add a server 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 account and password must be available on the PAS Server.
5. For **Root Dir**, enter the pathname where user job input and result files are stored.

    Ex: `/home, /users, /stage, /stage/$USER, /home/$USER/project, C:/stage, C:/Users, C:/%HOMEPATH%, C:/%HOMEPATH%/project`

    > 📒 **Note:** Add parent directory of all Users home during server registration to make home button work.

6. Click (+ Root Dir) if you want to add another **Root Dir** and enter the pathname.

Figure 8: Multiple Root Directory Entry

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

Figure 9: Default Root Directory

8. Optional: Click **Autocreate Directory** checkbox if you do not want the root directory to be created.

> ℹ️ **Tip:** By default, auto create directory option is set to true to create the root directory specified if it is not available in PAS.

9. Click **Save**.
If the server cluster is added successfully, then a notification is displayed.

△ **ALTAIR**

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

A list of server clusters that have been added is displayed.

Manage Servers					Add
🟩 Available 🟥 Not Available					
Name	**Url**	**Last Seen On**	**Last Modified**	**Details**	⚙
🟩 cluster	https://172.16.80.18:5243/pas	1/28/2021, 9:45:56 PM	1/28/2021, 9:45:31 PM	Available	
🟩 hpccluster	https://localhost:5243/pas	1/28/2021, 9:45:50 PM	1/28/2021, 1:23:40 PM	Available	

Figure 10: Server Clusters List

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

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

Manage Services					Add
🟩 Available 🟥 Not Available					
Name	**Url**	**Last Seen On**	**Last Modified**	**Details**	⚙
🟥 cluster	https://172.16.18.80:5243/pas	11/29/2019, 5:29:10 AM	11/29/2019, 5:23:59 AM	I/O error on GET request for "https://172.16.18.80:52	⋮
🟩 hpccluster	https://localhost:5243/pas	11/29/2019, 5:27:40 AM		I/O error on GET request for "https://172.16.18.80:5243/pas/restservice/jobs/select": Connect to 172.16.18.80:5243 [/172.16.18.80] failed: connect timed out; nested exception is org.apache.http.conn.ConnectTimeoutException: Connect to 172.16.18.80:5243 [/172.16.18.80] failed: connect timed out	

Figure 11: Server Cluster Details

10. Click **Add** and repeat steps 2 through 9 to add additional server clusters.

See Also
Troubleshoot PBS Application Services

△ ALTAIR

Downgrade Access Web **8**

Instructions for downgrading Access Web 2022.1.0 to a previous version.

Perform these steps to downgrade Access Web and PAS (single and different machine setup) 2022.1.0 to 2021.1.1 or 2021.1.2.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Stop Access Web 2022.1.0 using the following command:

   ```
   service pbsworks-pa stop
   ```
3. Create a backup of `/etc/pbsworks-pa.conf` as `/etc/pbsworks-pa.conf.2022.1.0`
4. Rename `/etc/pbsworks-pa.conf.CurrentlyInstalledVersion` to `/etc/pbsworks-pa.conf`
5. Copy `PreviousVersion_PA_EXEC/init/pbsworks-pa` script to `/etc/init.d/`
   You are successfully downgraded to previous version of Access Web.
6. Start Access Web using the following command:

   ```
   /etc/init.d/pbsworks-pa start
   ```
7. 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.
8. Enter your username and password.
9. Click **Log In.**

Uninstall Access Web and its Components

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

This chapter covers the following:

9.1 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 to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Navigate to the installation directory.

   The default installation directory is `/opt/altair/pbsworks/2022.1.0/access/`
   `Altair_Access_installation/`
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.

9.2 Uninstall Remote Desktops Agents on Linux

Instructions for uninstalling Remote Desktops Agent on Linux.

You need the `/tmp/desktop_<Installation Timestamp>` file that is taken as a backup during the installation.

For example, `/tmp/desktop_2022-02-16_19-54-23` is created during installation.

To uninstall, the configuration file must be copied back from the `/tmp/desktop_2022-02-16_19-54-23`.

Perform the following steps in all the PBS MoM's.

1. Login in to PBS MoM machine where Remote Desktops Agent is installed.
2. If you find `xserver` file in `/tmp/desktop_2022-02-16_19-54-23` folder then replace that in `/etc/pam.d/xserver`
3. If you find `permissions.local` file in `/tmp/desktop_2022-02-16_19-54-23` folder then replace that in `/etc/permissions.local`
4. If you find `Xwrapper.config` file in `/tmp/desktop_2022-02-16_19-54-23` folder then replace that in `/etc/X11/Xwrapper.config`

9.3 Uninstall Remote Desktops Agents on Windows

Instructions for uninstalling Remote Desktops Agent on Windows.

Perform the following steps in all the PBS MoM's.

1. Login in to graphical MoM machine where Remote Desktops Agent is installed as an administrator.
2. Open the command prompt.
3. Enter the following command to stop the Desktop Agent service:
   ```
   sc stop DesktopAgentService
   ```
4. Enter the following command to delete the Desktop Agent service:
   ```
   sc delete DesktopAgentService
   ```

Access Web Service Commands   **10**

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

This chapter covers the following:

The commands should be executed by the root or as a user with sudo permissions as defined in `/etc/pbsworks-pa.conf`.

10.1 Start Access Web

Start all Access Web services.

Access Web starts a watcher process which will monitor the status of Access Services. The watcher process automatically brings up any services that goes down abruptly.

Starting Access Web must be done as root or as a user with sudo permissions.

1. Login to the machine where Access Web is installed.
2. Enter the following command to start Access Web:

```
service pbsworks-pa start
```

10.2 Stop Access Web

Stop all Access Web services.

Stopping Access Web must be done as root or as a user with sudo permissions.

1. Login to the machine where Access Web is installed.
2. Enter the following command to stop Access Web:

    ```
    service pbsworks-pa stop
    ```

10.3 Restart Access Web

Restart all Access Web services.

Restarting Access Web must be done as root or as a user with sudo permissions.

1. Login to the machine where Access Web is installed.
2. Enter the following command to restart Access Web:

```
service pbsworks-pa restart
```

10.4 Determine the Status of all Access Web Services

Determine whether a Access Web service is up or down.

1. Login to the machine where Access Web is installed.
2. Enter the following command to display the status of each Access Web service:

   ```
   service pbsworks-pa status
   ```

   Messages similar to the following are displayed:

   ```
   PBSWORKS_EXEC =>/opt/altair/pbsworks/2022.1.0/access/exec
   PBSWORKS_HOME =>/var/spool/pbsworks/2022.1.0/access/home
   PAS_REPO: => /var/spool/pbsworks/2022.1.0/access/home//data/pas/

   api_gateway_service is Running (30056)  [OK]
   message_broker_service is Running (30257)        [OK]
   database_service is Running (Database)  [OK]
   ams_service is Running (30345)  [OK]
   pa_service is Running (30423)    [OK]
   executor_service is Running (30500)      [OK]
   desktop_service is Running (30615)       [OK]
   vnc_router_service is Running (30688)    [OK]
   desktop_edgeserver_service is Running (30744)    [OK]
   resultmanager_service is Running (31071)         [OK]
   pas_service is Running (31437)  [OK]
   ```

Manage Clusters

11

Add, update, or delete HPC clusters.

This chapter covers the following:

11.1 Add a Server Cluster

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

A server cluster is a PAS Server that is connected to an HPC complex. Before you can submit a job, a server cluster must be added. Only users that have been assigned a Manager role can add a server cluster. The Service User entered during the installation of Access Web is automatically assigned a Manager role.

When Access Web is installed on the same machine as PAS, a default local server is automatically added, eliminating the need for the Service User to add a server. Users can begin submitting jobs immediately. Access to the local filesystem is also configured through the default server. By default, users have access to the Job File Stage Area directory (default is `/stage`) that is defined at installation. Edit the default server to configure access to additional directories such as user home directories, for example `/home`.

Configure the default home location of users when you register a PAS server. Environment variables can be used to define the user's default home location like `$HOME` representing the PAS home directory and `$USER_HOME` or `%U` representing username. Access Web will display the contents of the user's home directory path when the user logs into Access Web.

The following are examples for setting the user's home directory:

In Linux : `/stage, /home, /stage/$USER, /home/$USER/project`

In Windows: `C:/stage, C:/Users, C:/%HOMEPATH%, C:/%HOMEPATH%/project`

> 📝 **Note:** The user's credentials should be available in the PAS server when you define the home location path with the user (`$USER_HOME` or `%U`).

An **Autocreate Directory** option is provided while adding a server cluster and it is set to true. This will create the root directory specified if it is not available in PAS.

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

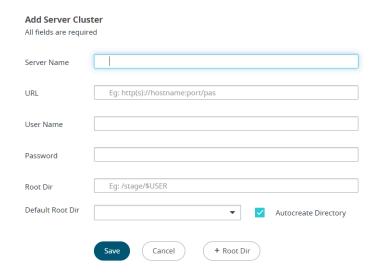   The Add Server Cluster screen is displayed.

△ **ALTAIR**

Figure 12: Add Server 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.

   > 📝 **Note:** It is recommended to add a server 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 account and password must be available on the PAS Server.

5. For **Root Dir**, enter the pathname where user job input and result files are stored.

   Ex: `/home, /users, /stage, /stage/$USER, /home/$USER/project, C:/stage, C:/Users, C:/%HOMEPATH%, C:/%HOMEPATH%/project`

   > 📝 **Note:** Add parent directory of all Users home during server registration to make home button work.

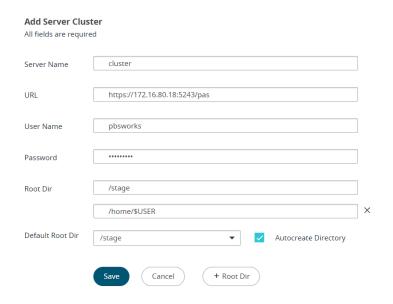6. Click (+ Root Dir) if you want to add another **Root Dir** and enter the pathname.

Figure 13: Multiple Root Directory Entry

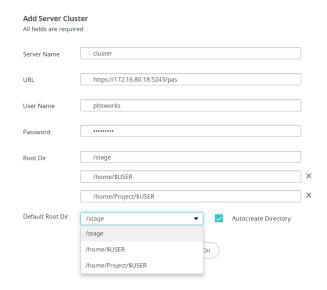**7.** Select the default root directory to be displayed from the **Default Root Dir** drop-down menu.
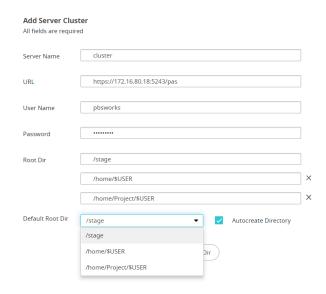
Figure 14: Default Root Directory

8. Optional: Click **Autocreate Directory** checkbox if you do not want the root directory to be created.

> ℹ️ **Tip:** By default, auto create directory option is set to true to create the root directory specified if it is not available in PAS.

9. Click **Save**.
If the server cluster is added successfully, then a notification is displayed.

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

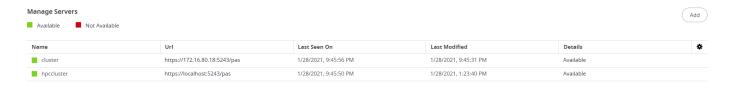A list of server clusters that have been added is displayed.

Figure 15: Server Clusters List

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

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

Figure 16: Server Cluster Details

10. Click **Add** and repeat steps 2 through 9 to add additional server clusters.

See Also
Troubleshoot PBS Application Services

11.2 Edit a Server Cluster

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

> 📄 **Note:** Only a user that has been assigned a Manager role can edit a server cluster.

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

Figure 17: Manage Servers

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

Figure 18: Cluster Edit Option

The Edit Server Cluster screen is displayed.

Figure 19: Edit Server Cluster

4. Update the server cluster information.

5. Click **Save**.

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

11.3 Delete a Server Cluster

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

> 📝 **Note:** Only a user that has been assigned a Manager role can delete a server cluster.

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

   Figure 20: Manage Servers

2. Right-click the cluster that you want to remove.
3. Click **Delete** from the context menu.
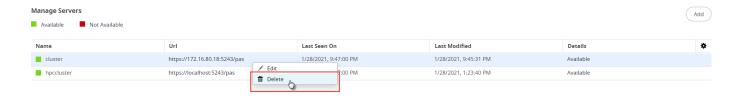
   Figure 21: Server Cluster Delete

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 server
   > cluster gets added, edited, deleted, if it goes down or if it is unreachable.

Advanced Configurations

Advanced configurations for Access Web and its services.

This chapter covers the following:

12.1 Configure Single Sign-On

Configure Single Sign-On (SSO) with Active Directory Federation Server (AD FS) 2.0 using Security Assertion Markup Language (SAML) protocol and with Okta integration using SAML 2.0.

Configuring Single Sign-On will enable clients who are logged into a domain to access the application without providing credentials a second time.

12.1.1 Prerequisite for Single Sign-On

Mandatory requirements for Single Sign-On.

The following are the ADFS prerequisites for installation:

- Active Directory Federation Server (AD FS) 2.0 should be installed and configured on Windows Server machine.
- Access server should be able to reach AD FS server using its Fully Qualified Domain Name (FQDN).
- Access server machine should be configured against AD for identity provider (System Security Services Daemon in Linux).
- To pull the jobs continuously register PAS Service with the Linux account before enabling SAML.

The following are the Okta prerequisites for installation:

- Okta account should be available and active.
- Okta username should match the username in HPC.

12.1.2 Configure Single Sign-On on Linux using ADFS

Configure Access Web using ADFS to allow clients who are logged into a domain to access the application without providing credentials a second time.

> 📝 **Note:** By default SSO is disabled.

The following steps must be done as root or as a user with sudo permissions.

1. Using a browser, navigate to the following URL to download the Identity Provider (IDP) metadata file:

   `https://<adfserver>/FederationMetadata/2007-06/FederationMetadata.xml`

   Where *adfsserver* is the AD FS server FQDN.

   The metadata file begins to download.
2. Login to the machine where Access Web is installed.
3. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:

   `source /etc/pbsworks-pa.conf`

4. Copy the downloaded metadata XML file to the location `$PA_HOME/config/sp/metadata/idp.xml`.

5. Edit the file `PA_HOME/config/ams/jaas-config/AA_jaas.config` to configure the Java Authentication and Authorization Service (JAAS) module.

    ```
    vi $PA_HOME/config/ams/jaas-config/AA_jaas.config
    ```

6. Add the below line under LoginModule.

    ```
    com.altair.jaas.module.PasswordLessLoginModule Sufficient;
    ```

    The `AA_jaas.config` file after adding the line will be as shown below:

    ```
    LoginModule {
                com.altair.jaas.module.PasswordLessLoginModule Sufficient;
                com.altair.jaas.module.SSHUnixLoginModule Required
                host="localhost"
                port="22"
                cache_ttl="60"
                debug=false;
            };
    ```

7. Edit the file `PA_HOME/config/sp/securityContext.xml` to configure the user-facing Access server domain name or IP address.

    ```
    vi $PA_HOME/config/sp/securityContext.xml
    ```

8. Update *entityBaseURL* with the domain name or IP address in *metadataGeneratorFilter* configuration.
    For example,

    ```
    <property name="entityBaseURL" value="https://localhost:4443/pbsworks/sp/auth"/>
    ```

9. Update *serverName* with your domain name or IP address in *contextProvider* configuration.
    For example,

    ```
    <property name="serverName" value="localhost"/>
    ```

10. Edit the file `PA_HOME/config/pa/configuration.json` to enable SSO.

    ```
    vi $PA_HOME/config/pa/configuration.json
    ```

11. Enable SSO by changing the value of the *enableSSO* key to true.

    ```
    enableSSO = true
    ```

12. Start Access Web using the following command:

    ```
    service pbsworks-pa start
    ```

12.1.3 Configure Single Sign-On on Linux using Okta

Configure Access Web using Okta to allow clients who are logged into a domain to access the application without providing credentials a second time.

The following details are required by Okta application configurations:

*   Single Sign-On URL, Recipient URL, and Destination URL: `https://<ACCESS_HOSTNAME>:<ACCESS_PORT>/pbsworks/sp/auth/saml/SSO`

*   SP Entity ID - provide your own SP entity ID or the default value is `https://<ACCESS_HOSTNAME>:<ACCESS_PORT>/pbsworks/sp/auth/saml/metadata`

*   Mapping HPC username field - if your Okta username is not matching HPC users, then configure the mapping field.

△ ALTAIR

> 📝 **Note:** By default SSO is disabled.

The following steps must be done as root or as a user with sudo permissions.

1. Download Identity Provider (IDP) metadata.

2. Login to the machine where Access Web is installed.

3. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:

    ```
    source /etc/pbsworks-pa.conf
    ```

4. Copy the downloaded metadata XML file to the location `$PA_HOME/config/sp/metadata/idp.xml`.

5. Edit the file `PA_HOME/config/ams/jaas-config/AA_jaas.config` to configure the Java Authentication and Authorization Service (JAAS) module.

    ```
    vi $PA_HOME/config/ams/jaas-config/AA_jaas.config
    ```

6. Add the below line under LoginModule.

    ```
    com.altair.jaas.module.PasswordLessLoginModule Sufficient;
    ```

    The `AA_jaas.config` file after adding the line will be as shown below:

    ```
    LoginModule {
                com.altair.jaas.module.PasswordLessLoginModule Sufficient;
                com.altair.jaas.module.SSHUnixLoginModule Required
                host="localhost"
                port="22"
                cache_ttl="60"
                debug=false;
           };
    ```

7. Edit the file `PA_HOME/config/sp/securityContext.xml` to configure the user-facing Access server domain name or IP address.

    ```
    vi $PA_HOME/config/sp/securityContext.xml
    ```

8. Update *entityBaseURL* with the domain name or IP address in *metadataGeneratorFilter* configuration.
    For example,

    ```
    <property name="entityBaseURL" value="https://localhost:4443/pbsworks/sp/auth"/>
    ```

9. Update *serverName* and *serverPort* with your domain name or IP address in *contextProvider* configuration.
    For example,

    ```
    <property name="serverName" value="localhost"/>
    <property name="serverPort" value="4443"/>
    ```

    > 📝 **Note:** The steps 10 to 13 are optional.

10. Uncomment *userDetails* property in *samlAuthenticationProvider* bean configuration to configure mapping field to match Okta username to HPC user.

    ```
    <property name="userDetails" ref="userDetailsProvider" />
    ```

11. Provide mapping *fieldName* as value in *userDetailsProvider* bean configuration.
    For example, if *HPCusername* is the mapping `fieldName` provided in Okta configuration then it will be displayed as:

```
<bean id="userDetailsProvider"
 class="com.altair.security.saml.UserDetailsProvider">
         <property name="accountName" value="HPCusername"/>
</bean>
```

12. Uncomment *entityId* property in *metadataGeneratorFilter* bean configuration to configure your own SP entity.

```
<property name="entityId" value="replaceWithUniqueIdentifier"/>
```

13. Update the *replaceWithUniqueIdentifier* with the SP entity.

```
<property name="entityId" value="replaceWithUniqueIdentifier"/>
```

14. Edit the file `PA_HOME/config/pa/configuration.json` to enable SSO.

```
vi $PA_HOME/config/pa/configuration.json
```

15. Enable SSO by changing the value of the *enableSSO* key to true.

```
enableSSO = true
```

16. Start Access Web using the following command:

```
service pbsworks-pa start
```

12.1.4 Configure AD FS as an Identity Provider

Configure AD FS as an identity provider using the service provider metadata.

The workflow to configure AD FS as an identity provider is as follows:

- Get the metadata.xml from Access Server.
- Import the data into AD FS.
- Add rules.
- Change to SHA-1.

> 📝 **Note:** These are general steps for configuring the AD FS as an identify provider. Steps may vary depending upon your site's version of Windows.

1. Using a browser, navigate to the following URL:

   `https://<accessserver>:4443/pbsworks/sp/auth/saml/metadata`

   where *accessserver* is the FQDN or IP Address of the machine where Access Web is installed..

2. Save the content of the metadata file as `metadata.xml`.

3. Upload the `metadata.xml` to the AD FS server.

4. Open AD FS 2.0 Management snap-in from **Start** > **Administrative Tools** > **ADFS 2.0 Management** > **Add Relying Party Trust**.

5. Select **Import data about the relying party from a file** and upload the metadata XML file. The wizard may complain that some content of metadata is not supported. You can safely ignore this warning.

6. On the **Ready to Add Trust** ensure that tab endpoints contains multiple endpoint values.

   If not, verify that your metadata was generated with HTTPS protocol URLs.

7. Verify that **Open the Edit Claim Rules dialog** is checked.

8. Select **Add Rule**, choose **Send LDAP Attributes as Claims** and click **Next**.

9. Update the following information:
    - Enter *NameID* for Claim rule name.
    - Choose *Active Directory* for Attribute store.
    - Choose *SAM-Account-Name* as LDAP Attribute.
    - Enter *Name ID* as Outgoing claim type.
10. Double-click the provider and select **Advanced** tab.
11. Change *Secure hash algorithm* to SHA-1.

12.1.5 Configure Signature Request Certificate

Configure your own Signature Request Certificate.

The configuration file to change the Signature Request Certificate is located at `PA_HOME/config/sp/` `securityContext.xml`.

Update the following parameters value:

- *constructor-arg value* - this argument points to the used key store file
- *constructor-arg type* - this argument contains password for the keystore
- *entry key* - this argument maps with passwords for private keys with alias-password value pairs
- *constructor-arg type* - this argument is the alias of the default certificate

1. Login to the machine where Access Web is installed.
2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:

    ```
    source /etc/pbsworks-pa.conf
    ```
3. Copy your signature certificate to `$PA_HOME/config/sp/security`.
4. Navigate to the location `PA_HOME/config/sp/`.

    ```
    cd $PA_HOME/config/sp/
    ```
5. Edit the `securityContext.xml` file.
6. Update the parameter values:

    ```
    <!-- Central storage of cryptographic keys -->
        <bean id="keyManager"
     class="org.springframework.security.saml.key.JKSKeyManager">
            <constructor-arg value="classpath:security/access.keystore"/>
            <constructor-arg type="java.lang.String"
     value="k86BCuq3mLrCqUGZVj3n9DupJ2ePqv"/>
            <constructor-arg>
            <map>
                    <entry key="pbsworks" value="k86BCuq3mLrCqUGZVj3n9DupJ2ePqv"/>
            </map>
            </constructor-arg>
            <constructor-arg type="java.lang.String" value="pbsworks"/>
        </bean>
    ```
7. Start Access Web using the following command:

    ```
    service pbsworks-pa start
    ```

12.1.6 Verify Single Sign-On on Linux

Verify if Single Sign-On is enabled on Linux.

1. Open a supported web browser.
2. Enter the URL `https://<hostname>:4443/pbsworks` in the address bar.
   where `<hostname>` is the IP address or hostname of the machine where Access Web is installed.
   The Access Web server will redirect to authenticate against the AD FS server or Okta.
3. Enter your credentials.
   The credentials will be authenticated and the Access Web user interface is displayed.

12.1.7 Disable Single Sign-On on Linux

Disable Single Sign-On so that users are forced to enter their credentials when logging into Access Web.

The following steps must be done as root or as a user with sudo permissions.

1. Login to the machine where the Access Web is installed.
2. Edit the file `PA_HOME/config/pa/configuration.json` to disable SSO.
3. Disable SSO by changing the value of the *enableSSO* key to false.
   ```
   enableSSO = false
   ```
4. Refresh the browser.

12.2 Configure the Access Web Component

Configurations required for Access Web component.

12.2.1 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 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.

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

Change 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.

For information on default port and port range, refer to Ports Used by Access Web.

The gateway port number has to be updated in the files:

- `nginx.conf`
- `rm_servers.xml`
- `desktop.properties`

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Edit the file `PA_HOME/config/api_gateway/nginx.conf`
3. Update the value of *listen* of *server*:
   ```
   server {
           listen         4443 ssl;
           server_name  localhost;
           add_header X-Frame-Options "SAMEORIGIN";
   ```
4. Edit the file `PA_HOME/config/resultmanager/rm_servers.xml`
5. Update the value in `<PAServerURL>`.
   ```
   <PAServerURL>https://localhost:4443/pbsworks/api</PAServerURL>
   ```
6. Edit the file `PA_HOME/config/desktop/desktop.properties`
7. Update value of *pbsaccess.api_gateway.service.host*.
   ```
   pbsaccess.api_gateway.service.host=https://localhost:4443/pbsworks/api
   ```
8. Start Access Web by entering the command:
   ```
   service pbsworks-pa start
   ```

Change 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. For information on default port and port range, refer to Ports Used by Access Web.

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

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

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Edit the file `PA_HOME/config/pa/tomcat/conf/server.xml`
3. Search for the Connector port and update with the new port number.
   ```
   <Connector port="4543" protocol="org.apache.coyote.http11.Http11NioProtocol"
   address="127.0.0.1"
                   scheme="https" compression="on" compressionMinSize="2048"
   ```
4. Edit the file `PA_HOME/config/api_gateway/nginx.conf`
5. Update the port number of *server*.
   ```
   upstream pbsaccess {
        server 127.0.0.1:4543;
   }
   ```
6. Edit the file `PA_HOME/config/desktop/desktop.properties`.
7. Update the port number of *pbsaccess.storage.service.host*.

   ```
   pbsaccess.storage.service.host=https://localhost:4543/storage
   ```
8. Edit the file `PA_HOME/config/desktop/desktop.properties.template`.
9. Update the port number of *pbsaccess.storage.service.host*.

   ```
   pbsaccess.storage.service.host=https://localhost:4543/storage
   ```
10. Start Access Web by entering the command:
    ```
    service pbsworks-pa start
    ```

Change 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. For information on default port and port range, refer to Ports Used by Access Web.

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`

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

Change the Message Broker Port Number

Change the port that the NATS message broker listens on.

You must stop Access Web before changing the port number. For more information about stopping Access Web, see Access Web Service Commands. For information on default port and port range, refer to Ports Used by Access Web.

The Message Broker port number has to be updated in the files:

- `nats-server.config`
- `app.properties`

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Edit the file `PA_HOME/config/shared/nats-server.config`.
3. Update the port number of *port*.
   ```
   port: 4743
   ```
4. Edit the file `PA_HOME/config/shared/app.properties`.
5. Update the port number in *nats.broker.url*.
   ```
   nats.broker.url=nats://vm8cent74-rvsdev:4743
   ```
6. Start Access Web by entering the command:
   ```
   service pbsworks-pa start
   ```

Change the Remote Desktops Web Server Port Number

Change the port that the Remote Desktops 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. For information on default port and port range, refer to Ports Used by Access Web.

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

- `server.xml`
- `nginx.conf`

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.

2. Edit the file `PA_HOME/config/desktop/tomcat/conf/server.xml`.

3. Search for the Connector port and update the new port number.

    ```
    <Connector port="4843" protocol="HTTP/1.1" address="127.0.0.1"
                    scheme="https" compression="on" compressionMinSize="2048"
    ```

4. Edit the file `PA_HOME/config/api_gateway/nginx.conf`.

5. Update the port number of *server localhost*.

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

6. Start Access Web by entering the command:

    ```
    service pbsworks-pa start
    ```

Change 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. For information on default port and port range, refer to Ports Used by Access Web.

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

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.

2. Edit the file `PA_HOME/config/resultmanager/tomcat/conf/server.xml`.

3. Search for the Connector port and update the new port number.

    ```
    <Connector port="5043" maxThreads="200" address="127.0.0.1" scheme="https"
        compression="on" compressionMinSize="2048" noCompressionUserAgents="gozilla,
     traviata"
        compressableMimeType="text/html,text/xml,text/javascript,text/
    css,application/javascript,text/plain,application/json"
        useSendfile="false" secure="true" SSLEnabled="true" clientAuth="false"
     sslProtocols="TLSv1, TLSv1.1, TLSv1.2"
        keystoreFile="${PBSWORKS_HOME}/config/shared/access.keystore"
     keystorePass="k86BCuq3mLrCqUGZVj3n9DupJ2ePqv" keyAlias="pbsworks"
        ciphers="TLS_RSA_WITH_AES_128_CBC_SHA,SSL_RSA_WITH_RC4_128_SHA"/>
    ```

4. Edit the file Navigate to `PA_HOME/config/api_gateway/nginx.conf`.

5. Update the port number of *server localhost*.

    ```
    upstream resultmanager {
        server localhost:5043;
    }
    ```

6. Start Access Web by entering the command:

    ```
    service pbsworks-pa start
    ```

△ALTAIR

Change 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. For information on default port and port range, refer to Ports Used by Access Web.

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

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.

2. Edit the file `PA_HOME/config/pas/tomcat/conf/server.xml`.

3. Search for the Connector port and update the new port number.

```
<Connector port="5243" protocol="HTTP/1.1" maxThreads="200"
              scheme="https" compression="on" compressionMinSize="2048"
              noCompressionUserAgents="gozilla, traviata"
              relaxedPathChars="[]|{}^&#x5c;&#x60;&quot;&lt;&gt;"
              relaxedQueryChars="[]|{}^&#x5c;&#x60;&quot;&lt;&gt;"
              compressableMimeType="text/html,text/xml,text/javascript,text/
css,application/javascript,text/plain,application/json"
              useSendfile="false" secure="true" SSLEnabled="true"
 clientAuth="false"
              sslEnabledProtocols="TLSv1,TLSv1.1,TLSv1.2"
              keystoreFile="${PBSWORKS_HOME}/config/shared/access.keystore"
 keystorePass="k86BCuq3mLrCqUGZVj3n9DupJ2ePqv" keyAlias="pbsworks"
              ciphers="TLS_RSA_WITH_AES_128_CBC_SHA,SSL_RSA_WITH_RC4_128_SHA"/>
```

4. Edit the file `PA_HOME/config/api_gateway/nginx.conf`.

5. Update the port number of `env PAS_SERVICE_PORT`.

```
env PAS_SERVICE_PORT=5243;
```

6. Start Access Web by entering the command:

```
service pbsworks-pa start
```

Change 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. For information on default port and port range, refer to Ports Used by Access Web.

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

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.

2. Edit the file `PA_HOME/config/jobprofiles/tomcat/conf/server.xml`.

3. Search for the Connector port and update the new port number.

```
<Connector port="5343" maxThreads="200" scheme="https"
      compression="on" compressionMinSize="2048" noCompressionUserAgents="gozilla,
 traviata"
      compressableMimeType="text/html,text/xml,text/javascript,text/
css,application/javascript,text/plain,application/json"
      useSendfile="false" secure="true" SSLEnabled="true" clientAuth="false"
 sslProtocols="TLSv1, TLSv1.1, TLSv1.2"
      keystoreFile="${PBSWORKS_HOME}/config/shared/access.keystore"
 keystorePass="k86BCuq3mLrCqUGZVj3n9DupJ2ePqv" keyAlias="pbsworks"
```

```
                ciphers="TLS_RSA_WITH_AES_128_CBC_SHA,SSL_RSA_WITH_RC4_128_SHA"/>
```

4. Edit the file `PA_HOME/config/api_gateway/nginx.conf`.

5. Update the port number of *server localhost*.

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

6. Start Access Web by entering the command:

```
service pbsworks-pa start
```

Change the AMS Port Number

Change the port that the Access Management 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. For information on default port and port range, refer to Ports Used by Access Web.

The AMS port number has to be updated in the files:

- `server.xml`

- `app.properties`

- `nginx.conf`

- `ServiceRegistry.json`.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.

2. Edit the file `PA_HOME/config/ams/tomcat/conf/server.xml`.

3. Search for the Connector port and update the new port number.

```
<Connector port="5543" protocol="HTTP/1.1" address="127.0.0.1"
              scheme="https" compression="on" compressionMinSize="2048"
              noCompressionUserAgents="gozilla, traviata"
              compressableMimeType="text/html,text/xml,text/javascript,text/
css,application/javascript,text/plain,application/json"
              useSendfile="false" secure="true" SSLEnabled="true"
              clientAuth="false" sslEnabledProtocols="TLSv1,TLSv1.1,TLSv1.2"
              keystoreFile="${PBSWORKS_HOME}/config/shared/access.keystore"
 keystorePass="k86BCuq3mLrCqUGZVj3n9DupJ2ePqv" keyAlias="pbsworks"
              ciphers="TLS_RSA_WITH_AES_128_CBC_SHA,SSL_RSA_WITH_RC4_128_SHA"/>
```

4. Edit the file `PA_HOME/config/shared/app.properties`.

5. Update the port number of *pbsworks.ams.url*.

```
pbsworks.ams.url = https://localhost:5543/AAService
```

6. Edit the file `PA_HOME/config/api_gateway/nginx.conf`.

7. Update the port number of *server localhost*.

```
upstream amsservice {
          server localhost:5543;
      }
```

8. Edit the file `PA_HOME/config/shared/token_handler_config/token_injector/ServiceRegistry.json`.

9. Update the port number in *service*.

```
{"service":[{"name":"ams","host":"localhost","port":"5543",
 "service_name":"AAService","scheme":"https"}]}
```

10. Start Access Web by entering the command:

```
service pbsworks-pa start
```

Change the Mobile Notification Service Port Number

Change the port that the mobile notification 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. For information on default port and port range, refer to Ports Used by Access Web.

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

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.

2. Edit the file `PA_HOME/config/mobile_notification_service/tomcat/conf/server.xml`.

3. Search for the Connector port and update the new port number.

```
<Connector port="5643" protocol="HTTP/1.1" address="127.0.0.1"
             scheme="https" compression="on" compressionMinSize="2048"
             noCompressionUserAgents="gozilla, traviata"
             compressableMimeType="text/html,text/xml,text/javascript,text/
css,application/javascript,text/plain,application/json"
             useSendfile="false" secure="true" SSLEnabled="true"
 clientAuth="false"
             sslEnabledProtocols="TLSv1,TLSv1.1,TLSv1.2"
             keystoreFile="${PBSWORKS_HOME}/config/shared/access.keystore"
 keystorePass="k86BCuq3mLrCqUGZVj3n9DupJ2ePqv" keyAlias="pbsworks"
             ciphers="TLS_RSA_WITH_AES_128_CBC_SHA,SSL_RSA_WITH_RC4_128_SHA"/>
```

4. Edit the file `PA_HOME/config/api_gateway/nginx.conf`.

5. Update the port number of `server localhost`.

```
upstream mobile_notification_service {
             server localhost:5643;
       }
```

6. Start Access Web by entering the command:

```
service pbsworks-pa start
```

Change the Edge Gateway Port Number for Web

Change the port number of Remote Desktops edge proxy for web-based connection when edge proxy is installed on a different machine.

You must stop Access Web before changing the port number. For more information about stopping Access Web, see Access Web Service Commands. For information on default port and port range, refer to Ports Used by Access Web.

The edge gateway port number has to be updated in the files `proxy_loadbalancing.properties` and `nginx.conf`.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.

2. Edit the file `PA_HOME/config/desktop/proxy_loadbalancing.properties`.

△ ALTAIR

3. Update the port number of `port`.

```
port:5943
```

4. Edit the file `PA_HOME/config/desktop_edgeserver/nginx.conf`.

5. Update the port number of `listen`.

```
server {
            listen        5943 ssl;
            server_name   localhost;
```

6. Start Access Web by entering the command:

```
service pbsworks-pa start
```

12.2.2 Change Memory used by the Services

Change the default memory value used by the services in Access Web.

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

To change the default memory value used by the service, follow the steps mentioned in below topics:

Change the Gateway Service Memory Value

Change the memory value of gateway service.

You must stop Access Web before changing the memory value. For more information about stopping Access Web, see Access Web Service Commands. For information on memory usage and its value, refer to Memory Usage by Service.

> 📝 **Note:** You can add the following line in the configuration file if the memory value is not found:
>
> ```
>     api_gateway:
>         max_memory: 512M
> ```

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.

2. Edit the file `PA_HOME/config/shared/deployment-config.yml`.

3. Update the value of *max_memory*:

```
   api_gateway:
            max_memory: 512M
```

4. Start Access Web by entering the command:

```
service pbsworks-pa start
```

Change the Access Web Server Service Memory Value

Change the memory value of Access Web server service.

You must stop Access Web before changing the memory value. For more information about stopping Access Web, see Access Web Service Commands. For information on memory usage and its value, refer to Memory Usage by Service.

> **Note:** You can add the following line in the configuration file if the memory value is not found:
>
> ```
> pa:
>         max_memory: 1024M
> ```

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Edit the file `PA_HOME/config/shared/deployment-config.yml`.
3. Update the value of *max_memory*:

   ```
   pa:
           max_memory: 1024M
   ```
4. Start Access Web by entering the command:

   ```
   service pbsworks-pa start
   ```

Change the PAS Service Memory Value

Change the memory value of PAS service.

You must stop Access Web before changing the memory value. For more information about stopping Access Web, see Access Web Service Commands. For information on memory usage and its value, refer to Memory Usage by Service.

> **Note:** You can add the following line in the configuration file if the memory value is not found:
>
> ```
> pas:
>         max_memory: 1024M
> ```

1. Login to the machine where PAS is installed as root or as a user with sudo permissions.
2. Edit the file `PA_HOME/config/shared/deployment-config.yml`.
3. Update the value of *max_memory*:

   ```
   pas:
             max_memory: 1024M
   ```
4. Start PAS by entering the command:

   ```
   service pbsworks-pa start
   ```

△ **ALTAIR**

Change the Remote Desktops Webserver Service Memory Value

Change the memory value of remote desktop webserver service.

You must stop Access Web before changing the memory value. For more information about stopping Access Web, see Access Web Service Commands. For information on memory usage and its value, refer to Memory Usage by Service.

> 📝 **Note:** You can add the following line in the configuration file if the memory value is not found:
>
> ```
> desktop:
>     max_memory: 512M
> ```

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Edit the file `PA_HOME/config/shared/deployment-config.yml`.
3. Update the value of *max_memory*:
   ```
   desktop:
       max_memory: 512M
   ```
4. Start Access Web by entering the command:
   ```
   service pbsworks-pa start
   ```

Change the Result Manager Services Memory Value

Change the memory value of result manager services.

You must stop Access Web before changing the memory value. For more information about stopping Access Web, see Access Web Service Commands. For information on memory usage and its value, refer to Memory Usage by Service.

> 📝 **Note:** You can add the following line in the configuration file if the memory value is not found:
>
> ```
> resultmanager:
>   max_memory: 512M
> ```

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Edit the file `PA_HOME/config/shared/deployment-config.yml`.
3. Update the value of *max_memory*:
   ```
   resultmanager:
     max_memory: 512M
   ```
4. Start Access Web by entering the command:
   ```
   service pbsworks-pa start
   ```

△ ALTAIR

Change the Job Profile Services Memory Value

Change the memory value of job profile services.

You must stop Access Web before changing the memory value. For more information about stopping Access Web, see Access Web Service Commands. For information on memory usage and its value, refer to Memory Usage by Service.

> 📋 **Note:** You can add the following line in the configuration file if the memory value is not found:
> ```
> jobprofiles:
>   max_memory: 512M
> ```

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Edit the file `PA_HOME/config/shared/deployment-config.yml`.
3. Update the value of *max_memory*:
   ```
   jobprofiles:
     max_memory: 512M
   ```
4. Start Access Web by entering the command:
   ```
   service pbsworks-pa start
   ```

Change the AMS Services Memory Value

Change the memory value of AMS services.

You must stop Access Web before changing the memory value. For more information about stopping Access Web, see Access Web Service Commands. For information on memory usage and its value, refer to Memory Usage by Service.

> 📋 **Note:** You can add the following line in the configuration file if the memory value is not found:
> ```
> ams:
>   max_memory: 512M
> ```

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Edit the file `PA_HOME/config/shared/deployment-config.yml`
3. Update the value of *max_memory*:
   ```
   ams:
     max_memory: 512M
   ```
4. Start Access Web by entering the command:
   ```
   service pbsworks-pa start
   ```

Change the Mobile Notification Services Memory Value

Change the memory value of mobile notification services.

You must stop Access Web before changing the memory value. For more information about stopping Access Web, see Access Web Service Commands. For information on memory usage and its value, refer to Memory Usage by Service.

> 📒 **Note:** You can add the following line in the configuration file if the memory value is not found:
>
> ```
>    mobile_notification_service:
>      max_memory: 512M
> ```

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Edit the file `PA_HOME/config/shared/deployment-config.yml`.
3. Update the value of *max_memory*:
   ```
    mobile_notification_service:
      max_memory: 512M
   ```
4. Start Access Web by entering the command:
   ```
   service pbsworks-pa start
   ```

Change the Executor Services Memory Value

Change the memory value of executor services.

You must stop Access Web before changing the memory value. For more information about stopping Access Web, see Access Web Service Commands. For information on memory usage and its value, refer to Memory Usage by Service.

> 📒 **Note:** You can add the following line in the configuration file if the memory value is not found:
>
> ```
>    executor:
>      max_memory: 512M
> ```

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Edit the file `PA_HOME/config/shared/deployment-config.yml`.
3. Update the value of *max_memory*:
   ```
    executor:
      max_memory: 512M
   ```
4. Start Access Web by entering the command:
   ```
   service pbsworks-pa start
   ```

12.2.3 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)

△ **ALTAIR**

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.

2. Navigate to `PA_HOME/config/pa/`

3. Edit the file `configuration.json`.

4. Change the value of the *doubleClickDelay*.

12.2.4 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 where Access Web is installed as root or as a user with sudo permissions.

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 6.

    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
    ```


12.2.5 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 default text viewer by double-clicking the files. 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.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.

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.

12.2.6 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 after clicking on the 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.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.

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.

12.2.7 Change the File Opening behavior of a Remote Desktops Application

Change the file opening behavior of a remote desktop 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 desktop 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.

If your site does not have a cross-mounted file system that is accessible to both the PBS MoMs and the PBS Server, then you have two options for job submission:

- Update the Application Definition Input file and change the default value of `Run from job directory` to *true*.
- 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 remote desktop only.

When submitting an interactive job via Access Web, the Run from job directory field must be unchecked. You will need to do this every time you submit a job.

12.2.8 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 **Preferences**.
   Access Web Preferences is displayed.

△ **ALTAIR**
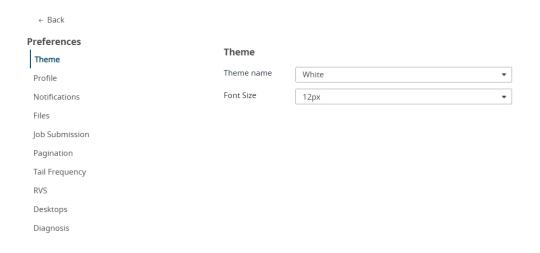
Figure 22: Preferences

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

4. Click **Job Submission** in the left menu.
   The Job Submission panel is displayed.
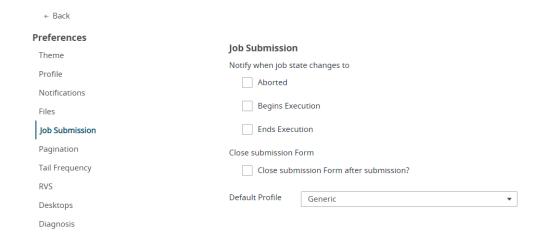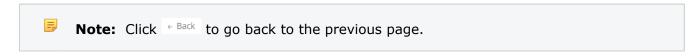
Figure 23: Job Submission

> 📝 **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.

12.2.9 Disable to View all Jobs

Restrict users to only be able to view their own jobs.

By default, users can view all jobs.

Managers and the users assigned the role of Managers will be able to view all the user's jobs.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:

    ```
    source /etc/pbsworks-pa.conf
    ```
3. Navigate to `PA_HOME/config/pa/`

    ```
    cd $PA_HOME/config/pa
    ```
4. Edit the `configuration.json` file.
5. Change the value of *restrictOthersJobs* to `true`.

    ```
        "restrictOthersJobs": false
    ```

    > 📝 **Note:** By default *restrictOthersJobs* is set to `false`.

6. Navigate to `PA_HOME/config/storage/app.properties`.

    ```
    cd $PA_HOME/config/storage
    ```
7. Edit the `app.properties` file.
8. Change the value of *pbsworks.storage.jobFilters.othersJobsRestricted* to `true`.

    ```
    # Restricting others jobs
    pbsworks.storage.jobFilters.othersJobsRestricted=false
    ```

    > 📝 **Note:** By default the value *pbsworks.storage.jobFilters.othersJobsRestricted* is set to `false`.

12.2.10 Add a Generic Action for a PAS server

Perform a generic action on a job.

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.

Define a Generic Action

Define a generic action JSON file.

Here is an example of a generic action for a cluster denoted by "hpccluster":

```
{
 "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"]
    }
  }
```

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

Name
> An internal name of the generic action.

DisplayName
> Name of the generic action that will be displayed to the user.

Description
> Description of the generic action.

ScriptLocation
> Location of the execution script.

Arguments
> Defines arguments (input fields) that a user will enter prior to executing the generic action. The following elements define an argument:

> *type*
>> Defines the type of the argument

DisplayName
> 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.

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.

Example of Generic Action for a Single Cluster

Below is an example of a generic action for the server "hpccluster", stored in a JSON file called `genericactions.json`.

```
{
 "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": [ ]
   }
  }
 ]
}
```

Example of Generic Action for a Multiple Cluster

Below is an example of a generic action for the server "hpccluster-1" and "hpccluster-2", stored in a JSON file called `genericactions.json`.

```
{
 "hpccluster-1": [{
   "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": [ ]
     }
    }
   ]
 },
 {
  "hpccluster-2": [{
     "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"]
    }
  }
 ]
}
```

Execution Script

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.

```python
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()
```

12.2.11 Map a File Extensions to an Icon

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

Access Web displays customized file types and icons when viewing remote files.

	▲Name	Created On	Owner	File Size	Type	⚙
☐	📁 Images	9/3/2020, 2:34:15 AM	pbsworks	--	Folder	
☐	📁 Inputs	9/3/2020, 4:52:47 AM	pbsworks	--	Folder	
☐	📁 Optistruct_Files	8/25/2020, 4:23:41 AM	pbsworks	--	Folder	
☐	📁 Optistruct_Master_File	9/3/2020, 10:52:27 PM	pbsworks	--	Folder	
☐	📁 Radioss_Files	8/25/2020, 4:34:27 AM	pbsworks	--	Folder	
☐	📁 ShellScript_Files	8/25/2020, 4:17:01 AM	pbsworks	--	Folder	
☐	📁 sleep_1598292540543	8/25/2020, 5:17:44 AM	pbsworks	--	Folder	
☐	AltairAccessWebReleaseNotes2019.4.pdf	8/25/2020, 4:18:01 AM	pbsworks	196.93 KB	File	
☐	Inputs.zip	9/3/2020, 4:35:11 AM	pbsworks	33.92 MB	File	
☐	Master_file.fem	9/3/2020, 4:37:46 AM	pbsworks	29 Bytes	File	
☐	Master_file_archive_1599068750459.zip	9/3/2020, 4:49:34 AM	pbsworks	1.01 MB	File	
☐	mbdsystemlvlopt.fem	9/3/2020, 4:37:48 AM	pbsworks	5.22 MB	File	
☐	model_1.png	8/25/2020, 4:18:01 AM	pbsworks	137.28 KB	File	
☐	model_2.png	8/25/2020, 4:18:01 AM	pbsworks	92.97 KB	File	
☐	rod.h3d	8/25/2020, 4:18:01 AM	pbsworks	6.21 MB	File	
☐	sleep.py	8/25/2020, 5:12:36 AM	pbsworks	28 Bytes	File	

Figure 24: File Icon Mapping

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/`

Below 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.

△ ALTAIR

displayText
>   Access Web will display this as the files type.

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

12.2.12 Set Maximum Page Size for Files

Set the maximum file size (in bytes) for displaying a file in a single page.

The default is 5000 bytes.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Navigate to `PA_HOME/config/pa/`
3. Open the `configuration.json` file and change the value of the *filechunksize*.
```
"filechunksize": 5000
```


12.2.13 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
```

12.2.14 Enable Mobile Notification Service

Enable mobile notification service in Access Web so that the job status notification is displayed in Access Mobile.

You must stop Access Web before enabling mobile notification service. For more information about stopping Access Web, see Access Web Service Commands.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Navigate to `PA_HOME/config/shared/`
3. Open the `deployment.ini` file.
4. Add the following line under `pa_deploy_options` section:

   ```
   option=("mobile_notification_service")
   ```
5. Start Access Web using the following command:

   ```
   service pbsworks-pa start
   ```

12.2.15 Customize Login Page

Customize Access Web login page with custom branding information.

Customize the Access Web login page with your company's logo, as well as a custom message that is displayed below the login prompt.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Add the branding logo at `PA_HOME/config/pa/branding/`
3. Edit `PA_HOME/config/pa/configuration.json`
4. Update the path of the branding logo in *appLogo*.

   For example:

   ```
   "appLogo": "/config/pa/branding/company-logo.png"
   ```
5. Update the description of the branding information in *appDescription*

   For example:

   ```
   "appDescription": "ALTAIR PBSWORKS HPC PORTAL"
   ```
6. Refresh your Access Web browser to reflect the login screen with branding logo and description. For example, Access Web with branding logo and description will be displayed as follows:

Figure 25: Access Web Login Page with Branding Logo and Description

12.2.16 Switch to Use the Locally Installed Webhelp

Configure Access Web to point to the locally installed webhelp when there is no internet connectivity and the internet version of the webhelp is not reachable.

Download or obtain the Access Web help using your Altair PBS Works Support (pbssupport@altair.com).

Access Web will now fetch and display the latest version of the User Guide (WebHelp) from Altair Connect. As a fall back, the application can also be configured to use a local copy of the help.

Perform these steps to configure and view the local copy of the help.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Copy the Access Web help zip file to `PA_HOME/data/pa/`
3. Unzip the Access Web help zip file.
4. Edit the `PA_HOME/config/pa/configuration.json` file.
5. Update the *helpurl* path to the local webhelp folder path.

   > 📝 **Note:** By default, the `helpurl` path mentioned is the Altair Connect path.

6. Change the Altair Connect default path to the local webhelp folder path as follows:
   ```
   "helpurl": "/data/pa/webhelp/index.htm"
   ```
7. Refresh the Access Web browser.

12.2.17 Shared File System Support

Support for sites that have deployed a shared file system on their HPC cluster.

Organizations may deploy a shared file system on their HPC cluster such that the file system is shared between the PBS Server and the PBS execution nodes, eliminating the need to stage in and stage out job files. Applications that run on the HPC cluster and need access to the shared file system will require a change to their corresponding application definition to set the following environment variables in the submittime script (`presubmit.py`) and a change to the runtime script (`start.py`):

ACCESS_INPUT_FILES
> Environment variable used by Access to establish the job's input files.

ACCESS_OUTPUT_FILES
> Environment variable used by Access to establish the job output directory.

ACCESS_RUNNING_FILES
> Environment variable used by Access to establish the job running directory.

For example, a site may have `/shared` mounted on a share file system that is shared between the PBS Server and the PBS execution nodes. The user tsmith moves job files to the directory `/stage/tsmith/opti_test1` and submits an Optistruct job. For Access to support the use of the shared file system by the job, the application definition associated with the Optistruct solver must be updated to set these three environment variables in the `presubmit.py` script.

To avoid copying the `runtime` and `pbsspawn` directories in share file system, update the parameters in `PA_HOME/config/pas/conf/server.conf` mentioned below:

IS_APPLICATION_STAGEIN_REQUIRED
> Specifies if stage in of application runtime and pbsspawn directories is required. By default, this is set to true.
> If it is disabled, then the APPLICATION_DEFINITION_EXECUTION_NODE_PATH specified is used for application definition related files.

APPLICATION_DEFINITION_EXECUTION_NODE_PATH
> Location of application definitions on the execution node. User must deploy application definitions under the application folder. PAS will look for the application definitions specified in this path if IS_APPLICATION_STAGEIN_REQUIRED is disabled. This location should be available on all the execution nodes.

STAGEOUT_ENABLED
> By default, the value is set to true indicating that the location for stage out files is taken care by Access Web.
> Set the value to false, if you want to maintain a different stage out location (example, Shared File Systems).

For more information about the specific changes that need to be made to the application definition see the recipe *How to Support a Shared File System* in *Diving Into Application Definitions*.

This feature is completely backward compatible. If your site is not using a shared file system, then your current application definitions do not require any changes.

12.2.18 Change the Maximum File Upload Size

Change the default file upload size based on site's requirements.

Before you begin

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

The default maximum file upload size is unlimited. Administrator can limit the file upload size by setting the value in the maximum file size. This will restrict the user not to upload a file size bigger than the value set in the maximum file size.

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

   > 📝 **Note:** By default, the file size is set to unlimited (`client_max_body_size 0;`).

   For example, if you want to restrict the file size to 4GB, then update the value as follows:
   ```
   client_max_body_size 4096m;
   ```
5. Start Access Web using the following command:
   ```
   service pbsworks-pa start
   ```

12.2.19 Configure to Handle Large File Uploads

Configure to handle large file uploads by enabling the modern file upload parameter.

Before you begin:

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

By default, the conventional file upload feature is enabled which will restrict the file upload to 4GB.

By enabling the modern file upload parameter there will be no upload restriction based on file size and the files will be uploaded in chunks.

You need to have administrative privileges to makes the changes.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Navigate to `PA_HOME/config/pa/`
3. Open the `configuration.json` file.
4. Change the *`enableModernFileUpload`* value to *true* to enable modern file upload.
   ```
   "enableModernFileUpload": false
   ```

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

5. Start Access Web using the following command:

△ **ALTAIR**

```
service pbsworks-pa start
```

12.2.20 Support for PBS Peering Based Setup

Access Web supports PBS Peering Scheduling by displaying the job that is moved from one complex to the other.

Peer scheduling allows separate PBS complexes to automatically run jobs from each other's queues. When Complex A pulls a job from Complex B, Complex A is the "pulling" complex and Complex B is the "furnishing" complex. These jobs can be monitored through Access Web through either the pulling complex or the furnishing complex using the Services filter that is available on the Job Monitoring page.

Ensure that the following prerequisites are met to support PBS Peering Scheduling:

- PBS should be configured for Peer Scheduling. For more information see *Peer Scheduling* in the *PBS Professional Administrator's Guide*.
- PAS must be installed on both the pulling and the furnishing HPC complex.
- Modern Communication Module should be setup on all execution nodes.
- Add a server cluster to Access Web for both the "pulling" complex and the "furnishing" complex.

In the following example, a ShellScript job is submitted to *troycompute* server. The *troycompute* server is the furnishing complex. The ShellScript job is pulled by the *syspc02compute* server and this server is the pulling complex. The job can be monitored through either the pulling or the furnishing server.

Using the **Job Monitoring** page, view those jobs that are submitted to the furnishing server (filter the jobs by enabling the *troycompute* Service) . The ShellScript job is listed.
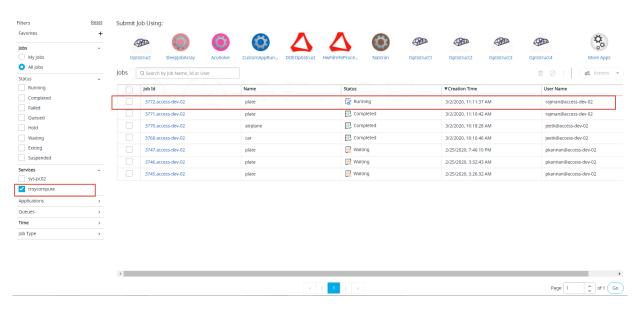
Figure 26: Jobs in troycompute Server

Using the **Job Monitoring** page, view those jobs that are submitted to the pulling server (filter the jobs by enabling the *syspc02compute* Service) . The ShellScript job is listed.
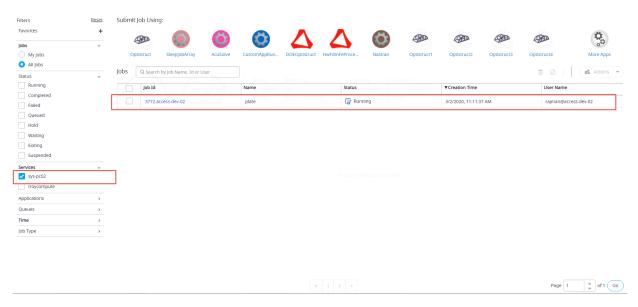
Figure 27: Jobs in syspc02compute Server

12.2.21 Enable and Disable the Modern Communication Module

Disable or enable the Modern Communication Module based on the needs of your site.

Disable the Modern Communication Module

Disable the Modern Communication Module and revert to PBS technology for performing file operations for running jobs.

> 📝 **Note:** Disabling the Modern Communication Module may degrade the performance of file operations on running jobs.

These steps assume that Access Web and PAS are installed on the same machine.

1. Login to Access Web as a user who has been assigned a Manager role.
2. Delete all server clusters.
3. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
4. Stop Access Web by entering the following command:

   `service pbsworks-pa stop`
5. Edit the `PA_HOME/config/pas/conf/server.conf` file.
6. Change the value of *MODERN_COMMUNICATION_ENABLED* to *false*.

   `MODERN_COMMUNICATION_ENABLED=false`
7. Navigate to `PA_HOME/config/api_gateway`.

△ ALTAIR

8. Copy `template_pyspawn.conf` to `template.conf`.

9. Start Access Web by entering the following command:

   ```
   service pbsworks-pa start
   ```

10. Login to Access Web as a user who has been assigned a Manager role.

11. Re-register all server clusters.

Enable the Modern Communication Module

Enable the Modern Communication Module, after it has been disabled, to improve file operation performance for running jobs.

The Modern Communication Module must be copied to a location accessible to the PBS execution hosts before it can be enabled. If this was not done when Access Web was installed or upgraded then refer to Modern Communication Module.

These steps assume that Access Web and PAS are installed on the same machine.

1. Login to Access Web as a user who has been assigned a Manager role.

2. Delete all Server Clusters.

3. Login to the machine where Access Web is installed as root or as a user with sudo permissions.

4. Stop Access Web by entering the following command:

   ```
   service pbsworks-pa stop
   ```

5. Edit the `PA_HOME/config/pas/conf/server.conf` file.

6. Change the value of *MODERN_COMMUNICATION_ENABLED* to *true*.

   ```
   MODERN_COMMUNICATION_ENABLED=true
   ```

7. Navigate to `PA_HOME/config/api_gateway`.

8. Copy `template_joboperation.conf` to `template.conf`.

9. Start Access Web by entering the following command:

   ```
   service pbsworks-pa start
   ```

10. Login to Access Web as a user who has been assigned a Manager role.

11. Re-register all server clusters.

12.2.22 Change the Locale

Change the locale to view user interface labels, buttons, tooltips, etc. in a different language.

Currently, the following languages are supported:

- English (*en_US*)
- French (*fr_FR*)
- Chinese (*zh_CN*)
- Japanese (*ja_JP*)

The default locale set in `PA_HOME/config/pa/configuration.json` file is English.

> 📝 **Note:** You can add your language bundle at `PA_HOME/data/pa/locale/` and update the `PA_HOME/config/pa/configuration.json` file.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:

   ```
   source /etc/pbsworks-pa.conf
   ```
3. Edit the file `PA_HOME/config/pa/configuration.json`.

   ```
   vi $PA_HOME/config/pa/configuration.json
   ```

   ```
   "locale": {
           "default": "en_US",
           "availableLocales": {
             "en_US": "/data/pa/locale/en_US.json",
             "fr_FR": "/data/pa/locale/fr_FR.json",
             "zh_CN": "/data/pa/locale/zh_CN.json"
             "ja_JP": "/data/pa/locale/ja_JP.json"
       }
     },
   ```
4. Choose one of the following options:
   - To set the locale to English, set the value of the parameter default to "en_US".
   - To set the locale to French, set the value of the parameter default to "fr_FR".
   - To set the locale to Chinese, set the value of the parameter default to "zh_CN".
   - To set the locale to Japanese, set the value of the parameter default to "ja_JP".
5. Refresh the browser to reflect the changes.

12.2.23 Add Site Specific Web Pages

Add site specific pages to share with all the users in Access Web by providing a URL link.

Some sites want to provide access to web sites via Access Web. These web sites may provide site-specific information that user's of Access Web require. For example, an internal web page describing the solver's available at each HPC complex.

Links to these site-specific web pages can be configured. They will appear as top-level options along with Files, Jobs and Desktops. The site-specific web pages will be opened in a new tab. A More drop-down option will be displayed if there are too many pages.

> 📝 **Note:** Provide the permission to access the site-specific pages for the user from the Access Management.

Figure 28: Pages Tab Context Menu

To configure the web pages, edit the `PA_HOME/config/pa/configuration.json` file and add JSON to describe the following parameters:

id

> The ID of an internal web page.

name

> The name of the web page that is displayed in the context menu accessible from the Pages tab.

url

> The URL of the web page. The URL should be in the format *https://<web page url link>*.

For example, to make the Altair and solidThinking web pages accessible through Access Web, provide the following JSON:

```
"pages": [
     {
       "id" : "altair",
       "name": "Altair",
       "url" : "https://www.altair.com/"
     },
     {
       "id" : "solidthinking",
       "name": "SolidThinking",
       "url" : "https://solidthinking.com/"
     }
     {
       "id" : "marketplace",
       "name": "AltairOneMarketplace",
       "url" : "https://marketplace.altairone.com/Marketplace"
     }
 ]
```

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:
3. Edit `PA_HOME/config/pa/configuration.json`

    ```
    vi $PA_HOME/config/pa/configuration.json
    ```
4. Add the JSON required to add the web page links between the below brackets:

    ```
    "pages": [
         ]
    ```
5. Refresh the browser to reflect the changes.

12.2.24 Configure Access Web Idle Session Timeout

Configure Access Web session timeout value to automatically logout a user when the session is idle.

The user is automatically logged out of Access Web session when the session is idle based on a configurable session timeout and token timeout value. The default session timeout value is 5 hours (300 minutes) and the token timeout value is 3 hours (180 minutes). Change these values per your site's requirements.

> 📝 **Note:** The session timeout value should always be greater than the token timeout value.

The Session Timeout specifies the number of minutes that an interactive session can remain idle before the user gets logged out of the application automatically.

The Token Timeout specifies the number of minutes in which the access token is refreshed. When the session is idle then the token is not refreshed and the user is automatically logged out.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.

2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:

   ```
   source /etc/pbsworks-pa.conf
   ```

3. Edit the file `PA_HOME/config/ams/AA_Service.properties`.

   ```
   vi $PA_HOME/config/ams/AA_Service.properties
   ```

4. Update the *java.sessionTimeout* value in minutes.

   ```
   java.sessionTimeout=300
   ```

   > 📝 **Note:** The minimum session timeout value that can be set is 30 minutes.

5. Edit the file `PA_HOME/config/api_gateway/nginx.conf`.

   ```
   vi $PA_HOME/config/api_gateway/nginx.conf
   ```

6. Update the *ACCESS_TOKEN_TIMEOUT* value in minutes.

   ```
   env ACCESS_TOKEN_TIMEOUT=180;
   ```

   > 📝 **Note:** The token timeout value should always be less than session timeout value. The minimum token timeout value that can be set is 20 minutes.

7. Edit the file `PA_EXEC/shared/scripts/start.sh`.

   ```
   vi $PA_EXEC/shared/scripts/start.sh
   ```

8. Update the *ACCESS_TOKEN_TIMEOUT* in minutes with the same value that is set in `PA_HOME/config/api_gateway/nginx.conf`

   ```
   env ACCESS_TOKEN_TIMEOUT=180;
   ```

9. Restart Access Web using the following command:

   ```
   service pbsworks-pa restart
   ```

12.2.25 Import Active Directory Users and Roles into Access Web

Export Active Directory roles, users, and user/role mappings to a CSV file and and use the file's contents to import the information into Access Web.

User a PowerShell script to export a site's Active Directory (AD) user and role data to a CSV file. The CSV file's contents are imported into Access Web using a Python script automatically, eliminating the need for an administrator to manually enter users and roles.

Export Active Directory Users and Roles to a CSV File

Export Active Directory users, roles and their mappings to a CSV file.

The export utility is a PowerShell script.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:

    ```
    source /etc/pbsworks-pa.conf
    ```
3. Navigate to the directory `PA_EXEC/ams/scripts/`

    ```
    cd $PA_EXEC/ams/scripts/
    ```
4. Copy the export script `ad_user_export.ps1` to your Active Directory machine.
5. Run the export script on PowerShell:

    ```
    ad_user_export.ps1
    ```

    This will generate `AD_Users_Roles_Data.csv` file in the current working directory.
6. Transfer the generated `AD_Users_Roles_Data.csv` file to Access Web machine.

The next step is to import the CSV file (`AD_Users_Roles_Data.csv`) into Access Web, refer to Import Active Directory Users and Roles into Access Web.

Import Active Directory Users and Roles into Access Web

Import a CSV file containing Active Directory users, roles and user/role mappings into Access Web.

Before importing the AD users and roles, they must be exported to a CSV file. Refer to Export Active Directory Users and Roles to a CSV File.

The import utility is a Python script which is compatible with Python 2.7.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:

    ```
    source /etc/pbsworks-pa.conf
    ```
3. Navigate to the directory `PA_EXEC/ams/scripts/`

    ```
    cd $PA_EXEC/ams/scripts/
    ```
4. Execute the following command to import the CSV file:

    ```
    python user_role_import.py --server_url=<accessurl> --csv_file=<csvfilelocation>
    ```

    > 📝 **Note:** Execute the command `python user_role_import.py -h` for help.

    An example of importing the active directory from Access Web:

    ```
    python user_role_import.py --server_url=https://localhost:4443 /
    --csv_file=AD_Users_Roles_Data.csv
    ```

△ **ALTAIR**

12.2.26 Enable Product Promotions from Access Web

Promote related products such as Access Desktop and Access Mobile from the About Access Web menu.

By default, the promotion screens are disabled in Access Web. Administrators can enable the promotion screens, so that the promotion screen will be displayed in the user interface.

For example, if both Access Mobile and Access Desktop are enabled, then the promotion of these products is displayed under About Access Web.

Hello, pbsworks

- App Composer
- Reload Apps
- Preferences
- Audit Logs
- Access Management
- Change Password
- Feedback
- Help
- About
  - Get Access Mobile
  - Get Access Desktop
- Logout
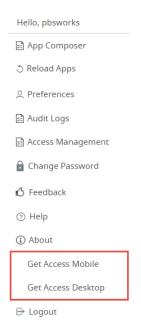
Figure 29: About Access Web Menu

If you click on **Get Access Mobile**, then the following promotion screen is displayed:

Figure 30: Access Mobile Promotion Screen

If you click on **Get Access Desktop**, then the following promotion screen is displayed:

Figure 31: Access Desktop Promotion Screen

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:
   ```
   source /etc/pbsworks-pa.conf
   ```
3. Edit the file `PA_HOME/config/pa/configurations.json`
   ```
   vi $PA_HOME/config/pa/configurations.json
   ```
   ```
   "promotions": [
   ```

```
      {
        "name": "Get Access Mobile",
        "config": "/config/pa/promotions/mobile/accessmobile.json",
        "enable": false
      },
      {
        "name": "Get Access Desktop",
        "config": "/config/pa/promotions/desktop/accessdesktop.json",
        "enable": false
      }
   ]
```

4. Choose the following:

    - To enable the Access Mobile promotion screen, set the value of *enable* to true.
    - To enable the Access Desktop promotion screen, set the value of *enable* to true.

    > 📝 **Note:** You can also set the value of *enable* to true for both Access Mobile and Access Desktop.

5. Refresh the browser to reflect the changes.

To allow Access Web users to install Access Desktop from its promotion screen requires additional configuration. Refer to Configure a Location to Distribute Access Desktop

12.2.27 Configure a Location to Distribute Access Desktop

Allow a user to download a preconfigured Access Desktop zip file package from the Access Web user interface.

Before you begin, the preconfigured Access Desktop zip file package has to be generated and placed in Access Web. Refer to *Install Access Desktop for Deployment Option 4* topic in *Access Desktop 2022.1.0 Administrator's Guide* for information about creating Access Desktop zip file package.

The Access Desktop zip file package contains Access Desktop executable file and a configuration zip file. The configuration zip file contains preconfigured clusters registered through HTTPS and SSH, application definitions and job profiles. Once you install Access Desktop, the clusters are registered automatically.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Copy the preconfigured Access Desktop zip file package using `scp`, `WinSCP` or some other kind of copy mechanism to the location `$PA_HOME/config/pa/promotions/desktop`.
3. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:

    `source /etc/pbsworks-pa.conf`
4. Edit the file `PA_HOME/config/pa/promotions/desktop/accessdesktop.json`

    `vi $PA_HOME/config/pa/promotions/desktop/accessdesktop.json`
5. Update the value of `downloadUrl` to point to the executable or binary that was copied in step 2. For example, the default zip file package name provided is as follows:

    `"downloadUrl": "/pbsworks/config/pa/promotions/desktop/AltairAccessDesktop.zip",`
6. Refresh the browser to reflect the changes.

△ **ALTAIR**

12.2.28 Disable File Listing of Users

Restrict the listing of user files from other users.

By default, the list of files can be viewed by other users.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:

   `source /etc/pbsworks-pa.conf`
3. Open the file `PA_HOME/config/pas/conf/server.conf`.
4. Update the *USER_SPACE_ENABLED* to *true.*

   `USER_SPACE_ENABLED=true`

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

5. Restart Access Web using the following command:

   `service pbsworks-pa restart`

12.2.29 Configure Deployment Profile

Configure deployment profile based on the site requirements.

Access Web provides 300, 500, and 1000 concurrent users deployment profile for batch jobs and provides support for 2000 concurrent remote desktop on access deployment. The profile can be updated based on your site requirement.

The default the configuration deployment profile (`deployment-config.yml`) is set to 300 concurrent users.

The following configuration deployment profiles is located at: `PA_HOME/config/shared/`:

- `deployment-config.yml.300` - deployment profile for 300 concurrent user
- `deployment-config.yml.500` - deployment profile for 500 concurrent user
- `deployment-config.yml.1k` - deployment profile for 1000 concurrent user
- `deployment-config.yml.2k` - deployment profile for 2000 concurrent user for Remote Desktops

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:

   `source /etc/pbsworks-pa.conf`
3. Navigate to `PA_HOME/config/shared/`.

   `cd $PA_HOME/config/shared/`

   > 📝 **Note:** By default, the configuration deployment profile (`deployment-config.yml`) is set to 300 concurrent users.

4. Take a backup of the default deployment profile `deployment-config.yml` file.

△ **ALTAIR**

For example:

```
cp deployment-config.yml deployment-config_default.yml
```

5. Copy the required deployment profile configuration to the default deployment profile name. For example, if you want to change it to 500 concurrent users, then copy `deployment-config.yml.500` to `deployment-config.yml`.

```
cp deployment-config.yml.500 deployment-config.yml
```

6. Restart Access Web using the following command:

```
service pbsworks-pa restart
```

12.2.30 Add Site Specific Name

Add site specific Access Web instance name so that it displays browser title.

By default, the browser page title displays the Tab title and the product name (Altair Access) in the format **<Tab Title>-Altair Access**

For example:

- if you are in Jobs tab the browser title will be displayed as follows:

- if you are in Desktops tab the browser title will be displayed as follows:

Access Web provides an option to configure the Site Name, so that it also gets displayed in the browser title page in the format: **<Tab Title>-<SiteName>-Altair Access**

Update the site name in the branding section of the configuration file (`configuration.json`) located at `PA_HOME/config/pa`.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:
3. Edit `PA_HOME/config/pa/configuration.json`

   ```
   vi $PA_HOME/config/pa/configuration.json
   ```

4. Add the site name in the branding section:

   ```
   "branding": {
       "appLogo": "",
       "appDescription": "",
       "siteName": ""
     }
   ```

5. Refresh the browser to reflect the changes.
   For example, if you have updated the *siteName* as *Altair 2022.1.0*, then the browser title page will be displayed as follows:

⚠ Jobs-Altair 2022.1.0-Altair Access   ✕

12.2.31 Change Audit Log Data Purging Period

Change audit log data purging period to clear the data after a specified period.

By default, the audit log data is stored for a period of 365 days.

For example, the following values provided indicates that:

- -1 - there will be no purging at all
- 0 - logs older than today only will be purged
- 1 - logs older than yesterday will be purged

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:
3. Edit `PA_HOME/config/executor/app.properties`

   ```
   vi $PA_HOME/config/executor/app.properties
   ```
4. Change the value of *pbsworks.executor.auditlog.purgingdays*.

   ```
   #Audit Log purging period in days
   pbsworks.executor.auditlog.purgingdays = 365
   ```

   > 📝 **Note:** By default, audit log purging period is set to 365 days.

5. Restart Access Web using the following command:

   ```
   service pbsworks-pa restart
   ```

12.2.32 Configure NATS Message Broker Connection

Configure NATS message broker connection details in Access Web.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:

   ```
   source /etc/pbsworks-pa.conf
   ```
3. Edit the file `PA_HOME/config/shared/app.properties`.

   ```
   vi $PA_HOME/config/shared/app.properties
   ```
4. Update `nats.broker.url` with the connection URL.

   ```
   #NATS server connection details
   nats.broker.url=nats://vm8cent74-rvsdev:4743
   ```
5. Restart Access Web using the following command:

   ```
   service pbsworks-pa restart
   ```

△ **ALTAIR**

12.2.33 Disable Access Web Page Title in the Browser Title

Disable the display of Access Web page title in the browser title.

By default, the browser page title displays the Tab title and the product name (Altair Access) in the format **<Tab Title>-Altair Access**.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:

   `source /etc/pbsworks-pa.conf`
3. Edit the file `PA_HOME/config/pa/configurations.json`

   `vi $PA_HOME/config/pa/configurations.json`
4. Change the value of *enablePageTitle* to *false* to hide the display of Access Web page title in browser title.

   `"enablePageTitle": true,`

   By default, the value of *enablePageTitle* is set to *true*.
5. Refresh the browser to reflect the changes.

12.2.34 Hide App Composer Option in Access Web Portal

Hide the App Composer feature option for the user in Access Web portal.

By default, the App Composer feature option is displayed for all the users logged in to Access Web.

The App Composer feature option is displayed when the user clicks ⚲ > **App Composer** in Access Web portal.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:

   `source /etc/pbsworks-pa.conf`
3. Edit the file `PA_HOME/config/pa/configurations.json`

   `vi $PA_HOME/config/pa/configurations.json`
4. Change the value of *enableAppComposer* to *false* to hide the display of App Composer feature in Access Web portal.

   `"enableAppComposer": true,`

   By default, the value of *enableAppComposer* is set to *true*.
5. Refresh the browser to reflect the changes.

△ALTAIR

12.2.35 Hide Reload Apps Option in Access Web Portal

Hide the Reload Apps option for the user in Access Web portal.

By default, the Reload Apps option is displayed for all the users logged in to Access Web.

The Reload Apps option is displayed when the user clicks ⌁ > **Reload Apps** in Access Web portal.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:

   ```
   source /etc/pbsworks-pa.conf
   ```
3. Edit the file `PA_HOME/config/pa/configurations.json`

   ```
   vi $PA_HOME/config/pa/configurations.json
   ```
4. Change the value of *enableReloadApps* to *false* to hide the display of Reload Apps option in Access Web portal.

   ```
   "enableReloadApps": true,
   ```

   By default, the value of *enableReloadApps* is set to *true*.
5. Refresh the browser to reflect the changes.

12.2.36 Hide Feedback Option in Access Web Portal

Hide the feedback option for the user in Access Web portal.

By default, the feedback option is displayed for all the users logged in to Access Web.

The feedback form is displayed when the user clicks ⌁ > **Feedback** in Access Web portal.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:

   ```
   source /etc/pbsworks-pa.conf
   ```
3. Edit the file `PA_HOME/config/pa/configurations.json`

   ```
   vi $PA_HOME/config/pa/configurations.json
   ```

   ```
   "feedback" : {
         "name": "Feedback",
         "url": "https://forms.office.com/Pages/ResponsePage.aspx?id
   =V1uuK7gO-0i6R5kCWdqJ0uT4gsljoi5IllH4MauaJGZUQ0NPRjk4OExFOUxBUkhVNUs1VU9LR0hEMy4u",
         "enable": true
       }
     },
   ```
4. Change the value of *enable* to *false* to hide the display of Feedback option in Access Web portal.

   By default, the value of *enable* is set to *true*.
5. Refresh the browser to reflect the changes.

△ **ALTAIR**

12.2.37 Add an External File Viewer

Add an external file viewer in Access Web.

The external file viewer folder must be created at `PA_HOME/data/pa/plugins/` and the following files of the external file viewer should be copied in that folder:

- `viewer.js` - This contains logic to populate open with context menu items for files and open with modal. The supported file extensions is also defined in this file.
- `render.js` - contains the logic to render the viewer in User Interface.

The details of the external file viewer must be added in `PA_HOME/config/pa/viewer.json` and can be enabled or disabled.

The following code has to be updated for the external file viewer and added in `PA_HOME/config/pa/viewer.json`:

```
{
    "id": "htmlviewer",
    "name": "HTML Viewer",
    "description": "Open any HTML file.",
    "enable": true,
    "native": false,
    "mainJsFile": "/data/pa/plugins/htmlviewer/viewer.js",
    "renderJsFile": "/data/pa/plugins/htmlviewer/render.js"
}
```

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:
3. Navigate to `PA_HOME/data/pa/plugins/`.
4. Create a folder for the external file viewer and copy the following files of the external file viewer:

    - `viewer.js`
    - `render.js`

    The folders that are currently available in this path are:
    ```
    3dmodalviewer
    htmlviewer
    imageviewer
    rvs
    textviewer
    ```
5. Edit `PA_HOME/config/pa/viewer.json`
    ```
    vi $PA_HOME/config/pa/viewer.json
    ```
6. Add the following code for the external file viewer
    ```
    {
        "id": "<external file viewer>",
        "name": "<name of the external file viewer>",
        "description": "<description of the external file viewer>",
        "enable": true,
        "native": false,
        "mainJsFile": "/data/pa/plugins/<external file viewer folder>/viewer.js",
        "renderJsFile": "/data/pa/plugins/<external file viewer folder>/render.js"
    }
    ```

> 📝 **Note:** Change the value of `enable` to *false* to disable the external file viewer.

The external file viewer will work in the same way as native viewers.

12.2.38 Configure Job Scheduler Intervals

Configure job scheduler intervals based on your site deployment.

Altair Access polls the jobs from PBS Professional at constant interval and it provides an option to change interval frequency based on deployments.

The default values set are tuned to make sure new jobs or updates are reflected promptly for the end users. The end users must be educated on what to expect when the default values are changed in the configuration file (`app.properties`).

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Stop Access Web using the following command:
   ```
   service pbsworks-pa stop
   ```
3. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:
   ```
   source /etc/pbsworks-pa.conf
   ```
4. Edit the file `PA_HOME/config/executor/app.properties`
   ```
   vi $PA_HOME/config/executor/app.properties
   ```
5. Change the priority scheduler used for bringing newly submitted jobs on priority.
   ```
   pbsworks.executor.scheduler.priorityjob.trigger = 0/2 * * * * ?
   ```
   By default, this runs in every 2 seconds.
6. Change jobs polling scheduler for polling all modified jobs since last fetch.
   ```
   pbsworks.executor.scheduler.normaljob.trigger = 0/5 * * * * ?
   ```
   By default, this runs in every 5 seconds.
7. Change subjobs polling scheduler for polling all modified subjobs since last fetch.
   ```
   pbsworks.executor.scheduler.subjob.trigger = 0/5 * * * * ?
   ```
   By default, this runs in every 5 seconds.
8. Start Access Web using the following command:
   ```
   service pbsworks-pa start
   ```

12.3 Configure PBS Application Services

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

12.3.1 PAS Configuration Files

Configuration files that are installed with PAS.

Application Definitions

Application definitions are a feature of PAS and are used to submit jobs using the power and flexibility of PAS. Application definitions are stored in the PAS application directory. The default location of this directory is:

`PA_HOME/data/pas/targets/localhost/repository/applications/`

Site Configuration File

A default site configuration file, `site-config.xml`, is installed in the PAS configuration directory. The location of this file for a typical installation of PAS is: `PA_HOME/data/pas/targets/localhost/repository/`

This file is a central repository for site specific information that can be used across all application definitions. The default site configuration file installed with PAS is basically a template. Initially, you may use the default site configuration file as is, but as you become accustomed to working with application definitions, you may choose to modify the site configuration file per your site's needs. A tutorial and several recipes are available within *Diving Into Application Definitions* showing several ways that the site configuration file can be used with an application definition. For an overview of the site configuration file see Sitewide Settings on page 57.

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. The location of this file for a typical installation of PAS is: `PA_HOME/config/pas/conf`

12.3.2 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. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Edit the `PA_EXEC/pas/scripts/File/FileCompress.py` file.
3. 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. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Navigate to `PA_EXEC/pas/bin/Linux-x86_64`.
3. Rename the existing zip utility to `zip_backup`.
4. Create a soft link to link to the system installed 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.

12.3.3 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. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Edit the `PA_EXEC/pas/scripts/File/FileUncompress.py` file.

3. 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. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Navigate to `PA_EXEC/pas/bin/Linux-x86_64`
3. Rename the existing unzip utility to `unzip_backup`.
4. Create a soft link to link to the system installed 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.

12.3.4 Configurable Parameters

Parameters that can be modified to configure the behavior of PAS.

PAS stores its configuration data in a file called `PA_HOME/config/pas/conf/server.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

A description of the standard parameters for configuration of the PAS Server.

LOGGING_CONFIG_FILE_RELATIVE
> 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
> Default value is `server-log.properties`.

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.

Default value is pbsadmin.

LOGIN_MODULE_NAME

The PAS authentication module. PAS uses the Java Authentication and Authorization Service (JAAS) as its authentication abstraction layer.
Default value is UnixLogin.

STAGE_ROOT

The PAS file staging location.

Default value is `/stage`.

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 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. Default value is 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.
The default value is -1.

ZIP_COMPRESSION_SPEED

Compression speed value for the zip compression utility. The value of -0 is the fastest with no compression (Store), the value of -9 is the slowest with maximum compression. Default value is -1.

MAX_LIST_FILES_COUNT

The number of files to be returned by the FileList API. Allows sites to impose limits on how many files are returned to improve the response time of Access Web remote file operations.

Default value is 1000.

BUFFER_SIZE
Maximum buffer size for file operations in bytes. Default value is 65536.

JSCH_REQUEST_TIMEOUT

> The JSCH request timeout value. The default value is 30000.

MAX_EXECUTION_TIME

> Maximum process execution time in minutes. The default value is 5 minutes.

MODERN_COMMUNICATION_ENABLED

> A modern communication infrastructure for faster running job operations. Enabling this will use modern communication layer for faster running job operations. Default value is True.
> For more information about modern communication, refer to Enable and Disable the Modern Communication Module.

MODERN_COMMUNICATION_SHARED_LIBS

> Location of python modules needed for modern communication for Job operations. This directory should be present on all MOMs.
> For more information about modern communication, refer to Enable and Disable the Modern Communication Module.

REPLACE_UNSUPPORTED_XML_CHARACTERS

> Replace unsupported XML charactes with "_" in qstat command output. By default the value of this is set to false indicating that the unsupported XML characters are not checked in the Qstat output. This improves the performance of the Qstat output.

USER_SPACE_ENABLED

> Configure this parameter by changing the value to *true* if you want to restrict the listing of user files from other users. By default, the value is set to *false*.

HTTP_FILE_TRANSFER_ENABLED

> Configuration to toggle HTTP based file transfer protocol. By default, the value is set to false.

PAS_TEMP_DIR

> Temporary directory used by PAS for files or scripts to download. The default value that is set is `${env.PBSWORKS_HOME}/data/pas/system/temp`.

WLM_ADAPTOR

> Script folder that must be used based on workload manager. The default value is set to pbs. Change the value to age if Altair Grid Engine is the workload manager.

CLUSTER_TYPE

> PAS supports clusters where head node is Linux and all execution hosts are Linux, Windows, or both. A cluster with both Linux and Windows execution nodes is supported. The default value is set to Linux. Change the value to windows if the execution node is Windows.

Advanced Configuration Parameters

Optional parameters available for advanced configurations of the PAS Server.

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.
>
> Default path is `$PBS_EXEC/bin/pbs_python`.

△ ALTAIR

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_DATA_REFRESH_TIME

Time interval in minutes to fetch PBS related data like qmgr, queues etc. Default value is 15.

SSH_LOGGER_ENABLE

SSH Log enabling. Default value is False.

STAGE_ROOT_TEMP_DIR

Temporary folder for zip/download API.

Default stage root temporary directory path is `/tmp`. The default path can be changed to `$STAGE_ROOT/tmp`.

JSON_SCHEMA_VALIDATION

JSON Schema Validation. Default value is False.

IS_APPLICATION_STAGEIN_REQUIRED

Specifies if stage in of application runtime and pbsspawn directories is required. By default, this is set to true.
If it is disabled, then the APPLICATION_DEFINITION_EXECUTION_NODE_PATH specified is used for application definition related files.

APPLICATION_DEFINITION_EXECUTION_NODE_PATH

Location of application definitions on the execution node. User must deploy application definitions under the application folder. PAS will look for the application definitions specified in this path if IS_APPLICATION_STAGEIN_REQUIRED is disabled. This location should be available on all the execution nodes.

STAGEOUT_ENABLED

By default, the value is set to true indicating that the location for stage out files is taken care by Access Web.
Set the value to false, if you want to maintain a different stage out location (example, Shared File Systems).

FILE_TRANSFER_BUFFER_SIZE

Configurable buffer size for reading of streams and files. The default value is set to 8 MB.

FILE_UPLOAD_CHUNKS_CLEANUP_WAIT_TIME

The maximum wait time after receiving most recent chunk for clean-up of chunked file. The default value is set to 60 minutes.

FILE_UPLOAD_CHUNKS_CLEANUP_TIME

The time interval for clean-up of chunked file upload that are not active for more than the time specified in FILE_UPLOAD_CLEANUP_WAIT_TIME. The default value is set to 15 minutes.

PAS_SERVICE_TIMEOUT

Set the PAS session idle time before the connection is closed. The default value is set to 5000 milliseconds.

LogLevel

> Configure SSH log level that is displayed. By default, it is set to ERROR.

SSH_PORT

> The port number that is used for SSH. The default port value is set to 22.

SSH_SESSION_TIME_OUT

> Set the SSH session idle time before it is closed. The default value is set to 60000 milliseconds.

serverURL

> Set the serverURL to fetch credentials for Windows MoM. The default value that is set is `https://localhost:4443/pbsworks/api/credentials`.

QMGR_CUSTOM_RESOURCES_LIST

> List of selected custom resources configured in PBS that can be viewed on
> Access Web. By default, it is set to * to list all custom resources. Use comma (,)
> separated custom resource names for listing only specific resources. For example,
> QMGR_CUSTOM_RESOURCES_LIST=ndesktops, GridWorks, floatlicense.

PAS_TEMP_AREA_FILE_EXPIRATION_TIME

> The maximum lifetime in days of a file in PAS temp directory. By default, the value is set to 1 day.
> If the value is set to <1, then PAS temp directory will not be cleaned up. If the value is set to 14,
> then the files will be cleaned up older than 14 days.

PAS_TEMP_AREA_CLEANUP_INTERVAL

> Schedule a cycle time in days for clean up of files in PAS temp directory that are older than time
> specified in PAS_TEMP_AREA_FILE_EXPIRATION_TIME. By default, the value is set to 1 day.

USER_NAME_LOGGING_ENABLED

> Enable display of user name when logging. The default value is set to true.

12.3.5 Restrict the Display of Custom Resource

Restrict the display of PBS custom resources in Access Web.

PBS custom resources are now displayed on the Job Listing grid in Job Monitoring page. The custom resources are not included in the default job attributes that are displayed for the job, but can be added via the **Job Select Columns** list.
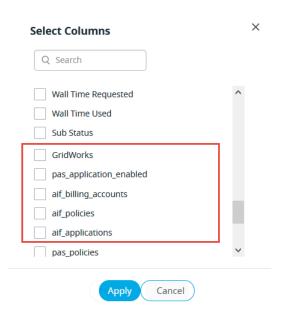
Figure 32: Job Attributes and Custom Resources

If your site would like to restrict displaying a custom resource in the Job Select Columns list, then follow these steps:

1. Login to the machine where PAS is installed as root or as a user with sudo permissions.
2. Source the Access Web configuration file to set up the environment variable `PA_HOME`.
   ```
   source /etc/pbsworks-pa.conf
   ```
3. Edit `$PA_HOME/config/pas/conf/server.conf`.
4. Locate the line:
   ```
   QMGR_CUSTOM_RESOURCES_LIST=*
   ```

   > 📝 **Note:** By default, the value of *QMGR_CUSTOM_RESOURCES_LIST* is set to *
   > indicating that all custom resource are displayed in the **Job Select Columns** list.

5. Update the value of *QMGR_CUSTOM_RESOURCES_LIST* to a comma-separated list of custom resources that should be displayed in the **Job Select Columns** list.
   For example, to display only the custom resources `pas_application_enabled` and `GridWorks`, set the value of *QMGR_CUSTOM_RESOURCES_LIST* as follows:
   ```
   QMGR_CUSTOM_RESOURCES_LIST=pas_application_enabled, GridWorks
   ```
6. Restart Access Web by entering the following command:
   ```
   service pbsworks-pa restart
   ```

   Now, the **Job Select Columns** list is restricted to only display the custom resources `pas_application_enabled` and `GridWorks`.
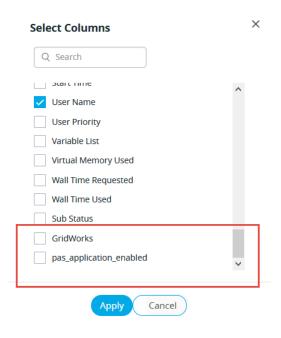
Figure 33: Custom Resources - Restricted

12.3.6 Change the Location of the PAS Repository

Change the default location of the PAS repository where application definitions and the site configuration file are stored.

The default location of the PAS repository is: `$PA_HOME/data/pas/`.

The environment variables `$PA_HOME` and `$PA_EXEC` can be used to specify the new location.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Stop Access Web using the following command:
   ```
   service pbsworks-pa stop
   ```
3. Edit the file `/etc/pbsworks-pa.conf`.
   ```
   vi /etc/pbsworks-pa.conf
   ```
4. Update the *PAS_REPO* variable with the custom directory location.
   ```
   PAS_REPO=$PA_HOME/data/pas/
   ```

   > 📝 **Note:** Ensure that the custom directory location is available.

5. Copy all the folders and file from `$PA_HOME/data/pas/` to the custom directory location.
6. Start Access Web using the following command:
   ```
   service pbsworks-pa start
   ```

12.4 Configure the Remote Desktops Component

Configurations when you install remote desktop component.

12.4.1 Configure the Desktop Manager for Interactive Applications

The Desktop Manager can be configured globally on all PBS execution hosts or on individual PBS execution host for interactive applications for Remote Desktops.

Globally update Desktop Manager information in: `/opt/altair/access/exec/joboperation/binaries/linuxmomclientmodules/desktop/core/xstartup` file.

> 📝 **Note:** Modern communication module and Change Remote Desktops to Use CPU Encoding must be performed after you make the changes globally.

Individually update Desktop Manager information in: `<MOM_CLIENT_MODULES>/momclientmodules/desktop/core/xstartup` file.

where `<MOM_CLIENT_MODULE>` is the location where Modern Communication Module is distributed.

1. Login to graphical PBS MoM.
2. If the Desktop Manager that is installed on the PBS MoM is GNOME:

    a) Verify that the *DESKTOP* variable is set to GNOME:

    ```
    #DESKTOP=${DESKTOP:-NONE}
    #DESKTOP=${DESKTOP:-MATE}
    #DESKTOP=${DESKTOP:-KDE}
    DESKTOP=${DESKTOP:-GNOME}
    ```

3. If the Desktop Manager that is installed on the PBS MoM is MATE:

    a) Verify that the *DESKTOP* variable is set to MATE:

    ```
    #DESKTOP=${DESKTOP:-NONE}
    DESKTOP=${DESKTOP:-MATE}
    #DESKTOP=${DESKTOP:-KDE}
    #DESKTOP=${DESKTOP:-GNOME}
    ```

4. If the Desktop Manager that is installed on the PBS MoM is KDE:

    a) Verify that the *DESKTOP* variable is set to KDE:

    ```
    #DESKTOP=${DESKTOP:-NONE}
    #DESKTOP=${DESKTOP:-MATE}
    DESKTOP=${DESKTOP:-KDE}
    #DESKTOP=${DESKTOP:-GNOME}
    ```

5. If the Desktop Manager that is installed on the PBS MoM is something other than GNOME, MATE, or KDE:

    a) Add a line to represent the Desktop Manager:

    ```
    #DESKTOP=${DESKTOP:-NONE}
    #DESKTOP=${DESKTOP:-MATE}
    #DESKTOP=${DESKTOP:-KDE}
    #DESKTOP=${DESKTOP:-GNOME}
    DESKTOP=${DESKTOP:-DESKTOP_MGR}
    ```

Where *DESKTOP_MGR* is an acronym for the installed Desktop Manager.

b) Add a line to start a Desktop Manager session:

```
MATE)  exec /bin/mate-session  ;;
KDE)   exec /bin/startkde ;;
DESKTOP_MGR)  exec /bin/DESKTOP_MGR_BIN  ;;
```

Where *DESKTOP_MGR* is an acronym for the installed Desktop Manager and *DESKTOP_MGR_BIN* is the executable for the Desktop Manager.

12.4.2 Configure the Desktop Manager for an Interactive Applications

If an application is optimized to run on a specific desktop manager like KDE, Gnome or Mate, Remote Desktops can be configured to launch the application with the suitable desktop manager.

Copy the `xstartup` file from a graphical node into the `runtime` folder path of the application definition and update the Desktop variable.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:
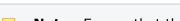
   `source /etc/pbsworks-pa.conf`

3. Copy the `<MOM_CLIENT_MODULES>/momclientmodules/desktop/core/xstartup` file from any graphical node to the application definition runtime folder to configure the desktop manager.
   For example, copy the `xstartup` file to the HyperWorks application definition at `$PA_HOME/data/pas/targets/localhost/repository/applications/HyperWorks/runtime/` folder.
4. Edit the `xstartup` file. To enable a desktop manager uncomment the corresponding statement in the file.
   For example, uncomment KDE *DESKTOP* statement in the `xstartup` file to configure KDE Desktop Manager as below:

   ```
   #DESKTOP=${DESKTOP:-NONE}
   #DESKTOP=${DESKTOP:-MATE}
   DESKTOP=${DESKTOP:-KDE}
   #DESKTOP=${DESKTOP:-GNOME}
   ```

5. Remove `timestamp.txt` file located at `$PA_HOME/data/pas/`.

12.4.3 Change Remote Desktops to Use CPU Encoding

Switch to CPU based encoding if the graphical node does not have a compatible GPU for Remote Desktops.

Change Remote Desktop configuration to use CPU based encoding in the following ways:

- Global configuration
- Individual graphical MoM

ALTAIR

> 📄 **Note:** The configuration on individual graphical MoM will take precedence over the global configuration.

Change Global Encoding Configuration

Change the global encoding configuration to use CPU based encoding.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:
   ```
   source /etc/pbsworks-pa.conf
   ```
3. Edit the file `PA_HOME/config/desktop/desktop.properties`.
   ```
   vi $PA_HOME/config/desktop/desktop.properties
   ```
4. Update the value of *jobaction.global_encoding_method* to *cpu*.
   ```
   jobaction.global_encoding_method=nvenc
   ```

   > 📄 **Note:** By default, the encoding method is set to *nvenc* which is NVIDIA's encoder.

5. Restart Access Web by entering the command:
   ```
   service pbsworks-pa restart
   ```

Change Encoding Configuration on Individual Graphical MoM

Change the desktop encoding node configuration on individual graphical MoM to use CPU based encoding.

> 📄 **Note:** The configuration on each of the PBS MoM will take precedence over the global configuration.

1. Login to the PBS MoM machine as root or as a user with sudo permissions.
2. Navigate to `$MOMCLIENTMODULES_PATH/config/`.

   `$MOMCLIENTMODULES_PATH` is the path where modern communication module is installed.
3. Edit the `platform.conf` file .
   ```
   vi platform.conf
   ```
4. Uncomment *encoding_method* variable and change its value to *cpu*
   ```
   #Encoding Methods: nvenc, cpu
   #encoding_method=nvenc
   ```

   > 📄 **Note:** By default, the encoding method is set to *nvenc* which is NVIDIA's encoder.

△ **ALTAIR**

12.4.4 Display Custom Message for Delayed Start of Interactive Desktop

Change the default message that is displayed when an interactive desktop takes a long time to start to a custom site specific message.

By default, interactive desktops are given 25 seconds to start before the following message is displayed:

```
Opening desktop is taking longer than expected, you can choose to wait for some more
time or kill desktop.
```

This message can be changed to provide a site-specific message explaining the cause for the delay. For example, in some cloud setup node is provisioned when a job comes to queue. The process of provisioning the node and to start a job might take time. In this case, administrator can provide custom message explaining the delay.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Open the file `PA_HOME/config/pa/configuration.json` to check the default locale file that is set.

    ```
    "locale": {
        "default": "en_US",
        "availableLocales": {
          "en_US": "/data/pa/locale/en_US.json",
          "fr_FR": "/data/pa/locale/fr_FR.json",
          "zh_CN": "/data/pa/locale/zh_CN.json",
          "ja_JP": "/data/pa/locale/ja_JP.json"
        }
      },
    ```

3. Open the default locale file from `PA_HOME/data/pa/locale/` .
    If English (en_US) is set as the default locale, then open the file `PA_HOME/data/pa/locale/en_US.json`.
4. Replace the text within the quotes after the colon with the custom message:

    ```
    "Opening desktop is taking longer than expected, you can choose to wait for
    some more time  or kill desktop": "Opening desktop is taking longer than expected,
    you can choose to wait for some more time  or kill desktop"
    ```

5. Refresh the browser to reflect the changes.

12.4.5 Change the Remote Desktops Timeout for Interactive Application

Change the remote desktop timeout when it is getting closed or killed because of inactivity over the client connection.

The default remote desktop timeout in the configuration file is 2 hours (7200 seconds). The remote desktop is killed if the user does not access the remote desktop within this time.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:

    ```
    source /etc/pbsworks-pa.conf
    ```

3. Naviate to `PA_HOME/config/desktop/`.

△ **ALTAIR**

```
cd $PA_HOME/config/desktop/
```

4. Edit the file `desktop.properties`.

5. Update the value of *jobaction.expiry_time* in seconds.

```
#desktop expiry time(seconds)
     jobaction.expiry_time=7200
```

6. Restart Access Web using the following command:

```
service pbsworks-pa restart
```

12.4.6 Configure GPU Limits when the Number of GPUs Change

Reconfigure the remote desktop 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 desktop 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.ndesktops = [o:PBS_ALL=2]
set queue iworkq resources_max.ndesktops = 1
set queue iworkq resources_min.ndesktops = 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
```

ALTAIR

The value of *max_queued_res.ndesktops* 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.ndesktops* using the command:

```
qmgr -c "set queue iworkq max_queued_res.ndesktops= [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 desktop 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. Login to the machine hosting the PBS Server as root or a user with sudo permissions.
2. 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.ndesktops = 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.ndesktops = 0
     resources_assigned.vmem = 0kb
     resv_enable = True
     sharing = default_shared
```

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

3. Change the value of *resources_available.ndesktops* using the command:

```
qmgr -c "set node <VNODENAME> resources_available.ndesktops=<NDESKTOPS>"
```

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

12.4.7 Configure Edge Gateway Proxy on a Separate Server

Install and configure edge gateway proxy on a different server to improve the scalability and performance of Remote Desktops.

The prerequisite is Access Web and the edge gateway proxy must be in the same domain.

> 📝 **Note:** Remote Desktops edge gateway proxy is always deployed in Linux machine.

Update the access portal with the edge gateway proxy to an external service.

Secure the Configured SSL Certificate

Secure the configured 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.

SSL certificate is a must to configure the edge gateway proxy. Generate the SSL Certificate with the domain name of the machine where remote desktop edge proxy is getting installed.

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
   ```

Install Remote Desktops Edge Proxy

Install and configure the Remote Desktops Edge Proxy edge proxy.

SSL certificate is a must to configure the edge gateway proxy. Refer to Secure the Configured SSL Certificate to generate the SSL certificate for edge gateway proxy.

1. Install the Access Web on separate Linux machine.

△ **ALTAIR**

> 📝 **Note:** Do not start the Access Web service.

2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:

```
source /etc/pbsworks-pa.conf
```

3. Edit the file `PA_HOME/config/shared/deployment.ini`.

```
vi $PA_HOME/config/shared/deployment.ini
```

4. Replace the contents of `deployment.ini` with the following:

```
pa_deploy_options=(
                    'option=("desktop_edgeserver")'
                )
    resultmanager_deploy_options=()
    pas_deploy_options=()
```

5. Edit the file `PA_HOME/config/desktop_edgeserver/nginx.conf`.

```
vi $PA_HOME/config/desktop_edgeserver/nginx.conf
```

6. Update the SSL certificate information which was generated in the following line:

```
ssl_certificate        certs/nginx.crt;
      ssl_certificate_key  certs/nginx.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. Update the URL1 and URL2 to the HTTP URL from which Access Web portal will be accessed.

```
($http_origin ~ (URL1|URL2))
```

> 📝 **Note:** The URL1 and URL2 are the URL that is used in the supported browser to access the Access Web portal.

   For example,

```
($http_origin ~ (https://access-dev-01.prog.altair.com:4443|
https://10.10.10.10:4443))
```

8. Remove `desktop.xml` from the location `PA_HOME/config/desktop/tomcat/conf/Catalina/localhost`

```
rm $PA_HOME/config/desktop/tomcat/conf/Catalina/localhost/desktop.xml
```

9. Start Access Web using the following command:

```
service pbsworks-pa start
```

Configure Access Web to Connect to an Edge Proxy

Update Access Web configuration to connect to an external edge proxy.

The prerequisite is Access Web and the edge gateway proxy must be in the same domain.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.

2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:

```
source /etc/pbsworks-pa.conf
```

3. Edit the file `PA_HOME/config/desktop/proxy_loadbalancing.properties`.

```
vi $PA_HOME/config/desktop/proxy_loadbalancing.properties
```

4. Update the *hostname* value with the hostname of an external edge proxy or the DNS load balancer.

```
hostname=localhost
```

5. Update the *port* value with the port to access the external edge proxy or the DNS load balancer.

```
port=5943
```

> 📝 **Note:** The default port used for edge proxy is 5943.

6. Change the value of *usedefault* to false to connect to the above configured edge proxy hostname and port.

```
usedefault=true
```

7. Edit the file `PA_HOME/config/api_gateway/default.d/desktop.conf`

```
vi $PA_HOME/config/api_gateway/default.d/desktop.conf
```

8. Comment the following line:

```
add_header Set-Cookie "access_token=$access_token;Path=/;HttpOnly;Max-Age=86400"
 always;
```

9. Add the following line with domain:

```
add_header Set-Cookie "access_token=
$access_token;Domain=.altair.com;Path=/;HttpOnly;Max-Age=86400" always;
```

The domain is where Access Web and edge gateway proxy is installed.

10. Restart Access Web using the following command:

```
service pbsworks-pa restart
```

12.4.8 Configure a Windows Graphical Node to Run a Single Job per User

To prevent a user from running multiple Remote Desktops on the same Windows graphics node, configure the graphics node at the PBS Professional level to limit the number of running jobs on the node to one per user.

Users cannot open multiple Remote Desktops on the same Windows graphical node because RDP does not allow more than one remote desktop per user per node.

Perform the below steps to configure PBS Professional to run a single remote desktop per user per graphics node.

> 📝 **Note:** Multiple remote desktop can be run on the same node for different users.

Repeat these steps for each Windows graphics node.

1. Login to the PBS Server as PBS Manager.

2. Open the command prompt.

△ **ALTAIR**

3. Enter the command:

```
qmgr -c "set node <vnode name>  max_user_run=1"
```

12.4.9 Disable Remote Desktops Statistics Panel

Disable Remote Desktops statistics panel to view the remote desktop throughput and frames per seconds in the user interface.

By default, the Remote Desktops statistics panel is enabled.

Use the Remote Desktops statistics panel to help diagnosis performance issues.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.

2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:

```
source /etc/pbsworks-pa.conf
```

3. Edit the file `PA_HOME/config/pa/configuration.json`.

```
vi $PA_HOME/config/pa/configuration.json
```

4. Update the value of *showStatisticsInfo* to false.

```
showStatisticsInfo=false
```

5. Refresh the browser to reflect the changes.

12.4.10 Disable User List for Sharing the Desktop

Configure share session user plugin to disable the user list for sharing the Desktop.

By default, the user list will be displayed to share the desktop. The share desktop user plugin class, *sharedesktopuser.plugin.classname* is specified as *com.altair.pbsacces.sharedesktop.ActiveUserCollectionImp* in the file `$PA_HOME/confg/desktop/desktop.properties`.

Update the *sharedesktopuser.plugin.classname* with *com.altair.pbsacces.sharedesktop.ShareSessionDisabledImp* to disable the user list for sharing the desktop.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.

2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:

```
source /etc/pbsworks-pa.conf
```

3. Open file `$PA_HOME/config/desktop/desktop.properties`.

4. Update the *sharedesktopuser.plugin.classname* with *com.altair.pbsacces.sharedesktop.ShareSessionDisabledImp*.

```
sharedesktopuser.plugin.classname=com.altair.pbsacces.sharedesktop.ShareSessionDisabledImp
```

> 📝 **Note:** This will disable the display of user list for sharing the desktop.

△ **ALTAIR**

5. Restart Access Web using the following command:
   ```
   service pbsworks-pa restart
   ```

12.4.11 Disable the Desktop Title Preview

Disable the desktop preview displayed in the Desktops list tab.

By default, the desktop preview is enabled and it gets displayed in the Desktops list.

To disable the remote desktop preview update the `isTilePreviewEnabled` to *false* in the configuration file (`configuration.json`) located at `PA_HOME/config/pa`.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:
3. Edit `PA_HOME/config/pa/configuration.json`
   ```
   vi $PA_HOME/config/pa/configuration.json
   ```
4. Change the value of `isTilePreviewEnabled` to *false*.
   ```
   "isTilePreviewEnabled": true
   ```
5. Refresh the browser to reflect the changes.
   The desktop preview will not be displayed in the Desktops list.

Open Desktop Using:

| HyperWorks | GlxSpheres | Gedit | HyperMath | HyperMesh | HyperView |

Desktops 🔍 Search by Desktop Name

No Preview Available
1: GlxSpheres
Timeout in: 118 min.
Started: 1/13/2022, 4:28:08 PM

No Preview Available
2: HyperWorks
Timeout in: 120 min.
Started: 1/13/2022, 4:29:10 PM

12.4.12 Hide Remote Desktops Sharing Option

Hide the Remote Desktops sharing option.

By default, the Remote Desktops sharing option is enabled.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:

    ```
    source /etc/pbsworks-pa.conf
    ```
3. Edit the file `PA_HOME/config/pa/configurations.json`

    ```
    vi $PA_HOME/config/pa/configurations.json
    ```
4. Change the value of *enableSharing* to *false* to disable the sharing option.

    ```
    "enableSharing": true
    ```

    By default, the value of *enableSharing* is set to *true*.
5. Refresh the browser to reflect the changes.

12.4.13 Disable Auto Adjust Quality Settings

Disable the remote desktops auto adjust quality settings.

By default, the Remote Desktops auto adjust quality settings is enabled. This ensures that there is a smooth experience of the Desktops and it improves the performance.

Turning off auto adjust quality settings will impact the performance as the frames per second and bitrate of Remote Desktops are getting changed from the default settings.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:

   ```
   source /etc/pbsworks-pa.conf
   ```
3. Edit the file `PA_HOME/config/pa/configurations.json`

   ```
   vi $PA_HOME/config/pa/configurations.json
   ```
4. Change the value of *disable_auto_adjust_quality* to *true* to disable the auto adjust quality setting.

   ```
   "disable_auto_adjust_quality": true
   ```

   By default, the value of *disable_auto_adjust_quality* is set to *false*.
5. Refresh the browser to reflect the changes.

12.5 Configure Results Visualization Service

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

12.5.1 Activate Solver Files Readers

You can activate solver file readers 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.

> ℹ️ **Tip:** If the configuration fails for any of the specified data files, then place the external reader line of code below the text column line of code. For example: The code looks like this when it is changed:
>
> ```
> *RegisterExternalColumnReader({external_readers_dir + "/hgtextcolumn.exe"})
> *RegisterExternalReader({external_readers_dir + "/
> hg<solverfilereadername>.exe"}, "", "", ascii)
> ```

Activate the Abaqus Reader

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

1. Edit the file at `<HyperWorks_INSTALL_DIR>/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 Abaqus 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 `<HyperWorks_INSTALL_DIR>/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 CFX 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 `<HyperWorks_INSTALL_DIR>/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 STAR-CCM 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 `<HyperWorks_INSTALL_DIR>/io/abf_readers/bin/linux64`

12.5.2 Supported Result File Types

The result file types supported by RVS for visualizing plot and animation.

Table 4: Supported Result File Types

Solver Name	Results File Format
Radioss Bulk	*.op2, *.h3d, *.res, *.pch, *.gz
Radioss Block	*A00#, *.T##, *.gz
Optistruct	*.op2, *.h3d, *.res, *.pch, *.hgdata
MotionSolve	*.mrf, *.plt, *.h3d, *.maf
Abaqus	*.odb, *.dat, *.out
ADAMS	*.req, *.res, *.nam, *.rsp, *.shl, *.sta
Ansys	*.rst, *.rth,
CFX	*.out
Fluent	*.out, output.*, *.trn, *.txt*
Ls-Dyna	d3plot, *dynain, *.fz, *.dyn, Intfor, Ptf, ABSTAT, BINOUT, BNDOUT, DBFSI, DEFORC, ELOUT, GLSTAT, GECOUT, JNTFORC, MATSUM, NCFORC, NODFOR, NODOUT, RBDOUT, RCFORC, RWFOC, SBTOUT, SECFORC, SLEOUT, SPHOUT, SWFORC
NanoFluidX	nFX
UltraFluidX	uFX
HyperXtrude	*.h3d, *.out
FEMZIP	*.fz, *d3plot*
HW ASCII	*.hwascii
MADYMO	*.fai, *.kn3
Marc	*.t16
Nastran	*.op2, *.pch
NIKE3D	*n3plot
Pamcrash	*.DSY, *.erfh5, *.THP, *.fz, *.h3d

Solver Name	Results File Format
Permas	*A##
Other Ascii formats	*.xgr, *.dat, *.col, *.csv, *.rvp

12.5.3 RVS Cache Data

Enable data caching for RVS. The fetched result file data is stored as a RVS cache data in the configured RVS server.

Set RVS 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 to Configure Data Directory.

Scheduling data cleanup avoids overlaing the database and server's disk space. 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.

Schedule RVS Data Cleanup

You can configure the recurring RVS data cleanup criteria settings in `site_config.xml` file.

Follow the steps given here:

1. Open the file at `PA_HOME/config/resultmanager/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 criteria 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 using the command, `service pbsworks-pa restart`.

12.5.4 Configure RVS Parameters

Main parameters related to RVS in Access Web.

Result Service Configuration

Result service configurations are defined in the *site_config.xml* file located at `PA_HOME/config/resultmanager/site_config.xml`. RVS has to be restarted after any changes to the site_config.xml

Configurations in site_config.xml

TOC Size

> The maximum TOC size-limit in bytes.
> Default value is 2097152 (In bytes) (i.e. 2MB).
> Configure the maximum TOC size using the parameter `<SizeLimit toc="2097152">`.
> If the TOC of the result file exceeds the size limit value, RVS will make multiple calls to retrieve the remaining data. To avoid this, increase the TOC size limit value to optimize data extraction time.
> Rename the result file after increasing the TOC size limit. This will ensure that RVS fetches the TOC data based on your settings instead of using cached data.

MaxProcessExecutionTime

> Maximum Compose process execution time in minutes. The default value is 5 minutes.
> Configure the Compose process execution time using `<MaxProcessExecutionTime time="5"/ >`

AIF Impersonation

> The result file computation and license checkout are processed for the user.
> Default value is set to true.
> If the value is changed to False, then the RVS result file computation and license check out are processed using the concern administrator's credentials.
> Configure the RVS impersonation using `<AIFImpersonation enabled ="true">`

Cache enabled

> Enable or disable data caching for the RVS data extraction requests. To improve performance, repeated RVS file requests are fetched from the cache. This will speed up the RVS performance.
> Default value is set to true.
> If set to false, result data is fetched from the database for every query. This reduces the speed of the application due to extra data transfer.
> Configure the RVS data caching using `<Cache enabled="true">`.

Session defaultTimeout

> The maximum amount of time the server should wait for a response from another application before disconnecting.
> Default value is 6000 milliseconds (i.e., 6 seconds).
> Configure the session default timeout using `<Session defaultTimeout="6000">`.

Operation defaultTimeout

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

Default value is 6000 milliseconds (i.e., 6 seconds).

Configure the operation default timeout using `<Operation defaultTimeout="6000">`.

Configuration in resultmanager.conf

The browser timeout value for RVS is configured in the resultmanager.conf file located at: PA_HOME/config/api_gateway/default.d/resultmanager.conf

Browser Timeout

The amount of time the browser should wait for a response from a request.

A browser timeout message is displayed if the server does not provide a response before this time limit is reached.

> 📑 **Note:** The timeout value doesn't interrupt any of the RVS background operations.

Default value is 600 seconds (i.e., 10 minutes).

Configure the browser timeout for RVS requests in the line `<proxy_read_timeout 600sec>` and enter the timeout value in seconds.

Parameters for Compose

The default value of the parameters are stored and listed in the following file *plugin_def.xml*. This file is located at `PA_HOME/config/resultmanager/plugins/compose_application/plugin_def.xml`

Parameters in plugin_def.xml

Socket timeout for Compose

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

Default value is 900000 milliseconds.

Configure the socket timeout for Compose in the line `<Application id="COMPOSE_APPLICATION">` and enter the `<socketTimeout="900000">` value.

Connection timeout for Compose

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

Default value is 900000 milliseconds.

Configure the connection timeout for Compose in the line `<Application id="COMPOSE_APPLICATION">` and enter the `<connectionTimeout="900000">` value.

Request timeout for PBS datasource

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

Default value is 6000 milliseconds.

Configure the request timeout for PAS in the line `<DataSourceHandler id="PBS_DATA_SOURCE_HANDLER">` and enter the `<requestTimeout="900000">` value.

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.

Default value is 6000 milliseconds.

Configure the connection timeout for PAS in the line `<DataSourceHandler id="PBS_DATA_SOURCE_HANDLER">` and enter the `<connectionTimeout="900000">` value.

Application Definitions

13

A predefined set of instructions 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:

To run a solver or application through Access Web requires an application definition. 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:

`PA_HOME/data/pas/targets/localhost/repository/applications`.

Application definitions now need to be compatible with Python 3.

> 📝 **Note:** Python usage in application definitions is usually minimal and straightforward. Validate the updated application definitions before using it.

For sites that are installing Access Web for the first time, default application definitions and a site configuration file are provided to get you up and running quickly. Obtain them through your usual Altair PBS Works Support (pbssupport@altair.com).

For sites that are using legacy products such as Compute Manager, legacy application definitions can be ported so that they can be used by Access Web.

Access Web supports application definitions in XML and JSON format.

Users can also have their own personal application definition repositories at `/home/$USER/userapps`.

The path of the user apps folder must be added as a Root Directory while registering a cluster, so that the users can view their application definition in Access Web user interface.

13.1 Application Definition Components

Overview of application definition components.

An application definition is comprised of a set of Python scripts and two XML or JSON 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*.

13.1.1 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` or `app-inp-applicationname.json` where `applicationname` is whatever name you choose to give your application.

13.1.2 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 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` or `app-conv-applicationname.json` where `applicationname` is whatever name you decide to give to your application.

13.1.3 Application Runtime Scripts

The application runtime scripts are what 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*.

13.1.4 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*.

The `site-config.xml` file must be updated manually if you add an application definition. Access Web does not create a backup the `site-config.xml` file. If you delete the `site-conifg.xml` file, then you must create it manually. Before making changes to the site configuration file, it is recommended to back it up.

13.2 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.

13.3 Display a Custom Icon for an Application

Provide a custom icon to represent an application or solver in the Access Web user interface.

By default, Access Web generates a unique generic icon for each of the application for visual distinction.

The generic icon is displayed in the Access Web user interface when displaying the list of available applications/solvers for submitting a job. To display a custom icon, place the icon in a special application definition directory.

A directory called `avatar` must be created in the application definition directory and the custom icon must be placed in this directory.

The custom icon should meet the following criteria:
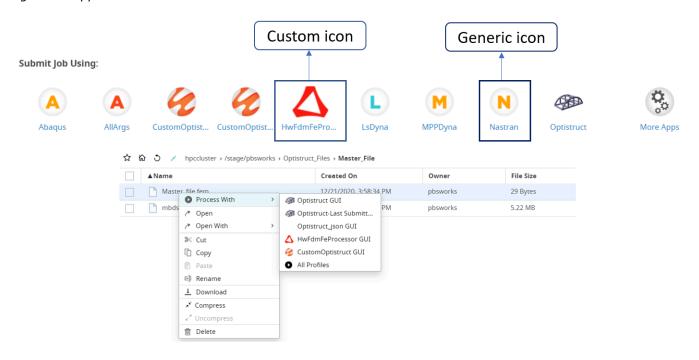
- supported icon formats are JPG, JPEG and PNG.
- the name of the custom icon must be `application.jpg`, `application.jpeg`, or `application.png`.
- minimum size of the icon should be 48x48 pixels.

The custom icon placed in the `avatar` directory gets converted to the following application icon set:

- `ApplicationName`-16.png
- `ApplicationName`-24.png
- `ApplicationName`-48.png
- `ApplicationName`.ico

The custom icon will be resized dynamically and it will be displayed in the application list menu and in the context menu.

Figure 34:Application Icon in Job List and Context Menu

To maintain backward compatibility, application definitions with the following XML elements `<ApplicationIconSmall>` (icon size 16x16), `<ApplicationIconMedium>` (icon size 48x48), `<ApplicationIconRegular>` (icon size 24x24) for the context menu and application list is still valid.

1. Log in to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Navigate to the required application directory to place the application icon.
3. Create an *avatar* directory.
4. Copy the custom icon to the `avatar` directory.
5. Repeat steps 2 through 4 for all applications.
6. Restart Access Web by entering the following command:

```
service pbsworks-pa restart
```

13.4 Define a Category for an Application Definition

Define a category in the application definition input file so that the application listed can be filtered based on the category.

The XML element `<TAGS>` or JSON element `"Tags"` defines the category under which an application is listed within the Access Web UI.
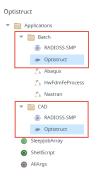
Figure 35: Application Definition Category

Applications can be filtered in the UI by selecting a category.

For example, add a category to the Optistruct application using the XML element `<TAGS>`:

```
<TAGS>
    <TAG>ApplicationType:Batch</TAG>
    <TAG>ApplicationType:CAD</TAG>
</TAGS>
```

For example, add a category to the Optistruct application using the JSON element `"Tags"`:

```
"Tags": {
    "type": "array",
    "items": {
      "type": "string"
    },
    "value": [
      "Application Type:Batch"
    ],
    "Displayable": false
  },
```

This tag indicates that the Optistruct application will be listed under the ApplicationType. The Batch and CAD will be the subcategory of ApplicationType. The category tags can be defined in other application definition input files so that the categories can be used to filter the application list.

13.5 Administration of Application Definitions

PAS has a central location for storing application definitions -

`PA_HOME/data/pas/targets/localhost/repository/applications`.

All application definition components are stored in this directory.

Access Web supports application definitions in XML and JSON format.

Application definitions now need to be compatible with Python 3.

> 📄 **Note:** Python usage in application definitions is usually minimal and straightforward. Validate the updated application definitions before using it.

13.5.1 Add a New Application Definition

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

Application definitions now need to be compatible with Python 3.

A default application definition called ShellScript is available after installing Access.

Follow these steps to add an application definition:

1. Login to the machine hosting the PAS Server as root or as a user with sudo permissions.
2. Navigate to `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` or `app-inp-appname.json`) and the application definition converter file (`app-conv-appname.xml` or `app-conv-appname.json`) 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. Navigate to `PA_HOME/data/pas/targets/localhost/repository/`.

   > 📄 **Note:** It is recommended to back up the site configuration file before making any changes to it.

9. Update the application definition information in `site-config.xml`.
10. Restart Access Web by entering the following command:
    ```
    service pbsworks-pa restart
    ```

△ ALTAIR

13.5.2 Application Definition Validation

When Access Web starts, it performs a validation of the existing application definitions. If application definitions fail to meet key criteria, error messages are displayed in the PAS log file indicating why the application definition was invalid.

If an application is not displayed in the Access Web user interface after adding a new application definition or making changes to an existing one and restarting Access Web, then most likely there were validation problems with the application definition or the site configuration file.

View the PAS log for any error messages and edit the application definition or site configuration file to take any required corrective action. An invalid application definition will not prevent the PAS Server from starting up.

13.5.3 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 Access Web.

Update an Application Definition

You can easily modify an existing application definition using your favorite XML or JSON editor.

Application definitions now need to be compatible with Python 3.

> 📒 **Note:** Python usage in application definitions is usually minimal and straightforward. Validate the updated application definitions before using it.

1. Login to the machine hosting the PAS Server as root or as a user with sudo permissions.
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 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 as root or as a user with sudo permissions.
2. Navigate to the applications directory located at:

   `PA_HOME/data/pas/targets/localhost/respository/applications`.

△ **ALTAIR**

3. Remove the application directory and all of its content including the `runtime` subdirectory and associated files.

4. Remove references to the application definition from the `site-config.xml` file located at:

`PA_HOME/data/pas/targets/localhost/repository/`.

5. Restart Access Web by entering the following command:

```
service pbsworks-pa restart
```

13.6 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`.

13.6.1 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 os JSON 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.

The `site-config.xml` file must be updated manually if you add an application definition. Access Web does not create a backup of `site-config.xml` file. If you delete the `site-conifg.xml` file, then you must create it manually. Before making changes to the site configuration file, it is recommended to back it up.

13.6.2 Site Configuration File Backup

Upon server start-up, PAS will validate the content of the site configuration file and the content of application definitions. Any validation errors are written to the PAS log file.

If an application is not displayed in the Access Web user interface after adding a new application definition or making changes to an existing one and restarting PAS, then most likely there were validation problems with the application definition or the site configuration file.

View the PAS log for any error messages and edit the application definition or site configuration file to take any required corrective action.

As of Access 2022.1.0, the site configuration file is no longer backed up during its validation process.

13.6.3 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>
```

13.6.4 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>
```

13.6.5 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*.

13.6.6 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.

13.7 Interactive Application Definitions

Interactive application definition mandatory and special arguments.

An interactive application runs a 3D and graphics-intensive application on a graphical node.

Application definitions now need to be compatible with Python 3.

> 📝 **Note:** Python usage in application definitions is usually minimal and straightforward. Validate the updated application definitions before using it.

13.7.1 Special Interactive Application Arguments

Arguments that can be added to an interactive application definition.

For an application to be identified as interactive, the corresponding application definition must contain the `<Interactive>` XML element and its value must be set to `true` in the application input file.

```
<Interactive>true</Interactive>
```

An example of JSON tag is as follows:

```
"Interactive": {
     "type": "boolean",
     "value": true,
     "Displayable": false
   },
```

In the JSON format, the input required fields are mentioned as below:

```
"required": [
    "JOB_NAME",
    "VERSION",
    "APP_GEOMETRY"
  ]
}
```

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 *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>UI_APP_ARGS</Name>
        <Description>'\n' separated args</Description>
        <DisplayName>Arguments</DisplayName>
        <InputRequired>false</InputRequired>
    </ArgumentString>
</ArgumentChoice>
```

An example of JSON format is:

```
"UI_APP_ARGS": {
      "type": "string",
      "Displayable": true,
      "DisplayName": "Arguments",
      "Description": "'\n' separated args"
    },
```

Environment

A special string argument having the name *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>APP_ENVS</Name>
        <Description>'\n' separated envs</Description>
        <DisplayName>Environments</DisplayName>
        <InputRequired>false</InputRequired>
    </ArgumentString>
</ArgumentChoice>
```

An example of JSON format is:

```
"APP_ENVS": {
      "type": "string",
      "Displayable": true,
      "DisplayName": "Environments",
      "Description": "'\n' separated envs"
    },
```

WorkDirectory

A special string argument having the name *APP_WDIR* can be added to an application definition so a job working directory is created when the job is submitted.

```
<ArgumentChoice>
    <ArgumentString>
        <Name>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`.

An example of JSON format is:

```
"APP_WDIR": {
      "type": "string",
      "Displayable": true,
      "DisplayName": "Working Dir",
      "Description": "Working dir"
    },
```

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 where to stageout job input files.

An example of JSON format when the *COPY_BACK_FILES* argument is enabled:

```
"COPY_BACK_FILES": {
    "type": "boolean",
    "Displayable": true,
    "DisplayName": "Copy back files",
    "Description": "Should job file(s) be staged out",
    "FeatureEnabled": true
    "RefreshOnUpdate": true
},
```

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="DESKTOP_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`.

An example of JSON format is:

```
{
        "Name": "DESKTOP_JOB",
        "Value": "True"
    },
```

13.7.2 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 HyperWorks application definition which is available after installing the Remote Desktops components of Access Web.

The default application definitions are provided to get you up and running quickly. Obtain them through your usual Altair PBS Works Support (pbssupport@altair.com).

△ **ALTAIR**

1. Navigate to `PA_HOME/data/pas/targets/localhost/repository/applications/`

2. Copy the HyperWorks 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 HyperWorks HyperView
   ```

3. Rename the HyperWorks application definition files to the name of the new application.

   ```
   mv app-actions-HyperWorks.xml app-actions-HyperView.xml
   mv app-conv-HyperWorks.xml app-conv-HyperView.xml
   mv app-inp-HyperWorks.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='HyperWorks']/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. Remove `time_stamp.txt` from `PA_HOME/data/pas/`

The new application will be available in Access Web.

App Composer

14

Access Web provides an easy way to create a user defined application definition (app def) using the App Composer.

This chapter covers the following:

The App Composer provides an easy way to:
- Add executable commands
- Add pre and post job execution scripts
- Manage app def parameters
- Test and onboard app defs

You can now create application definitions in Access Web without depending on the site administrator or editing XML files.

The user defined application definitions will be listed in the Jobs tab. They will also be displayed in the context menu when you right-click on an input file.

- Edit the application definitions
- Publish a local application definition
- Publish and Unpublish a global application definition
- Share an application definition
- View a shared application definition
- Clone an application definition
- Delete an application definition

14.1 Application Definitions Composer Components

Overview of application definition composer components.

To launch the App Composer, click your user profile and select App Composer.

The main panels of the App Composer are:

- Applications panel displays a list of user defined app defs.
- Compose panel enables you to define the components of the app def
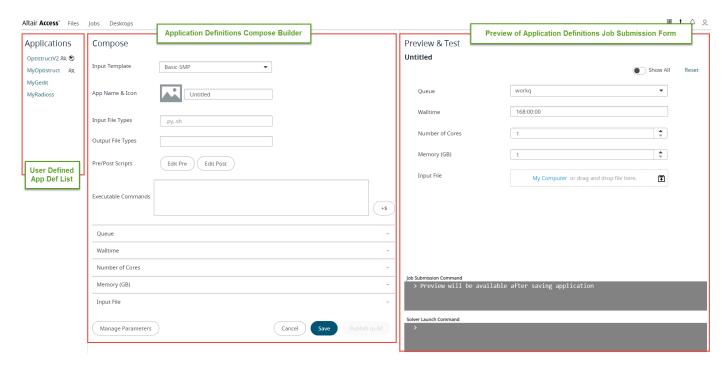- Preview & Test panel displays the Job Submission form for the app def.

Figure 36: Application Definitions Composer

Applications Panel

The Applications panel provides all the application definitions that is created using the App Composer. The application definitions listed can be edited, published, shared, and cloned. The application definition can be published to admin to make it a global app def and removed if it is no longer required.

Figure 37:Applications Panel

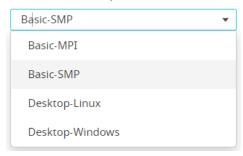The 👥 indicates that it is a shared application definition.

The 🌐 indicates that it is a global application definition which is published and available to all the users.
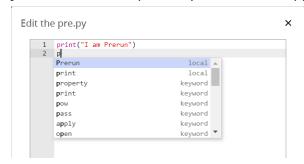
Compose Panel

The application definitions composer builder provides the following:

- Starter templates to compose application definitions - choose basic MPI or SMP based template for your solver. If you are creating remote desktop application definitions, then choose a Desktops-Linux or Desktops-Windows template based on the target Operating System (Windows or Linux).

    Figure 38:Starter Templates

- Common parameters for all the template - the following are some of the common parameters:
    - Application Name and Icon
    - Input File Types - associated with the application definitions that is created
    - Output File Types
    - Pre and Post Scripts - specify Pre and Post python script that can be run before and after the job is executed respectively. The editor supports syntax highlighting and code completion.

◦ Executable Commands - specify the command to run the application definition executable. The syntax of shell script commands can be used. Multi line commands are also supported. Click

(+$)

to get a list of parameters which can be used to build the executable commands.
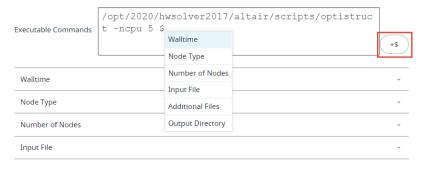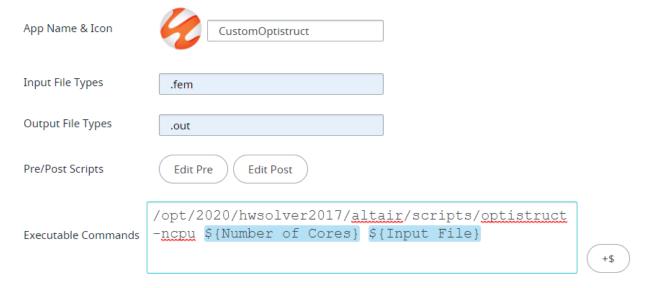

Figure 39:Common Parameters

- Parameters based on the template - these parameters for the application definition are defined by the selected starter template. Some of the parameters are Walltime, Number of Cores, Input File, Additional Files, GPUs, Memory and so on.
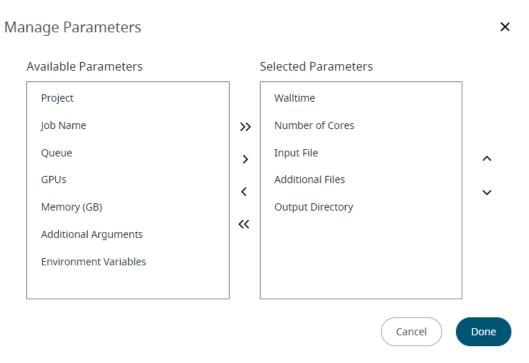
Figure 40:Variable Parameters

You can view or manage the list of parameters by clicking **Manage Parameters**.

Figure 41:Manage Parameters

You can move the parameters from Available Parameters to Selected Parameters using ❯ and ❮ buttons. Rearrange the order of the parameters in the Selected Parameters column using ∧ and ∨ buttons. The Job Submission form will display the parameters in the same order.

Specify if the variable parameter is an optional or required field. By default, only the required are displayed in the Job Submission form. You can also set default value for these parameters.

Figure 42:Required or Optional Parameters

Preview and Test Panel

The Preview and Test panel displays the job submission form for the app def. You can edit in the Compose form and review the changes in the Preview & Test panel.

Figure 43: Preview and Test

The Preview and Test panel also provides the Job Submission Command that is used to submit the job in PBS along with the Solver Launch Command.

Figure 44:Job Submission Command and Solver Launch Command

By default, only the required fields are displayed. Move the toggle switch ⬤ Show All to the right to display all available application arguments.

Tooltip displays the description of the application arguments when you mouseover the argument field.

Click **Reset** at the top right-hand corner of the Job Submission Form to reset the values of application arguments.

Fill the required arguments in the Job Submission form and click **Submit & Test**. The job is submitted and a notification is displayed providing the status of the job.

14.2 Create Solver Application Definitions

Create your own solver application definitions using the Basic-SMP or Basic-MPI starter template.

1.   Click ⚲.

2.   Select **App Composer**.
     The App Composer is displayed.

     The App Composer UI is made up of three main areas. On the left panel, all your custom applications are listed. In the centre panel build your own application definition and save it. The right panel provides a preview of our application definition in the Job Submission form.
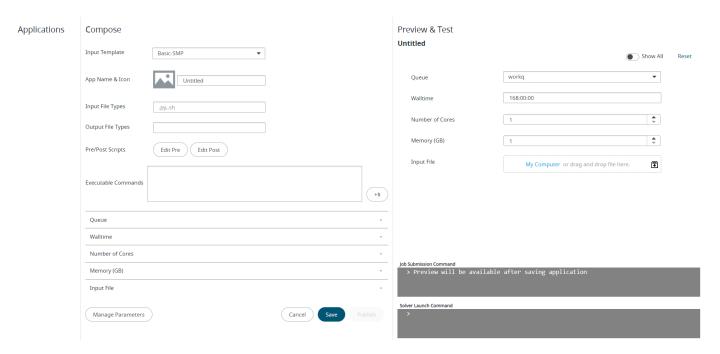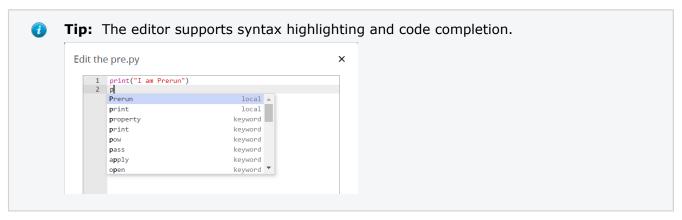

Figure 45: App Composer

3.   Select a starter template from the **Input Template** drop down list.
     To create solver based application definitions, select either Basic-SMP or Basic-MPI starter template.

4.   Click 🖼 to browse and select an icon for the application definition.

5.   Enter a name for the application definition.

6.   Enter all the input file type extensions that will be supported by the application definition.
     Access Web links a job input file to a specific application or solver via the file's extension. This set up will determine which solvers are available for a file extension.

7.   Enter the output file types that has to be associated with the application definition.

8.   Click **Edit Pre** to provide a python script to run before the job is executed.

> **Tip:** The editor supports syntax highlighting and code completion.
>
> Edit the pre.py   ✕
>
> ```
> 1  print("I am Prerun")
> 2  p
> ```
> | Prerun | local |
> | print | local |
> | property | keyword |
> | print | keyword |
> | pow | keyword |
> | pass | keyword |
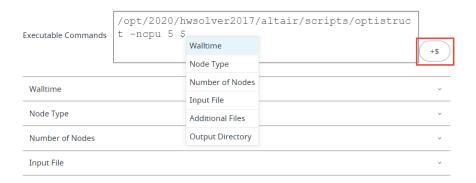> | apply | keyword |
> | open | keyword |

9. Click **Edit Post** to provide a python script to run after the job is executed.

> **Tip:** The editor supports syntax highlighting and code completion.

10. Enter the executable command to run the job.

The executable command can be a basic shell scripts and multi-line shell scripts.

> **Tip:** The command box provides the list of parameters when you click (+$) or when you enter the syntax $ which helps to write the executable command.

Executable Commands
```
/opt/2020/hwsolver2017/altair/scripts/optistruc
t -ncpu 5 $
```
(+$)

- Walltime
- Node Type
- Number of Nodes
- Input File
- Additional Files
- Output Directory
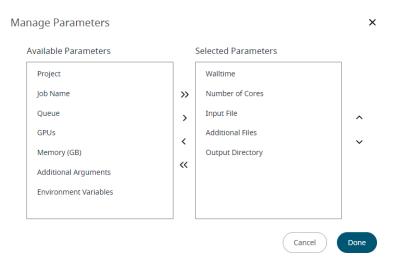
Walltime

Node Type

Number of Nodes

Input File

11. Choose the parameters that are provided by the selected starter template required for the application definition.

You can specify if the parameters are optional or required field in the job submission form. You can also set the default values for the parameters.
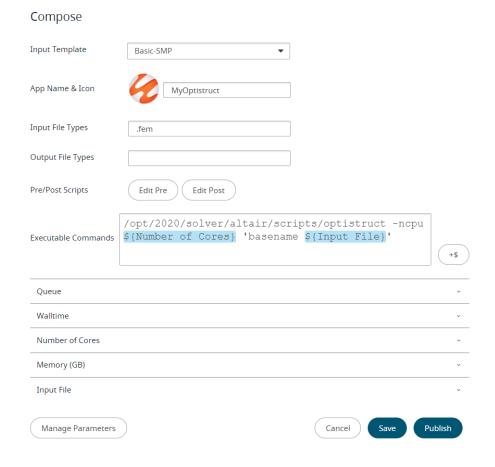
Walltime

Number of Cores

Default [2] ✓ Required

12. Click **Manage Parameters** to add or remove parameters.

△ ALTAIR

Manage Parameters   ✕

Available Parameters   Selected Parameters

Project   Walltime
Job Name   »   Number of Cores
Queue   ›   Input File
GPUs   Additional Files   ^
Memory (GB)   ‹   Output Directory
Additional Arguments   «   ⌄
Environment Variables

                                                        Cancel   Done

You can move the parameters from Available Parameters to Selected Parameters using ⟩ and ⟨ buttons. Rearrange the order of the parameters in the Selected Parameters column using ^ and ⌄ buttons. The Job Submission form will display the parameters in the same order.

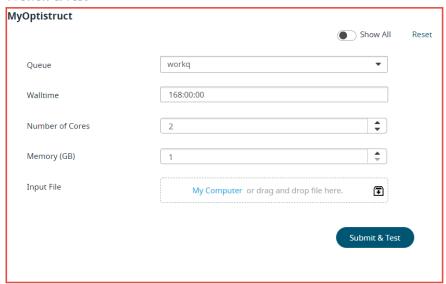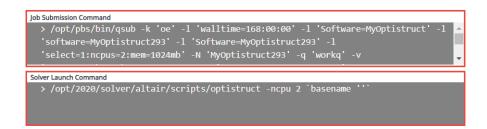**13.** Verify if you have provided all the details required for the application definition.

Compose

Input Template   Basic-SMP   ▼

App Name & Icon   🌀 MyOptistruct

Input File Types   .fem

Output File Types

Pre/Post Scripts   Edit Pre   Edit Post

Executable Commands   /opt/2020/solver/altair/scripts/optistruct -ncpu
                      ${Number of Cores} 'basename ${Input File}'   +$

Queue   ⌄

Walltime   ⌄

Number of Cores   ⌄

Memory (GB)   ⌄

Input File   ⌄

Manage Parameters   Cancel   Save   Publish

△ALTAIR

14. Click **Save**.

The Preview & Test panel on the right displays the job submission form, Job Submission command and Solver Launch Command information for the application definition we have composed.

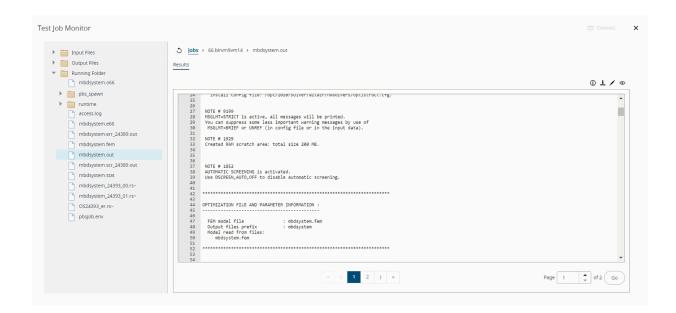**15.** Fill in the required application or profile arguments in the Job Submission form.

16. Click **Submit & Test**.

> ⓘ **Tip:** Click on the Job ID link provided in the notification to view the job status.

The job is submitted and a Test Job Monitor window is displayed.

> **ⓘ Tip:** From the files list on the left panel, you can select other files to view the job details.

The application definition is added to the existing application definitions list.

Submit Job Using:

| MyOptistruct | Optistruct | Abaqus | AllArgs | HwFdmFePro... | LsDyna | MPPDyna | Nastran |

> **☰ Note:** The application definitions that you have created are in JSON format and is stored at `/home/$USER/userapps`. The System Administrator can configure the path so that you can edit it in the Text Viewer.

14.3 Create Remote Desktops Application Definitions

Create your own Remote Desktops application definitions using the Desktop-Linux or Desktop-Windows starter template.

1. Click 👤 .

2. Select **App Composer (Beta)**.
   The App Composer is displayed.

   The App Composer UI is made up of three main areas. On the left panel, all your custom applications are listed. In the centre panel build your own application definition and save it. The right panel provides a preview of our application definition in the Job Submission form.
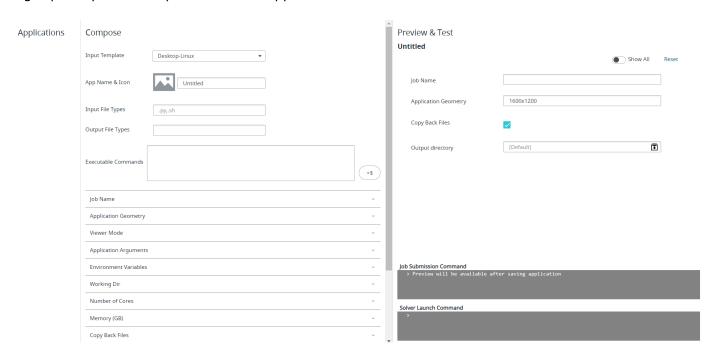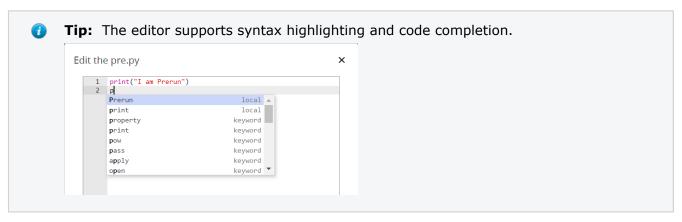
Figure 46: App Composer

3. Select **Input Template** from the drop down list.

   To create remote desktop based application definitions, select either Desktops-Linux or Desktops-Windows starter template.

4. Click 🖼 to browse and select an icon for the application definition.

5. Enter a name for the application definition.

6. Enter all the input file type extensions that will be supported by the application definition.

   Access Web links a job input file to a specific application or solver via the file's extension. This set up will determine which solvers are available for a file extension.

7. Enter the output file types that has to be associated with the application definition.

8. Click **Edit Pre** to provide a python script to run before the job is executed.
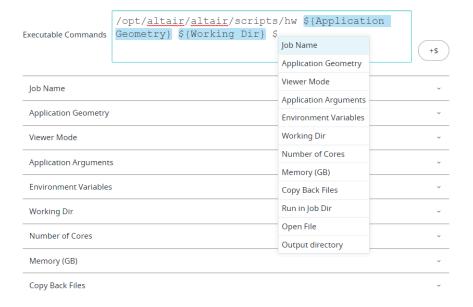
> **Tip:** The editor supports syntax highlighting and code completion.
>
> Edit the pre.py   ✕
>
> ```
> 1  print("I am Prerun")
> 2  p
> ```
> | Prerun | local |
> | print | local |
> | property | keyword |
> | print | keyword |
> | pow | keyword |
> | pass | keyword |
> | apply | keyword |
> | open | keyword |

9. Click **Edit Post** to provide a python script to run after the job is executed.

> **Tip:** The editor supports syntax highlighting and code completion.

10. Enter the executable command to run the job.

The executable command can be a basic shell scripts and multi-line shell scripts.

> **Tip:** The command box provides the list of parameters when you click ⊕+$ or when you enter the syntax $ which helps to write the executable command.

Executable Commands

```
/opt/altair/altair/scripts/hw ${Application
Geometry} ${Working Dir} $
```

| Job Name |
| Application Geometry |
| Viewer Mode |
| Application Arguments |
| Environment Variables |
| Working Dir |
| Number of Cores |
| Memory (GB) |
| Copy Back Files |
| Run in Job Dir |
| Open File |
| Output directory |

Job Name

Application Geometry

Viewer Mode

Application Arguments

Environment Variables

Working Dir

Number of Cores

Memory (GB)

Copy Back Files

11. Choose the parameters required for the application definition that are provided by the selected starter template.

You can specify if the parameters are optional or required field in the job submission form. You can also set the default values for the parameters.
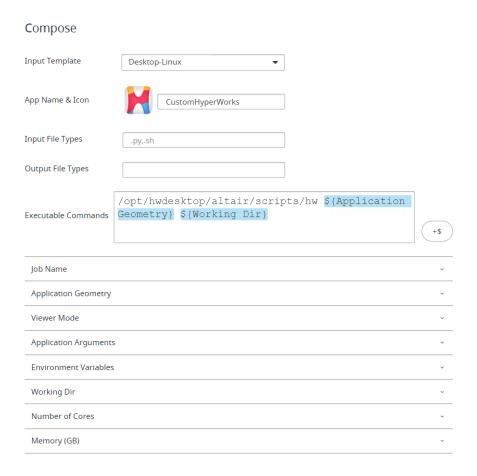
△ALTAIR

Application Geometry ⌄

Default [   ] ☑ Required

12. Click **Manage Parameters** to add or remove parameters.

Manage Parameters   ✕

Available Parameters	Selected Parameters

Selected Parameters:
- Job Name
- Application Geometry
- Viewer Mode
- Application Arguments
- Environment Variables
- Working Dir
- Number of Cores
- Memory (GB)
- Copy Back Files
- Run in Job Dir
- Open File
- Output directory

≫ ＞ ＜ ≪   ^ ⌄

Cancel   **Done**

You can move the parameters from Available Parameters to Selected Parameters using ＞ and ＜ buttons. Rearrange the order of the parameters in the Selected Parameters column using ^ and ⌄ buttons. The Job Submission form will display the parameters in the same order.

13. Verify if you have provided all the details required for the application definition.

Compose

Input Template   Desktop-Linux   ▼

App Name & Icon   CustomHyperWorks

Input File Types   .py,.sh

Output File Types

Executable Commands   /opt/hwdesktop/altair/scripts/hw ${Application
Geometry} ${Working Dir}

+$

Job Name	⌄
Application Geometry	⌄
Viewer Mode	⌄
Application Arguments	⌄
Environment Variables	⌄
Working Dir	⌄
Number of Cores	⌄
Memory (GB)	⌄

14. Click **Save**.

The Preview & Test panel on the right displays the job submission form, Job Submission command and Solver Launch Command information for the application definition we have composed.

Preview & Test
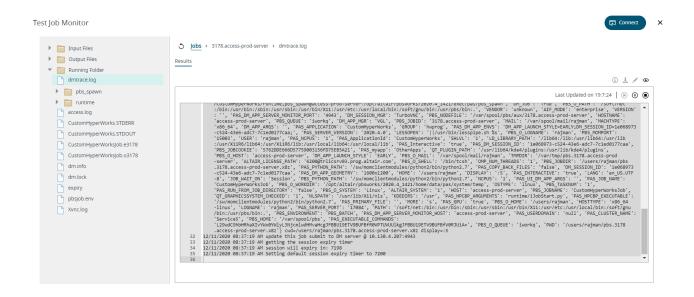
CustomHyperWorks

15. Fill in the required application or profile arguments in the Job Submission form.

16. Click **Submit & Test**.

> ⓘ   **Tip:** Click on the Job ID link provided in the notification to view the job status.

The job is submitted and a Test Job Monitor window is displayed.

△ ALTAIR
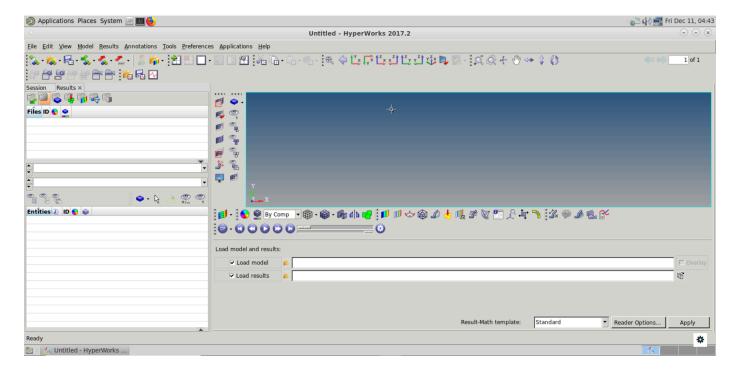
> **ℹ Tip:** From the files list on the left panel, you can select other files to view the desktops details.

17. Click **Connect**.

The desktops will be launched in a new tab.

The application definition is added to the existing application definitions list.

> 📝 **Note:** The application definitions that you have created are in JSON format and is stored at `/home/$USER/userapps`. The System Administrator can configure the path so that you can edit it in the Text Viewer.

14.4 Edit an Application Definition

Edit and save the application definitions.

The Applications panel lists all the application definitions that you have created. You can edit and save the application definitions.

1. Click 👤 .

2. Select **App Composer**.
   The Applications panel lists all the application definitions that you have created.
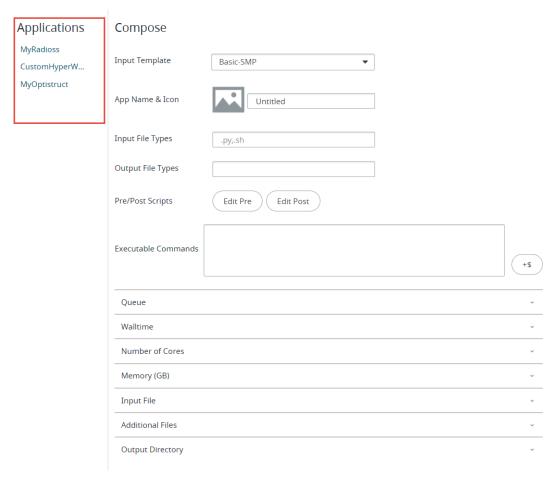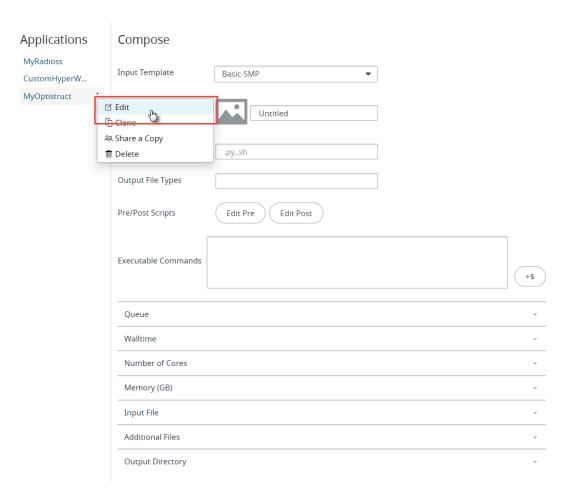
Figure 47: App Composer

3. Click ⋮ next to the application definition that you want to edit.

> 📝 **Note:** You can also click on the application definition to edit.
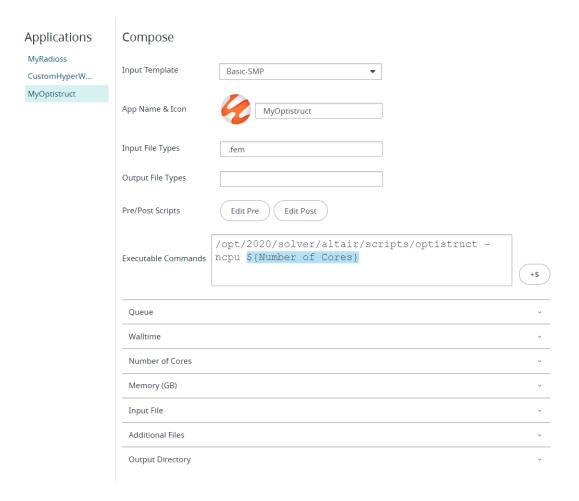
4. Select **Edit** in the context menu.

The application definition parameters are displayed in the Compose panel.

5. Update the application definition parameters as per your requirement.

6. Click **Save**.

> 📝 **Note:** The application definitions that you have created are in JSON format and is stored at `/home/$USER/userapps`. The System Administrator can configure the path so that you can edit it in the Text Viewer.

14.5 Publish a Local Application Definition

Publish a local application definition to the administrator to make it as a global application definition.

The application definition created using the app composer is a local application definition stored in your home directory. The application definitions are stored at `/home/$USER/userapps` and they are in the JSON format.

The local application definition created can be made available to all the users of Access Web by making a global application.

You must publish your local application definition to the administrator. The administrator will review the application definition and publish it all the users making it a global application definition.

1. Click 👤.

2. Select **App Composer**.
   The Applications panel lists all the application definitions that you have created.
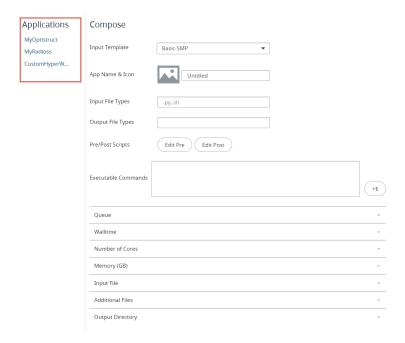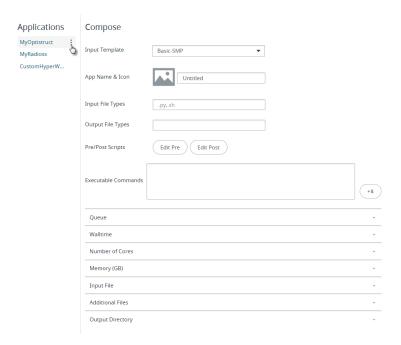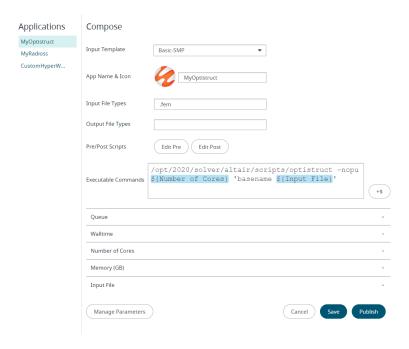
Figure 48: App Composer

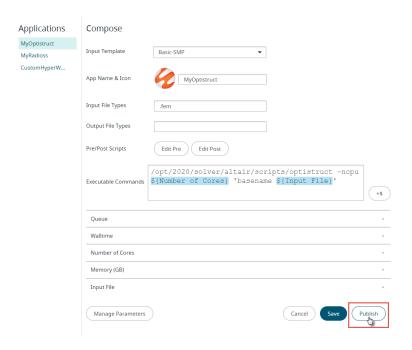3. Select the application definition that you want to publish.

The application definition parameters are displayed in the Compose panel.

Review the application definition and edit the application definition if required before publishing the application definition.
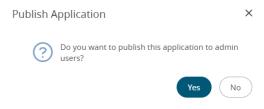
4. Click **Publish**.

Publish Application dialog box is displayed.
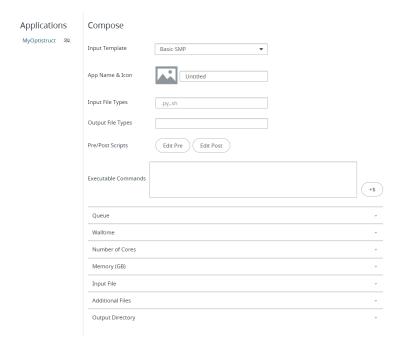
Figure 49:Publish Application

5. Click **Yes**.
The application definition will be published to the administrator and a notification will be displayed.

The app composer application list of the administrator will get updated with the application definition.

Applications

MyOptistruct

Compose

Input Template	Basic-SMP ▼
App Name & Icon	🖼 Untitled
Input File Types	.py,.sh
Output File Types	
Pre/Post Scripts	Edit Pre Edit Post
Executable Commands	+$

Queue	⌄
Walltime	⌄
Number of Cores	⌄
Memory (GB)	⌄
Input File	⌄
Additional Files	⌄
Output Directory	⌄

📝 **Note:** The administrator can click 👥 to view the user who has requested to publish the application definition.

14.6 Publish a Global Application Definition

Publish a local application definition that is shared by the user as a global application definition.

Users must publish the local application definition to the administrator to make it as a global application definition and accessible to all the users.

> 📝 **Note:** Only administrator can publish the application definition to all the users.

Administrators must save the application definition locally before publishing it to all the users.

Administrators can make the following changes to an application definition before publishing to all the users:

- View an application definition
- Edit an application definition

If there are multiple administrators, then the first administrator who saves it as local copy can publish the application definition to all the users.

> 📝 **Note:** Only the administrator who has published the application definition can remove it from the global list.

Once the application definition is published to all the user, then it becomes a global app def and it will be stored at `PA_HOME/data/pas/targets/localhost/repository/applications/` in the JSON format.

1. Click 👤 .
2. Select **App Composer**.
   The Applications panel lists all the application definitions that is created, shared and published.
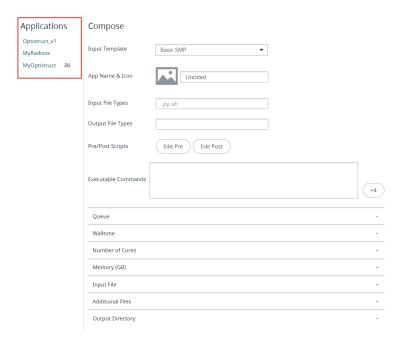
Figure 50: App Composer

The ⚎ indicates that it is a shared application definition.

> 📝 **Note:** Delete is the only option that will be available till you save the application definition locally.

3. Click ⚎ to view the user who has shared the application definition.

4. Click the application definition to save it as a local copy.
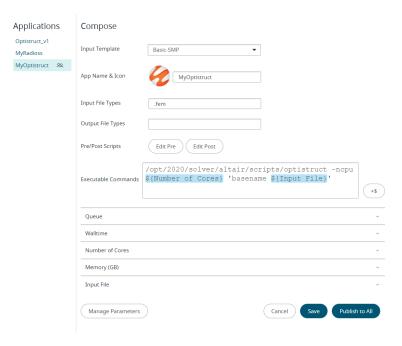The application definition parameters are displayed in the Compose panel.

Figure 51: Application Definition Published

Review the application definition and edit the application definition if required before publishing the application definition to all the users.

5. Click **Publish to All**.

The application definition will be published to all the users making it a global application definition and a notification will be displayed.
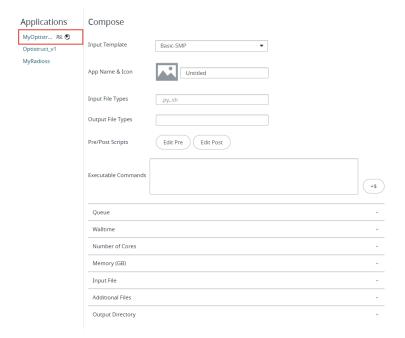

Access Web will get a notification that application defintion has been updated.

The application definition that is published will get updated in the application list with 🌐 icon.

Figure 52:Global Application Definition

Applications

MyOptistr...

Optistruct_v1

MyRadioss

Compose

Input Template	Basic-SMP
App Name & Icon	Untitled
Input File Types	.py,.sh
Output File Types	
Pre/Post Scripts	Edit Pre Edit Post
Executable Commands	+$

Queue	
Walltime	
Number of Cores	
Memory (GB)	
Input File	
Additional Files	
Output Directory	

> **Note:** The administrator cannot edit, update or view the application definition after publishing it to all the users.

14.7 Unpublish a Global Application Definition

Remove a published application definition from the global application definition list.

> 📝 **Note:** Only administrator can remove the global application definition.

The administrator who has published the application definition can remove the application definition from the global list.

1. Click 👤 .

2. Select **App Composer**.

    The Applications panel lists all the application definitions that is created, shared and published.
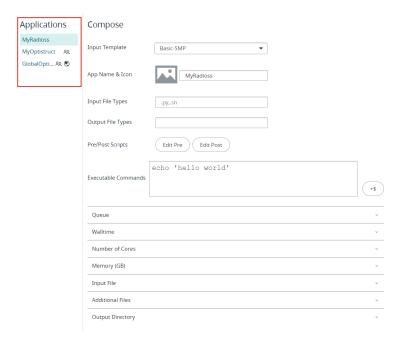
Figure 53: App Composer

The 👥 indicates that it is a shared application definition.

The 🌐 indicates that it is a published application definition which is global app def available to all the users.

3. Click ⋮ next to the application definition that you want to unpublish.
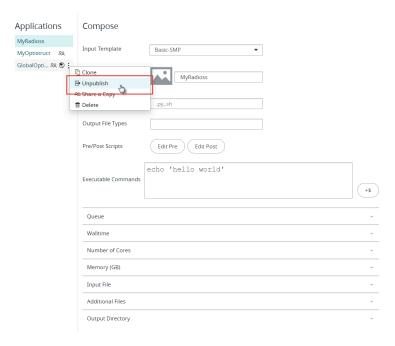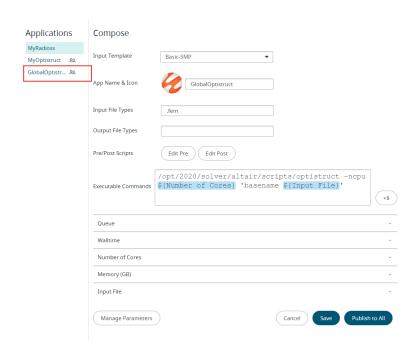
4. Select **Unpublish** in the context menu.

Figure 54: Unpublish Application Definition

A notification is displayed.

The application list is updated and

is removed.

14.8 Share an Application Definition

Share the application definition that you have created to your team members.

The Applications panel lists all the application definitions that you have created. You can share a copy of your application definition to your team members.

1. Click ⚲.

2. Select **App Composer**.

The Applications panel lists all the application definitions that you have created.
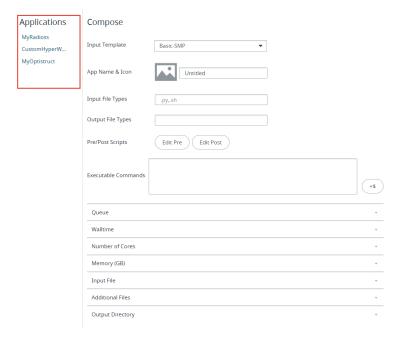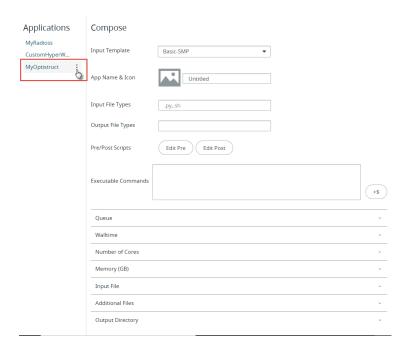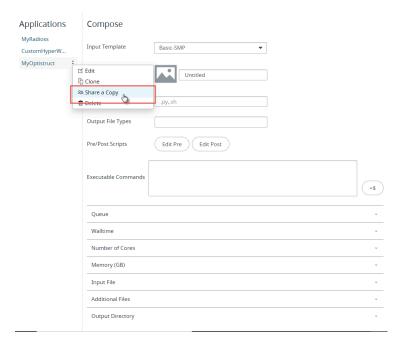

Figure 55: App Composer

3. Click ⋮ next to the application definition that you want to clone.

4. Click **Share a Copy** in the context menu.

Share Application dialog box is displayed.

Figure 56: Share Application

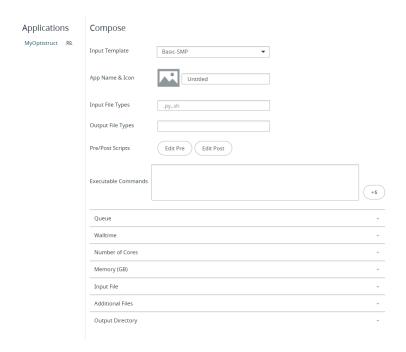**5.** Enter the user names to share the application definitions.

6. Click **Share**.
The application definition will be shared and the following notification is displayed.

The app composer application list of the user to whom it is shared will get updated with the shared application definition.

Note: The user can click to view the user who has shared the application definition.

14.9 View a Shared Application Definition

View the shared application definition and save it as a local application definition.

The Applications panel lists all the application definitions that you have created and the shared application definition. You can check the user who has shared the application definition by clicking 👥.

The application definition must be saved locally to view, edit, clone, share, and publish.

1. Click 👤.

2. Select **App Composer**.
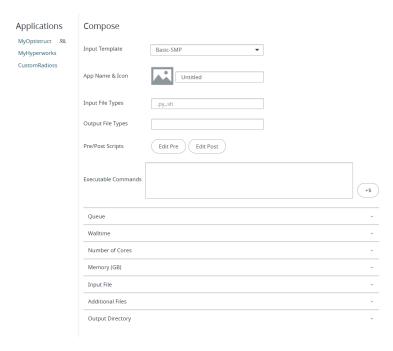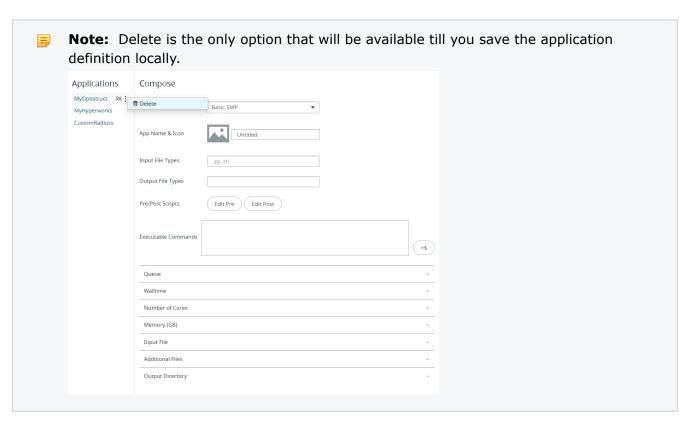   The Applications panel lists all the application definitions that you have created and shared by others.

Figure 57: App Composer

3. Click 👥 to view the user who has shared the application definition.

> **Note:** Delete is the only option that will be available till you save the application definition locally.

4. Click the shared application definition to view its parameters in the Compose panel and to save the application definition to locally.

A notification will be displayed.

MyOptistruct application is saved successfully.
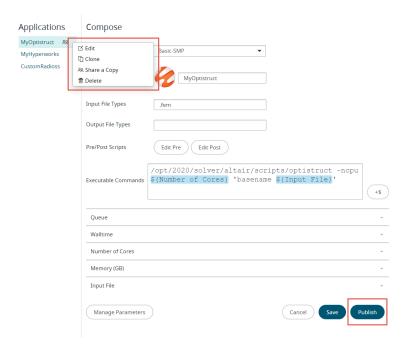
5. Click ⋮ next to the application definition to view the complete options of the application definition.

14.10 Clone an Application Definition

Clone or copy an application definition that is created.

The Applications panel lists all the application definitions that you have created. You can make a copy by cloning the application definitions.

1. Click 👤.

2. Select **App Composer**.
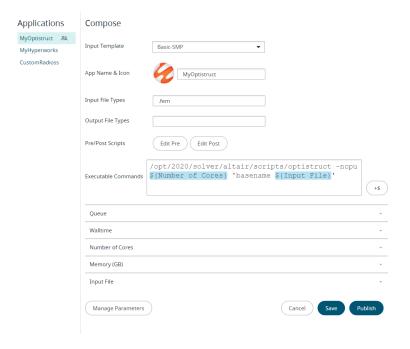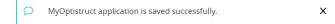   The Applications panel lists all the application definitions that you have created.

Figure 58: App Composer

3. Click ⋮ next to the application definition that you want to clone.

4. Click **Clone** in the context menu.

Clone Application dialog box is displayed.
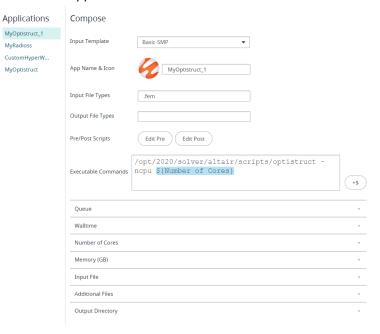
Figure 59: Clone Application

5. Enter a name for a application definition.
The cloned application definition will be displayed in the application list.

Figure 60:Cloned Application Definition

14.11 Delete an Application Definition

Delete unwanted application definition.

The Applications panel lists all the application definitions that you have created. You can delete the unwanted application definitions.

1. Click ⚇.

2. Select **App Composer**.
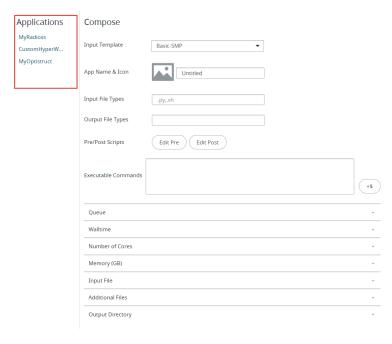The Applications panel lists all the application definitions that you have created.
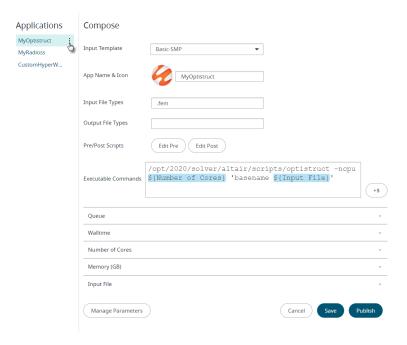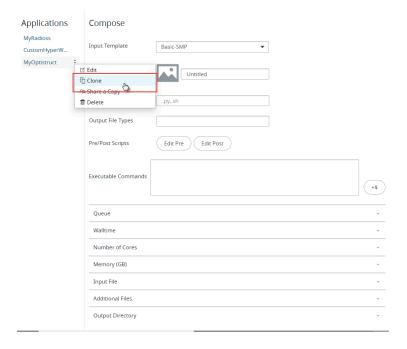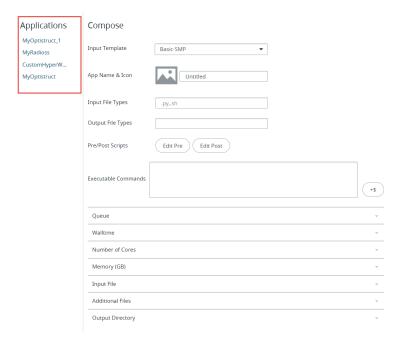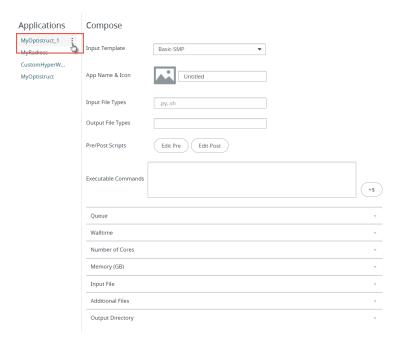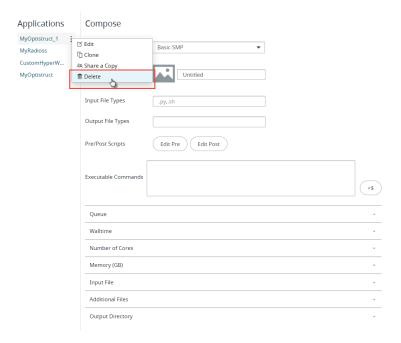

Figure 61: App Composer

3. Click ⋮ next to the application definition that you want to delete.

4. Click **Delete** in the context menu.

The application definition is removed from the list.

Establish Access Controls   **15**

Create roles and add users to these roles to grant and restrict access to the various features of Access Web.

This chapter covers the following:

Access Web roles and privileges are defined by clicking ⚲ > **Access Management** located in the upper right-hand of the web page once you have logged into Access Web. This option is only displayed for users who have been assigned the role of Manager.

15.1 Manage Roles

Add a role, change a role's privileges or delete a role.

15.1.1 Default Roles and Resources

Overview of the default Access Web roles.

Access Web Roles

By default, the only role available in Access Web is Manager. This role cannot be deleted and their assigned privileges cannot be changed, however additional users can be added to these roles to provide manager access.

Manager

    A Manager has the highest level of access privilege. Managers can add, edit, or remove clusters, can view and act upon jobs and files, and can grant and restrict access to the various features of Access Web. By default, the Service User entered during the installation of Access Web is assigned to the Manager role and cannot be removed from this role.

Privileges

By default, the following privilege levels are available:

System Administrator

    This privilege allows user management activities like Create, Modify, and Delete Users. It also offers all functions available to Portal Administrator except Management of Roles and Privileges.

Security Administrator

    This privilege allows full access to manage user access controls via roles and privileges.

Audit Administrator

    This privilege allows read-only access to Audit Logs.

Application User

    This privilege restricts application access for jobs, desktops and to the resources.

> 📝 **Note:** Portal administrator account is created at the time of deployment and by default all the roles are assigned to Portal Administrator. The portal administrator configures the appropriate roles after the deployment.

Resources

By default, the following resource levels are available:

Applications

> This privilege allows access to the applications and its respective profiles available in Access Web. It limits the users to access applications for job submission, create job profiles, use the application for remote desktop and perform custom action.

Pages

> This privilege allows access to the site-specific pages that is created and available in Access Web. It limits the users to access and view the site-specific pages.

15.1.2 Add a New Role

Create a custom role for your site.

Review the default roles and privileges before adding a new role.

1. Click ⚬ > **Access Management**.
2. Click **Roles** from the Access Management menu located on the left-hand side of the web page.
3. Click **Add Role**.
   A role is created with a default role name and undefined privileges.
4. Click ✎ located to the right of the name of the role.
   a) For Role Name, enter a name that describes the role.
   b) Click **OK**.
5. Choose **System Administrator** privilege if you want to provide full permission.
   By default, only **Application User** privilege is provided to the new Role.
6. Click **+** at the top next to the **Assigned Resources**.
   The **Available Resources** menu is displayed.
7. Choose the resources to give to this role by clicking the check-box located to the right of the resources.
   Resources are not mutually exclusive, so you may assign more than one.
8. Click **Ok** in the **Available Resources** menu.
9. Click **Save**.
   The new role is displayed in the Roles list.

15.1.3 Change the Resources of a Role

Add or remove access resources for a role.

Review the default roles and privileges before adding a new role.

1. Click ⚬ > **Access Management**.
2. Click **Roles** from the Access Management menu located on the left-hand side of the web page.
3. Click the name of the role.
   The **Assigned Resources** menu is displayed.

4. Click **+** at the top next to the **Assigned Resources**.
   The **Available Resources** menu is displayed.

5. Enable the checkbox next to the Resource to assign the resource to the role.

6. Click **Ok** in the **Available Resources** menu.

7. Click **Save**.

15.1.4 Change the Name of a Role

Rename a role.

1. Click 👤 > **Access Management**.

2. Click **Roles** from the Access Management menu located on the left-hand side of the web page.

3. Click the name of the role.

4. Click 🖉 located to the right of the name of the role.

   a) For Role Name, enter a new role name.

   b) Click **Ok**.

5. Click **Save**.

15.1.5 Delete a Role

Delete a role that is no longer needed.

1. Click 👤 > **Access Management**.

2. Click **Roles** from the Access Management menu located on the left-hand side of the web page.

3. Select a role by enabling the checkbox next to the role's name.

   > ℹ️ **Tip:** Select multiple roles so that they can be deleted in a single click.

4. Click 🗑.

5. Click **Yes**.

15.2 Manage Users

Add a user, assign or remove a role from a user, revoke access to a cluster, or delete a user.

15.2.1 Add a User

Add a user so that the user can access the features of Access Web.

1. Click ⚲ > **Access Management**.
2. Click **Users** from the Access Management menu located on the left-hand side of the web page.
3. Click **Add User**.
   a) For First Name, enter the first name of the user.
   b) For Last Name, enter the last name of the user.
   c) For User Name, enter the user's username.
   d) Click **Save**.

15.2.2 Assign a Role to a User

Assign a role to the user to establish user privileges.

1. Click ⚲ > **Access Management**.
2. Click **Users** from the Access Management menu located on the left-hand side of the web page.
3. Select a user by enabling the checkbox next to the user's name.

   > ℹ️ **Tip:** Select multiple users when you want to assign the same roles to multiple users.

4. Click **Assign Roles**.
5. Enable the checkbox next to the Role Name to assign the role to the user.

   > ℹ️ **Tip:** More than one role can be assigned to the user.

6. Click **OK**.

15.2.3 Remove a Role from a User

Remove a role from a user to limit access to certain features.

1. Click ⚲ > **Access Management**.
2. Click **Users** from the Access Management menu located on the left-hand side of the web page.
3. Click the name of the user.
   The roles assigned to the user are displayed.

△ ALTAIR

4. Click 🗑 to delete the role.

5. Click **Save**.

15.2.4 Change the User Name

Change the first or last name of a user.

1. Click ⚬ > **Access Management**.

2. Click **Users** from the Access Management menu located on the left-hand side of the web page.

3. Click the name of the user.
   The roles assigned to the user is displayed.

4. Change the first or last name of the user.

5. Click **Save**.

15.2.5 Delete a User

Delete a user when the user no longer needs access to Access Web.

1. Click ⚬ > **Access Management**.

2. Click **Users** from the Access Management menu located on the left-hand side of the web page.

3. Select a user by enabling the checkbox next to the user's name.

> ℹ **Tip:** Select multiple users so that they can be deleted in a single click.

4. Click 🗑.

5. Click **Yes**.

Access Web High Availability   **16**

Access Web provides high availability setup to ensure uninterrupted service and response time for the users.

This chapter covers the following:

16.1 Prerequisites

Prerequisites for High Availability setup.

The following are the prerequisites for setting up High Availability (HA):

- Ensure that a supported version of PBS Professional is installed.

- Hardware and VMs are available to setup nodes required for Access HA Setup.

- Network Information Service (NIS) or Authentication setup are ready between Access HA node and PBS Professional complex.

- Shared File System (SFS) using Network File System (NFS) or any alternatives among head nodes of PBS Professional Cluster and Access HA nodes.

- It is recommended to have a RAID (Redundant Array of Independent Disks), standard distributed data storage technology solution for your setup.

16.2 High Availability Infrastructure
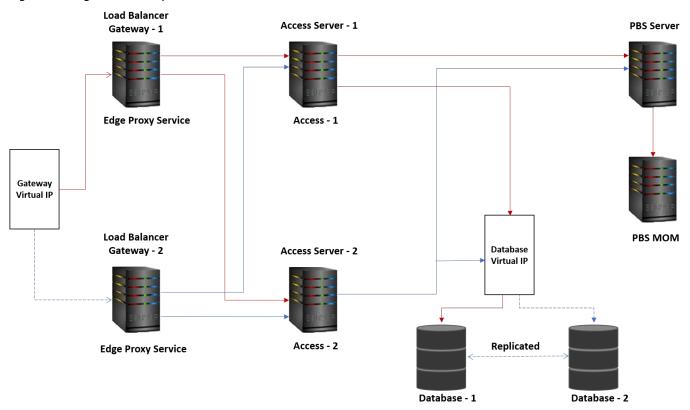
Overview of the infrastructure and hardware requirements.

Access Web High Availability is supported only on RHEL and CentOS platforms.

The infrastructure setup provided below shows failover with two nodes. You can have more than one network card in each of the node to active network failover.

The following is an example of the Access HA illustration:

Figure 62:High Availability Illustration

The description of the various components are as follows:
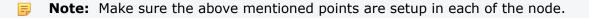
- Gateway Virtual IP – its a publicly available IP address which user will point in browser to get to Altair Access.
- Load Balancer – will have Edge Proxy Service
- Access Server – this will have Access Web and its components installed:
  - Gateway
  - Storage
  - Preferences
  - Meta Templates
  - Result Service
  - Executor

- ◦ AMS
  - ◦ PAS + PBS Client
- Database Virtual IP – its a IP address which application will point to get database access.
- Database – the information in the database will be replicated
- PBS Server – contains the following components installed:
  - ◦ PBS Server
  - ◦ NIS
  - ◦ Shared File System
  - ◦ HyperWorks
  - ◦ Compose
- PBS MOM – executable node

As described in the illustration, prepare Access HA as follows:

- short host names are provided as described in machine name column.
- each machine should be able to ping other machine using these short host names, this can be achieved by making an entry on `/etc/hosts` for each machine.
- NIS/Common authentication is configured on each of the machine.
- each node has access to shared file system under `/shared` directory.

> 📝 **Note:** Make sure the above mentioned points are setup in each of the node.

△ ALTAIR

16.3 System and Hardware Requirements

The system and hardware requirement for each of the component required for Access HA.

Hardware Requirement for Gateway 1 and Gateway 2

The hardware requirement for Gateway 1 and Gateway 2 are as follows:

Table 5: Gateway 1 and Gateway 2

Hardware Requirement	Network	Description
At least 4 Core 16 GB	10 Gbps	This is an entry point gateway

Hardware Requirement for Access 1 and Access 2

The hardware requirement for Access Web and Remote Desktops are as follows:

Table 6: Access Web

Hardware	Requirement	Description
CPU	4 CPU with a minimum speed of 2.5 GHz	
Memory (Physical)	16 GB	
Disk Space	500 GB	
Network Connectivity	10 Gigabit	

Table 7: Remote Desktops Proxy

Hardware	Requirement	Description
CPU	1 core per remote desktop and a total of 8 cores for scalability	
Memory (Physical)	16 to 32 GB	
Disk Space	100 GB	
Network Connectivity	15 to 20 Mbps per remote desktop	

Hardware Requirement for Database 1 and Database 2

The hardware requirement for Database 1 and Database 2 are as follows:

Table 8: Database 1 and Database 2

Hardware Requirement	Network	Description
At least 4 Core 16 GB	10 Gbps	This is an entry point gateway

16.4 Configure Database HA Servers

Configure database HA servers by installing database components.

The following components are required to be configure in database HA servers:

- Install Postgres
- Replicate the database
- Install and configure pacemaker
- Enable and configure firewall in the database
- Test fail over of the nodes

The prerequisites are:

- database 1 machine (For example, 10.75.32.51)
- database 2 machine (For example, 10.75.32.52)
- database floating IP (For example, 10.75.32.53) - Access components will be connecting to database using this IP

> 📝 **Note:** In case of any errors, please view PCSD status message and POSTGRES SQL logs, if applicable they will have all the required information for debugging.

1. Install Postgres 9.4 on Database 1 and Database 2 machines using the following command:

   ```
   yum install -y https://download.postgresql.org/pub/repos/yum/9.4/redhat/rhel-7-
   x86_64/pgdg-centos94-9.4-3.noarch.rpm

   yum install -y postgresql94 postgresql94-contrib.x86_64 postgresql94-server.x86_64
   # Initialize the database cluster in Database
   /usr/pgsql-9.4/bin/postgresql94-setup initdb
   # Start the database cluster in Database
   systemctl start postgresql-9.4.service
   # Enable the unit file to start PostgreSQL on boot
   systemctl enable postgresql-9.4.service
   # Stop the database cluster in Database
   systemctl stop postgresql-9.4.service
   # Change listen_address to * from localhost in vi /var/lib/pgsql/9.4/data/
   postgresql.conf
   systemctl start postgresql-9.4.service
   # Configure pg_ctl which will be later used by pacemaker
   /usr/sbin/update-alternatives --install /usr/bin/pg_ctl pgsql-pg_ctl /usr/
   pgsql-9.4/bin/pg_ctl 940
   ```

2. Create a replication user on Master.

3. Configure *pg_hba.conf* on both the database machines to allow connections from Master, and Access 1 and Access 2 machines.

4. Configure and test the database replication.

5. Install pacemaker *corosync* and *pcsd* on both Database 1 and Database 2

   ```
   yum -y install epel-release

   yum -y install corosync pacemaker pcs
   # Enable pacemaker
   systemctl enable pcsd corosync pacemaker
   # Start pacemaker
   ```

△ **ALTAIR**

```
systemctl start pcsd
# Set the password in the pacemaker cluster
passwd hacluster
```

6. Perform the following steps in Database

```
# Authenticate the pacemaker cluster to the database
pcs cluster auth database1 database2
# Setup the username to the pacemaker cluster
pcs cluster setup --name database_cluster database1 database2
# Start the pacemaker cluster services
pcs cluster start --all
# Enable the pacemaker cluster services
pcs cluster enable --all
# Check the status of the pacemaker cluster
pcs status cluster
# Disable STONITH (fencing) property
pcs property set stonith-enabled=false
```

> 📑 **Note:** By default pacemaker enables STONITH (Shoot The Other Node In The Head) /
> Fencing in an order to protect the data. Fencing is mandatory when you use the shared
> storage to avoid the data corruptions.

```
# Set the QUORUM policy to be ignored
pcs property set no-quorum-policy=ignore
# View the pacemaker property list
pcs property list
# Create a virtual IP address resource in pacemaker
pcs resource create virtual_ip ocf:heartbeat:IPaddr2 ip=10.75.32.53 cidr_netmask=24
 op monitor interval=30s
```

7. Create resource agent in pacemaker using the command:

```
pcs resource create pgsql pgsql \
    pgctl="/usr/bin/pg_ctl" \
    psql="/usr/bin/psql" \
    pgdata="/var/lib/pgsql/9.4/data/" \
    rep_mode="sync" \
    node_list="database1 database2" \
    restore_command="cp /var/lib/pgsql/9.4/archive/%f %p" \
    primary_conninfo_opt="host=10.75.32.53 user=replica password=pwd@123
 keepalives_idle=60 keepalives_interval=5 keepalives_count=5" \
    master_ip="10.75.32.53" \
    restart_on_promote='true' \
    op start   timeout="60s" interval="0s"  on-fail="restart" \
    op monitor timeout="60s" interval="4s"  on-fail="restart" \
    op monitor timeout="60s" interval="3s"  on-fail="restart" role="Master" \
    op promote timeout="60s" interval="0s"  on-fail="restart" \
    op demote  timeout="60s" interval="0s"  on-fail="stop" \
    op stop    timeout="60s" interval="0s"  on-fail="block" \
    op notify  timeout="60s" interval="0s"
pcs resource master msPostgresql pgsql
    master-max=1
    master-node-max=1
    clone-max=2
    clone-node-max=1
    notify=true
pcs resource group add master-group virtual_ip
pcs constraint colocation add master-group with Master msPostgresql INFINITY
pcs constraint order promote msPostgresql then start master-group
    symmetrical=false
    score=INFINITY
pcs constraint order demote  msPostgresql then stop  master-group
```

```
    symmetrical=false
    score=0
```

8. Enable and configure firewall:

```
# Configure firewall
yum -y install firewalld
# Start the firewall service
systemctl start firewalld
# Enable the firewall service
systemctl enable firewalld
# Add services to the firewall
firewall-cmd --permanent --add-service=high-availability
firewall-cmd --permanent --add-service=http
firewall-cmd --permanent --add-service=https
# Reload the firewall service
firewall-cmd --reload
# Check the firewall service list
firewall-cmd --list-all
# Verify if pcsd is still able to coordinate and reach out to nodes
pcs status nodes
crm_mon -Arf
```

The following node status message will be displayed in case of successful configurations:

```
[root@database2 ~]# history | grep Arf
  270  crm_mon -Arf
  320  history | grep Arf
[root@database2 ~]# crm_mon -Arf

Stack: corosync
Current DC: database2 (version 1.1.23-1.el7_9.1-9acf116@22) - partition with quorum
Last updated: Mon Feb  8 10:50:42 2021
Last change: Tue Feb  2 21:19:50 2021 by root via crm_attribute on database1

2 nodes configured
3 resource instances configured

Online: [ database1 database2 ]

Full list of resources:

 Resource Group: master-group
     virtual_ip (ocf::heartbeat:IPaddr2):        Started database1
 Master/Slave Set: msPostgresql [pgsql]
     Masters: [ database1 ]
     Slaves: [ database2 ]

Node Attributes:
* Node database1:
    + master-pgsql                     : 1000
    + pgsql-data-status                : LATEST
    + pgsql-master-baseline            : 0000000028000090
    + pgsql-status                     : PRI
* Node database2:
    + master-pgsql                     : 100
    + pgsql-data-status                : STREAMING|SYNC
    + pgsql-status                     : HS:sync

Migration Summary:
* Node database1:
* Node database2:
```

9. Test fail over of the nodes and by bringing down primary node and observing primary node is migrated to other node without any manual interventions.

10. Test fail over of the nodes and by bringing down primary node and observing primary node is migrated to other node without any manual interventions.

11. As per access section please make sure database connection is configured correctly on Access side to point to this postgres HA cluster using virtual IP.

16.5 Configure Access Servers

Configure Access servers and its components for HA.

Perform the following tasks to setup the Access servers for HA:

1. Install Access Web
2. Post Installation
3. Configure the Database
4. Configure Access Gateway
5. Setup NATS Message Broker
6. Distribute Modern Communication Module
7. Setup Application Definitions
8. Configure Results Visualization Service

16.5.1 Install Access Web

Instructions for installing Access Web.

Before you begin:

- Review system requirements of Access Web.
- Review prerequisites of Access Web.

Ensure that you have the following information before starting the installation:

- license server details
- an user account that will be the Service User
- HyperWorks location
- Compose location

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

Installation must be done as root or as a user with sudo permissions.

The Access Web 2022.1.0 installer provides an option for installing Access Web, PAS, or both. Install both Access Web and PAS on the same machine by choosing an option 3 in the Access server machine.

1. Login to the machine where Access Web/PAS is to be installed.
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. 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`. You can provide multiple license server details separated by a colon in Linux.

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 PAS and press `ENTER`.

The staging directory is where user's job files will be staged prior to execution. Provide stage directory path which is on shared file system.

9. Enter the Service User and press `ENTER`.

The default Service User is 'pbsworks' and the Access Web services are 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. Enter the HyperWorks Desktop location for visualizing results.

> 📝 **Note:** It is recommended to install Hyperworks on shared file system, so that the same installation can be accessed by multiple Access instances.

`<Hyperworks_INSTALL_DIR>/altair`

For example, `/shared/access/HyperWorks/altair`.

> 📝 **Note:** You can also configure this path after the installation by editing `PA_HOME/config/resultmanager/site_config.xml`.

13. Enter the Compose location for visualizing plot results.

> 📝 **Note:** It is recommended to install Compose on shared file system, so that the same installation can be accessed by multiple Access instances.

`<COMPOSE_INSTALL_DIR>/altair/Compose2021.2`

For example, `/shared/access/Compose/altair/Compose2021.2`.

> 📝 **Note:** You can also configure this path after the installation by editing `PA_HOME/config/resultmanager/site_config.xml`.

14. Review the installation summary and press `ENTER`.

The installation starts. It may take a few minutes for the installation to complete.

> 📝 **Note:** Access Web starts a watcher process which will monitor the status of Access Services. The watcher process automatically brings up any services that goes down abruptly.

16.5.2 Post Installation of Access Web

Post installation steps required for Access Web.

Authentication and Authorization Management System (AMS) meta information must be placed on shared disk, so multiple installations can access A&A information.

The post-installation steps to be performed are as follows:

- Copy `local-datastore` and `policies` directories from `PA_HOME/data/ams` to the shared disk.
- Remove `local-datastore` and `policies` directories from the local installation (`PA_HOME/data/ams`)
- Create softlink to the shared location
- Provide the ownership to the service user

1. Login to the machine where Access Web is installed as root or as a s user with sudo permissions.
2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:
   ```
   source /etc/pbsworks-pa.conf
   ```
3. Copy `local-datastore` and `policies` directories from `PA_HOME/data/ams` to the shared disk location.
   For example, the shared disk location can be `/shared/access/ams/`.
4. Remove `local-datastore` and `policies` directories from the local installation (`PA_HOME/data/ams`).
5. Create a softlink to the shared disk location.
   For example, create a softlink to the shared disk location `/shared/access/ams/`.
6. Execute the following command:
   ```
   ln -s /shared/access/ams/local-datastore $PA_HOME/data/ams/local-datastore
   ```
   ```
   chown -h pbsadmin $PA_HOME/data/ams/local-datastore
   ```
   ```
   ln -s /shared/access/ams/policies $PA_HOME/data/ams/policies
   ```
   ```
   chown -h pbsadmin $PA_HOME/data/ams/policies
   ```

16.5.3 Configure the Database

Update the Access Web database information to point to HA database.

Update the database information in the following files:

- `app.properties` file located at `PA_HOME/config/shared`
- `site_config.xml` file located at `PA_HOME/config/resultmanager`

1. Login to the machine where Access Web is installed as root or as a s user with sudo permissions.

2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:
   ```
   source /etc/pbsworks-pa.conf
   ```
3. Navigate to `PA_HOME/config/shared`
   ```
   cd $PA_HOME/config/shared
   ```
4. Update the value of *spring.datasource.url* and *spring.datasource.username* in `app.properties` file as follows:
   ```
   spring.datasource.url=jdbc:postgresql://<Database Virtual IP>:5432/pbsworks
   spring.datasource.username=pbsworks
   ```
5. Navigate to `PA_HOME/config/resultmanager`
   ```
   cd $PA_HOME/config/resultmanager
   ```
6. Update the property value in `site_config.xml` file as follows:
   ```
   <Property name="hibernate.connection.url" value="jdbc:postgresql://<Database
    Virtual IP>:5432/pbsworks"/>
   <Property name="hibernate.connection.username" value="pbsworks"/>
   ```

16.5.4 Configure Access Gateway

Configure Access gateway with the PAS services so that multiple installations can use a common configuration.

The following steps must be performed to configure Access gateway:

- Copy `conf.d` directory from `PA_HOME/config/api_gateway` to the shared disk
- Remove `conf.d` directory from the local installation (`PA_HOME/config/api_gateway`)
- Create softlink to the shared location
- Provide the ownership to the service user

1. Login to the machine where Access Web is installed as root or as a s user with sudo permissions.
2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:
   ```
   source /etc/pbsworks-pa.conf
   ```
3. Copy `conf.d` directory from `PA_HOME/config/api_gateway` to the shared disk location.
   For example, the shared disk location can be `/shared/access/gateway/`.
4. Remove `conf.d` directory from the local installation (`PA_HOME/config/api_gateway`).
5. Create a softlink to the shared disk location.
   For example, create a softlink to the shared disk location `/shared/access/gateway/`.
6. Execute the following command:
   ```
   ln -s /shared/access/gateway/conf.d $PA_HOME/config/api_gateway/conf.d
   ```
   ```
   chown -h pbsadmin $PA_HOME/config/api_gateway/conf.d
   ```

16.5.5 Configure NATS Message Broker

Configure the NATS message broker in clustering mode.

Update the host, port, routes in the `nats-server.config` file of Access 1 and Access 2 located at `PA_HOME/config/shared`.

1. Login to the Access 1 machine where Access Web is installed as root or as a s user with sudo permissions.

2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:

```
source /etc/pbsworks-pa.conf
```

3. Navigate to `PA_HOME/config/shared`.

```
cd $PA_HOME/config/shared
```

4. Add the clustering configuration in `nats-server.config` file as follows:

```
cluster {
    host: 0.0.0.0
    port: 5248
}
```

5. Login to the Access 2 machine where Access Web is installed as root or as a s user with sudo permissions.

6. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:

```
source /etc/pbsworks-pa.conf
```

7. Navigate to `PA_HOME/config/shared`.

```
cd $PA_HOME/config/shared
```

8. Add the clustering configuration in `nats-server.config` file as follows:

```
cluster {
    host: 0.0.0.0
    port: 5248
    routes = [
        nats://accessha37(Access-1):5248
    ]
}
```

9. Navigate to `PA_HOME/config/shared/`

```
cd $PA_HOME/config/shared/
```

10. Update *nats.broker.url* value in `app.properties`.

```
nats.broker.url=nats://accessha37.4743, nats://accessha39:4743
```

16.5.6 Execute Modern Communication Module

The Modern Communication Module is a set of python libraries that are copied to the HPC execution hosts to improve file operations on running jobs.

Once Access Web is installed, the Modern Communication Module must be distributed to the HPC execution hosts. These libraries improve file operations for job operations on running jobs, such as:

- uploading and downloading files to the running directory.

- listing files from the running directory.
- custom actions on running jobs.
- viewing results visualization animation files from the running directory.
- starting remote desktop.

The Modern Communication Module replaces PBS technology for performing these same types of operations.

The script to distribute the Modern Communication Module to all of the PBS MoM hosts is located at `PA_EXEC/joboperation/scripts/distribute_modern_comm_modules.py`.

Before running this script, ensure the following prerequisites have been met:

- The Modern Communication Module distribution script must be run as the root user.
- The script must be run from the system where PAS is installed.
- The distribution script uses PBS Client commands. Hence the PBS Client must be installed on the PAS server.
- The distribution script asks for a location (a directory on the PAS Server) into which it will copy the Modern Communication Module before distributing it to the PBS MoMs. The permissions on this directory need to allow read and execute permissions for all users. This directory must be created before running the script. Once the script is run, the Modern Communication Module is copied to this directory on the PAS Server. The directory is then copied to each execution host and placed in the same location on each execution host.
- The distribution script asks for a user account. This user account will own the directory containing the Modern Communication Module on the PBS MoMs. This user must:
  - exist on the PAS server and all of the PBS MoMs
  - be able to SSH passwordlessly from the PAS Server into each MoM so that the Modern Communication Module can be securely copied to the PBS MoM.

1. Login to the Access 1 machine where PAS is installed as the root user.
2. Source the Access Web configuration file to set up the environment variable `PA_EXEC`:
   ```
   source /etc/pbsworks-pa.conf
   ```
3. Create a directory where the distribution script can copy the Modern Communication Module before distribution to the PBS MoMs:
   ```
   mkdir -m 755 -p /access/mcm
   ```
4. Navigate to the directory `$PA_EXEC/joboperation/scripts`.
5. Execute the script to distribute the Modern Communication Module:
   ```
   python distribute_modern_comm_modules.py
   ```
6. Enter the directory that meets the stated prerequisites.

   This is where the script will copy the Modern Communication Module before distributing it to the PBS MoMs.

   For example, `/shared/access/mcm`
7. Enter a user that meets the stated prerequisites.
   For example, `pbsworks`.
   The following message is displayed:
   ```
   Do you want to override if modules already exist (Y/N). [Default: N]?
   ```

△ALTAIR

8. Choose one of the following:

   - Enter `N` or press `ENTER` (to accept the default value) to skip copying the Modern Communication Module to the PBS MoM when it already exists on the MoM.

   - Enter `Y` to overwrite the modules. The distribution script will copy the modules even if it detects that Modern Communication Module already exists on the MoM.

   The script will display the inputs provided.

9. Enter `Y` to continue with the inputs provided.

   Messages similar to the below are displayed:

   ```
   Transferring Client Modules to n2.lab.com...
   Successfully transferred to 'n2.lab.com'
   Transferring Client Modules to n1.lab.com...
   Successfully transferred to 'n1.lab.com'
   Updated PAS Configuration file successfully
   ```

10. Login to the Access 2 machine where PAS is installed as the root user.

11. Source the Access Web configuration file to set up the environment variable `PA_EXEC`:

    ```
    source /etc/pbsworks-pa.conf
    ```

12. Update the Access 1 Modern Communication Module path in the Access 2 `server.conf` file located at `PA_HOME/config/pas/conf/`

    ```
    MODERN_COMMUNICATION_SHARED_LIBS=/shared/access/mcm/momclientmodules
    ```

16.5.7 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 Altair PBS Works Support (pbssupport@altair.com).

Copy the Application Definitions and Site Configuration Files to a shared location and update the *PAS_REPO* path in all the Access instance. This will make it easy to update the application definition and site configuration file.

Similarly, update the *USER_APP_REPO* path in all the Access instance.

Perform the following steps:

- Copy `PAS_REPO` directory to a shared location (`/shared/access/repo/`)
- Configure the `PAS_REPO` path in `/etc/pbsworks-pa.conf` file of Access 1 and Access 2 installation with the shared location path
- Copy `USER_APP_REPO` directory to a shared location (`/shared/access/scratch/\$user`)
- Configure the `USER_APP_REPO` path in `/etc/pbsworks-pa.conf` file of Access 1 and Access 2 installation with the shared location path

1. Login to the machine where PAS is installed as root or as a user with sudo permissions.

2. Navigate to `/etc/pbsworks-pa.conf`.

   ```
   cd /etc/pbsworks-pa.conf
   ```

3. Update the *PAS_REPO* path with the shared location path:

```
PAS_REPO=/app/repo/
```

4. Update the *USER_APP_REPO* path with the shared location path:

```
USER_APPS_REPO=/apps/scratch/\$user/userapps
```

5. Perform step 1 to step 4 in all the access instance.

16.5.8 Configure Results Visualization Service

Configure Results Visualization Service so that multiple installations can use a common configuration.

The following steps must be performed to configure Results Visualization Service:

- Copy `data` directory from `PA_HOME/data/resultmanager` to the shared disk
- Remove `data` directory from the local installation (`PA_HOME/data/resultmanager`)
- Create softlink to the shared location
- Provide the ownership to the service user

Create a `temp` directory in the shared disk and update that location in `site-config.xml` located at `PA_HOME/config/resultmanager`.

1. Login to the machine where Access Web is installed as root or as a s user with sudo permissions.
2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:

```
source /etc/pbsworks-pa.conf
```

3. Copy `data` directory from `PA_HOME/data/resultmanager` to the shared disk location.
    For example, the shared disk location can be `/shared/access/resultmanager/data`.
4. Remove `data` directory from the local installation (`PA_HOME/data/resultmanager`),
5. Create softlink to the shared disk location.
    For example, create a softlink to the shared disk location `/shared/access/resultmanager/data`.
6. Execute the following command:

```
ln -s /shared/access/resultmanager/data $PA_HOME/data/resultmanager/data

chown -h pbsadmin $PA_HOME/data/resultmanager/data
```

7. Navigate to `PA_HOME/config/resultmanager`.

```
cd $PA_HOME/config/resultmanager
```

8. Update *HWE_RM_DATA_LOC* value in `site_config.xml` file.

```
<HWE_RM_DATA_LOC>/shared/access/resultmanager/tmp</HWE_RM_DATA_LOC>
```

16.5.9 Start Access Web

Start all Access Web services.

Access Web starts a watcher process which will monitor the status of Access Services. The watcher process automatically brings up any services that goes down abruptly.

Starting Access Web must be done as root or as a user with sudo permissions as defined in `/etc/pbsworks-pa.conf`.

1. Login to the machine where Access Web is installed.

2. Enter the following command to start Access Web:

    ```
    service pbsworks-pa start
    ```

    > 📝 **Note:** If administrators adds or removes PAS server, then all the instances of Access gateway service needs to be restarted.

3. Enter the following command to start `api_gateway`:

    ```
    /etc/init.d/pbsworks-pa restart api_gateway
    ```

△ ALTAIR

16.6 Configure Gateway HA Servers

Configure gateway HA servers by installing gateway components.

The following components are required to be configure in gateway HA servers:

- Install Epel release, Nginx, Corosync, Pacemaker and Pcsd
- setup password for user hacluster on both the nodes

The prerequisites are:

- gateway 1 server (For example, 10.75.32.48)
- gateway 2 server (For example, 10.75.32.49)
- gateway floating IP (For example, 10.75.32.50)
- access 1 server is setup and working
- access 2 server is setup and working

1. On Gateway 1 and Gateway 2 machines perform below steps to install EPEL release , Nginx web server, Corosync, Pacemaker and Pcsd. After installation we will setup same password for user hacluster on both the nodes.

   ```
   # Install EPEL repository package in Gateway 1 and Gateway 2
   yum -y install epel-release
   # Install Nginx web server, Pacemaker, Corosync, and Pcsd
   yum -y install nginx corosync pacemaker pcs
   # Enable pacemaker
   systemctl enable pcsd corosync pacemaker
   # Start pacemaker
   systemctl start pcsd
   # Set the password in the pacemaker cluster
   passwd hacluster
   Enter new password:
   ```

2. On Gateway 1 machine perform below steps to authorize Gateway 1 and Gateway 2 nodes to a pcs cluster called gateway_cluster

   ```
   # Authenticate the pacemaker cluster to the gateway
   pcs cluster auth gateway1 gateway2
   # Setup the username to the pacemaker cluster
   pcs cluster setup --name gateway_clustergateway_cluster gateway1 gateway2
   # Start the pacemaker cluster services
   pcs cluster start --all
   # Enable the pacemaker cluster services
   pcs cluster enable --all
   # Check the status of the pacemaker cluster
   pcs status cluster
   # Disable STONITH (fencing) property
   pcs property set stonith-enabled=false
   ```

   > 📝 **Note:** By default pacemaker enables STONITH (Shoot The Other Node In The Head) / Fencing in an order to protect the data. Fencing is mandatory when you use the shared storage to avoid the data corruptions.

   ```
   # Set the QUORUM policy to be ignored
   pcs property set no-quorum-policy=ignore
   # View the pacemaker property list
   pcs property list
   ```

```
# Create a virtual IP address resource in pacemaker
pcs resource create virtual_ip ocf:heartbeat:IPaddr2 ip=10.75.32.50 cidr_netmask=24
 op monitor interval=30s
# Create pacemaker resource
pcs resource create webserver ocf:heartbeat:nginx configfile=/etc/nginx/nginx.conf
 op monitor timeout="5s" interval="5s"
# Check the pacemaker resource status
pcs status resources
# Add webserver virtual IP in pacemaker
pcs constraint colocation add webserver virtual_ip INFINITY
# Constraint pacemaker virtual IP
pcs constraint order virtual_ip then webserver
# Stop the pacemaker cluster
pcs cluster stop --all
# Start the pacemaker cluster
pcs cluster start --all
```

3. On gateway1 and gateway2

```
# Configure firewall
yum -y install firewalld
# Start the firewall service
systemctl start firewalld
# Enable the firewall service
systemctl enable firewalld
# Add services to the firewall
firewall-cmd --permanent --add-service=high-availability
firewall-cmd --permanent --add-service=http
firewall-cmd --permanent --add-service=https
# Reload the firewall service
firewall-cmd --reload
# Check the firewall service list
firewall-cmd --list-all
# Verify if pcsd is still able to coordinate and reach out to nodes
pcs status nodes
```

4. Configure *https* certificate on both gateway servers and configure routes to access1 and access2 servers.

   A sample configuration will look like this:

```
#disable request body size check
client_max_body_size 0;

upstream altairaccess {
    least_conn;
    server access1:4443;
    server access2:4443;
}

server {
    listen       443 ssl http2 default_server;
    listen       [::]:443 ssl http2 default_server;
    server_name  _;
    root         /usr/share/nginx/html;

    ssl_certificate "/etc/nginx/certs/nginx.crt";
    ssl_certificate_key "/etc/nginx/certs/nginx.key";
    ssl_session_cache shared:SSL:1m;
    ssl_session_timeout  10m;
    ssl_ciphers HIGH:!aNULL:!MD5;
    ssl_prefer_server_ciphers on;

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;
```

![ALTAIR logo]

```
        location / {
            proxy_http_version 1.1;
            proxy_set_header Upgrade $http_upgrade;
            proxy_set_header Connection "upgrade";
            proxy_set_header   X-Real-IP $remote_addr;
            proxy_set_header   Host       $http_host;
            proxy_pass https://altairaccess;
        }

        error_page 404 /404.html;
        location = /404.html {
        }

        error_page 500 502 503 504 /50x.html;
        location = /50x.html {
        }
}
```

Troubleshoot Issues 17

Troubleshooting tips and tricks.

This chapter covers the following:

17.1 Use the Diagnosis Script to Troubleshoot Issues

Use the diagnosis script to gather logs and system data to help troubleshoot issues with Access Web.

The diagnosis script must be run as root or as a user with sudo permissions using the `sudo` command.

The diagnosis script will create a zip archive that can be shared with Altair support team for troubleshooting issues.

> 📝 **Note:** If PAS is installed on a separate machine, run the diagnosis script on the PAS server as well.

The zip archive contains the Access Web configuration folder, logs folder and system information as mentioned below:

Table 9: pa-diagnosis Zip Archive Contents

Folder or File Name	Description
config folder	Access Web configuration folders
logs folder	Access Web components logs folders
cpuinfo	CPU information
diskinfo	Disk information
meminfo	Memory information
os	Operating System information
pbsworks-pa.conf	Access Web configuration file
process	Process information running from `PA_EXEC` location.
desktop-diagnosis	Remote Desktops diagnosis information
selinuxstatus	SE Linux status information
topoutput	The top command is used to show the Linux processes. It provides a dynamic real-time view of the running system. It shows the summary information of the system and the list of processes or threads which are currently managed by the Linux Kernel.
VERSION	Access Web Version and Build information

△ **ALTAIR**

1. Login to the machine where you have installed Access Web or PAS server.

2. Navigate to `PA_EXEC/init/`

3. Run the following command:

   ```
   ./pa-diagnosis.py
   ```

   A zip archive is created in `/tmp` called `pbsworks-pa-diagnosis_DATETIMESTAMP.zip` where *DATETIMESTAMP* is the file creation timestamp in the format YYYYMMDD-HHMMSS.

17.2 Produce Log Summary Report

Execute log summary report to analyze and troubleshoot the logs generated with Access Web.

The log analysis report can be shared with Altair support team for troubleshooting issues.
After execution of the script, the Analysis folder is created in the logs folder that includes a `Globalsummary_YYYYMMDDHHMMSS.csv` file. This file includes all logs that are categorized based on the errors and the information collected. The *YYYYMMDDHHMMSS* is the file creation timestamp.

The global summary report is generated at `$PA_HOME/logs/Analysis`.

1. Login to the machine where Access Web is installed as root or as a user with sudo permissions.
2. Source the Access Web configuration file to set up the environment variables `PA_HOME` and `PA_EXEC`:
    ```
    source /etc/pbsworks-pa.conf
    ```
3. Navigate to `PA_EXEC/init`.
    ```
    cd $PA_EXEC/init/
    ```
4. Run the following command:
    ```
    python LogAnalysis.py
    ```
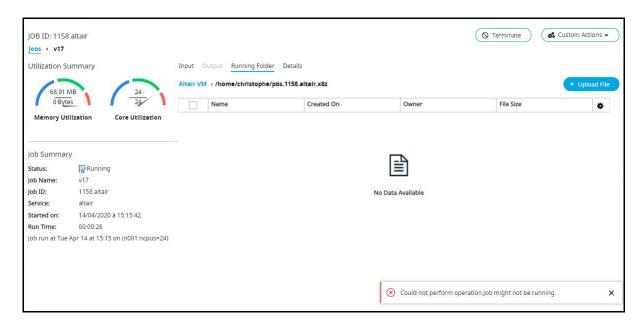   The global summary report is generated at `$PA_HOME/logs/Analysis`.

17.3 Troubleshoot Access Web

Troubleshooting information and steps for Access Web.

17.3.1 Unable to View the Job Files for a Running Job

Condition

After submitting a job, I am unable to view the job files for the running job and the following message is displayed `Could not perform operation. Job might not be running.`

The following error is displayed in the PAS server log:

```
ERROR com.altair.pas.joboperations.messagebroker.NATSServices - No Response
message received from mom. Could not perform operation.Job might not be running.
```

Cause

If the Access Web Server is not reachable or resolvable from PBS execution hosts, then the job directory will not be visible and no file listing will be seen from the jobs running directory.

Remedy

Verify that the Access Web Server is reachable and its hostname resolvable from the HPC complex execution hosts. Additionally, verify that all ports used for external communication are opened.

See Also
Update Access Web Server Hostname
Open Ports

ALTAIR

17.4 Troubleshoot PBS Application Services

Troubleshooting information and steps for PAS.

17.4.1 PAS Status Page

Use the PAS status page to get information about PAS such as JVM data, memory and disk usage data, and PBS information.

A status page is available through the following URL to monitor the status of PAS:

`https://HOSTNAME:5243/pas/pasStatus`

Where *HOSTNAME* is the hostname of the machine where the PAS Server is installed.

PAS Status at time: 3:18:51 PM

JVM Platform OS Name (system property os.name)	Linux - OK
JVM Platform OS Version (system property os.version)	3.10.0-1062.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	11.0.11 - OK
Python	OK
Dependent shared objects	OK
Available heap memory in MB	629 MB - OK
Memory used by AIF server in MB	343 MB - OK
PAS_EXEC directory disk space.	OK
PAS_HOME directory disk space.	OK
PBS status	PBS server is Running - OK
PBS Version	2021.1 - OK
PBS execution speed	OK

Figure 63: PAS Status Page

Green indicates that the system is functioning properly. Red indicates an issue that should be investigated.

For the PBS execution speed, PAS retrieves the PBS version by executing `wrapper_get_PBS_version.py` located in `PA_EXEC/pas/scripts/wlmadaptor/pbs`. If the response takes more than 5 seconds, the execution speed is considered slow and will be displayed in red.

△ ALTAIR

17.4.2 PAS Log File Contains OutofMemory Errors

Condition

There are `OutOfMemoryErrors` in the PAS Server log.

Cause

Usually, this error is thrown when the Java Virtual Machine cannot allocate an object because it is out of memory.

Remedy

Adjust the Java Virtual Machine (JVM) heap size of PAS.

Out of memory errors can indicate a underlying problem, therefore it is recommended to report these errors to the Altair support team.

See Also

Change the PAS Service Memory Value

17.4.3 Verify the Installation of PBS Application Services

Verify PAS installation and server status.

1. Open any of the supported browsers.
2. Enter URL (`https://<HOSTNAME>:<PORT>/pas`)

   Where *HOSTNAME* is the hostname of the machine where PAS is installed and *PORT* is the port that PAS listens on.

   > 📝 **Note:** The default port is 5243.

   The browser will display the PAS information.

   Messages similar to the following are displayed:

PBSWorks Application Services

Version: 2022.1.0.0
Build: 20220207

REST Services

17.4.4 Troubleshoot PAS Job Submission Issues

Information about troubleshooting PAS job submission issues.

Change the Logging Level to Troubleshoot Job Submission Issues

Change the PAS logging level to get fine-grained information that is more useful to debug a job submission issue.

> 📝 **Note:** To gather debugging information for other functional areas of PAS, see Logging Behavior.

To troubleshoot errors during job submission and job monitoring, relevant information must be collected:

- user input provided for the job submission
- submission environment
- status (success or failure) of all the dependencies
- steps which happen during the job submission
- job submission attributes generated by the PAS Server for the workload manager (PBS)

Follow these steps to gather relevant data:

1. Check the PAS server log `PA_HOME`/logs/pas/pas-server.log for errors.
2. Check the Tomcat log file `PA_HOME`/logs/pas/catalina.out for network or security errors.
3. Check the system logs.

   > 📝 **Note:** For advanced debugging, contact the Altair support team.

If you cannot determine the cause of the issue after checking the PAS Server log files, the Tomcat log files and the system logs, change the PAS logging level to get fine-grained information that is more useful to debug a job submission issue.

4. Navigate to `PA_HOME/config/pas/conf`.
5. Change the value of `rootLogger.level` from *info* to `debug` in `server-log.properties` file.
   ```
   # Configure root logger
   rootLogger.level = info
   ```
6. Add the following lines to the `server-log.properties` file to get detailed logging information about user inputs and to see how long it takes to execute a submission request:
   ```
   logger.rjpi.name=com.altair.gw.aif.rest.RESTJobsPortImpl
   logger.rjpi.level=debug
   ```
7. Add the following lines to the `server-log.properties` file to get detailed logging information about the process of creating PBS job attributes from user inputs as well as information about the application definition:
   ```
   logger.pngju.name=com.altair.gw.aif.rest.util.PASNextGenJobUtils
   logger.pngju.level=debug
   ```

   PAS supports multiple adapters to communicate with the workload manager.

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

    ```
    logger.sshi.name=com.altair.gw.aif.ssh.implementation.SSHImplementation
    logger.sshi.level=debug
    ```

9. Edit the PAS job script `PA_EXEC/pas/scripts/wlmadaptor/pbs/job.py`.

10. Enable debug logging 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.

11. Restart Access Web by entering the following command:

    ```
    service pbsworks-pa restart
    ```

12. Resubmit the job.

13. Check the log files described in steps 1 through 3.

14. Check the job output file for information about the job process arguments and the job environment.

If this troubleshooting steps do not provide enough information to debug the issue, then submit a job script directly to PBS.

Troubleshoot Job Submission Issues by Submitting a Job Directly to PBS

Debug job submission issues by submitting the job script directly to PBS.

Submit a job directly to PBS to determine if the problem is a PBS issue or a PAS issue.

1. Edit the file *PA_HOME*`/config/pas/conf/server.conf`.

2. Enable debug mode for the PAS Server by setting *DEBUG* to `true`.

    ```
    DEBUG=TRUE
    ```

3. Restart Access Web by entering the 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.
    If the job runs successfully with no errors, then the problem is originating from PAS.

Job Fails With "Bad UID for job execution" Error

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 error is displayed when jobs are submitted by root. If the failed job was not submitted by root, then the PBS Professional *flatuid* may be set to False.

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.

1. Login to the PBS Server as root or a user with sudo permissions.

2. At the command line, enter the command:

    `qmgr`

3. Enter the command:

    `print server`

4. 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:

    `set server flatuid = True`

Job Fails With Unknown Resource Error

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 the job is 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.

1. Login to the PBS Server as root or a user with sudo permissions.

2. Edit the PBS resource definition file `PBS_HOME`/`server_priv/resourcedef`.

3. Add the resource definitions to the `resourcedef` file.

4. Save the file.

5. Restart the PBS Professional server.

6. At the command line, enter the command:

    `qmgr`

7. Enter the command:

    `print server`

8. 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:

    `set server flatuid = True`

17.4.5 Troubleshoot Issues During the Installation of PAS

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.

PBS Professional Features are Not Working after Upgrade

Condition

I have upgraded PBS Professional to the newest version and now the new features are not working.

Cause

Whenever PAS starts, it saves PBS configuration information in memory. After PBS Professional is upgraded, the new configuration information is not available to PAS.

Remedy

After installing a new version of PBS Professional, you must restart PAS.

△ ALTAIR

1. Login to the PAS Server.
2. Restart PAS.

17.5 Troubleshoot Remote Desktops Components

Troubleshoot problems related to Remote Desktops and interactive applications.

17.5.1 Remote Desktops Precheck Diagnosis Script

A script that captures Remote Desktops information to help troubleshoot issues in PBS MoM.

Name

`remotesession-precheck.py`

Description

Capture information about GPU Nodes and X Server in PBS MoM before installing Remote Desktops. This information will be useful to troubleshooting issues.

Running the Script

This command must be executed as root.

This diagnosis script must be run on all the PBS MoM.

Python 2.4 or later is required to run the script.

Script Location

Download or obtain the Remote Desktops precheck diagnosis script folder (`remotesession_precheck`) using your usual Altair support channels.

The Remote Desktops precheck diagnosis script folder will have to be copied to all the PBS MoM to troubleshoot those machines.

Run the `remotesession-precheck.py` script that is available in the `remotesession_precheck` folder.

Output on the Machine Hosting the PBS MoM

After running the diagnosis script on the machine hosting the PBS MoM, messages similar to the below are displayed. Information that is of interest in this output:

Script Location
> Provides the path of the script location.
> ```
> Script Location: /tmp/remotesession-utilityscript-master/remotesession_precheck
> ```

Log Location
> Location of the log file is provided for troubleshooting. A separate directory is created each time when the script is run and the logs are stored in that directory.
> ```
> Log Location: /tmp/remotesession-utilityscript-master/remotesession_precheck/
> scratch/scratch_20190628140246
> ```

GPU Harware
> Print the configured GPU hardware if available.
> ```
> GPU Hardware:
> ```

```
     - output: 00:01.3 Non-VGA unclassified device: Intel Corporation 82371AB/EB/MB
 PIIX4 ACPI (rev 08)

00:03.0 VGA compatible controller: Amazon.com, Inc. Device 1111

00:1e.0 3D controller: NVIDIA Corporation TU104GL [Tesla T4] (rev a1)
```

X Server Status

Provides the status of X Server.

```
X Server status: Running
```

NVDIA Driver Installation Check

Check the NVDIA driver installation status.

```
- Nvidia driver is installed! Make sure latest driver is installed.
```

Desktop Manager Environment

Status about Desktop Manager environment is provided.

```
- Desktop Manager Environment installed:

- output: /usr/share/xsessions/1-kde-plasma-standard.desktop:Exec=/usr/bin/
startkde
/usr/share/xsessions/2-kde-plasma-safe.desktop:Exec=/usr/bin/startkde --failsafe
/usr/share/xsessions/gnome-classic.desktop:Exec=env
 GNOME_SHELL_SESSION_MODE=classic gnome-session --session gnome-classic
/usr/share/xsessions/gnome.desktop:Exec=gnome-session
/usr/share/xsessions/mate.desktop:Exec=mate-session
/usr/share/xsessions/xfce.desktop:Exec=startxfce4
/usr/share/xsessions/xinit-compat.desktop:Exec=/usr/libexec/xinit-compat
```

Complete Output of the Precheck Diagnosis Script

The complete display of the Precheck diagnosis script:

```
- Script Location: /home/ec2-user

- Log Location: /home/ec2-user/scratch/scratch_20220125141455

- GPU Hardware:
     - output: 00:01.3 Non-VGA unclassified device: Intel Corporation 82371AB/EB/MB
 PIIX4 ACPI (rev 08)
00:03.0 VGA compatible controller: Amazon.com, Inc. Device 1111
00:1e.0 3D controller: NVIDIA Corporation TU104GL [Tesla T4] (rev a1)

- Nvidia driver is installed! Make sure latest driver is installed.

- Desktop Manager Environment installed:
      - output: /usr/share/xsessions/1-kde-plasma-standard.desktop:Exec=/usr/bin/
startkde
/usr/share/xsessions/2-kde-plasma-safe.desktop:Exec=/usr/bin/startkde --failsafe
/usr/share/xsessions/gnome-classic.desktop:Exec=env GNOME_SHELL_SESSION_MODE=classic
 gnome-session --session gnome-classic
/usr/share/xsessions/gnome.desktop:Exec=gnome-session
/usr/share/xsessions/mate.desktop:Exec=mate-session
/usr/share/xsessions/xfce.desktop:Exec=startxfce4
/usr/share/xsessions/xinit-compat.desktop:Exec=/usr/libexec/xinit-compat

Congratulation! System look good for remote session.
```

17.5.2 Troubleshoot a Connection Error

Condition

After submitting an interactive job, job is running but not able to connect to the session.

Cause

This may be caused when:

- the graphics node is not reachable.
- problem with the X Server startup, encoder, or agent proxy

General Troubleshooting Steps

1. Login to the Access Web portal.
2. Check for errors in the encoder.log, proxy.log, xserver.log, and access.log available in Job Directory.

Remedy - Verify that the Agent Proxy Port Number is Not Blocked through the Firewall

1. Login to the graphical PBS MoM.
2. View the information in the `info` file.

   This file is located in the interactive application's job's execution directory on the graphic node.

   The below lines display the IP address and port on which the agent proxy server is listening.
   ```
   wshost=172.31.35.217
   wsport=4944
   ```
3. Login to the Access Web server or to edge proxy.
4. Attempt to connect to the agent proxy server through the agent proxy server IP address (wshost) and port (wsport):
   ```
   telnet <IP address> <port>
   ```
5. If a connection cannot be established:
   a) Login to the graphical PBS Mom.
   b) Open the port through the firewall to allow access to the agent proxy server from the Access Web server.

17.5.3 Desktop Manager Is Not Displaying

Condition

I have opened a Remote Desktops and the application is displayed, but I cannot see the Desktop Manager.

Cause

- The Desktop Manager is not installed.
- The interactive application is not configured for the installed Desktop Manager.

△ ALTAIR

- A graphic card compatibility issue.

Remedy - General Troubleshooting Steps

1. Login to the Access Web portal.

2. Check the `xserver.log` and the `desktopManager.log` files for errors.

    These files are located in the interactive application's job's execution directory on the graphic node where the job is running.

Remedy - Verify that a Desktop Manager is Installed

Follow the below steps to determine if a Desktop Manager is installed on the PBS MoM:

1. Login to the PBS MoM where the interactive application is running as root or a user with sudo permissions.

2. Perform Remote Desktops Precheck Diagnosis Script and verify if configured Desktop Manager is installed.

3. If a Desktop Manager has not been installed, then install a Desktop Manager such as GNOME, MATE, or KDE.

    If you install a different Desktop Manager, then you will have to edit all of the interactive application definitions to reconfigure them to use a different Desktop Manager. Refer to Configure the Desktop Manager for Interactive Applications for more information.

See Also

Graphic Card Compatibility Issues

17.5.4 Graphic Card Compatibility Issues

Condition

Several conditions may be observed that may be caused by graphic card compatibility issues:

- Job is failing and agent encoder is failing.

Cause

- A supported graphic card is not installed.
- The wrong drivers have been installed for the graphics card.
- Full 3D acceleration is not enabled by the graphic card drivers.
- X Server is not correctly configured with GPU hardware.

Remedy

1. Ensure that the prerequisites are met.

2. The X Server is 3D accelerated.

    If your GPU is not supported, then switch to CPU encoding. Refer to Change Remote Desktops to Use CPU Encoding for more information.

17.5.5 Interactive Application Job is in a Wait State

Condition

After submitting a job to start a remote desktop, the job has gone into a wait state.

Cause

There is an issue at the PBS Professional level, for example a file transfer issue.

Remedy

1. Login to the PBS Server as root or a user with sudo permissions.
2. Check the PBS Server logs for any errors related to the job.

17.5.6 Interactive Application Job is in a Queued State

Condition

After submitting a job to start a remote desktop, the job has gone into a queued state.

Cause

- ndesktops are currently not available to run the job.
- An insufficient amount of ndesktops are available to run the job.

Remedy

1. Login to the PBS Server as root or a user with sudo permissions.
2. Check the PBS Server logs for any errors related to the job.
3. Execute the command:
   ```
   qstat -xf <jobid>
   ```
4. View the *comment* parameter at the end of the `qstat` command's output.

   The following error indicates that there are not enough ndesktops available to run this job currently. The job remains in the queued state until ndesktops are free to run the job.
   ```
   Not Running: Insufficient amount of resource: ndesktops
   ```

   The following error indicates that there is an insufficient amount of GPUs to run the job. For example, the job requests 2 GPUs and there is only a single GPU in the cluster or there are no execution nodes having a ndesktops.
   ```
   Can Never Run: Insufficient amount of resource: ndesktops
   ```

17.5.7 Interactive Application Job Fails

Condition

After submitting a job to start a remote desktop, the job fails.

Cause

Interactive jobs may fail because:

- there is an application specific error.
- there is an issue with the associated application definition, such pointing to the wrong application path.
- the Remote Desktops component is not installed on the PBS MoM.

Remedy

Resubmit the job but enable the feature that copies back all job files to the job's output directory.

1. Login to Access Web portal.
2. Navigate to Job Details page.
3. Once the job fails, view the following files in the job's Output tab to help diagnose the problem.

    - `access.log`
    - `<jobname>.e<jobid>`
    - `<jobname>.o<jobid>`
    - `*.log`

17.6 Troubleshoot Results Visualization Service

Troubleshooting information and steps for RVS.

The following section provides the information about troubleshooting information and steps for RVS.

17.6.1 Checklist for Troubleshooting

Access should be installed in a supported operating system. A supported web browser should be used to visualize results. 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 Results Visualization System Requirements.

   > 📄 **Note:** Turn off pop-up blockers in your browser to view the progress of auto-refreshing plots.

2. On Linux platform, install these library packages for Compose and Altair HyperWorks Desktop.
   a) For SLES 12 and SLES 15, install the library `libgomp` using the command `zypper install libgomp` and the library `libXss` using the command `zypper install libXss`.

3. Ensure HyperWorks Desktop and Compose is installed in a location where all the Access Web users have read and write permissions.

   These are recommended to install in the `Home` location, which enables read permission to all the Access Web users.

4. The Access Web user should have read access to results files. This indicates the user's read access to the PBS MOM's execution directory (the staging directory set during the installation) to view the running job result files.

5. To check if there any of the HyperViewTrans processes that are running, use the command:
   ```
   ps -ef | grep hvtrans
   ```
   The list of HyperViewTrans process ids is displayed.

6. To check if there any of the Compose processes that are running, use the command:
   ```
   ps -ef | grep composeserv
   ```
   The list of Compose process ids is displayed.

7. To kill a process id, specify the `process id` in the following command:
   ```
   kill -9 <process id>
   ```

△ **ALTAIR**

17.6.2 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 Results Visualization Service Features

Unable to view or perform any of the RVS options.

The causes could be:

- RVS Options are Not Visible when a Result File is Right-Clicked

RVS Options are Not Visible when a Result File is Right-Clicked

RVS options are not available when attempting to view or create a plot or animation.

Condition

I am attempting to create or view a plot or animation by right-clicking a results file and the RVS options are not visible on the sub-menu.

Cause

The results file type may be unsupported or the result file type reader may need to be activated.

Remedy

Check that the file is a supported results file type.

- If the results file type is supported:
    1. Check if the result file type is associated to one of the following solvers: Abaqus, CFX, Fluent, or STAR-CCM+.
    2. If the result file type is associated to one of the above solvers, activate the solver file reader.
- If the result file type is not supported, then post-processing by RVS is not supported.

Compose License Error

Unable to post process the result files due to Compose license error.

Condition

Compose licenses are not available in the license server.

Cause

I receive the following error while plotting a result file: `License for Compose is required for post processing of results and is not available in the License Server`.

Remedy

Follow the given steps:

1. Login to Access Web server.

2. Navigate to RVS license configuration file location at `PA_HOME/config/license/app.properties`

3. Use the Altair license server hostname.

4. Ping the hostname of the license server to determine if it is pingable.

5. If the server responds to the ping, navigate to the license file location at: `/usr/local/altair/licensing14.5.1`

6. Verify the validity dates of Compose and HyperViewTrans licenses.
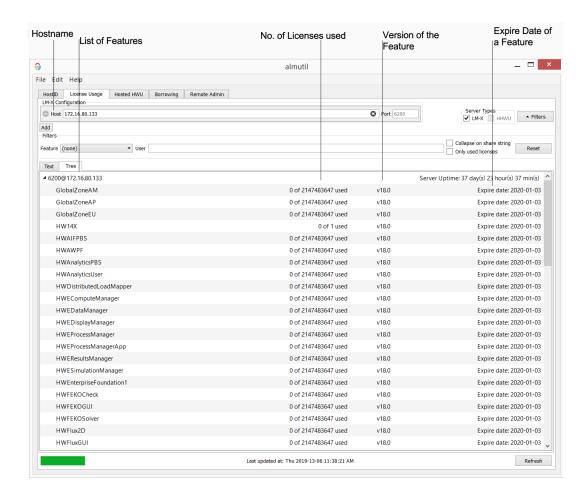
Cause

`License for Compose is required` indicates Compose feature is not available or insufficient licenses are available.

Remedy

Verify the details of licenses using Altair Utility:

1. Download and install the Altair License Utility Software.
   A shortcut, **Altair License Utility** is displayed.

2. Double-click and launch the utility.

3. Enter the Altair License server host name.
   The utility will provide the list of features with version, number of licenses used and unused, and expiry date of the feature.

Cause

If the error still persist even after the above checks, verify if there are any Compose process ids running. Kill the processs id and try again.

Remedy

Follow the given steps:

1. To check if there any of the Compose processes that are running, use the command:

```
ps -ef | grep composeserv
```

   The list of Compose process ids is displayed.

2. To kill a process id, specify the `process id` in the following command:

```
kill -9 <process id>
```

HVTrans License Error

Unable to post process the result files due to HyperViewTrans license error.

Condition

HWHyperViewTrans licenses are not available in the license server.

Cause

I receive the following error while viewing an animation result file: `License for HyperViewTrans is required for post processing of results and is not available in the License Server.`

Remedy

Follow the given steps:

1. Login to Access Web server.
2. Navigate to RVS license configuration file location at `PA_HOME/config/license/app.properties`
3. Use the Altair license server hostname.
4. Ping the hostname of the license server to determine if it is pingable.
5. If the server responds to the ping, navigate to the license file location at: `/usr/local/altair/licensing14.5.1`
6. Verify the validity dates of Compose and HyperViewTrans licenses.

Cause

`License for HVTrans is required` indicates HyperViewTrans feature is not available or insufficient licenses are available.
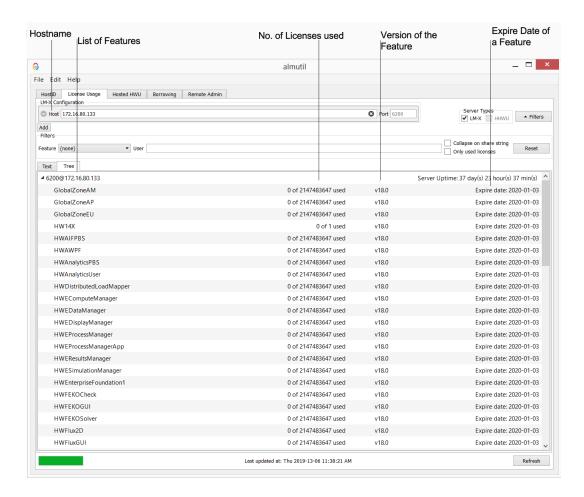
Remedy

Verify the details of HyperViewTrans feature and its licenses using Altair Utility:

1. Download and install the Altair License Utility Software.
   A shortcut, **Altair License Utility** is displayed.
2. Double-click and launch the utility.
3. Enter the Altair License server host name.
   The utility will provide the list of features with version, number of licenses used and unused, and expiry date of the feature.

Cause

If the error still persist even after the above checks, verify if there are any HyperViewTrans process ids running. Kill the running processes and try again.

Remedy

Follow the given steps:

1. To check if there any of the HyperViewTrans processes that are running, use the command:

    ```
    ps -ef | grep hvtrans
    ```

    The list of HyperViewTrans process ids is displayed.

2. To kill a process id, specify the `process id` in the following command:

    ```
    kill -9 <process id>
    ```

Result File Reader is Unrecognized

Unable to post process the result files when you try to plot or animation.

Condition

The supported file type reader is not configured in HyperWorks.

Cause

I am trying to open a result file using RVS and I receive the following error: `The result file reader is not configured in HyperWorks.`

Remedy

Follow the givens 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.

See Also

Activate Solver Files Readers
Supported Result File Types

Unable to Extract TOC of a Result File

Condition

RVS is unable to post process the plot or animation result files.

Cause

Insufficient permissions to read the result file.

Remedy

Ensure Compose and HyperWorks Desktop is installed in a location where all the Access Web users have read and write permissions.

- Compose and HyperWorks Desktop are recommended to install in the `Home` location, which enables read and write permission to all the Access Web users.
- Navigate to `ALTAIR_HOME` and verify the permissions.
  For example, the image displays Access Web users having executable permissions for the

  HyperWorks Desktop application. `drwxr-xr-x.  5 root root       4096 May  6 10:27 hw2017.3`.

Cause

The reason could be Compose is not accessible.

Remedy

Verify if Compose is installed properly.

- Verify if Compose is accessible by running the Compose script at: `<COMPOSE_INSTALL_DIR>/altair/Compose2021.2/scripts/Compose`.

Cause

The reason could be that the shared library package is not installed.

Remedy

Verify if the library packages are installed properly for Compose and HyperWorks Desktop.

On Linux platform, install these library packages for Compose and Altair HyperWorks Desktop.

a) For SLES 12 and SLES 15, install the library `libgomp` using the command `zypper install libgomp` and the library `libXss` using the command `zypper install libXss`.

See Also

Checklist for Troubleshooting

17.7 Logging

Information about defining PAS logging behavior and Log Files.

17.7.1 Logging Behavior

Information about defining PAS logging behavior.

PAS logging behavior is defined in the file `PA_HOME/config/pas/conf/server-log.properties`.

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.properties` file and set the `rootLogger.level` attribute value 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.

```
# Configure root logger
rootLogger.level = info
```

Remote File operations, Job submission, and Job Status

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

1. Navigate to `PA_HOME/config/pas/conf/`
2. Update `server-log.properties` with the following line:
   ```
   logger.bci.name=com.altair.gw.aif.cli.implementation.BasicCommandsImpl
   logger.bci.level=debug
   ```
3. Update `server-log.properties` to get detailed troubleshooting information for the job status:
   ```
   logger.pbseifl.name=com.altair.gw.aif.pbs.ifl.implementations.PbsEiflWs
   logger.pbseifl.level=debug
   ```

File Operations (local and remote)

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

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

2. Update `server-log.properties` to change the *priority value*:

```
logger.fiop.name=com.altair.gw.aif.fileop.implementation.FileOperations
logger.fiop.level=debug

logger.fiophl.name=com.altair.gw.aif.fileop.implementation.FileOperationsHelper
logger.fiophl.level=debug
```

Job Submission

Configure logging behavior for job submission.

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

2. Update `server-log.properties` to get detailed logging information about user inputs and to see how long it takes to execute a submission request by adding the following line:

```
logger.rjpi.name=com.altair.gw.aif.rest.RESTJobsPortImpl
logger.rjpi.level=debug
```

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

```
logger.pngju.name=com.altair.gw.aif.rest.util.PASNextGenJobUtils
logger.pngju.level=debug
```

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 line to the `server-log.properties` file to troubleshoot job submission:

```
logger.sshi.name=com.altair.gw.aif.ssh.implementation.SSHImplementation
logger.sshi.level=debug
```

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

Table 10: 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/pas/conf/`

2. Update `server-log.properties` with the following line:

    ```
    logger.jsr.name=com.altair.gw.aif.pbs.ifl.responses.JobsStatus_Response
    logger.jsr.level=debug

    logger.pbsos.name=com.altair.gw.aif.pbs.ifl.responses.PBSObjectStatus
    logger.pbsos.level=debug
    ```

    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/pas/conf/`

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

    ```
    logger.conv.name=com.altair.gw.aif.converter
    logger.conv.level=debug
    ```

Dynamic Application Refresh Script

Configure logging behavior for the dynamic application refresh script.

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

2. Update `server-log.properties` to get detailed logging information about dynamic application refresh script by adding the following line:

    ```
    logger.dab.name=com.altair.gw.aif.applications.utils.DynamicApplicationBuilder
    logger.dab.level=debug
    ```

During the Job Status Retrieval

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

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

2. Update `server-log.properties` to get user information and the total time of the PAS server operation by adding the following line:

    ```
    logger.rjpi.name=com.altair.gw.aif.rest.RESTJobsPortImpl
    logger.rjpi.level=debug
    ```

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

```
logger.jsr.name=com.altair.gw.aif.pbs.ifl.responses.JobsStatus_Response
logger.jsr.level=debug

logger.pbsos.name=com.altair.gw.aif.pbs.ifl.responses.PBSObjectStatus
logger.pbsos.level=debug
```

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

```
logger.sshi.name=com.altair.gw.aif.ssh.implementation.SSHImplementation
logger.sshi.level=debug
```

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/pas/conf/`

2. Update the `server-log.properties` to get more details on user information and the total time of the PAS server operation by adding the following line:

```
logger.rfpi.name=com.altair.gw.aif.rest.RESTFilePortImpl
logger.rfpi.level=debug
```

3. Update the `server-log.properties` 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 line:

```
logger.fiop.name=com.altair.gw.aif.fileop.implementation.FileOperations
logger.fiop.level=debug
```

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

```
logger.fiophl.name=com.altair.gw.aif.fileop.implementation.FileOperationsHelper
logger.fiophl.level=debug
```

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

```
logger.prwr.name=com.altair.gw.aif.process.ProcessWrapper
logger.prwr.level=debug
```

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 line to the `server-log.properties` file:

```
logger.rfotb.name=com.altair.gw.aif.fileop.implementation.cli.RemoteFileOpsTunnel-
Based
logger.rfotb.level=debug
```

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.

△ **ALTAIR**

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

```
logger.fdm.name=com.altair.gw.aif.fileop.implementation.FileDownloadModel
logger.fdm.level=debug
```

Application Definition Related Errors

Configure logging behavior to troubleshoot application definition related errors.

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

2. Update the `server-log.properties` to get more details on user information and the total time of the PAS server operation by adding the following line:

```
logger.ads.name=com.altair.gw.aif.rest.ApplicationDefinitionService
logger.ads.level=debug
```

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

```
logger.aprs.name=com.altair.gw.aif.applications.utils.ApplicationsRepository
logger.aprs.level=debug
```

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.properties` to get more information on cache debug logging by adding the following line:

```
logger.chmr.name=com.altair.gw.aif.utils.generic.CacheManager
logger.chmr.level=debug
```

User Profile

Configure logging behavior to troubleshoot user profile errors.

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

2. Update the `server-log.properties` to get more details on user information and the total time of the PAS Server operation by adding the following line:

```
logger.prsr.name=com.altair.gw.aif.rest.ProfileService
logger.prsr.level=debug
```

Server Registration

Configure logging behavior to troubleshoot server registration errors.

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

2. Update the `server-log.properties` to get more details on user information and the total time of the PAS Server operation by adding the following line:

```
logger.srvsr.name=com.altair.gw.aif.rest.ServerService
```

```
logger.srvsr.level=debug
```

User Account

Configure logging behavior to troubleshoot user account related errors.

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

```
logger.usrsr.name=com.altair.gw.aif.rest.UserService
logger.usrsr.level=debug
```

Remote Desktops Errors

Configure logging behavior to troubleshoot Remote Desktops related errors.

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

```
logger.sessr.name=com.altair.gw.aif.rest.SessionService
logger.sessr.level=debug
```

17.7.2 Locate RVS 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) Result Manager logs: `PA_HOME/logs/resultmanager/resultmanager.log`
b) RVS web server logs: `PA_HOME/logs/resultmanager/catalina.out`

17.7.3 Log Files

Information about the log files.

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/`

Other Log Files

The PAS installation log file is located in: `/opt/altair/pbsworks/pas/2022.1.0/_PAS Services_installation/Logs`

The log files for the Apache Tomcat web server are located in: `PBSWORKS_EXEC/pas/bin/pas-server/ logs`